User's Manual

RZ/T2H and RZ/N2H Groups

User's Manual: Hardware

RZ/T Series for Real-Time Control
RZ/N Series for Industrial Network
RZ Family
64-Bit & 32-Bit Arm®-Based High-End MPUs

Renesas Confidential

Renesas Electronics
www.renesas.com

Rev.1.10 Oct 2024

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Preface

1. About This Document

This manual is generally organized into an overview of the product, descriptions of the CPU, system control functions, peripheral functions, electrical characteristics, and usage notes. This manual describes the product specification of the microprocessors (MPUs) superset. Depending on your product, some pins, registers, or functions might not exist. Address space that store unavailable registers are reserved.

2. Audience

This manual is written for system designers who are designing and programming applications using the Renesas MPU. The user is expected to have basic knowledge of electrical circuits, logic circuits, and the MPU.

3. Renesas Publications

Renesas provides the following documents. Before using any of these documents, visit www.renesas.com for the most up-to-date version of the document.

Component	Document Type	Description
Hardware	Datasheet	Features, overview, and electrical characteristics
	User's Manual: Hardware	Specifications such as pin assignments, memory maps, peripheral functions, electrical characteristics, timing diagrams, and operation descriptions
	Application Notes	Technical notes, board design guidelines, and software migration information
	Technical Update (TU)	Preliminary reports on product specifications such as restriction and errata
Software	User's Manual: Software	API reference and programming information
	Application Notes	Project files, guidelines for software programming, and application examples to develop embedded software applications
Tools & Kits, Solutions	User's Manual: Development Tools	User's manuals and quick start guides for developing embedded software applications with Software Packages, Development Kits, Starter Kits, Promotion Kits, Product Examples, and Application Examples
	Quick Start Guide	
	Application Notes	Project files, guidelines for software programming, and application examples to develop embedded software applications

4. Numbering Notation

The following numbering notation is used throughout this manual:

Example	Description
011b	Binary number. For example, the binary equivalent of the number 3 is 011b.
0x1F	Hexadecimal number. For example, the hexadecimal equivalent of the number 31 is described 0x1F. In some cases, a hexadecimal number is shown with the suffix "h".
1234	Decimal number. A decimal number is followed by this symbol only when the possibility of confusion exists. Decimal numbers are generally shown without a suffix.

5. Typographic Notation

The following typographic notation is used throughout this manual:

Example	Description
AAA.BBB.CCC	Periods separated a function module symbol (AAA), register symbol (BBB), and bit field symbol (CCC).
AAA.BBB	A period separated a function module symbol (AAA) and register symbol (BBB).
BBB.DDD	A period separated a register symbol (BBB) and bit field symbol (DDD).
EEE[3:0]	Numbers in brackets expresses a bit number. For example, EEE[3:0] occupies bits 3 to 0.

6. Unit and Unit Prefix

The following units and unit prefixes are sometimes misleading. Those unit prefixes are described throughout this manual with the following meaning:

Symbol	Name	Description
b	Binary Digit	Single 0 or 1
B	Byte	This unit is generally used for memory specification and address space.
k	kilo-	$1000 = 10^3$. k is also used to denote 1024 (2^{10}) but this unit prefix is used to denote 1000 (10^3) throughout this manual.
K	Kilo-	$1024 = 2^{10}$. This unit prefix is used to denote 1024 (2^{10}) not 1000 (10^3) throughout this manual.

7. Special Terms

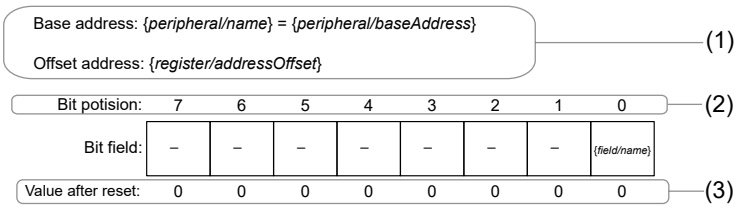
The following terms have special meanings.

Term	Description
NC	Not connected pin. This pin should be left floating unless specified otherwise.
Hi-Z	High impedance.
x	Don't care or undefined.

8. Register Description

Each register description includes both a register diagram that shows the bit assignments and a register bit table that describes the content of each bit. The example of symbols used in these tables are described in the sections that follow. The following is an example of a register description and associated bit field definition.

XX.X.X {*register/name*} : {*register/description*}



(4) (5) (6)

Bit	Symbol	Function	R/W
0	{*field/name*}	{*field/description*} 0: {*enumeratedValue/description*} 1: {*enumeratedValue/description*}	{*access*}
7:1	–	These bits are read as 0. The write value should be 0.	R/W

(1) Function module symbol, register symbol, and address assignment

Function module symbol, {*pheripheral/name*}, register symbol, {*register/name*}, and address assignment of this register are generally expressed. Base address and Offset address mean {*register/name*} : {*register/description*} of {*peripheral/name*} is assigned to address {*peripheral/baseAddress*} + {*register/addressOffset*}.

(2) Bit number

This number indicates the bit number. This bits are shown in order from bits 31 to 0 for 32-bit register, from bits 15 to 0 for 16-bit register, and from bits 7 to 0 for 8-bit register.

(3) Value after reset

This symbol or number indicate the value of each bit after a hard reset. The value is shown in binary unless specified otherwise.

- 0: Indicates that the value is 0 after a reset.
- 1: Indicates that the value is 1 after a reset.
- x: Indicates that the value is undefined after a reset.

(4) Symbol

{*filed/name*} indicates the short name of bit field. Reserved bit is expressed with a —.

(5) Function

Function indicates the full name of the bit field, {*field/description*}, and enumerated values.

(6) R/W

The R/W column indicates the access type whether the bit field is readable or writable.

- R/W: The bit field is readable and writable.
- R: The bit field is readable only. Writing to this bit field has no effect.
- W: The bit field is writable only. The read value is the same as after a reset unless specified otherwise.

9. Abbreviations

The abbreviations used in this document are shown in the following table.

Abbreviation	Description
AES	Advanced Encryption Standard
AHB	Advanced High-performance Bus
AHB-AP	AHB Access Port
APB	Advanced Peripheral Bus
ATB	Advanced Trace Bus
AXI	Advanced eXtensible Interface
BCD	Binary Coded Decimal
BSDL	Boundary Scan Description Language
CTI	Cross Trigger Interface
DAP	Coresight Debug Access Port
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
ETM	Embedded Trace Macrocell
JTAG	Joint Test Action Group
LSB	Least Significant Bit
MSB	Most Significant Bit
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
RSA	Rivest Shamir Adleman
S/H	Sample and Hold
SHA	Secure Hash Algorithm
SWD	Serial Wire Debug
TRNG	True Random Number Generator
UART	Universal Asynchronous Receiver/Transmitter

10. Proprietary Notice

11. Feedback on the Product

If you have any comments or suggestions about this product, go to Contact Us.

Contents

RZ/T2H and RZ/N2H Groups
User's Manual

600/1200 MHz, Cortex-A55 Quad/Dual/Single MPCore, 500/1000 MHz, Dual Arm® Cortex®-R52 on-chip FPU and NEON™, 2.0 MB of on-chip SRAM, LPDDR4-3200 with 32-bit, Ethernet MAC, EtherCAT, USB 2.0 high-speed, PCI Express Gen3, SD card host interface, CAN/CANFD, various communications interfaces such as an xSPI and ΔΣ interface, encoder interfaces, and security functions

Features

On-chip 64-bit Arm Cortex-A55 processor

- Quad/Dual/Single MPCore cores
- Maximum operating frequency:
  - Core: 600/1200 MHz
  - DSU: 500/1000 MHz
- 64-bit Arm Cortex-A55 Quad/Dual/Single MPCore cores (revision r2p0)
- Address space: 32 Gbytes
- L1 cache:
  - I-cache: 32 Kbytes (with parity)
  - D-cache: 32 Kbytes (with ECC)
- L2 cache: 0 bytes
- L3 cache: 1024 Kbytes (with ECC)
- NEON/FPU supported
- Cryptographic extension supported (Security product only)
- Arm V8.2-A architecture

On-chip 32-bit Arm Cortex-R52 processor

- High-speed realtime control with operating frequency of 500/1000 MHz
- On-chip Dual 32-bit Arm Cortex-R52 (revision r1p4)
- Tightly coupled memory (TCM) with ECC
  - CPU0, CPU1: 512 KB/64 KB
- Instruction cache/data cache with ECC
  - CPU0, CPU1: 16 KB per cache
- High-speed interrupt
- NEON/FPU supported
- Harvard architecture with 8-stage pipeline
- Supports the memory protection unit (MPU)
- Arm CoreSight architecture, includes support for debugging through JTAG and SWD interfaces.
- No DCLS (Dual Core Lock Step) support

Low power consumption

- Standby mode and module stop function

On-chip SRAM

- 2.0 MB of the on-chip SRAM with ECC
- 250 MHz

LPDDR4 SDRAM Memory Interface

- Data transfer: 3.2 Gbps
- Bus width: 32-bit
- Rank: 1, 2
- Size: up to 64 Gb

Data transfer

- DMAC: 16 channels × 3 units

Event link controller

- Module operations can be started by event signals rather than by interrupt handlers.
- Linked operation of modules is available even while the CPU is in the standby state.

Reset and power supply voltage control

- Ten reset sources including a pin reset

Clock functions

- External clock/oscillator input frequency: 25 MHz
- CPU clock frequency: 600/1200 MHz (Cortex-A55), 500/1000 MHz (Cortex-R52)
- System clock frequency: 400 MHz (A-Bus), 250 MHz (R-Bus)
- Low-speed on-chip oscillator (LOCO): 1 MHz

Safety functions

- Register write protection, input clock oscillation stop detection and CRC
- Master Memory Protection Unit (MPU)

Security functions (optional)

- Boot mode with security through encryption
- JTAG authentication
- Cryptologic accelerator
- TRNG
- Cortex-A55 Crypto Extension
- Arm® TrustZone® technology

Encoder interfaces

- 16 channels
- EnDat 2.2, BiSS-C, A-format, and HIPERFACE DSL-compliant interfaces
- Frequency-divided output from an encoder

Various communications interfaces

- Ethernet
  - EtherCAT slave Controller: 3 ports
  - Ethernet switch: 3 ports
  - Ethernet MAC: 1 port × 3 units
- USB 2.0 high-speed host/functions: 1 channel
- CAN/CANFD (compliant with ISO11898-1): 2 channels
- SCI with 16-byte transmission and reception FIFOs: 6 channels + 12 channels (for encoder)
- I2C bus interface: 3 channel for transfer at up to 400 kbps
- SPI: 4 channels
- xSPI: 2 channels
- PCI Express Gen3: 2 lane × 1 port or 1 lane × 2 ports
- SD card host interface: 2 channels

External host interface

- Serial host interface

External address space

- Buses for high-speed data transfer at up to 125 MHz
- Support for up to 4 CS areas
- 8-, 16- or 32-bit bus space is selectable per area

Up to 73 extended-function timers

- 16-bit × 8 + 32-bit MTU3 (9 channels), 32-bit GPT (56 channels)
- 16-bit CMT (6 channels), 32-bit CMTW (2 channels)

ΔΣ interface

- Up to 30 ΔΣ modulators are connectable externally.

Trigonometric function unit

- 2 units
- Simultaneous calculation of sine and cosine
- Simultaneous calculation of arctangent and hypot_k

12-bit A/D converter

- 12 bits × 3 unit (4 channels for unit 0, 1, 15 channels for unit 2)

Temperature sensor for measuring temperature within the chip

LCD Controller

- Support Image Processing (Dither, Clipping, Gamma Correction)
- Parallel output interface
- Output Data Format: RGB666 / RGB888
- Support WXGA (1280 × 800), 60 fps

General-purpose I/O ports

- Input pull-up/pull-down
- The locations of input/output functions for peripheral modules are selectable from among multiple pins.

Operating temperature range

- Tj = -40 to +125°C

1. Overview

1.1 Outline of Specifications

The MPU is a high-performance ASSP that has Cortex-A55 quad MPCore and dual Arm Cortex®-R52 processor with Floating-Point Unit (FPU) and NEON™. It incorporates integrated peripheral functions necessary for system configuration.

Table 1.1 CPU

Feature	Functional description
Arm® Cortex®-A55	• One Processor which consist of Quad/Dual/Single MPCore cores • Operating frequency – Core: 600 MHz/1200 MHz – DSU: 500 MHz/1000 MHz • 64-bit CPU Cortex-A55 designed by Arm (core revision r2p0) • Address space: 32 GB • L1 cache – I-cache: 32 KB (with parity) each core – D-cache: 32 KB (with ECC) each core • L2 cache: 0 bytes • L3 cache: 1024 KB (with ECC) • Arm V8.2-A architecture • NEON/FPU supported • Cryptographic extension supported (Security product only)
Arm® Cortex®-R52	• Two processors which consist of single core • Operating frequency – Each CPU0 and CPU1: 500 MHz/1000 MHz • 32-bit CPU Cortex-R52 designed by Arm (core revision r1p4) • Address space: 4 GB • Instruction cache – Each CPU0 and CPU1: 16 KB (with ECC) • Data cache – Each CPU0 and CPU1: 16 KB (with ECC) • Tightly coupled memory (TCM) – Each CPU0 and CPU1 ATCM: 512 KB (with ECC) 0 wait (500 MHz) / 1 wait (1000 MHz) BTCM: 64 KB (with ECC) 0 wait CTCM: 0 KB • Instruction set: ARMv8-R architecture, so support includes Thumb® and Thumb-2 • Two stage Memory protection unit (MPU) • NEON/FPU supported

Table 1.2 Memory

Feature	Functional description
On-chip system SRAM with ECC	• Capacity: 2.0 MB (512 KB × 4 units) (with ECC) • Operating frequency: 250 MHz • SEC-DED (single error correction/double error detection) Error injection supported
One-time programmable memory	• Overwrite protection • Redundancy support • Available information – Unique ID – Authentication settings – Trimming data – Boot mode setting – User area
LPDDR4 SDRAM memory interface	• Data transfer: 3.2 Gbps • Bus width: 32 bits • Rank: 1, 2 • Size: up to 64 Gb • Inline ECC • No support of LPDDR4X

Table 1.3 System

Feature	Functional description
Operating modes	The operating mode can be selected from the following seven boot modes: ● xSPI0 boot mode (CS0 × 1 boot Serial Flash) ● xSPI0 boot mode (CS0 × 8 boot Serial Flash) ● eMMC boot mode ● SD boot mode ● xSPI1 boot mode (CS0 × 1 boot Serial Flash) ● SCI boot mode ● USB boot mode
Clock generation circuit	● The input clock can be selected from an external clock or external resonator. ● Detection of input clock oscillation stopping ● The following clocks are generated: – Cortex-A55 clock: 600 or 1200 MHz (selectable each core) – Cortex-R52 clock: 500 or 1000 MHz (selectable each unit) – A-Bus clock: 400 MHz – R-Bus clock: 250 MHz – High-speed peripheral module clock A (PCLKAH): 400 MHz – Middle-speed peripheral module clock A (PCLKAM): 200 MHz – Low-speed peripheral module clock A (PCLKAL): 100 MHz – High-speed peripheral module clock R (PCLKH): 250 MHz – Middle-speed peripheral module clock R (PCLKM): 125 MHz – Low-speed peripheral module clock R (PCLKL): 62.5 MHz – ADC clock in the 12-bit A/D converter: 62.5 MHz – External bus clock 125 MHz – Low-speed on-chip oscillator: 1 MHz
Reset	RES# pin reset, software reset, error reset, Cortex-A55 cluster software reset, Cortex-A55 core0 software reset, Cortex-A55 core1 software reset, Cortex-A55 core2 software reset, Cortex-A55 core3 software reset, Cortex-R52 CPU0 software reset, Cortex-R52 CPU1 software reset
Low-power consumption function	● Standby mode (Cortex-A55, Cortex-R52) ● Module stop function
Interrupt controller (ICU)	● Connect an interrupt to the GIC600 for Cortex-A55 ● Connect an interrupt to the GIC (Generic Interrupt Controller) for Cortex-R52 CPU0 and CPU1 ● Connect an activating trigger to the DMAC and ELC ● Peripheral function interrupts: 394 sources ● External interrupts: 16 sources (IRQ0 to IRQ15 pins) ● Software interrupts: 16 sources ● System error interrupts: 1 source ● 32 levels specifiable for the order of priority in the GIC
Bus state controller (BSC)	● The external address space is divided into four areas (CS0, CS2, CS3, and CS5) for management. ● The following features are configurable for each area independently: Bus size (8, 16, or 32 bits): Available sizes depend on the area. Number of access wait cycles (different wait cycles can be specified for read and write access cycles in some areas). Idle wait cycle insertion (between same area access cycles or different area access cycles). Specifying the memory to be connected to each area enables direct connection to SRAM, SRAM with byte selection, and burst ROM (clocked synchronous or asynchronous). The address/data multiplexed I/O (MPX) interface is also available. ● Outputs a chip select signal (CS0# to CS5#) according to the target area (CS assert or negate timing can be selected by software).

Table 1.4 Direct memory access

Feature	Functional description
Direct memory access controller (DMAC)	● 3 units (16 channels each unit) ● Transfer modes: Single transfer mode and block transfer mode ● Transfer size – Unit 0: 1/2/4/8/16/32/64 bytes – Unit 1, 2: 1/2/4/8/16/32 bytes ● Activation sources: Software trigger, external DMA requests (DREQ), external interrupts, and interrupt requests from peripheral functions

Table 1.5 I/O Ports

Feature	Functional description
General-purpose I/O ports	● 729-pin FCBGA – I/O pins: 287 – Input pins: 0 – Pull-up/pull-down resistors: 287 ● 576-pin FCBGA – I/O pins: 189 – Input pins: 0 – Pull-up/pull-down resistors: 189 ● The locations of input/output functions are selectable from among multiple pins.

Table 1.6 Event link

Feature	Functional description
Event link controller (ELC)	● Up to 648 event signals can be interlinked with the operation of modules. ● In particular, the operation of timer modules can be started by input event signals. ● Event-linked operation of signals of ports 14 and port 30 is to be possible.

Table 1.7 Timers (1 of 2)

Feature	Functional description
Multi-function timer pulse unit 3 (MTU3)	● 9 channels (16 bits × 8 channels, 32 bits × 1 channel) ● Maximum of 28 pulse-input/output and 3 pulse-input possible ● Select from among 10, 11, 12, or 14 counter-input clock signals for each channel (with maximum operating frequency of 250 MHz) ● Input capture function ● 39 output compare/input capture registers ● Counter clear operation (synchronous clearing by compare match/input capture) ● Simultaneous writing to multiple timer counters (TCNT) ● Simultaneous register input/output by synchronous counter operation ● Buffered operation ● Support for cascade-connected operation ● Automatic transfer of register data ● Pulse output mode Toggle/PWM/complementary PWM/reset-synchronized PWM ● Complementary PWM output mode – Outputs non-overlapping waveforms for controlling 3-phase inverters – Automatic specification of dead times – PWM duty cycle: Selectable as any value from 0% to 100% – Delay can be applied to requests for A/D conversion. – Non-generation of interrupt requests at peak or trough values of counters can be selected. – Double buffer configuration ● Reset synchronous PWM mode Six phases of positive and negative PWM waveforms can be output with desired duty cycles. ● Phase-counting mode: 16-bit mode (channels 1 and 2), 32-bit mode (channels 1 and 2 in cascade connection) ● Counter functionality for dead-time compensation ● Generation of triggers for A/D converter conversion ● A/D converter start triggers can be skipped ● Digital noise filter function for signals on the input capture and external counter clock pins ● Event linking by the ELC

Table 1.7 Timers (2 of 2)

Feature	Functional description
General PWM timer (GPT)	• 32 bits × 56 channels (5 channels × 9 units + 7 channels × 1 unit + 4 channels × 1 unit) • Counting up or down (saw-wave), counting up and down (triangle-wave) selectable for all channels • Select from among four counter-input clock signals for each channel (with maximum operating frequency of 500 MHz for LLPP) • 2 input/output pins per channel • 2 output compare/input capture registers per channel • For the 2 output compare/input capture registers of each channel, 4 registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use. • In output compare operation, buffer switching can be at peaks or troughs, enabling the generation of laterally asymmetrically PWM waveforms. • Registers for setting up frame intervals on each channel (with capability for generating interrupts on overflow or underflow) • Synchronizable operation of the several counters • Modes of synchronized operation (synchronized, or displaced by desired times for phase shifting) • Generation of dead times in PWM operation • Through combination of 3 counters, generation of automatic 3-phase PWM waveforms incorporating dead times • Starting, clearing, stopping, switching, up/down counters, and input capture in response to external or internal triggers • Starting, clearing, stopping, switching, up/down counters, and input capture in response to input level comparison • Internal trigger sources: software and compare-match • Generation of triggers for A/D converter conversion • Digital noise filter function for signals on the input capture and external trigger pins • Event linking by the ELC • Function of output duty 0% and 100% is selectable from troughs or crests/troughs.
Compare match timer (CMT)	• (16 bits × 2 channels) × 3 units • Select from among four counter-input clock signals for each channel
Compare match timer W (CMTW)	• (32 bits × 1 channel) × 2 units • Compare-match, input-capture input, and output-comparison output are available. • Select from among four counter-input clock signals for each channel • Interrupt requests can be output in response to compare-match, input-capture, and output-comparison events.
Watchdog timer (WDT)	• 14 bits × 6 channels • Select from among six counter-input clock signals for each channel
Port output enable 3 (POE3)	• Control of the high-impedance state of the MTU3 waveform output pins • 5 pins for input from signal sources: POE0#, POE4#, POE8#, POE10#, POE11# • Initiation on detection of short-circuited outputs (detection of simultaneous PWM output to the active level) • Initiation by input clock oscillation-stoppage detection, PLL oscillation anomaly detection, two types of DSMIF error detection, or software • Additional programming of output control target pins is enabled
Port output enable for GPT (POEG)	• Controlling the output disable for GPT waveform output • Initiation by input level detection of GTETRG pins • Initiation by output disable request from GPT • Initiation by detection of oscillation stop, two types of DSMIF error detection, or by software
Real time clock (RTC)	• A 100 year calendar from 2000 to 2099 • BCD code display • Clock source is division of main oscillator. • Automatic adjustment function for leap years

Table 1.8 Communication interfaces (1 of 3)

Feature	Functional description
Ethernet MAC (GMAC)	• 1 port × 3 units • IEEE802.3 • IEEE1588-2008 • IEEE802.3-az-2010 for EEE • Support for 10/100/1000 Mbps data transfer • Full duplex and half duplex are supported • Programmable frame length to support both standard and jumbo frames up to 16 KB • 32 MAC address registers for the address filter block • Variety of flexible addresses filtering modes are supported • Advanced IEEE 1588-2002 & 2008 Ethernet frame time-stamping supported • MII/RMII/RGMII interface is supported by RMII/RGMII converter • TSN features (IEEE802.1Qbv, IEEE802.1Qbu/802.3br) • Multi queues support up to 8 RX and 8 TX queues • DMA channels: 8 RX and 8 TX channels • VLAN Filtering, L3 and L4 Filtering
Ethernet switch (ETHSW)	• 3-port PHY interfaces • IEEE802.3 • Support for 10/100/1000 Mbps data transfer • Full and half duplex (1000 Mbps supports full-duplex only) • Hardware switching, lookup, and filtering • QoS with frame prioritization • Priority control based on VLAN Priority (IEEE802.1q), which enables priority reassignment • Classification and priority assignment based on IPv4 DiffServ Code Point Field, IPv6 Class of Service • Queue with eight priority levels • Multicasting and broadcasting • VLAN frame • IEEE 1588-2008 compatible • Programmable addition, removal and manipulation of ingress and egress VLAN tags, supporting single and double-tagged VLAN frames on each port • Cut-through and hub features • Device level ring (DLR) • Programmable egress rate limit per port • Ingress Configurable Broadcast/multicast storm protection per port • IEEE802.1X source address authentication supported • IEEE802.1X guest VLAN supported • PRP functionality (IEC 62439-3 edition 2.0- 2012) • Configurable Time Multiplexed (TDMA) output queue scheduler supporting real-time network infrastructures using time slots for bandwidth reservation enabling deterministic delays • Frame preemption • Cyclic queuing and forwarding • Forwarding rule for the TSN-IA profile • A MAC source address filtering • Pattern matchers 12 channels • Independent two timer module are available for timestamping and time for TDMA. • Remote monitoring through SNMP • Powerlink capable hub • Ingress filtering and frame header manipulation (active stream identification, flow metering) with Enhanced Frame Parser • 4 additional PTP timer pulse generators • MII/RMII/RGMII interface is supported by RMII/RGMII converter
EtherCAT slave controller (ESC)*1	• 1 channel (3 ports) • EtherCAT Slave Controller IP core (made by Beckhoff Automation GmbH) implemented • MII interface is supported. RMII/RGMII interface is supported by RGMII converter in Ethernet Subsystem.
USB 2.0 HS host/function module	• 1 port • Compliance with the USB 2.0 specification • OTG support • Transfer rate High speed (480 Mbps), full speed (12 Mbps), low speed (1.5 Mbps, host only) • Communications buffer – Incorporates 1 KB of RAM for host mode – Incorporates 8 KB of RAM for function mode • DMAC (2 channels) incorporated

Table 1.8 Communication interfaces (2 of 3)

Feature	Functional description
Serial communication interface (SCI)	• 6 channels + 12 channels (for encoder) • 6 communication mode – Asynchronous interfaces – 8-bit clock synchronous interface – Simple I^2C (master-only) – Simple SPI – Smart card interface – Manchester mode (For channel for encoder, Simple I^2C, Simple SPI, and Smart card interface mode are not supported.) • Clock source is select from among four internal clock signals • Bit rate specifiable with the on-chip baud rate generator • Full-duplex and half-duplex communication • Data length: 7 to 9 bits (Asynchronous mode) • Bit rate modulation • Double speed mode (Asynchronous, Clock Synchronous, Simple SPI mode, Manchester mode) • RS-485 driver control function (Asynchronous mode) • Loopback function to enable self-diagnosis (Asynchronous, Clock synchronous mode)
I^2C bus interface (IIC)	• 3 channels • Communication formats: I^2C bus format or SMBus format • Master or slave mode selectable • Supports the multi-master • Maximum transfer rate: 400 kbps (Standard mode and Fast mode)
CAN-FD module (CANFD)	• 2 channels • Comply with CAN-FD ISO 11898-1 (2015) • Communication speed – Classical CAN mode: 1 Mbps – CAN FD mode: Nominal bit rate: max. 1 Mbps Data bit rate: max. 8 Mbps • Total 192 message buffers (in case frame size is 76 bytes) – Individual buffers: 64 for TX – Shared buffers: 128 for TX and RX including FIFO • Selectable ID type with 11-bit Standard and 18-bit Extended • Selectable Frame type: Data Frame and Remote Frame • Up to 256 receive rules
Serial peripheral interface (SPI)	• 4 channels • SPI transfer facility Using the MOSI (master out slave in), MISO (master in slave out), SSL (slave select), and RSPCK (SPI clock) signals enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines) capable of handling serial transfer as a master or slave • Data formats – Switching between MSB first and LSB first – Transfer bit length selectable to 4 - 32 bits – 32 bits × 4-stage FIFO transmit and receive buffers – Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits) • RSPCK can be stopped automatically with the reception buffer full for master reception

Table 1.8 Communication interfaces (3 of 3)

Feature	Functional description
Expanded serial peripheral interface (xSPI)	• 2 channels • Comply with JESD251 • Multiple slave up to 2 slaves • Protocol mode: 1/4/8pin with SDR/DDR 1S-1S-1S, 4S-4D-4D, 8D-8D-8D • Support OctaFlash, OctaRAM, HyperFlash and HyperRAM • Protocol mode: 2/4pin with SDR compatible with QSPI 1S-2S-2S, 2S-2S-2S 1S-4S-4S, 4S-4S-4S • Configurable address length • Configurable initial access latency cycle • Support XiP mode • Support up to 256 MB address space • Prefetch function for burst-read with low latency • Outstanding buffer for burst-write with high throughput • Manual command configurable up to 4 commands • Output clock/input strobe port timing shift • Automatic command after released reset: up to 4 commands
SD/eMMC Host Interface (SDHI)	• SD memory/IO card interface (1-bit/4-bit SD bus) • SD, SDHC, and SDXC SD memory card access supported • Default, high-speed, UHS-I/SDR12, SDR25, SDR50, SDR104, and DDR50 transfer modes supported • Error check function: CRC7 (for command/response), CRC16 (for data) • Card detect function • Write protect supported • MMC interface (1-/4-/8-bit MMC bus) • e-MMC device access supported • Backward-compatible, high-speed (SDR/DDR), and HS200 transfer modes supported • High-priority interrupt (HPI) supported
PCI Express Gen3 (PCIE)	• PCI Express Gen1 (2.5 [GT/s]) / Gen2 (5.0 [GT/s]) / Gen3 (8.0 [GT/s]) • Root Complex / Endpoint Applications, Type 0/1 Configuration Register • Lane/Port: 1 lane × 2 ports or 2 lanes × 1 port selectable • Support Polarity inversion • Maximum data payload of 256 bytes, Maximum read request size 512 bytes • Number of outstanding 1-8 • Dynamic control of speed/width up/down configuration • Power Management (not support ASPM L1-Substate) • Error handling/logging (AER Support) • Replay FIFO with ECC • Number of Support Functions 2

Note 1. EtherCAT is a registered trademark of Beckhoff Automation GmbH, Germany.

Table 1.9 Analog

Feature	Functional description
12-bit A/D converter (ADC12)	• 12 bits × 3 units (unit 0, 1: 4 channels, unit 2: 15 channels) • 12-bit resolution • Conversion time 0.32 µs per channel • Operating mode Scan mode (single scan mode, continuous scan mode, or 3 group scan mode) Group priority control • Sample-and-hold function Common sample-and-hold circuit included In addition, channel-dedicated sample-and-hold function (3 channels: in all units) included • Sampling variable Sampling time can be set up for each channel • Double trigger mode (A/D conversion data duplicated) • Three ways to start A/D conversion Software trigger, timer (MTU3, ELC) trigger, external trigger • Event linking by the ELC
Temperature sensor unit (TSU)	• 1 channel • Relative precision: ±2°C (typ)

Table 1.10 Hardware accelerator for industrial interfaces

Feature	Functional description
ΔΣ interface (DSMIF)	• 3 channels × 10 units • Selectable 2 inputs (U/V) or 3 inputs (U/V/W) • Up to 6 ΔΣ modulators are externally connectable • Sinc filter can be selected as first, second or third order • Direct error connection to POE3 and POEG • Enhancement of current error detection • Core clock: 250 MHz/400 MHz selectable
Trigonometric function unit (TFU)	• 2 units • Calculation of sine, cosine, arctangent, hypot_k ($\sqrt{x^2+y^2}/k$) • Simultaneous calculation of sine and cosine • Simultaneous calculation of arctangent and hypot_k
Encoder interfaces	• EnDat 2.2 (16 units) • BiSS-C (16 units) • A-format (16 units) • HIPERFACE DSL (16 units) • ENCOUT (1 unit)

Table 1.11 Safety

Feature	Functional description
Memory protection unit (MPU)	• Cortex-R52 MPU Two stages MPUs (EL2 and EL1) 24 regions each MPU • Master MPU Memory protection for masters except Cortex-A55 and Cortex-R52 (DMAC, USB, Ethernet MAC, CoreSight, SHOSTIF, LCDC, SDHI, PCIE)
Register write protection function	Protects important registers from being overwritten for in case a program runs out of control.
CRC calculator (CRC)	• 2 channels • CRC code generation for arbitrary amounts of data in 8-, 16-, or 32-bit units • Select any of five generating polynomials: – $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (32-Ethernet) – $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ (CRC-32C) – $X^{16} + X^{15} + X^2 + 1$ (CRC-16) – $X^{16} + X^{12} + X^5 + 1$ (CRC-CCITT) – $X^8 + X^2 + X + 1$ (CRC-8)
Clock monitor circuit (CLMA)	• Monitors the abnormal output clock frequency from the input clock (main clock oscillator), PLL circuit, or low-speed on-chip oscillator. • Input clock oscillation stop detection: Available
Data operation circuit (DOC)	The function to compare, add, or subtract 16-bit data
Isolated peripherals	• Safety dedicated peripherals are available: – GPT: 4 ch – SCI: 1 ch – IIC: 1 ch – SPI: 1 ch – CRC: 1 unit – RTC: 1 unit – GPIO: Sharable with normal GPIO – On-chip system SRAM with ECC • They are mapped independently from normal peripherals so that access protection can be done by EL2 MPU.

Table 1.12 Security

Feature	Functional description
Security*1	• Secure boot • JTAG authentication • SCI/USB boot authentication • Cryptographic accelerators – Symmetric Cipher: AES 128/192/256 bits with CBC/ECB/CTR/GCM/XTS – Asymmetric Cipher: ECC 256 bits, RSA 1024/2048/3072 bits, RSAES-OAEP – Hash: SHA-1, SHA-2 – Message authentication: HMAC, CMAC, GMAC – Signature algorithms: ECDSA with NIST P-256, RSASSA-PSS, RSASSA-PKCS1 • TRNG • Cortex-A55 Crypto Extension • Arm® TrustZone® technology

Note 1. For details, contact our sales representative.

Table 1.13 Debug

Feature	Functional description
Debugging interface	• CoreSight architecture designed by Arm • Debugging function by the JTAG/SWD interface, and trace function by ETF and system bus by ETR

Table 1.14 External host interface

Feature	Functional description
Serial host interface (SHOSTIF)	• Serial communication is possible in slave mode. • Supported interface – Motorola Serial Peripheral Interface (4-wire SPI) – Enhanced SPI Modes with Dual, Quad, or Octal SPI • Serial clock polarity switching • Serial clock phase switching • Single Data Transfer • Data size is up to 32 bits × 64 burst
Mailbox and semaphore (MBXSEM)	• Eight semaphores for external host CPU and Cortex-A55/Cortex-R52 • Four 32-bit mailboxes for both external host CPU to Cortex-A55/Cortex-R52 and Cortex-A55/Cortex-R52 to external host CPU • Interrupts can be generated and cleared from both external host CPU and Cortex-A55/Cortex-R52 • Semaphores and Mailboxes with exclusive access among Cortex-A55, Cortex-R52 CPU0 and CPU1.

Table 1.15 Display interface

Feature	Functional description
LCD Controller (LCDC)	• 2 planes blending (can blend 2 different size images) • Support Image Processing: – Dither processing (RGB666) – Clipping – RGB Gamma Correction LUT • Support Input Data Format: – RGB565 / RGB666 / RGB888 – ARGB1555 / ARGB4444 / ARGB8888 – YcbCr444 8-bit / YcbCr422 8-bit / YcbCr420 8-bit • Support WXGA (1280 × 800), 60 fps • Support Output Data Format: – RGB666 / RGB888 • CLK / HD / VD timing signal supported

Table 1.16 Others

Feature	Functional description
Power supply voltage	0.8 V: Core (Digital and Analog (PLL, TSU, OTP, USB, PCIE, ADC)) 1.1 V: DDR 1.8 V: PLL, OSC, USB, ADC, TSU, OTP, PCIE, DDR, RGMII 3.3 V: GPIO (in 3.3 V fixed domain), USB, OSC, RMII/MII, other peripherals 1.8 V/3.3 V selectable: xSPI, SDHI, GPIO (in 1.8 V/3.3 V selectable domains)
Operating temperature	Tj = −40 to +125°C
Packages	RZ/T2H: 729 pin FCBGA 23 × 23 mm, 0.8-mm pitch RZ/N2H: 576 pin FCBGA 21 × 21 mm, 0.8-mm pitch

1.2 Function Comparison

Table 1.17 Function comparison (1 of 2)

Module/Function		RZ/T2H (729-pin FCBGA)	RZ/N2H (576-pin FCBGA)
CPU	ARM Cortex-A55	Quad/Dual/Single	
	ARM Cortex-R52	Two	
Memory	System SRAM	2.0 MB	
External bus	External bus width	32 bits	
Interrupt	External interrupt	SEI, IRQ0 to IRQ15	
DMA	DMA controller (DMAC)	3 units (16 channels each unit)	
Timer	Multi-function timer pulse unit 3 (MTU3)	1 unit (9 channels)	
	General PWM timer (GPT)	11 units (56 channels)	11 units (56 channels)*1
	Compare match timer (CMT)	3 units (6 channels)	
	Compare match timer W (CMTW)	2 channels	
	Watchdog timer	6 channels	
	Port output enable 3 (POE3)	1 unit	
	Port output enable for GPT (POEG)	3 unit	
	Real time clock (RTC)	1 unit	
Communication function	Ethernet MAC (GMAC)	3 units	3 units*1
	Ethernet switch (ETHSW)	1 unit (3 external ports)	1 unit (3 external ports)*1
	EtherCAT slave controller (ESC)	1 unit (3 external ports)	
	USB 2.0 HS host/function module (USB)	1 unit (1 port)	
	Serial communication interface (SCI, SCIE)	6 channels + 12 channels	
	I2C bus interface (IIC)	3 channels	
	CANFD module (CANFD)	2 channels	
	Serial peripheral interface (SPI)	4 channels	4 channels*1
	Expanded serial peripheral interface (xSPI)	2 channels	2 channels*1
	PCI Express Gen3 (PCIE)	1 unit (1 lane × 2 ports or 2 lanes × 1 port)	
	SD/eMMC Host Interface (SDHI)	2 channels	
12-bit A/D converter (ADC12)		3 units (4 channels for unit 0 and unit 1, 6 channels for unit 2)	3 units (4 channels for unit 0 and unit 1, 15 channels for unit 2)
Temperature sensor unit (TSU)		1 unit	
ΔΣ interface (DSMIF)		10 units (30 channels)	8 units (23 channels)*2
Trigonometric function unit (TFU)		2 units	
Encoder I/F	EnDat 2.2 (ENDAT)	16 units	14 units*3
	BiSS-C (BISS)	16 units	14 units*3
	A-format (AFMT)	16 units	14 units*3
	Hiperface DSL (HDSL)	16 units	14 units*1 *3
	ENCOUT	1 unit	1 unit
CRC calculator (CRC)		2 channels	
Clock monitor circuit (CLMA)		7 units	
Data operation circuit (DOC)		1 unit	
Security*4		Optional	

Table 1.17 Function comparison (2 of 2)

Module/Function	RZ/T2H (729-pin FCBGA)	RZ/N2H (576-pin FCBGA)
One-time programmable memory (OTP)	Available	
Serial host interface	1 unit	
Mailbox and semaphore (MBXSEM)	Available	
Event link controller (ELC)	Available	
LPDDR4 SDRAM Subsystem (DDRSS)	32 bits	
LCD Controller (LCDC)	1 unit	

Note 1. A part of external signals are not available.
Note 2. Unit 6, 9 and ch 2 of unit 8 are not available due to lack of mandatory external signals.
Note 3. Unit 8 and 15 are not available due to lack of mandatory external signals.
Note 4. For details, contact our sales representative.

1.3 Product Lineup

Table 1.18 shows a product lineup.

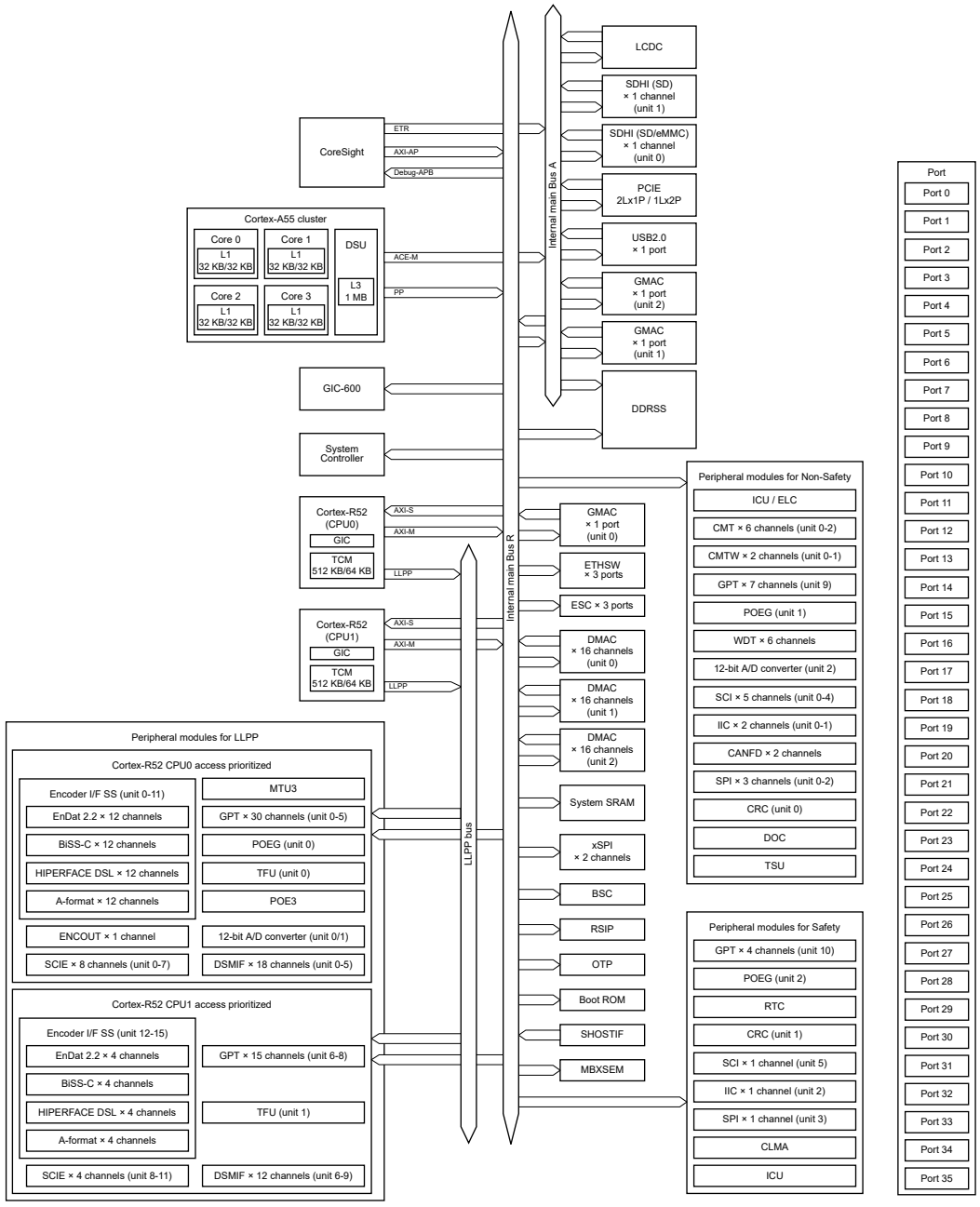
Table 1.18 Product lineup

Product	Part Number	Package	Cortex-A55	Cortex-R52	Security*1
RZ/T2H	R9A09G077M48GBG	729-pin FCBGA	Quad cores	Two CPUs	Available
	R9A09G077M28GBG	729-pin FCBGA	Dual cores	Two CPUs	Available
	R9A09G077M08GBG	729-pin FCBGA	Single core	Two CPUs	Available
	R9A09G077M44GBG	729-pin FCBGA	Quad cores	Two CPUs	Not available
	R9A09G077M24GBG	729-pin FCBGA	Dual cores	Two CPUs	Not available
	R9A09G077M04GBG	729-pin FCBGA	Single core	Two CPUs	Not available
RZ/N2H	R9A09G087M48GBG	576-pin FCBGA	Quad cores	Two CPUs	Available
	R9A09G087M28GBG	576-pin FCBGA	Dual cores	Two CPUs	Available
	R9A09G087M08GBG	576-pin FCBGA	Single core	Two CPUs	Available
	R9A09G087M44GBG	576-pin FCBGA	Quad cores	Two CPUs	Not available
	R9A09G087M24GBG	576-pin FCBGA	Dual cores	Two CPUs	Not available
	R9A09G087M04GBG	576-pin FCBGA	Single core	Two CPUs	Not available

Note 1. Except for TrustZone. TrustZone is available for all part number.

1.4 Block Diagram

Figure 1.1 shows a block diagram.



GMAC	: Ethernet MAC
ETHSW	: Ethernet switch
ESC	: EtherCAT slave controller
DMAC	: DMA controller
BSC	: Bus state controller
xSPI	: Expanded serial peripheral interface
WDT	: Watchdog timer
SCI	: Serial communication interface
SCIE	: Serial communication interface for encoder
SPI	: Serial peripheral interface
USB	: USB 2.0 HS host/function module
GIC	: Generic interrupt controller
ELC	: Event link controller
MTU3	: Multi-function timer pulse unit 3
SHOSTIF	: Serial host interface
MBXSEM	: Mailbox and Semaphore
SDHI	: SD/eMMC host interface
PCIE	: PCI Express Gen 3
GPT	: General PWM timer
CMT	: Compare match timer
CMTW	: Compare match timer W
POE3	: Port output enable 3
POEG	: Port output enable for GPT
RTC	: Real time clock
IIC	: I^2C bus interface
CANFD	: CANFD module
DSMIF	: ΔΣ interface
TFU	: Arithmetic unit for trigonometric functions
CRC	: CRC (cyclic redundancy check) calculator
CLMA	: Clock monitor circuit
DOC	: Data operation circuit
TSU	: Temperature sensor unit
RSIP	: Renesas Security IP
OTP	: One-time programmable memory
LCDC	: LCD Controller
DDRSS	: LPDDR4 SDRAM memory interface

Figure 1.1 Block diagram

1.5 Pin Functions

Table 1.19 lists the pin functions.

Table 1.19 Pin functions (1 of 10)

Classification	Pin name	I/O	Description
Power supply	VDD33	Input	3.3 V power supply pin for I/O
	VDD1833_0 to VDD1833_7	Input	Power supply pins for each I/O domains. (1.8 V or 3.3 V) VDD1833_0: ETH0 domain VDD1833_1: ETH1 domain VDD1833_2: ETH2 domain VDD1833_3: ETH3 domain VDD1833_4: xSPI0 domain VDD1833_5: xSPI1 domain VDD1833_6: SDHI0 domain VDD1833_7: SDHI1 domain
	VDDP_18_33, VDDP_18_0 to VDDP18_7	Input	1.8 V power supply pins for I/O
	VDD08	Input	0.8 V power supply pin.
	VDD18_PLL0 to VDD18_PLL4	Input	1.8 V power supply pins for PLL
	VDD08_PLL0 to VDD08_PLL4	Input	0.8 V power supply pins for PLL
	VDD33_X	Input	3.3 V power supply pin for Oscillator
	VDDP_18_X	Input	1.8 V power supply pin for Oscillator
	AVDD18A_TSU	Input	1.8 V power supply pin for Thermal sensor
	DVDD08A_TSU	Input	0.8 V power supply pin for Thermal sensor
	OTPVDD18	Input	1.8 V power supply pin for OTP
	OTPVDD08	Input	0.8 V power supply pin for OTP
	VSS	Input	Ground pin. Connect to the system power supply (0 V).
	VSS_PLL0 to VSS_PLL4	Input	Ground pin for PLL. Connect to the system power supply (0 V).
Clock	XTAL	Output	Connected to a crystal resonator. When external clock signal is used, EXTAL pin should be driven low. XTAL pin should never be driven or loaded by anything other than crystal oscillator. The voltage level of EXTAL must not exceed VDD18 (1.8 V).
	EXTAL	Input	
	EXTCLKIN	Input	Inputs the external clock. When a crystal resonator is connected, it should be driven low.
	XTALSEL	Input	Main clock source select pin (low: EXTCLKIN, high: XTAL/EXTAL)
	CKIO	Output	Outputs the external bus clock for external devices.
	ETH0_REFCLK	Output	Outputs 25 MHz clock for EtherPHY 0
	ETH1_REFCLK	Output	Outputs 25 MHz clock for EtherPHY 1
	ETH2_REFCLK	Output	Outputs 25 MHz clock for EtherPHY 2
	ETH3_REFCLK	Output	Outputs 25 MHz clock for EtherPHY 3
	RMII0_REFCLK	Output	Outputs 50 MHz clock for RMII0
	RMII1_REFCLK	Output	Outputs 50 MHz clock for RMII1
	RMII2_REFCLK	Output	Outputs 50 MHz clock for RMII2
	RMII3_REFCLK	Output	Outputs 50 MHz clock for RMII3

Table 1.19 Pin functions (2 of 10)

Classification	Pin name	I/O	Description
Operating mode control	MDX	Input	This signal should be driven low.
	MD0 to MD2	Input	Input the operating mode select signal. The signal level on these pins must not be changed during operation mode transition on release from the reset state.
	MDV	Input	Input the operating voltage select signal for boot peripheral. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
	MDW0, MDW1	Input	Input the ATCM wait cycle select signal. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
	MDD	Input	Input the enabling JTAG authentication by hash signal. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
System control	RES#	Input	Inputs the reset signal. This MPU enters the reset state when this signal goes low.
	BSCANP	Input	Inputs the boundary scan enable signal. Boundary scan is enabled when this pin goes high. When boundary scan is not used, this pin should be driven low.
	RSTOUT#	Output	Outputs the reset signal externally
Debugging interface	TRST#	Input	Test reset pin for the on-chip emulator
	TMS	I/O	Test mode select pin for the on-chip emulator Functions as the SWDIO pin in serial wire debug (SWD) mode
	TDI	Input	Test data input pin for the on-chip emulator
	TDO	Output	Test data output pin for the on-chip emulator
	TCK	Input	Test clock pin for the on-chip emulator Functions as the SWCLK pin in serial wire debug (SWD) mode
Bus state controller (BSC)	A25 to A0	Output	Output the address
	D31 to D0	I/O	Input and output the data
	CS0#, CS2#, CS3#, CS5#	Output	Output the chip select signal for the external memory or device.
	RD#	Output	Outputs the strobe signal which indicates a read is in progress.
	RD/WR#	Output	Outputs the strobe signal which indicates a read or write access
	BS#	Output	Outputs the status signal which indicates the start of the bus cycle
	AH#	Output	Outputs the address hold signal for the device that uses the multiplexed I/O interface
	WAIT#	Input	Inputs the external wait control signal which inserts a wait cycle into the bus cycle
	WE0#	Output	Outputs the write strobe signal to D7 to D0
	WE1#	Output	Outputs the write strobe signal to D15 to D8
	WE2#	Output	Outputs the write strobe signal to D23 to D16
	WE3#	Output	Outputs the write strobe signal to D31 to D24

Table 1.19 Pin functions (3 of 10)

Classification	Pin name	I/O	Description
Direct memory access controller (DMAC)	DREQ	Input	Inputs the DMA transfer request signal from the external device
	DACK	Output	Outputs the acknowledge signal which indicates acceptance of the DMA transfer request from the external device
	TEND	Output	Outputs the DMA transfer end signal
Interrupt	SEI	Input	Input the system error interrupt signal
	IRQ0 to IRQ15	Input	Input the external interrupt request signal
Multi-function timer pulse unit 3 (MTU3)	MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D	I/O	TGRA0 to TGRD0 input capture input, output compare output, and PWM output pins
	MTIOC1A, MTIOC1B	I/O	TGRA1 and TGRB1 input capture input, output compare output, and PWM output pins
	MTIOC2A, MTIOC2B	I/O	TGRA2 and TGRB2 input capture input, output compare output, and PWM output pins
	MTIOC3A, MTIOC3B, MTIOC3C, MTIOC3D	I/O	TGRA3 to TGRD3 input capture input, output compare output, and PWM output pins
	MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D	I/O	TGRA4 to TGRD4 input capture input, output compare output, and PWM output pins
	MTIC5U, MTIC5V, MTIC5W	Input	TGRU5, TGRV5, and TGRW5 input capture input and dead time compensation input pins
	MTIOC6A, MTIOC6B, MTIOC6C, MTIOC6D	I/O	TGRA6 to TGRD6 input capture input/output compare output/PWM output pins
	MTIOC7A, MTIOC7B, MTIOC7C, MTIOC7D	I/O	TGRA7 to TGRD7 input capture input/output compare output/PWM output pins
	MTIOC8A, MTIOC8B, MTIOC8C, MTIOC8D	I/O	TGRA8 to TGRD8 input capture input/output compare output/PWM output pins
	MTCLKA, MTCLKB, MTCLKC, MTCLKD	Input	External clock input pins for MTU3
Port output enable 3 (POE3)	POE0#, POE4#, POE8#, POE10#, POE11#	Input	Input the request signal to place the MTU3 in the high-impedance state

Table 1.19 Pin functions (4 of 10)

Classification	Pin name	I/O	Description
General PWM timer (GPT)/ Port output enable for GPT (POEG)	GTETRGA, GTETRGB, GTETRGC, GTETRGD	Input	External trigger input and output-disable request input pins
	GTETRGSA, GTETRGSB	Input	External trigger input and output-disable request input pins (SAFETY)
	GTIOC00_0A to GTIOC00_4A, GTIOC00_0B to GTIOC00_4B, GTIOC01_0A to GTIOC01_4A, GTIOC01_0B to GTIOC01_4B, GTIOC02_0A to GTIOC02_4A, GTIOC02_0B to GTIOC02_4B, GTIOC03_0A to GTIOC03_4A, GTIOC03_0B to GTIOC03_4B, GTIOC04_0A to GTIOC04_4A, GTIOC04_0B to GTIOC04_4B, GTIOC05_0A to GTIOC05_4A, GTIOC05_0B to GTIOC05_4B, GTIOC06_0A to GTIOC06_4A, GTIOC06_0B to GTIOC06_4B, GTIOC07_0A to GTIOC07_4A, GTIOC07_0B to GTIOC07_4B, GTIOC08_0A to GTIOC08_4A, GTIOC08_0B to GTIOC08_4B, GTIOC09_0A to GTIOC09_6A, GTIOC09_0B to GTIOC09_6B, GTIOC10_0A to GTIOC10_3A, GTIOC10_0B to GTIOC10_3B	I/O	Input capture input/output compare output/PWM output pins
	GTADSM00_0 to GTADSM00_1, GTADSM01_0 to GTADSM01_1, GTADSM02_0 to GTADSM02_1, GTADSM03_0 to GTADSM03_1, GTADSM04_0 to GTADSM04_1, GTADSM05_0 to GTADSM05_1, GTADSM06_0 to GTADSM06_1, GTADSM07_0 to GTADSM07_1, GTADSM08_0 to GTADSM08_1, GTADSM09_0 to GTADSM09_1	Output	Output pins for monitoring A/D conversion start requests
Compare match timer W (CMTW)	CMTW0_TIC0, CMTW0_TIC1, CMTW1_TIC0, CMTW1_TIC1	Input	CMTW input capture input pins
	CMTW0_TOC0, CMTW0_TOC1, CMTW1_TOC0, CMTW1_TOC1	Output	CMTW output compare output pins
Real time clock (RTC)	RTCAT1HZ	Output	RTC 1 Hz output pin

Table 1.19 Pin functions (5 of 10)

Classification	Pin name	I/O	Description
Serial communication interface (SCI)	SCK0 to SCK5	I/O	Clock I/O pins (clock synchronous mode/simple SPI mode/smart card mode)
	RXD0 to RXD5	Input	Input the receive data (asynchronous mode/clock synchronous mode/smart card mode)
	TXD0 to TXD5	Output	Output the transmit data (asynchronous mode/clock synchronous mode/smart card mode)
	CTS0# to CTS5#	Input	Input the start of transmission (asynchronous mode/ clock synchronous mode) active-low
	RTS0# to RTS5#	Output	Output the reception (asynchronous mode/clock synchronous mode) active-low
	SCL0 to SCL5	I/O	Input/output the I2C clocks (simple I2C mode)
	SDA0 to SDA5	I/O	Input/output the I2C data (simple I2C mode)
	MISO0 to MISO5	I/O	Input/output the data for slave transmission (simple SPI mode)
	MOSI0 to MOSI5	I/O	Input/output the data for master transmission (simple SPI mode)
	SS0# to SS5#	Input	Chip-select input pins (simple SPI mode) active-low
	DE0 to DE5	Output	Driver enable output pins (asynchronous mode)
Serial communication interface for Encoder (SCIE)	SCKE00 to SCKE11	Output	Clock I/O pins (clock synchronous mode)
	RXDE00 to RXDE11	Input	Input the receive data (asynchronous mode/clock synchronous mode)
	TXDE00 to TXDE11	Output	Output the transmit data (asynchronous mode/clock synchronous mode)
	DEE00 to DEE11	Output	Driver enable output pins (asynchronous mode)
I^2C bus interface (IIC)	IIC_SCL0 to IIC_SCL2	I/O	Clock I/O pins
	IIC_SDA0 to IIC_SDA2	I/O	Data I/O pins

Table 1.19 Pin functions (6 of 10)

Classification	Pin name	I/O	Description
Ethernet	ETH0_TXCLK to ETH3_TXCLK	I/O	TX clock input pins (MII mode) TX clock output pins (RGMII mode)
	ETH0_TXD0 to ETH3_TXD0	Output	TX data 0 pins (RGMII, RMII, and MII modes)
	ETH0_TXD1 to ETH3_TXD1	Output	TX data 1 pins (RGMII, RMII, and MII modes)
	ETH0_TXD2 to ETH3_TXD2	Output	TX data 2 pins (RGMII and MII modes)
	ETH0_TXD3 to ETH3_TXD3	Output	TX data 3 pins (RGMII and MII modes)
	ETH0_TXEN to ETH3_TXEN	Output	TX data enable pins (RMII and MII modes) TX data enable/TX data error (TX_CTL) pins (RGMII mode)
	ETH0_TXER to ETH3_TXER	Output	TX data error pins (MII mode)
	ETH0_RXCLK to ETH3_RXCLK	Input	RX clock pins (RGMII, RMII, and MII modes)
	ETH0_RXD0 to ETH3_RXD0	Input	RX data 0 pins (RGMII, RMII, and MII modes)
	ETH0_RXD1 to ETH3_RXD1	Input	RX data 1 pins (RGMII, RMII, and MII modes)
	ETH0_RXD2 to ETH3_RXD2	Input	RX data 2 pins (RGMII and MII modes)
	ETH0_RXD3 to ETH3_RXD3	Input	RX data 3 pins (RGMII and MII modes)
	ETH0_RXDV to ETH3_RXDV	Input	RX data valid pins (MII mode) Carrier sense/RX data valid (CRS_DV) pins (RMII mode) RX data valid/RX error (RX_CTL) pins (RGMII mode)
	ETH0_RXER to ETH3_RXER	Input	RX data error pins (RMII and MII modes)
	ETH0_CRS to ETH3_CRS	Input	Carrier sense pins (MII mode)
	ETH0_COL to ETH3_COL	Input	Collision detection pins (MII mode)
Ethernet MAC (GMAC)	GMAC0_PTPTRG0 to GMAC2_PTPTRG0	Input	PTP timer trigger external input 0
	GMAC0_PTPTRG1 to GMAC2_PTPTRG1	Input	PTP timer trigger external input 1
	GMAC0_MDC to GMAC2_MDC	Output	Management data clock output pin
	GMAC0_MDIO to GMAC2_MDIO	I/O	Management data I/O pin
Ethernet switch (ETHSW)	ETHSW_LPI0	Output	Port 0 MAC status indicates that it is currently receiving low-power-idle sequences from the PHY
	ETHSW_LPI1	Output	Port 1 MAC status indicates that it is currently receiving low-power-idle sequences from the PHY
	ETHSW_LPI2	Output	Port 2 MAC status indicates that it is currently receiving low-power-idle sequences from the PHY
	ETHSW_PTPOUT0 to ETHSW_PTPOUT3	Output	Ethernet switch timer pulse output pins
	ETHSW_TDMAOUT0 to ETHSW_TDMAOUT3	Output	Ethernet switch TDMA timer output pins
	ETHSW_PHYLINK0 to ETHSW_PHYLINK2	Input	Ethernet switch PHY link status input pins
	ETHSW_MDC	Output	Management data clock output pin
	ETHSW_MDIO	I/O	Management data I/O pin

Table 1.19 Pin functions (7 of 10)

Classification	Pin name	I/O	Description
EtherCAT slave controller (ESC)	ESC_LEDRUN	Output	Outputs the EtherCAT RUN LED signal
	ESC_IRQ	Output	Outputs the EtherCAT IRQ signal
	ESC_LEDSTER	Output	Outputs the EtherCAT Dual-color state LED signal
	ESC_LEDERR	Output	Outputs the EtherCAT error LED signal
	ESC_LINKACT0 to ESC_LINKACT2	Output	Output the EtherCAT link/activity LED signal
	ESC_SYNC0, ESC_SYNC1	Output	Output the EtherCAT SYNC signal
	ESC_LATCH0, ESC_LATCH1	Input	Input the EtherCAT LATCH signal
	ESC_RESETOUT#	Output	Output the EtherCAT reset signal
	ESC_I2CCLK	Output	Outputs the EtherCAT EEPROM I2C clock signal
	ESC_I2CDATA	I/O	Inputs and outputs the EtherCAT EEPROM I2C data signal
	ESC_PHYLINK0 to ESC_PHYLINK2	Input	Inputs the EtherCAT PHY link status signal.
	ESC_MDC	Output	Management data clock output pin
	ESC_MDIO	I/O	Management data I/O pin
USB 2.0 host/function module	USB_USVDD33	Input	3.3 V power supply input pin for USB
	USB_USVDD18	Input	1.8 V power supply input pin for USB
	USB_USDVDD	Input	0.8 V power supply
	USB_TXRTUNE	Input	USB2 transmitter tune pin. Connect this pin to GND via 200 Ω (±1%).
	USB_QDP	I/O	USB bus D+ data I/O pin
	USB_QDM	I/O	USB bus D- data I/O pin
	USB_VBUSEN	Output	Outputs the VBUS power enable signal for USB
	USB_OVRCUR	Input	Inputs the overcurrent signal for USB
	USB_VUBUSIN	Input	USB cable connection/disconnection detection input pin
	USB_EXICEN	Output	OTG power supply IC control pin
	USB_OTG_ID	Input	OTG ID pin
CANFD module (CANFD)	CANRX0, CANRX1	Input	Receive data input pins
	CANTX0, CANTX1	Output	Transmit data output pins
	CANRXDP0, CANRXDP1	Output	Receive data phase output pins
	CANTXDP0, CANTXDP1	Output	Transmit data phase output pins
Serial peripheral interface (SPI)	SPI_RSPCK0 to SPI_RSPCK3	I/O	Clock I/O pins
	SPI_MOSI0 to SPI_MOSI3	I/O	Master transmit data I/O pins
	SPI_MISO0 to SPI_MISO3	I/O	Slave transmit data I/O pins
	SPI_SSL00 to SPI_SSL30	I/O	Slave select signal I/O pins
	SPI_SSL01 to SPI_SSL31, SPI_SSL02 to SPI_SSL32, SPI_SSL03 to SPI_SSL33	Output	Slave select signal output pins

Table 1.19 Pin functions (8 of 10)

Classification	Pin name	I/O	Description
Expanded serial peripheral interface (xSPI)	XSPI0_CKP, XSPI1_CKP, XSPI0_CKN	Output	Clock output pins
	XSPI0_CS0#, XSPI0_CS1#, XSPI1_CS0#, XSPI1_CS1#	Output	Chip select output pins
	XSPI0_DS, XSPI1_DS	I/O	Read data strobe/write data mask input/output pin
	XSPI0_IO0 to XSPI0_IO7, XSPI1_IO0 to XSPI1_IO7	I/O	Data0 to Data7 input/output pins
	XSPI0_RESET0#, XSPI0_RESET1#	Output	Master reset status output pins
	XSPI0_RSTO0#, XSPI0_RSTO1#	Input	Slave reset status input pins
	XSPI0_INT0#, XSPI0_INT1#	Input	Interrupt input pins
	XSPI0_ECS0#, XSPI0_ECS1#	Input	Error correction status input pins
	XSPI0_WP0#, XSPI0_WP1#	Output	Write protect output pins
ΔΣ interface (DSMIF)	MCLK00 to MCLK90, MCLK01 to MCLK91, MCLK02 to MCLK92	I/O	Clock I/O pins
	MDAT00 to MDAT90, MDAT01 to MDAT91, MDAT02 to MDAT92	Input	Data input pins
12-bit A/D converter (ADC12)	AVDD_ADC0 to AVDD_ADC2	Input	0.8 V power supply pins for ADC
	AVDDIO_ADC0 to AVDDIO_ADC2	Input	1.8 V power supply pins for ADC
	AVSS_ADC0 to AVSS_ADC2	Input	Ground pins for ADC
	AVSSIO_ADC0 to AVSSIO_ADC2	Input	Ground pins for ADC
	AVDDREF_ADC0 to AVDDREF_ADC2	Input	Voltage Reference Analog Supply
	AN000 to AN003, AN100 to AN103, AN200 to AN214	input	Analog input pins for the A/D converter
	ADTRG0# to ADTRG2#	input	External trigger input pins for the start of A/D conversion
Serial host interface (SHOSTIF)	HSPI_CK	Input	Clock input pin
	HSPI_CS#	Input	Chip select input pin
	HSPI_IO0 to HSPI_IO7	I/O	Data0 to Data7 input/output pins
	HSPI_INT#	Output	Interrupt output pin
Mailbox and Semaphore (MBXSEM)	MBX_HINT#	Output	Mailbox (Cortex-A55/Cortex-R52 to Host CPU) interrupt output pin
Encoder I/F common	ENCIFCK00 to ENCIFCK15	Output	Encoder I/F clock output pins
	ENCIFOE00 to ENCIFOE15	Output	Encoder I/F data output enable pins
	ENCIFDO00 to ENCIFDO15	Output	Encoder I/F data output pins
	ENCIFDI00 to ENCIFDI15	Input	Encoder I/F data input pins
EnDat 2.2 (ENDAT)	DUEI00 to DUEI15	Output	EnDat 2.2 Data transfer
	TST_OUT00 to TSTOUT15	Output	EnDat 2.2 Data input after internal synchronization
	SI00# to SI15#	Output	EnDat 2.2 Start pulse

Table 1.19 Pin functions (9 of 10)

Classification	Pin name	I/O	Description
Hiperface DSL (HDSL)	HDSL00_LINK to HDSL15_LINK	Output	HDSL LINK
	HDSL00_SMPL to HDSL15_SMPL	Output	HDSL Test signal line sampler
	HDSL00_CLK1 to HDSL15_CLK1	Input	HDSL SPI clock safe 1
	HDSL00_SEL1 to HDSL15_SEL1	Input	HDSL SPI selection safe 1
	HDSL00_MISO1 to HDSL15_MISO1	Output	HDSL SPI data output safe 1
	HDSL00_MOSI1 to HDSL15_MOSI1	Input	HDSL SPI data input safe 1
	HDSL00_CLK2 to HDSL15_CLK2	Input	HDSL SPI clock safe 2
	HDSL00_SEL2 to HDSL15_SEL2	Input	HDSL SPI selection safe 2
	HDSL00_MISO2 to HDSL15_MISO2	Output	HDSL SPI data output safe 2
	HDSL00_MOSI2 to HDSL15_MOSI2	Input	HDSL SPI data input safe 2
ENCOUT	POUTA	Output	ENCOUT A-phase output pin
	POUTB	Output	ENCOUT B-phase output pin
	POUTZ	Output	ENCOUT Z-phase output pin
LCD Controller (LCDC)	DISP_CLK	Output	Display Parallel Interface pixel clock
	DISP_HSYNC	Output	Display Parallel Interface Horizontal sync pulse
	DISP_VSYNC	Output	Display Parallel Interface Vertical sync pulse
	DISP_DE	Output	Display Parallel Interface data enable
	DISP_DATAR0 to DISP_DATAR7	Output	Display Parallel Interface pixel data output Red
	DISP_DATAG0 to DISP_DATAG7	Output	Display Parallel Interface pixel data output Green
	DISP_DATAB0 to DISP_DATAB7	Output	Display Parallel Interface pixel data output Blue
SD/eMMC Host Interface (SDHI)	SD0_CLK, SD1_CLK	Output	SD/MMC clock output
	SD0_CMD, SD1_CMD	I/O	SD/MMC command output, response input
	SD0_DATA0 to SD0_DATA7, SD1_DATA0 to SD1_DATA3	I/O	SD/MMC Data
	SD0_CD, SD1_CD	Input	SD card detection
	SD0_WP, SD1_WP	Input	SD write protection
	SD0_RST#	Output	MMC reset
	SD0_PWEN, SD1_PWEN	Output	SD power enable
	SD0_IOVS, SD1_IOVS	Output	SD voltage select
PCI Express Gen3 (PCIE)	PCIE_VDD18A_CMN, PCIE_VDD18A_L0, PCIE_VDD18A_L1	Input	1.8 V power supply pins for PCIE
	PCIE_VDD08A_L0, PCIE_VDD08A_L1	Input	0.8 V power supply pins for PCIE
	PCIE_REFCLK_P0, PCIE_REFCLK_N0, PCIE_REFCLK_P1, PCIE_REFCLK_N1	Input	Reference clock input
	PCIE_RXDP_L0, PCIE_RXDN_L0, PCIE_RXDP_L1, PCIE_RXDN_L1	Input	Serial data input
	PCIE_TXDP_L0, PCIE_TXDN_L0, PCIE_TXDP_L1, PCIE_TXDN_L1	Output	Serial data output
	PCIE_RSTOUT0B, PCIE_RSTOUT1B	Output	PCIE Reset output for Root Complex

Table 1.19 Pin functions (10 of 10)

Classification	Pin name	I/O	Description
LPDDR4 SDRAM Subsystem (DDRSS)	DDR_VAA	Input	1.8 V power supply pin for DDRPHY
	DDR_VDDQ	Input	1.1 V power supply pin for DDRPHY
	DDR_ZN	Output	DDRSS calibration external reference resistor
	DDR_DTEST	I/O	DDRSS Digital test point for debug
	DDR_ATEST	I/O	DDRSS Analog test point for debug
	DDR_RESET_N	Output	DDRSS DRAM reset
	DDR_CKA_T, DDR_CKA_C, DDR_CKB_T, DDR_CKB_C	I/O	DDRSS DRAM Ch A/B Clock
	DDR_CKEA[1:0], DDR_CKEB[1:0]	I/O	DDRSS DRAM Ch A/B Clock Enable
	DDR_CSA[1:0], DDR_CSB[1:0]	I/O	DDRSS DRAM Ch A/B Chip Select
	DDR_CAA[5:0], DDR_CAB[5:0]	I/O	DDRSS DRAM Ch A/B Command/Address
	DDR_DQA[15:0], DDR_DQB[15:0]	I/O	DDRSS DRAM Ch A/B Data
	DDR_DMIA[1:0], DDR_DMIB[1:0]	I/O	DDRSS DRAM Ch A/B Data Mask Inversion
	DDR_DQSA_T[1:0], DDR_DQSB_T[1:0]	I/O	DDRSS DRAM Ch A/B Data Strobe (positive)
	DDR_DQSA_C[1:0], DDR_DQSB_C[1:0]	I/O	DDRSS DRAM Ch A/B Data Strobe (negative)
I/O ports	P00_0 to P35_6	I/O	General-purpose input/output pins

1.6 RZ/T2H FCBGA 729 Pin Assignments

Figure 1.2 shows the pin arrangement. Table 1.20 shows the lists of pins and the pin functions.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
A	VSS	P11_4	P11_6	P11_7	P10_1	P10_6	P14_0	P13_6	P14_4	P14_7	P16_2	P16_4	P17_0	P17_5	P19_4	P19_5	P20_1	P20_0	P21_3	P23_1	P23_3	P23_2	P24_6	P25_2	P26_3	P25_1	VSS	A
B	P10_3	P10_2	P10_5	P11_5	P10_4	P12_4	P13_5	P13_4	P14_6	VSS	P16_1	P16_3	P17_1	VSS	P18_2	P18_6	P20_5	VSS	P21_2	P24_3	P23_6	VSS	P26_1	P26_2	P25_0	P25_3	P28_3	B
C	P11_1	P09_5	P11_2	P10_7	P11_0	VSS	P13_0	P13_7	P14_1	P14_2	P14_5	P16_6	P17_3	P19_2	P19_3	P18_7	P20_3	P20_6	P21_4	P23_4	P23_5	P22_6	P24_7	P24_5	P26_4	P26_5	P27_7	C
D	P11_3	P09_4	P09_6	P10_0	P09_7	P12_6	P12_0	VSS	P14_3	P15_6	P16_0	VSS	P18_3	P19_0	P18_0	VSS	P21_7	P20_7	P21_6	VSS	P22_1	P24_1	P25_6	VSS	P26_0	P26_6	P28_5	D
E	DDR_DQA12	DDR_DQA9	VSS	DDR_DQA13	VSS	P12_7	P12_3	P12_2	P15_7	P15_1	P15_5	P17_2	P18_1	P19_1	P17_7	P19_6	P20_2	P21_0	P22_0	P22_4	P22_5	P22_2	P25_4	P25_7	P28_4	P28_1	P28_6	E
F	VSS	DDR_DQA8	DDR_DQA11	DDR_DQA14	DDR_DQSA_C1	P13_3	P13_1	P13_2	P15_4	VSS	P15_2	P16_7	P17_6	VSS	P18_4	P19_7	P20_4	VSS	P24_0	P22_3	P24_2	VSS	P25_5	P28_0	P27_5	VSS	P27_3	F
G	DDR_DQA15	VSS	DDR_DQA10	DDR_DMIA1	DDR_DQSA_T1	DDR_VDDQ	P12_5	P12_1	P15_0	VSS	P15_3	P16_5	P17_4	VSS	P18_5	P21_5	P21_1	VSS	P23_7	P23_0	P22_7	P24_4	P27_0	P27_1	P28_2	P27_2	P27_6	G
H	DDR_DQA6	DDR_DQA5	VSS	VSS	VSS	DDR_VDDQ	VSS	VSS	VDDP_18_33	VSS	VDDP_18_6	VDDP_18_33	VSS	VDDP_18_7	VSS	VDDP_18_33	VSS	VDDP_18_0	VSS	VDDP_18_1	VDD1833_1	P26_7	P28_7	VSS	P29_0	P27_4	P29_3	H
J	DDR_DQA4	VSS	DDR_DMIA0	DDR_DQA7	DDR_DQSA_C0	DDR_VDDQ	VDD33	VSS	VDD1833_6	AVDD18A_TSU	VDD33	VSS	VDD1833_7	VSS	VDD33	VSS	VDD1833_0	VSS	VDDP_18_33	VSS	VDD1833_1	P30_5	P29_6	P30_7	P29_1	P31_1	P30_3	J
K	DDR_DQA2	DDR_DQA0	DDR_DQA1	DDR_DQA3	DDR_DQSA_T0	VSS	VDD33	VSS	VDD1833_6	DVDD08A_TSU	VDD33	VSS	VDD1833_7	VSS	VDD33	VSS	VDD1833_0	VDD33	VDD33	VDDP_18_2	P30_6	VSS	P29_2	P30_4	P30_1	VSS	P30_0	K
L	VSS	DDR_CAA1	VSS	VSS	VSS	DDR_CKEA1	VSS	VDD18_PLL2	VDD08_PLL2	VSS	VDD18_PLL3	VSS_PLL3	VDD08_PLL3	VSS	VSS	VSS	VSS	VSS	VSS	VDD1833_2	VDD1833_2	P29_4	P29_5	P31_0	P29_7	P30_2	P32_6	L
M	DDR_CKA_C	DDR_CAA3	DDR_CAA4	DDR_CSA0	DDR_CSA1	DDR_CKEA0	VSS	VSS_PLL2	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08_PLL0	VSS	VSS	P32_2	P32_1	VSS	P32_0	P31_7	P32_5	M
N	DDR_CKA_T	VSS	DDR_CAA2	VSS	DDR_CAA5	VSS	DDR_VDDQ	DDR_VAA	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS_PLL0	VDD18_PLL0	VSS	P32_4	P32_3	P32_7	P31_4	P31_3	P31_2	N
P	VSS	DDR_CKEB1	DDR_CAB0	DDR_CAA0	DDR_VDDQ	VSS	DDR_RESET_N	DDR_ATEST	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VSS	VSS	VDDP_18_33	VSS	P33_1	P33_0	P31_6	P31_5	P34_4	P
R	DDR_CKB_T	DDR_CKEB0	VSS	VSS	DDR_VDDQ	VSS	DDR_DTEST	DDR_ZN	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VDDP_18_X	VDD33	VSS	P34_1	P34_6	P33_2	P33_6	P33_7	P34_5	R
T	DDR_CKB_C	DDR_CAB1	DDR_VDDQ	DDR_CAB2	DDR_CAB5	DDR_CSB0	VSS	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD33_X	VSS	VDD1833_3	VDDP_18_3	P34_2	VSS	P33_5	P35_1	XTALSEL	T
U	DDR_CAB3	VSS	DDR_CAB4	DDR_DQB0	VSS	DDR_CSB1	VSS	VSS	VSS	VDD08	VSS	VDD08	VSS_PLL1	VSS_PLL4	VSS	VDD08	VSS	VSS	VSS	VSS	VDD1833_3	P35_6	P34_3	P33_4	P33_3	VSS	EXTCLKIN	U
V	DDR_DQB2	DDR_DQB1	VSS	DDR_DQB3	DDR_DQSB_T0	VSS	VSS	VSS	VDD1833_4	VDD33	VDDP_18_4	VDDP_18_5	VDD18_PLL1	VDD18_PLL4	VDD08	VSS	OTPVDD08	VSS	PCIE_VDD08A_L0	VSS	VSS	P35_3	P34_0	P35_5	P35_0	VSS	XTAL	V
W	VSS	DDR_DQB4	DDR_DQB7	DDR_DMIB0	DDR_DQSB_C0	VSS	VSS	VSS	VDD1833_4	VDD33	VDDP_18_33	VDDP_18_33	VDD08_PLL1	VDD08_PLL4	VSS	VSS	OTPVDD18	VSS	PCIE_VDD08A_L0	VSS	VSS	VSS	P34_7	P35_4	P35_2	VSS	EXTAL	W
Y	DDR_DQB6	DDR_VDDQ	DDR_DQB5	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VDD33	VDD33	VDD1833_5	VDD1833_5	VSS	USB_USVDD18	USB_USVDD33	PCIE_VDD08A_L1	PCIE_VDD18A_L1	PCIE_VDD18A_L0	VSS	VSS	VSS	VSS	VSS	VSS	Y
AA	DDR_DQB8	VSS	DDR_DQB15	VSS	DDR_DQSB_T1	VSS	P06_5	P05_1	P05_2	P05_0	VSS	P04_2	P02_2	VSS	VSS	USB_USDVDD	USB_USVDD18	USB_USVDD33	PCIE_VDD08A_L1	PCIE_VDD18A_L1	PCIE_VDD18A_L0	AVSS_ADC0	AVDD_ADC0	AVSSIO_ADC0	AVSSIO_ADC0	AN002	AN000	AA
AB	DDR_DQB14	DDR_DQB9	DDR_DMIB1	DDR_DQB10	DDR_DQSB_C1	BSCANP	P06_3	P06_4	P07_7	P03_4	P03_3	P03_7	P02_1	P01_3	P00_5	USB_USDVDD	VSS	VSS	VSS	VSS	PCIE_VDD18A_CMN	AVSS_ADC1	AVDD_ADC1	AVDDIO_ADC0	AVDDREF_ADC0	AN001	AN003	AB
AC	DDR_DQB12	VSS	DDR_DQB13	DDR_DQB11	VSS	P06_1	P07_6	P07_4	P03_2	P03_1	P04_7	P03_5	P01_7	P02_0	P00_2	VSS	USB_TXRTUNE	VSS	PCIE_REFCLK_N1	PCIE_REFCLK_P0	PCIE_VDD18A_CMN	AVDD_ADC2	AVDDIO_ADC2	AVDDIO_ADC1	AVDDREF_ADC1	AN103	AN100	AC
AD	MDX	P08_6	P08_3	RES#	P08_2	P05_6	P08_0	VSS	P02_6	P04_6	P02_5	VSS	P01_6	P01_2	P00_1	USB_VUBUSIN	USB_OTG_ID	VSS	PCIE_REFCLK_P1	PCIE_REFCLK_N0	VSS	AVSS_ADC2	AVSSIO_ADC2	AVSSIO_ADC1	AVSSIO_ADC1	AN102	AN101	AD
AE	P09_0	P09_2	P08_5	P08_4	P06_0	P06_2	P05_7	P07_3	P02_7	P03_0	P04_0	P02_4	P01_5	P00_7	P00_0	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	AVSSIO_ADC2	AN202	AN201	AE
AF	P09_3	P08_7	TRST#	P07_1	P05_5	VSS	P06_7	P07_0	P04_5	VSS	P04_1	P02_3	P01_0	VSS	P00_4	VSS	USB_QDP	VSS	PCIE_RXDN_L1	PCIE_RXDN_L0	VSS	PCIE_TXDN_L1	PCIE_TXDN_L0	VSS	AVDDREF_ADC2	AN204	AN200	AF
AG	VSS	P09_1	P08_1	P05_3	P07_2	P07_5	P05_4	P06_6	P03_6	P04_3	P04_4	P01_1	P01_4	P00_6	P00_3	VSS	USB_QDM	VSS	PCIE_RXDP_L1	PCIE_RXDP_L0	VSS	PCIE_TXDP_L1	PCIE_TXDP_L0	VSS	AN205	AN203	VSS	AG
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	

Figure 1.2 Pin arrangement (RZ/T2H 729-pin FCBGA) (top view)

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (1 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
A1	—	VSS	—	—	—	—	—	—	—	—	—
A2	VDD33	IRQ7	P11_4	—	—	DE1	MCLK31	—	—	—	HDSL09_SMPL
A3	VDD33	—	P11_6	—	GTIOC05_0A	—	—	—	—	—	TST_OUT00 / HDSL09_SEL1
A4	VDD33	—	P11_7	—	GTIOC05_0B	—	—	—	—	—	SI00# / HDSL09_MISO1

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (2 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
A5	VDD33	IRQ7	P10_1	WAIT#	MTIOC7D / GTIOC04_2B / GTIOC10_2B	SCK0	MDAT72	DISP_DATAR1	—	—	SI14# / HDSL08_LINK
A6	VDD33	IRQ0	P10_6	A3	MTIOC0B / GTIOC05_0A	DE0	MCLK21	DISP_DATAR6	—	—	HDSL08_MOSI1 / POUTA
A7	VDD1833_6	IRQ5	P14_0	A0	GTIOC06_4B	ETHSW_PTPOUT2 / ESC_SYNC0 / DE3	MCLK42	—	—	—	HDSL11_SMPL
A8	VDD1833_6	—	P13_6	D30	GTIOC06_3B / GTIOC04_3A	SS3#/CTS3#/ RTS3# / SPI_SSL23	MCLK41	—	—	—	ENCIFDO13 / TXDE09 / HDSL10_MOSI2
A9	VDD33	—	P14_4	DACK	POE4# / GTIOC06_1B / GTIOC09_1B / GTIOC06_3A / CMTW0_TIC0	ESC_IRQ / SS4#/ CTS4#/RTS4# / SD1_WP	—	DISP_DATAG4	—	MBX_HINT#	ENCIFDO00 / TXDE00 / HDSL11_MOSI1
A10	VDD33	IRQ9	P14_7	—	POE11# / GTIOC09_3A / CMTW0_TOC1	ESC_I2CDATA / IIC_SDA0 / SD0_IOVS	MCLK32	—	—	—	SI02# / HDSL11_MISO2
A11	VDD33	—	P16_2	—	—	SCK5	MDAT51	—	—	—	SI03# / HDSL12_MOSI2
A12	VDD33	IRQ11	P16_4	—	GTETRGSB	ESC_LINKACT1 / TXD5/SDA5/MOSI5	—	—	—	—	TST_OUT04 / HDSL13_SMPL
A13	VDD1833_7	IRQ12	P17_0	—	GTIOC03_1B	SD1_DATA1	—	—	—	—	SI05# / HDSL13_MOSI1
A14	VDD33	—	P17_5	A7 / DACK	GTADSM00_1 / GTETRGC / CMTW1_TOC0	SCK0 / CANTX0 / SD1_WP	—	—	—	—	TST_OUT07 / HDSL14_LINK
A15	VDD33	—	P19_4	—	GTIOC07_2A	—	—	—	—	—	TST_OUT10 / HDSL15_MOSI1
A16	VDD33	—	P19_5	—	GTIOC07_2B	—	—	—	—	—	SI10# / HDSL15_CLK2
A17	VDD1833_0	MDV	P20_1	—	—	ETH0_TXD0	—	—	—	—	—
A18	VDD1833_0	—	P20_0	—	—	ETH0_TXCLK	—	—	—	—	HDSL15_MOSI2
A19	VDD1833_0	—	P21_3	—	—	ETH0_RXDV	—	—	—	—	DUEI13 / HDSL00_CLK2
A20	VDD33	—	P23_1	—	GTIOC06_1A	ESC_IRQ	—	—	—	—	DUEI00 / HDSL02_LINK
A21	VDD33	—	P23_3	—	GTIOC06_2A	ESC_I2CCLK / IIC_SCL0	—	—	—	—	SI00# / HDSL02_CLK1
A22	VDD33	—	P23_2	—	GTIOC06_1B	ESC_RESETOUT#	—	—	—	—	TST_OUT00 / HDSL02_SMPL
A23	VDD1833_1	MD0	P24_6	—	—	ETH1_TXD0	—	—	—	—	—
A24	VDD1833_1	MDW1	P25_2	—	—	ETH1_TXEN	—	—	—	—	—
A25	VDD1833_1	—	P26_3	—	—	ETHSW_PHYLINK1 / ESC_PHYLINK1	—	—	—	—	HDSL04_SMPL
A26	VDD1833_1	MDW0	P25_1	—	—	ETH1_TXD3 / CANTXDP0	—	—	—	—	—
A27	—	VSS	—	—	—	—	—	—	—	—	—
B1	VDD33	IRQ2	P10_3	RD#	MTCLKD / MTIOC2B / GTIOC04_3B / GTIOC10_3B	TXD0/SDA0/MOSI0	MDAT10 / MDAT00	DISP_DATAR3	—	—	ENCIFOE04 / DEE04 / HDSL08_CLK1
B2	VDD33	IRQ1	P10_2	CS0#	MTCLKC / MTIOC2A / GTIOC04_3A / GTIOC10_3A	RXD0/SCL0/MISO0	MCLK10 / MCLK00	DISP_DATAR2	—	—	ENCIFCK04 / SCKE04 / HDSL08_SMPL
B3	VDD33	—	P10_5	A2	MTIOC1B / MTIOC0A / GTIOC04_4B	CTS0#	MDAT11 / MDAT01	DISP_DATAR5	—	—	ENCIFDI04 / RXDE04 / HDSL08_MISO1
B4	VDD33	—	P11_5	—	—	—	MDAT31	—	—	—	DUEI00 / HDSL09_CLK1

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (3 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
B5	VDD33	IRQ3	P10_4	A1	MTIOC1A / GTIOC04_4A	SS0#/CTS0#/RTS0#	MCLK11 / MCLK01	DISP_DATAR4	—	—	ENCIFDO04 / TXDE04 / HDSL08_SEL1
B6	VDD1833_6	IRQ1	P12_4	D20	GTIOC05_3A / CMTW1_TIC0	RXD2/SCL2/MISO2 / SD0_DATA2	MCLK02	—	—	—	ENCIFCK05 / SCKE05 / HDSL09_MOSI2
B7	VDD1833_6	IRQ4	P13_5	D29	GTIOC06_3A	TXD3/SDA3/MOSI3 / SPI_SSL32	MDAT40	—	—	—	ENCIFOE13 / DEE09 / HDSL10_MISO2
B8	VDD1833_6	—	P13_4	D28	GTIOC03_3B	RXD3/SCL3/MISO3 / SPI_SSL31	MCLK40	—	—	—	ENCIFCK13 / SCKE09 / HDSL10_SEL2
B9	VDD33	IRQ8	P14_6	—	POE10# / GTIOC06_2B / GTIOC09_2B / CMTW0_TIC1	ESC_I2CCLK / DE4 / IIC_SCL0 / SD0_PWEN	—	DISP_DATAG6	—	—	TST_OUT02 / HDSL11_SEL2
B10	—	VSS	—	—	—	—	—	—	—	—	—
B11	VDD33	—	P16_1	—	—	DE5	MCLK51	—	—	—	TST_OUT03 / HDSL12_MISO2
B12	VDD33	IRQ10	P16_3	—	GTETRGSA	ESC_LINKACT0 / RXD5/SCL5/MISO5	—	—	—	—	DUEI04 / HDSL13_LINK
B13	VDD1833_7	IRQ13	P17_1	—	GTIOC03_2A	SD1_DATA2	—	—	—	—	DUEI06 / HDSL13_CLK2
B14	—	VSS	—	—	—	—	—	—	—	—	—
B15	VDD33	SEI	P18_2	A10	GTADSM03_0 / GTIOC07_3B	ETH1_CRS / GMAC1_MDC / SCK1 / CANRX0 / SD1_PWEN	MCLK10	DISP_DATAB3	—	—	HDSL14_MOSI1
B16	VDD33	IRQ3	P18_6	A14	GTIOC07_4A / GTADSM05_0	CTS1# / CANRXDP1	MCLK12	DISP_DATAB7	—	—	ENCIFDO13 / ENCIFDO14 / TXDE09 / TXDE10 / HDSL14_MOSI2
B17	VDD1833_0	—	P20_5	—	—	ETH0_TXEN	—	—	—	—	DUEI11 / HDSL00_LINK
B18	—	VSS	—	—	—	—	—	—	—	—	—
B19	VDD1833_0	—	P21_2	—	—	ETH0_RXD3 / CANTXDP0	—	—	—	—	SI12# / HDSL00_MOSI1
B20	VDD33	IRQ14	P24_3	—	—	ESC_I2CCLK / IIC_SCL1 / CANRX0	MCLK70	—	—	—	HDSL03_LINK
B21	VDD33	—	P23_6	—	—	ETHSW_LPI0	MDAT60	—	—	—	SI01# / HDSL02_MOSI1
B22	—	VSS	—	—	—	—	—	—	—	—	—
B23	VDD1833_1	—	P26_1	—	—	GMAC1_MDC / ETHSW_MDC / ESC_MDC / CANRXDP1	—	—	—	—	HDSL03_MOSI2
B24	VDD1833_1	—	P26_2	—	—	GMAC1_MDIO / ETHSW_MDIO / ESC_MDIO / CANTXDP1	—	—	—	—	HDSL04_LINK
B25	VDD1833_1	MD2	P25_0	—	—	ETH1_TXD2 / CANRXDP0	—	—	—	—	—
B26	VDD1833_1	—	P25_3	—	—	ETH1_RXCLK	—	—	—	—	DUEI03 / HDSL03_SEL1
B27	VDD33	—	P28_3	—	GTIOC08_2A	SPI_SSL11	—	—	—	—	TST_OUT06 / HDSL05_CLK2
C1	VDD33	IRQ4	P11_1	—	—	ESC_LEDRUN / TXD1/SDA1/MOSI1	MDAT22	—	—	—	DUEI15 / HDSL08_MISO2
C2	VDD33	—	P09_5	D14	MTIOC6D / GTIOC04_0B / GTIOC10_0B	—	MDAT70	DISP_HSYNC	—	—	TST_OUT13 / HDSL07_CLK2
C3	VDD33	IRQ5	P11_2	—	—	SS1#/CTS1#/RTS1#	MCLK30	—	—	—	TST_OUT15 / HDSL08_MOSI2

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (4 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
C4	VDD33	IRQ9	P10_7	A4	MTIC5U / GTIOC05_0B / GTIOC00_3A	SCK1	MDAT21	DISP_DATAR7	—	—	HDSL08_CLK2 / POUTB
C5	VDD33	IRQ13	P11_0	A5	GTIOC00_3B	ESC_RESETOUT# / RXD1 / SCL1 / MISO1	MCLK22	DISP_DATAG0	—	—	HDSL08_SEL2 / POUTZ
C6	—	VSS	—	—	—	—	—	—	—	—	—
C7	VDD1833_6	—	P13_0	D24	GTIOC02_3A	DE2 / SPI_RSPCK3 / SD0_DATA6	MCLK00	—	—	—	ENCIFCK12 / ENCIFCK03 / SCKE08 / SCKE03 / HDSL10_SEL1
C8	VDD1833_6	IRQ14	P13_7	D31	GTIOC06_4A / GTIOC04_3B	CTS3#	MDAT41	—	—	—	ENCIFDI13 / RXDE09 / HDSL11_LINK
C9	VDD33	—	P14_1	RD/WR#	GTIOC06_0A / GTIOC09_0A / GTIOC05_3A / RTCAT1HZ	SCK4 / SD0_CD	MDAT42	DISP_DATAG1	—	—	DUEI02 / HDSL11_CLK1
C10	VDD33	—	P14_2	BS#	GTIOC06_0B / GTIOC09_0B / GTIOC05_3B	RXD4/SCL4/MISO4 / SD0_WP	—	DISP_DATAG2	—	—	ENCIFCK00 / SCKE00 / HDSL11_SEL1
C11	VDD33	—	P14_5	TEND	POE8# / GTIOC06_2A / GTIOC09_2A / GTIOC06_3B / CMTW0_TOC0	ESC_RESETOUT# / CTS4#	—	DISP_DATAG5	—	—	ENCIFDI00 / RXDE00 / HDSL11_CLK2
C12	VDD1833_7	—	P16_6	—	GTIOC03_0B	SD1_CMD	—	—	—	—	DUEI05 / HDSL13_SEL1
C13	VDD1833_7	IRQ15	P17_3	—	GTETRGA	—	—	—	—	—	SI06# / HDSL13_MISO2
C14	VDD33	—	P19_2	—	GTIOC07_1A	—	—	—	—	—	SI09# / HDSL15_SEL1
C15	VDD33	—	P19_3	—	GTIOC07_1B	—	—	—	—	—	DUEI10 / HDSL15_MISO1
C16	VDD33	IRQ4	P18_7	A15	GTIOC07_4B / GTADSM05_1	ETHSW_PTPOUT3 / ESC_SYNC1 / DE1 / CANTXDP1	MDAT12	—	—	—	ENCIFDI13 / ENCIFDI14 / RXDE09 / RXDE10 / HDSL15_LINK
C17	VDD1833_0	—	P20_3	—	—	ETH0_TXD2 / CANRX0	—	—	—	—	—
C18	VDD1833_0	—	P20_6	—	—	ETH0_RXCLK	—	—	—	—	TST_OUT11 / HDSL00_SMPL
C19	VDD1833_0	—	P21_4	—	—	GMAC0_MDC / ETHSW_MDC / ESC_MDC / CANRX1	—	—	—	—	TST_OUT13 / HDSL00_SEL2
C20	VDD33	—	P23_4	—	GTIOC06_2B	ESC_I2CDATA / IIC_SDA0	—	—	—	—	DUEI01 / HDSL02_SEL1
C21	VDD33	—	P23_5	—	—	ESC_LINKACT2	MCLK60	—	—	—	TST_OUT01 / HDSL02_MISO1
C22	VDD33	IRQ8	P22_6	A19	GTETRGSB	GMAC0_PTPTRG1 / ESC_LATCH1 / DE5 / CANTX1 / SD0_WP	—	—	—	—	DUEI15 / HDSL01_SEL2
C23	VDD1833_1	MD1	P24_7	—	—	ETH1_TXD1	—	—	—	—	—
C24	VDD1833_1	—	P24_5	—	—	ETH1_TXCLK	—	—	—	—	HDSL03_CLK1
C25	VDD1833_1	ETH1_REFCLK / RMII1_REFCLK	P26_4	—	—	—	—	—	—	—	—
C26	VDD1833_1	IRQ12	P26_5	—	—	CANTX0	—	—	—	—	ENCIFCK01 / SCKE01 / HDSL04_CLK1
C27	VDD33	IRQ4	P27_7	—	GTIOC08_0A	ETHSW_TDMAOUT0 / SPI_RSPCK1	—	—	—	—	DUEI05 / HDSL05_CLK1

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (5 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
D1	VDD33	IRQ6	P11_3	—	—	CTS1#	MDAT30	—	—	—	SI15# / HDSL09_LINK
D2	VDD33	—	P09_4	D13	MTIOC6B / GTIOC04_0A / GTIOC10_0A	—	MCLK70	DISP_CLK	—	—	DUEI13 / HDSL07_MOSI1
D3	VDD33	—	P09_6	D15	MTIOC7A / GTIOC04_1A / GTIOC10_1A	—	MCLK71	DISP_VSYNC	—	—	SI13# / HDSL07_SEL2
D4	VDD33	IRQ4	P10_0	WE1#	MTIOC7B / GTIOC04_2A / GTIOC10_2A	—	MCLK72	DISP_DATAR0	—	—	TST_OUT14 / HDSL07_MOSI2
D5	VDD33	—	P09_7	WE0#	MTIOC7C / GTIOC04_1B / GTIOC10_1B	—	MDAT71	DISP_DE	—	—	DUEI14 / HDSL07_MISO2
D6	VDD1833_6	—	P12_6	D22	GTIOC05_4A / GTIOC01_3B / CMTW1_TIC1	SS2#/CTS2#/ RTS2# / SD0_DATA4	MCLK10	—	—	—	ENCIFDO05 / TXDE05 / HDSL10_SMPL
D7	VDD1833_6	—	P12_0	D16	MTIC5V / GTIOC05_1A / CMTW0_TIC0	CANRX1 / SD0_CLK	—	—	—	—	DUEI01 / HDSL09_MOSI1
D8	—	VSS	—	—	—	—	—	—	—	—	—
D9	VDD33	IRQ6	P14_3	DREQ	POE0# / GTIOC06_1A / GTIOC09_1A	ESC_LINKACT2 / TXD4/SDA4/MOSI4 / SD1_CD	—	DISP_DATAG3	—	—	ENCIFOE00 / DEE00 / HDSL11_MISO1
D10	VDD33	IRQ1	P15_6	—	GTIOC09_6B	—	MDAT42	—	—	—	ENCIFDO07 / TXDE07 / HDSL12_MOSI1
D11	VDD33	IRQ2	P16_0	—	—	CTS5#	MDAT50	—	—	—	TXDE07 / DUEI03 / HDSL12_SEL2
D12	—	VSS	—	—	—	—	—	—	—	—	—
D13	VDD33	IRQ0	P18_3	A11	GTADSM03_1 / RTCAT1HZ	ETH1_COL / GMAC1_MDIO / RXD1/SCL1/MISO1 / CANTX0 / SD1_IOVS	MDAT10	DISP_DATAB4	—	—	HDSL14_CLK2
D14	VDD33	—	P19_0	—	GTIOC07_0A	—	—	—	—	—	DUEI09 / HDSL15_SMPL
D15	VDD33	IRQ7	P18_0	A8 / TEND	GTADSM02_0	ESC_LEDRUN / SS0#/CTS0#/ RTS0# / CANRXDP0 / SD1_PWEN	—	DISP_DATAB1	—	—	TST_OUT08 / HDSL14_SEL1
D16	—	VSS	—	—	—	—	—	—	—	—	—
D17	VDD1833_0	ETH0_REFCLK / RMII0_REFCLK	P21_7	—	—	CANTXDP1	—	—	—	—	HDSL01_LINK
D18	VDD1833_0	—	P20_7	—	—	ETH0_RXD0	—	—	—	—	SI11# / HDSL00_CLK1
D19	VDD1833_0	—	P21_6	—	—	ETHSW_PHYLINK0 / ESC_PHYLINK0 / CANRXDP1	—	—	—	—	HDSL00_MOSI2
D20	—	VSS	—	—	—	—	—	—	—	—	—
D21	VDD33	—	P22_1	—	GTETRGA	ETH0_TXER / TXD5/ SDA5/MOSI5 / CANTX0	—	—	—	—	HDSL01_CLK1
D22	VDD33	IRQ12	P24_1	—	—	—	MCLK62	—	—	—	SI02# / HDSL02_MISO2
D23	VDD1833_1	—	P25_6	—	—	ETH1_RXD2 / CANRX1	—	—	—	—	DUEI04 / HDSL03_CLK2
D24	—	VSS	—	—	—	—	—	—	—	—	—
D25	VDD1833_1	—	P26_0	—	—	ETH1_RXDV	—	—	—	—	SI04# / HDSL03_MISO2

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (6 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
D26	VDD33	SEI	P26_6	CS2#	—	ETH1_TXER / ESC_RESETOUT# / CANRX0	—	—	—	—	ENCIFOE01 / DEE01 / HDSL04_SEL1
D27	VDD33	—	P28_5	—	—	CANRX0 / SPI_SSL13	MCLK71	—	—	—	ENCIFCK08 / ENCIFCK00 / SCKE08 / SCKE00 / HDSL05_MISO2
E1	—	—	—	—	—	—	—	—	DDR_DQA12	—	—
E2	—	—	—	—	—	—	—	—	DDR_DQA9	—	—
E3	—	VSS	—	—	—	—	—	—	—	—	—
E4	—	—	—	—	—	—	—	—	DDR_DQA13	—	—
E5	—	VSS	—	—	—	—	—	—	—	—	—
E6	VDD1833_6	IRQ2	P12_7	D23	GTIOC05_4B / CMTW1_TOC1	CTS2# / SD0_DATA5	MDAT10	—	—	—	ENCIFDI05 / RXDE05 / HDSL10_CLK1
E7	VDD1833_6	—	P12_3	D19	GTIOC05_2B / CMTW0_TOC1	SCK2 / CANTXDP1 / SD0_DATA1	—	—	—	—	HDSL09_MISO2
E8	VDD1833_6	—	P12_2	D18	GTIOC05_2A / CMTW0_TIC1	CANRXDP1 / SD0_DATA0	—	—	—	—	SI01# / HDSL09_SEL2
E9	VDD33	—	P15_7	—	—	SS5#/CTS5#/RTS5#	MCLK50	—	—	—	ENCIFDI07 / RXDE07 / HDSL12_CLK2
E10	VDD33	—	P15_1	—	GTIOC09_4A	—	MCLK40	—	—	—	ENCIFOE06 / DEE06 / HDSL12_LINK
E11	VDD33	IRQ0	P15_5	—	GTIOC09_6A	—	MCLK42	—	—	—	ENCIFOE07 / DEE07 / HDSL12_MISO1
E12	VDD1833_7	IRQ14	P17_2	—	GTIOC03_2B	SD1_DATA3	—	—	—	—	TST_OUT06 / HDSL13_SEL2
E13	VDD33	IRQ15	P18_1	A9	GTADSM02_1 / GTIOC07_3A	ESC_LEDERR / CTS0# / CANTXDP0 / SD1_IOVS	—	DISP_DATAB2	—	—	SI08# / HDSL14_MISO1
E14	VDD33	—	P19_1	—	GTIOC07_0B	—	—	—	—	—	TST_OUT09 / HDSL15_CLK1
E15	VDD33	—	P17_7	WE3#/AH#	GTADSM01_1 / CMTW1_TOC1	ETHSW_PTPOUT1 / ESC_SYNC1 / TXD0/SDA0/MOSI0 / SD1_IOVS	—	DISP_DATAB0	—	—	DUEI08 / HDSL14_CLK1
E16	VDD33	—	P19_6	—	—	—	MCLK52	—	—	—	HDSL15_SEL2
E17	VDD1833_0	—	P20_2	—	—	ETH0_TXD1	—	—	—	—	—
E18	VDD1833_0	—	P21_0	—	—	ETH0_RXD1	—	—	—	—	DUEI12 / HDSL00_SEL1
E19	VDD1833_0	IRQ11	P22_0	—	—	—	—	—	—	—	HDSL01_SMPL
E20	VDD33	IRQ6	P22_4	A21	GTETRGD	ETH0_COL / SS5#/ CTS5#/RTS5# / CANTXDP0	—	—	—	—	TST_OUT14 / HDSL01_MOSI1
E21	VDD33	IRQ7	P22_5	A20	GTETRGSA	GMAC0_PTPTRG0 / ESC_LATCH0 / CTS5# / CANRX1 / SD0_CD	—	—	—	—	SI14# / HDSL01_CLK2
E22	VDD33	—	P22_2	A23	GTETRGB	ETH0_RXER / RXD5/SCL5/MISO5 / CANRX0	—	—	—	—	HDSL01_SEL1
E23	VDD1833_1	—	P25_4	—	—	ETH1_RXD0	—	—	—	—	TST_OUT03 / HDSL03_MISO1
E24	VDD1833_1	—	P25_7	—	—	ETH1_RXD3 / CANTX1	—	—	—	—	TST_OUT04 / HDSL03_SEL2
E25	VDD33	—	P28_4	—	GTIOC08_2B	SPI_SSL12	—	—	—	—	SI06# / HDSL05_SEL2

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (7 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
E26	VDD33	IRQ6	P28_1	—	GTIOC08_1A	ETHSW_TDMAOUT 2 / SPI_MISO1	—	—	—	—	SI05# / HDSL05_MISO1
E27	VDD33	—	P28_6	—	—	CANTX0	MDAT71	—	—	—	ENCIFOE08 / ENCIFOE00 / DEE08 / DEE00 / HDSL05_MOSI2
F1	—	VSS	—	—	—	—	—	—	—	—	—
F2	—	—	—	—	—	—	—	—	DDR_DQA8	—	—
F3	—	—	—	—	—	—	—	—	DDR_DQA11	—	—
F4	—	—	—	—	—	—	—	—	DDR_DQA14	—	—
F5	—	—	—	—	—	—	—	—	DDR_DQSA_C 1	—	—
F6	VDD1833_6	—	P13_3	D27	GTIOC03_3A	SCK3 / SPI_SSL30	MDAT01	—	—	—	ENCIFDI12 / ENCIFDI03 / RXDE08 / RXDE03 / HDSL10_CLK2
F7	VDD1833_6	—	P13_1	D25	GTIOC02_3B	SPI_MOSI3 / SD0_DATA7	MDAT00	—	—	—	ENCIFOE12 / ENCIFOE03 / DEE08 / DEE03 / HDSL10_MISO1
F8	VDD1833_6	IRQ3	P13_2	D26	—	SPI_MISO3 / SD0_RST#	MCLK01	—	—	—	ENCIFDO12 / ENCIFDO03 / TXDE08 / TXDE03 / HDSL10_MOSI1
F9	VDD33	—	P15_4	—	GTIOC09_5B	—	MDAT41	—	—	—	ENCIFCK07 / SCKE07 / HDSL12_SEL1
F10	—	VSS	—	—	—	—	—	—	—	—	—
F11	VDD33	—	P15_2	—	GTIOC09_4B	—	MDAT40	—	—	—	ENCIFDO06 / TXDE06 / HDSL12_SMPL
F12	VDD1833_7	—	P16_7	—	GTIOC03_1A	SD1_DATA0	—	—	—	—	TST_OUT05 / HDSL13_MISO1
F13	VDD33	—	P17_6	WE2#	GTADSM01_0 / GTETRGD / CMTW1_TIC1	ETHSW_PTPOUT0 / ESC_SYNC0 / RXD0/SCL0/MISO0 / SD1_PWEN	—	DISP_DATAG7	—	—	SI07# / HDSL14_SMPL
F14	—	VSS	—	—	—	—	—	—	—	—	—
F15	VDD33	IRQ1	P18_4	A12	GTIOC07_3A / GTADSM04_0	ESC_LEDSTER / TXD1/SDA1/MOSI1 / CANRX1	MCLK11	DISP_DATAB5	—	—	ENCIFCK13 / ENCIFCK14 / SCKE09 / SCKE10 / HDSL14_SEL2
F16	VDD33	—	P19_7	—	—	—	MDAT52	—	—	—	HDSL15_MISO2
F17	VDD1833_0	—	P20_4	—	—	ETH0_TXD3 / CANTX0	—	—	—	—	—
F18	—	VSS	—	—	—	—	—	—	—	—	—
F19	VDD33	IRQ11	P24_0	—	—	ETHSW_LPI2	MDAT61	—	—	—	TST_OUT02 / HDSL02_SEL2
F20	VDD33	IRQ5	P22_3	A22	GTETRGC	ETH0_CRS / SCK5 / CANRXDP0	—	—	—	—	DUEI14 / HDSL01_MISO1
F21	VDD33	IRQ13	P24_2	—	—	—	MDAT62	—	—	—	HDSL02_MOSI2
F22	—	VSS	—	—	—	—	—	—	—	—	—
F23	VDD1833_1	—	P25_5	—	—	ETH1_RXD1	—	—	—	—	SI03# / HDSL03_MOSI1
F24	VDD33	IRQ5	P28_0	—	GTIOC08_0B	ETHSW_TDMAOUT 1 / SPI_MOSI1	—	—	—	—	TST_OUT05 / HDSL05_SEL1
F25	VDD33	—	P27_5	—	MTIOC1A / GTIOC08_4A / GTIOC02_2A	TXD0/SDA0/MOSI0 / SPI_SSL00	—	—	—	HSPI_IO3	ENCIFDO14 / TXDE10 / HDSL05_LINK

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (8 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
F26	—	VSS	—	—	—	—	—	—	—	—	—
F27	VDD33	—	P27_3	—	MTIOC2A / GTIOC08_3A / GTIOC02_1A	GMAC1_PTPTRG1 / SCK0 / CANRXDP1 / SPI_MOSI0	—	—	—	HSPI_IO1	ENCIFCK14 / SCKE10 / HDSL04_MISO2
G1	—	—	—	—	—	—	—	—	DDR_DQA15	—	—
G2	—	VSS	—	—	—	—	—	—	—	—	—
G3	—	—	—	—	—	—	—	—	DDR_DQA10	—	—
G4	—	—	—	—	—	—	—	—	DDR_DMIA1	—	—
G5	—	—	—	—	—	—	—	—	DDR_DQSA_T 1	—	—
G6	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
G7	VDD1833_6	—	P12_5	D21	GTIOC05_3B / GTIOC01_3A / CMTW1_TOC 0	TXD2/SDA2/MOSI2 / SD0_DATA3	MDAT02	—	—	—	ENCIFOE05 / DEE05 / HDSL10_LINK
G8	VDD1833_6	—	P12_1	D17	MTIC5W / GTIOC05_1B / CMTW0_TOC 0	CANTX1 / SD0_CMD	—	—	—	—	TST_OUT01 / HDSL09_CLK2
G9	VDD33	—	P15_0	—	GTIOC09_3B	—	MDAT32	—	—	—	ENCIFCK06 / SCKE06 / HDSL11_MOSI2
G10	—	VSS	—	—	—	—	—	—	—	—	—
G11	VDD33	—	P15_3	—	GTIOC09_5A	—	MCLK41	—	—	—	ENCIFDI06 / RXDE06 / HDSL12_CLK1
G12	VDD1833_7	—	P16_5	—	GTIOC03_0A	SD1_CLK	—	—	—	—	SI04# / HDSL13_CLK1
G13	VDD33	—	P17_4	A6 / DREQ	GTADSM00_0 / GTETRGB / CMTW1_TIC0	DE0 / CANRX0 / SD1_CD	—	—	—	—	DUEI07 / HDSL13_MOSI2
G14	—	VSS	—	—	—	—	—	—	—	—	—
G15	VDD33	IRQ2	P18_5	A13	GTIOC07_3B / GTADSM04_1	SS1#/CTS1#/ RTS1# / CANTX1	MDAT11	DISP_DATAB6	—	—	ENCIFOE13 / ENCIFOE14 / DEE09 / DEE10 / HDSL14_MISO2
G16	VDD1833_0	—	P21_5	—	—	GMAC0_MDIO / ETHSW_MDIO / ESC_MDIO / CANTX1	—	—	—	—	SI13# / HDSL00_MISO2
G17	VDD1833_0	—	P21_1	—	—	ETH0_RXD2 / CANRXDP0	—	—	—	—	TST_OUT12 / HDSL00_MISO1
G18	—	VSS	—	—	—	—	—	—	—	—	—
G19	VDD33	—	P23_7	—	—	ETHSW_LPI1	MCLK61	—	—	—	DUEI02 / HDSL02_CLK2
G20	VDD33	IRQ10	P23_0	A17	GTIOC06_0B	ETH1_COL / ETHSW_TDMAOUT 3 / ESC_LINKACT1 / CANTXDP1	—	—	—	—	SI15# / HDSL01_MOSI2
G21	VDD33	IRQ9	P22_7	A18	GTIOC06_0A	ETH1_CRS / ETHSW_TDMAOUT 2 / ESC_LINKACT0 / CANRXDP1	—	—	—	—	TST_OUT15 / HDSL01_MISO2
G22	VDD33	IRQ15	P24_4	—	—	ESC_I2CDATA / IIC_SDA1 / CANTX0	MDAT70	—	—	—	HDSL03_SMPL
G23	VDD33	IRQ1	P27_0	CS5#	—	ETH1_CRS / CANTXDP0 / SPI_SSL02	—	—	—	HSPI_INT#	ENCIFDI01 / RXDE01 / HDSL04_MOSI1
G24	VDD33	IRQ2	P27_1	—	GTIOC02_0A	ETH1_COL / CANRX1 / SPI_SSL03	—	—	—	HSPI_CS#	HDSL04_CLK2
G25	VDD33	IRQ7	P28_2	—	GTIOC08_1B	ETHSW_TDMAOUT 3 / SPI_SSL10	—	—	—	—	DUEI06 / HDSL05_MOSI1

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (9 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
G26	VDD33	IRQ3	P27_2	—	GTIOC02_0B	GMAC1_PTPTRG0 / ESC_LEDERR / CANTX1 / SPI_RSPCK0	—	—	—	HSPI_IO0	HDSL04_SEL2
G27	VDD33	—	P27_6	—	MTIOC1B / GTIOC08_4B / GTIOC02_2B	—	—	—	—	HSPI_CK	ENCIFDI14 / RXDE10 / HDSL05_SMPL
H1	—	—	—	—	—	—	—	—	DDR_DQA6	—	—
H2	—	—	—	—	—	—	—	—	DDR_DQA5	—	—
H3	—	VSS	—	—	—	—	—	—	—	—	—
H4	—	VSS	—	—	—	—	—	—	—	—	—
H5	—	VSS	—	—	—	—	—	—	—	—	—
H6	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
H7	—	VSS	—	—	—	—	—	—	—	—	—
H8	—	VSS	—	—	—	—	—	—	—	—	—
H9	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
H10	—	VSS	—	—	—	—	—	—	—	—	—
H11	—	VDDP_18_6	—	—	—	—	—	—	—	—	—
H12	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
H13	—	VSS	—	—	—	—	—	—	—	—	—
H14	—	VDDP_18_7	—	—	—	—	—	—	—	—	—
H15	—	VSS	—	—	—	—	—	—	—	—	—
H16	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
H17	—	VSS	—	—	—	—	—	—	—	—	—
H18	—	VDDP_18_0	—	—	—	—	—	—	—	—	—
H19	—	VSS	—	—	—	—	—	—	—	—	—
H20	—	VDDP_18_1	—	—	—	—	—	—	—	—	—
H21	—	VDD1833_1	—	—	—	—	—	—	—	—	—
H22	VDD33	IRQ0	P26_7	CS3#	—	ETH1_RXER / ESC_LEDSTER / CANRXDP0 / SPI_SSL01	—	—	—	—	ENCIFDO01 / TXDE01 / HDSL04_MISO1
H23	VDD33	—	P28_7	—	—	CANRXDP0	MCLK72	—	—	—	ENCIFDO08 / ENCIFDO00 / TXDE08 / TXDE00 / HDSL06_LINK
H24	—	VSS	—	—	—	—	—	—	—	—	—
H25	VDD33	—	P29_0	—	—	CANTXDP0	MDAT72	—	—	—	ENCIFDI08 / ENCIFDI00 / RXDE08 / RXDE00 / HDSL06_SMPL
H26	VDD33	—	P27_4	—	MTIOC2B / GTIOC08_3B / GTIOC02_1B	RXD0/SCL0/MISO0 / CANTXDP1 / SPI_MISO0	—	—	—	HSPI_IO2	ENCIFOE14 / DEE10 / HDSL04_MOSI2
H27	VDD1833_2	—	P29_3	—	GTIOC09_1A	ETH2_TXD1	—	—	—	—	ENCIFDO09 / TXDE09 / HDSL06_MISO1
J1	—	—	—	—	—	—	—	—	DDR_DQA4	—	—
J2	—	VSS	—	—	—	—	—	—	—	—	—
J3	—	—	—	—	—	—	—	—	DDR_DMIA0	—	—
J4	—	—	—	—	—	—	—	—	DDR_DQA7	—	—
J5	—	—	—	—	—	—	—	—	DDR_DQSA_C 0	—	—
J6	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
J7	—	VDD33	—	—	—	—	—	—	—	—	—

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (10 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
J8	—	VSS	—	—	—	—	—	—	—	—	—
J9	—	VDD1833_6	—	—	—	—	—	—	—	—	—
J10	—	AVDD18A_TSU	—	—	—	—	—	—	—	—	—
J11	—	VDD33	—	—	—	—	—	—	—	—	—
J12	—	VSS	—	—	—	—	—	—	—	—	—
J13	—	VDD1833_7	—	—	—	—	—	—	—	—	—
J14	—	VSS	—	—	—	—	—	—	—	—	—
J15	—	VDD33	—	—	—	—	—	—	—	—	—
J16	—	VSS	—	—	—	—	—	—	—	—	—
J17	—	VDD1833_0	—	—	—	—	—	—	—	—	—
J18	—	VSS	—	—	—	—	—	—	—	—	—
J19	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
J20	—	VSS	—	—	—	—	—	—	—	—	—
J21	—	VDD1833_1	—	—	—	—	—	—	—	—	—
J22	VDD1833_2	—	P30_5	—	GTIOC09_6A	GMAC2_MDC / ETHSW_MDC / ESC_MDC / SPI_RSPCK3	—	—	—	—	DUEI07 / HDSL07_MISO1
J23	VDD1833_2	—	P29_6	—	GTIOC09_2B	ETH2_TXEN / SPI_SSL22	—	—	—	—	ENCIFOE10 / DEE10 / HDSL06_SEL2
J24	VDD1833_2	IRQ14	P30_7	—	—	ETHSW_PHYLINK2 / ESC_PHYLINK2 / SPI_MISO3 / SD1_IOVS	MCLK30	—	—	—	SI07# / HDSL07_CLK2
J25	VDD1833_2	—	P29_1	—	GTIOC09_0A	ETH2_TXCLK	—	—	—	—	ENCIFCK09 / SCKE09 / HDSL06_CLK1
J26	VDD1833_2	IRQ13	P31_1	—	GTETRGSB	ETH2_RXER / SPI_SSL31	—	—	—	—	HDSL07_MISO2
J27	VDD1833_2	IRQ11	P30_3	—	GTIOC09_5A	ETH2_RXD3 / SPI_MISO2	—	—	—	—	ENCIFDO11 / TXDE11 / HDSL07_CLK1
K1	—	—	—	—	—	—	—	—	DDR_DQA2	—	—
K2	—	—	—	—	—	—	—	—	DDR_DQA0	—	—
K3	—	—	—	—	—	—	—	—	DDR_DQA1	—	—
K4	—	—	—	—	—	—	—	—	DDR_DQA3	—	—
K5	—	—	—	—	—	—	—	—	DDR_DQSA_T0	—	—
K6	—	VSS	—	—	—	—	—	—	—	—	—
K7	—	VDD33	—	—	—	—	—	—	—	—	—
K8	—	VSS	—	—	—	—	—	—	—	—	—
K9	—	VDD1833_6	—	—	—	—	—	—	—	—	—
K10	—	DVDD08A_TSU	—	—	—	—	—	—	—	—	—
K11	—	VDD33	—	—	—	—	—	—	—	—	—
K12	—	VSS	—	—	—	—	—	—	—	—	—
K13	—	VDD1833_7	—	—	—	—	—	—	—	—	—
K14	—	VSS	—	—	—	—	—	—	—	—	—
K15	—	VDD33	—	—	—	—	—	—	—	—	—
K16	—	VSS	—	—	—	—	—	—	—	—	—
K17	—	VDD1833_0	—	—	—	—	—	—	—	—	—
K18	—	VDD33	—	—	—	—	—	—	—	—	—
K19	—	VDD33	—	—	—	—	—	—	—	—	—

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (11 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
K20	—	VDDP_18_2	—	—	—	—	—	—	—	—	—
K21	VDD1833_2	—	P30_6	—	GTIOC09_6B	GMAC2_MDIO / ETHSW_MDIO / ESC_MDIO / SPI_MOSI3	—	—	—	—	TST_OUT07 / HDSL07_MOSI1
K22	—	VSS	—	—	—	—	—	—	—	—	—
K23	VDD1833_2	—	P29_2	—	GTIOC09_0B	ETH2_TXD0	—	—	—	—	ENCIFOE09 / DEE09 / HDSL06_SEL1
K24	VDD1833_2	—	P30_4	—	GTIOC09_5B	ETH2_RXDV	—	—	—	—	ENCIFDI11 / RXDE11 / HDSL07_SEL1
K25	VDD1833_2	—	P30_1	—	GTIOC09_4A	ETH2_RXD1	—	—	—	—	ENCIFCK11 / SCKE11 / HDSL07_LINK
K26	—	VSS	—	—	—	—	—	—	—	—	—
K27	VDD1833_2	—	P30_0	—	GTIOC09_3B	ETH2_RXD0	—	—	—	—	ENCIFDI10 / RXDE10 / HDSL06_MOSI2
L1	—	VSS	—	—	—	—	—	—	—	—	—
L2	—	—	—	—	—	—	—	—	DDR_CAA1	—	—
L3	—	VSS	—	—	—	—	—	—	—	—	—
L4	—	VSS	—	—	—	—	—	—	—	—	—
L5	—	VSS	—	—	—	—	—	—	—	—	—
L6	—	—	—	—	—	—	—	—	DDR_CKEA1	—	—
L7	—	VSS	—	—	—	—	—	—	—	—	—
L8	—	VDD18_PLL2	—	—	—	—	—	—	—	—	—
L9	—	VDD08_PLL2	—	—	—	—	—	—	—	—	—
L10	—	VSS	—	—	—	—	—	—	—	—	—
L11	—	VDD18_PLL3	—	—	—	—	—	—	—	—	—
L12	—	VSS_PLL3	—	—	—	—	—	—	—	—	—
L13	—	VDD08_PLL3	—	—	—	—	—	—	—	—	—
L14	—	VSS	—	—	—	—	—	—	—	—	—
L15	—	VSS	—	—	—	—	—	—	—	—	—
L16	—	VSS	—	—	—	—	—	—	—	—	—
L17	—	VSS	—	—	—	—	—	—	—	—	—
L18	—	VSS	—	—	—	—	—	—	—	—	—
L19	—	VSS	—	—	—	—	—	—	—	—	—
L20	—	VDD1833_2	—	—	—	—	—	—	—	—	—
L21	—	VDD1833_2	—	—	—	—	—	—	—	—	—
L22	VDD1833_2	IRQ8	P29_4	—	GTIOC09_1B	ETH2_TXD2 / SPI_SSL20	—	—	—	—	ENCIFDI09 / RXDE09 / HDSL06_MOSI1
L23	VDD1833_2	IRQ9	P29_5	—	GTIOC09_2A	ETH2_TXD3 / SPI_SSL21	—	—	—	—	ENCIFCK10 / SCKE10 / HDSL06_CLK2
L24	VDD1833_2	ETH2_REFCLK / RMII2_REFCLK	P31_0	—	GTETRGSA	SPI_SSL30	—	—	—	—	HDSL07_SEL2
L25	VDD1833_2	—	P29_7	—	GTIOC09_3A	ETH2_RXCLK / SPI_SSL23	—	—	—	—	ENCIFDO10 / TXDE10 / HDSL06_MISO2
L26	VDD1833_2	IRQ10	P30_2	—	GTIOC09_4B	ETH2_RXD2 / SPI_MOSI2	—	—	—	—	ENCIFOE11 / DEE11 / HDSL07_SMPL

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (12 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
L27	VDD33	—	P32_6	—	GTIOC10_2A / GTIOC01_2A	SPI_SSL10	—	—	—	—	ENCIFCK11 / SCKE11 / HDSL09_SMPL
M1	—	—	—	—	—	—	—	—	DDR_CKA_C	—	—
M2	—	—	—	—	—	—	—	—	DDR_CAA3	—	—
M3	—	—	—	—	—	—	—	—	DDR_CAA4	—	—
M4	—	—	—	—	—	—	—	—	DDR_CSA0	—	—
M5	—	—	—	—	—	—	—	—	DDR_CSA1	—	—
M6	—	—	—	—	—	—	—	—	DDR_CKEA0	—	—
M7	—	VSS	—	—	—	—	—	—	—	—	—
M8	—	VSS_PLL2	—	—	—	—	—	—	—	—	—
M9	—	VDD08	—	—	—	—	—	—	—	—	—
M10	—	VSS	—	—	—	—	—	—	—	—	—
M11	—	VDD08	—	—	—	—	—	—	—	—	—
M12	—	VSS	—	—	—	—	—	—	—	—	—
M13	—	VDD08	—	—	—	—	—	—	—	—	—
M14	—	VSS	—	—	—	—	—	—	—	—	—
M15	—	VDD08	—	—	—	—	—	—	—	—	—
M16	—	VSS	—	—	—	—	—	—	—	—	—
M17	—	VDD08	—	—	—	—	—	—	—	—	—
M18	—	VSS	—	—	—	—	—	—	—	—	—
M19	—	VDD08_PLL0	—	—	—	—	—	—	—	—	—
M20	—	VSS	—	—	—	—	—	—	—	—	—
M21	—	VSS	—	—	—	—	—	—	—	—	—
M22	VDD33	—	P32_2	—	GTIOC10_0A / GTIOC01_0A	SPI_SSL03	—	—	—	—	ENCIFCK10 / SCKE10 / HDSL08_SEL2
M23	VDD33	—	P32_1	—	—	SPI_SSL02	—	—	—	—	ENCIFDI15 / ENCIFDI01 / RXDE11 / RXDE01 / HDSL08_CLK2
M24	—	VSS	—	—	—	—	—	—	—	—	—
M25	VDD33	—	P32_0	—	—	SPI_SSL01	—	—	—	—	ENCIFDO15 / ENCIFDO01 / TXDE11 / TXDE01 / HDSL08_MOSI1
M26	VDD33	—	P31_7	—	—	GMAC2_PTPTRG1 / SPI_SSL00	—	—	—	—	ENCIFOE15 / ENCIFOE01 / DEE11 / DEE01 / HDSL08_MISO1
M27	VDD33	—	P32_5	—	GTIOC10_1B / GTIOC01_1B	SPI_MISO1	—	—	—	—	ENCIFDI10 / RXDE10 / HDSL09_LINK
N1	—	—	—	—	—	—	—	—	DDR_CKA_T	—	—
N2	—	VSS	—	—	—	—	—	—	—	—	—
N3	—	—	—	—	—	—	—	—	DDR_CAA2	—	—
N4	—	VSS	—	—	—	—	—	—	—	—	—
N5	—	—	—	—	—	—	—	—	DDR_CAA5	—	—
N6	—	VSS	—	—	—	—	—	—	—	—	—
N7	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
N8	—	DDR_VAA	—	—	—	—	—	—	—	—	—
N9	—	VSS	—	—	—	—	—	—	—	—	—
N10	—	VDD08	—	—	—	—	—	—	—	—	—

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (13 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
N11	—	VSS	—	—	—	—	—	—	—	—	—
N12	—	VDD08	—	—	—	—	—	—	—	—	—
N13	—	VSS	—	—	—	—	—	—	—	—	—
N14	—	VDD08	—	—	—	—	—	—	—	—	—
N15	—	VSS	—	—	—	—	—	—	—	—	—
N16	—	VDD08	—	—	—	—	—	—	—	—	—
N17	—	VSS	—	—	—	—	—	—	—	—	—
N18	—	VDD08	—	—	—	—	—	—	—	—	—
N19	—	VSS_PLL0	—	—	—	—	—	—	—	—	—
N20	—	VDD18_PLL0	—	—	—	—	—	—	—	—	—
N21	—	VSS	—	—	—	—	—	—	—	—	—
N22	VDD33	—	P32_4	—	GTIOC10_1A / GTIOC01_1A	SPI_MOSI1	—	—	—	—	ENCIFDO10 / TXDE10 / HDSL08_MOSI2
N23	VDD33	—	P32_3	—	GTIOC10_0B / GTIOC01_0B	SPI_RSPCK1	—	—	—	—	ENCIFOE10 / DEE10 / HDSL08_MISO2
N24	VDD33	—	P32_7	—	GTIOC10_2B / GTIOC01_2B	SPI_SSL11	—	—	—	—	ENCIFOE11 / DEE11 / HDSL09_CLK1
N25	VDD33	—	P31_4	DREQ	POE8#	ETH2_CRS / ETHSW_PTPOUT2 / ESC_SYNC0 / SPI_RSPCK0 / SPI_SSL30	MCLK81 / MDAT31	—	—	HSPI_IO6	ENCIFDO09 / TXDE09 / HDSL08_SMPL / POUTB
N26	VDD33	—	P31_3	—	POE4#	ETH2_RXER / ETHSW_TDMAOUT 1 / ESC_LEDERR / SPI_SSL33	MDAT80 / MCLK31	—	—	HSPI_IO5	ENCIFOE09 / DEE09 / HDSL08_LINK
N27	VDD33	—	P31_2	—	POE0#	ETH2_TXER / SPI_SSL32	MCLK80 / MDAT30	—	—	HSPI_IO4	ENCIFCK09 / SCKE09 / HDSL07_MOSI2 / POUTA
P1	—	VSS	—	—	—	—	—	—	—	—	—
P2	—	—	—	—	—	—	—	—	DDR_CKEB1	—	—
P3	—	—	—	—	—	—	—	—	DDR_CAB0	—	—
P4	—	—	—	—	—	—	—	—	DDR_CAA0	—	—
P5	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
P6	—	VSS	—	—	—	—	—	—	—	—	—
P7	—	—	—	—	—	—	—	—	DDR_RESET_N	—	—
P8	—	—	—	—	—	—	—	—	DDR_ATEST	—	—
P9	—	VDD08	—	—	—	—	—	—	—	—	—
P10	—	VSS	—	—	—	—	—	—	—	—	—
P11	—	VDD08	—	—	—	—	—	—	—	—	—
P12	—	VSS	—	—	—	—	—	—	—	—	—
P13	—	VDD08	—	—	—	—	—	—	—	—	—
P14	—	VSS	—	—	—	—	—	—	—	—	—
P15	—	VDD08	—	—	—	—	—	—	—	—	—
P16	—	VSS	—	—	—	—	—	—	—	—	—
P17	—	VDD08	—	—	—	—	—	—	—	—	—
P18	—	VSS	—	—	—	—	—	—	—	—	—
P19	—	VSS	—	—	—	—	—	—	—	—	—
P20	—	VSS	—	—	—	—	—	—	—	—	—
P21	—	VDDP_18_33	—	—	—	—	—	—	—	—	—

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (14 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
P22	—	VSS	—	—	—	—	—	—	—	—	—
P23	VDD33	—	P33_1	—	GTIOC10_3B	SPI_SSL13	MDAT82	—	—	—	ENCIFDI11 / RXDE11 / HDSL09_MISO1
P24	VDD33	—	P33_0	—	GTIOC10_3A	SPI_SSL12	MCLK82	—	—	—	ENCIFDO11 / TXDE11 / HDSL09_SEL1
P25	VDD33	—	P31_6	A16 / TEND	POE11#	GMAC2_PTPTRG0 / ETHSW_TDMAOUT 0 / ESC_LEDRUN / SPI_MISO0	MDAT32	—	—	—	ENCIFCK15 / ENCIFCK01 / SCKE11 / SCKE01 / HDSL08_SEL1
P26	VDD33	—	P31_5	DACK	POE10#	ETH2_COL / ETHSW_PTPOUT3 / ESC_SYNC1 / SPI_MOSI0 / SPI_SSL31	MDAT81 / MCLK32	—	—	HSPI_IO7	ENCIFDI09 / RXDE09 / HDSL08_CLK1 / POUTZ
P27	VDD1833_3	—	P34_4	CS2#	GTADSM05_0 / GTIOC03_2A	ETH3_RXD3 / RXD3/ SCL3/MISO3 / SPI_SSL22 / SD1_IOVS	—	—	ADTRG0#	—	ENCIFDO07 / TXDE07 / HDSL10_MOSI1
R1	—	—	—	—	—	—	—	—	DDR_CKB_T	—	—
R2	—	—	—	—	—	—	—	—	DDR_CKEB0	—	—
R3	—	VSS	—	—	—	—	—	—	—	—	—
R4	—	VSS	—	—	—	—	—	—	—	—	—
R5	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
R6	—	VSS	—	—	—	—	—	—	—	—	—
R7	—	—	—	—	—	—	—	—	DDR_DTEST	—	—
R8	—	—	—	—	—	—	—	—	DDR_ZN	—	—
R9	—	VSS	—	—	—	—	—	—	—	—	—
R10	—	VDD08	—	—	—	—	—	—	—	—	—
R11	—	VSS	—	—	—	—	—	—	—	—	—
R12	—	VDD08	—	—	—	—	—	—	—	—	—
R13	—	VSS	—	—	—	—	—	—	—	—	—
R14	—	VDD08	—	—	—	—	—	—	—	—	—
R15	—	VSS	—	—	—	—	—	—	—	—	—
R16	—	VDD08	—	—	—	—	—	—	—	—	—
R17	—	VSS	—	—	—	—	—	—	—	—	—
R18	—	VDD08	—	—	—	—	—	—	—	—	—
R19	—	VDDP_18_X	—	—	—	—	—	—	—	—	—
R20	—	VDD33	—	—	—	—	—	—	—	—	—
R21	—	VSS	—	—	—	—	—	—	—	—	—
R22	VDD1833_3	—	P34_1	A23	GTADSM03_1 / GTIOC03_0B	ETH3_RXD0 / SPI_MISO2	—	—	—	—	ENCIFDI06 / RXDE06 / HDSL10_CLK1
R23	VDD1833_3	ETH3_REFCLK / RMII3_REFCLK	P34_6	CS5#	—	ETH1_RXER / ESC_I2CDATA / IIC_SDA1 / SPI_RSPCK3	—	—	ADTRG2#	—	DUEI08 / HDSL10_SEL2
R24	VDD1833_3	—	P33_2	A16	GTADSM00_0	ETH3_TXCLK / SCK1 / SPI_RSPCK1 / SPI_SSL30	MCLK50	—	—	—	ENCIFCK01 / SCKE01 / HDSL09_MOSI1
R25	VDD1833_3	IRQ15	P33_6	A20	GTADSM02_0	ETH3_TXD3 / TXD2/ SDA2/MOSI2 / SPI_SSL11 / SPI_SSL00	MCLK52	—	—	—	ENCIFCK06 / SCKE06 / HDSL09_MOSI2
R26	VDD1833_3	—	P33_7	A21	GTADSM02_1	ETH3_TXEN / SPI_RSPCK2	MDAT52	—	—	—	ENCIFOE06 / DEE06 / HDSL10_LINK

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (15 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
R27	VDD1833_3	—	P34_5	CS3#	GTADSM05_1 / GTIOC03_2B	ETH3_RXDV / ESC_I2CCLK / TXD3/SDA3/MOSI3 / IIC_SCL1 / SPI_SSL23	—	—	ADTRG1#	—	ENCIFDI07 / RXDE07 / HDSL10_CLK2
T1	—	—	—	—	—	—	—	—	DDR_CKB_C	—	—
T2	—	—	—	—	—	—	—	—	DDR_CAB1	—	—
T3	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
T4	—	—	—	—	—	—	—	—	DDR_CAB2	—	—
T5	—	—	—	—	—	—	—	—	DDR_CAB5	—	—
T6	—	—	—	—	—	—	—	—	DDR_CSB0	—	—
T7	—	VSS	—	—	—	—	—	—	—	—	—
T8	—	VSS	—	—	—	—	—	—	—	—	—
T9	—	VDD08	—	—	—	—	—	—	—	—	—
T10	—	VSS	—	—	—	—	—	—	—	—	—
T11	—	VDD08	—	—	—	—	—	—	—	—	—
T12	—	VSS	—	—	—	—	—	—	—	—	—
T13	—	VDD08	—	—	—	—	—	—	—	—	—
T14	—	VSS	—	—	—	—	—	—	—	—	—
T15	—	VDD08	—	—	—	—	—	—	—	—	—
T16	—	VSS	—	—	—	—	—	—	—	—	—
T17	—	VDD08	—	—	—	—	—	—	—	—	—
T18	—	VSS	—	—	—	—	—	—	—	—	—
T19	—	VDD33_X	—	—	—	—	—	—	—	—	—
T20	—	VSS	—	—	—	—	—	—	—	—	—
T21	—	VDD1833_3	—	—	—	—	—	—	—	—	—
T22	—	VDDP_18_3	—	—	—	—	—	—	—	—	—
T23	VDD1833_3	—	P34_2	A24	GTADSM04_0 / GTIOC03_1A	ETH3_RXD1 / SPI_SSL20	—	—	—	—	ENCIFCK07 / SCKE07 / HDSL10_SEL1
T24	—	VSS	—	—	—	—	—	—	—	—	—
T25	VDD1833_3	IRQ14	P33_5	A19	GTADSM01_1	ETH3_TXD2 / RXD2/ SCL2/MISO2 / SPI_SSL10 / SPI_MISO0	MDAT51	—	—	—	ENCIFDI01 / RXDE01 / HDSL09_MISO2
T26	VDD33	—	P35_1	TEND	GTADSM07_0	ETH3_CRS / SPI_SSL30 / SPI_MISO1	MCLK90	—	—	—	DUEI09 / HDSL11_LINK
T27	VDD33_X	XTALSEL	—	—	—	—	—	—	—	—	—
U1	—	—	—	—	—	—	—	—	DDR_CAB3	—	—
U2	—	VSS	—	—	—	—	—	—	—	—	—
U3	—	—	—	—	—	—	—	—	DDR_CAB4	—	—
U4	—	—	—	—	—	—	—	—	DDR_DQB0	—	—
U5	—	VSS	—	—	—	—	—	—	—	—	—
U6	—	—	—	—	—	—	—	—	DDR_CSB1	—	—
U7	—	VSS	—	—	—	—	—	—	—	—	—
U8	—	VSS	—	—	—	—	—	—	—	—	—
U9	—	VSS	—	—	—	—	—	—	—	—	—
U10	—	VDD08	—	—	—	—	—	—	—	—	—
U11	—	VSS	—	—	—	—	—	—	—	—	—
U12	—	VDD08	—	—	—	—	—	—	—	—	—
U13	—	VSS_PLL1	—	—	—	—	—	—	—	—	—

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (16 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
U14	—	VSS_PLL4	—	—	—	—	—	—	—	—	—
U15	—	VSS	—	—	—	—	—	—	—	—	—
U16	—	VDD08	—	—	—	—	—	—	—	—	—
U17	—	VSS	—	—	—	—	—	—	—	—	—
U18	—	VSS	—	—	—	—	—	—	—	—	—
U19	—	VSS	—	—	—	—	—	—	—	—	—
U20	—	VSS	—	—	—	—	—	—	—	—	—
U21	—	VDD1833_3	—	—	—	—	—	—	—	—	—
U22	VDD33	—	P35_6	—	GTADSM09_1	TXD4/SDA4/MOSI4 / SPI_SSL12	MDAT92	—	—	—	SI10# / HDSL11_MOSI1
U23	VDD1833_3	—	P34_3	A25	GTADSM04_1 / GTIOC03_1B	ETH3_RXD2 / SPI_SSL21 / SD1_PWEN	—	—	—	—	ENCIFOE07 / DEE07 / HDSL10_MISO1
U24	VDD1833_3	IRQ13	P33_4	A18	GTADSM01_0	ETH3_TXD1 / TXD1/ SDA1/MOSI1 / SPI_MISO1 / SPI_MOSI0	MCLK51	—	PCIE_RSTOUT1B	—	ENCIFDO01 / TXDE01 / HDSL09_SEL2
U25	VDD1833_3	IRQ12	P33_3	A17	GTADSM00_1	ETH3_TXD0 / RXD1/ SCL1/MISO1 / SPI_MOSI1 / SPI_RSPCK0	MDAT50	—	PCIE_RSTOUT0B	—	ENCIFOE01 / DEE01 / HDSL09_CLK2
U26	—	VSS	—	—	—	—	—	—	—	—	—
U27	VDD33_X	EXTCLKIN	—	—	—	—	—	—	—	—	—
V1	—	—	—	—	—	—	—	—	DDR_DQB2	—	—
V2	—	—	—	—	—	—	—	—	DDR_DQB1	—	—
V3	—	VSS	—	—	—	—	—	—	—	—	—
V4	—	—	—	—	—	—	—	—	DDR_DQB3	—	—
V5	—	—	—	—	—	—	—	—	DDR_DQSB_T0	—	—
V6	—	VSS	—	—	—	—	—	—	—	—	—
V7	—	VSS	—	—	—	—	—	—	—	—	—
V8	—	VSS	—	—	—	—	—	—	—	—	—
V9	—	VDD1833_4	—	—	—	—	—	—	—	—	—
V10	—	VDD33	—	—	—	—	—	—	—	—	—
V11	—	VDDP_18_4	—	—	—	—	—	—	—	—	—
V12	—	VDDP_18_5	—	—	—	—	—	—	—	—	—
V13	—	VDD18_PLL1	—	—	—	—	—	—	—	—	—
V14	—	VDD18_PLL4	—	—	—	—	—	—	—	—	—
V15	—	VDD08	—	—	—	—	—	—	—	—	—
V16	—	VSS	—	—	—	—	—	—	—	—	—
V17	—	OTPVDD08	—	—	—	—	—	—	—	—	—
V18	—	VSS	—	—	—	—	—	—	—	—	—
V19	—	PCIE_VDD08A_L0	—	—	—	—	—	—	—	—	—
V20	—	VSS	—	—	—	—	—	—	—	—	—
V21	—	VSS	—	—	—	—	—	—	—	—	—
V22	VDD33	—	P35_3	—	GTADSM08_0	SPI_SSL32 / SPI_MOSI1	MCLK91	—	ADTRG0#	—	SI09# / HDSL11_CLK1
V23	VDD1833_3	—	P34_0	A22	GTADSM03_0 / GTIOC03_0A	ETH3_RXCLK / SPI_MOSI2	—	—	—	—	ENCIFDO06 / TXDE06 / HDSL10_SMPL
V24	VDD33	—	P35_5	—	GTADSM09_0	RXD4/SCL4/MISO4 / SPI_RSPCK1	MCLK92	—	—	—	TST_OUT10 / HDSL11_MISO1
V25	VDD33	—	P35_0	DACK	GTADSM06_1	ETH3_RXER / SPI_MISO3	—	—	—	—	SI08# / HDSL10_MOSI2

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (17 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
V26	—	VSS	—	—	—	—	—	—	—	—	—
V27	—	XTAL	—	—	—	—	—	—	—	—	—
W1	—	VSS	—	—	—	—	—	—	—	—	—
W2	—	—	—	—	—	—	—	—	DDR_DQB4	—	—
W3	—	—	—	—	—	—	—	—	DDR_DQB7	—	—
W4	—	—	—	—	—	—	—	—	DDR_DMIB0	—	—
W5	—	—	—	—	—	—	—	—	DDR_DQSB_C 0	—	—
W6	—	VSS	—	—	—	—	—	—	—	—	—
W7	—	VSS	—	—	—	—	—	—	—	—	—
W8	—	VSS	—	—	—	—	—	—	—	—	—
W9	—	VDD1833_4	—	—	—	—	—	—	—	—	—
W10	—	VDD33	—	—	—	—	—	—	—	—	—
W11	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
W12	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
W13	—	VDD08_PLL1	—	—	—	—	—	—	—	—	—
W14	—	VDD08_PLL4	—	—	—	—	—	—	—	—	—
W15	—	VSS	—	—	—	—	—	—	—	—	—
W16	—	VSS	—	—	—	—	—	—	—	—	—
W17	—	OTPVDD18	—	—	—	—	—	—	—	—	—
W18	—	VSS	—	—	—	—	—	—	—	—	—
W19	—	PCIE_VDD08A _L0	—	—	—	—	—	—	—	—	—
W20	—	VSS	—	—	—	—	—	—	—	—	—
W21	—	VSS	—	—	—	—	—	—	—	—	—
W22	—	VSS	—	—	—	—	—	—	—	—	—
W23	VDD33	IRQ14	P34_7	DREQ	GTADSM06_0	ETH3_TXER / ESC_RESETOUT# / SPI_MOSI3	—	—	—	—	TST_OUT08 / HDSL10_MISO2
W24	VDD33	—	P35_4	—	GTADSM08_1	SPI_SSL33 / SPI_SSL11	MDAT91	—	ADTRG1#	—	DUEI10 / HDSL11_SEL1
W25	VDD33	—	P35_2	—	GTADSM07_1	ETH3_COL / SPI_SSL31 / SPI_SSL10	MDAT90	—	ADTRG2#	—	TST_OUT09 / HDSL11_SMPL
W26	—	VSS	—	—	—	—	—	—	—	—	—
W27	—	EXTAL	—	—	—	—	—	—	—	—	—
Y1	—	—	—	—	—	—	—	—	DDR_DQB6	—	—
Y2	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
Y3	—	—	—	—	—	—	—	—	DDR_DQB5	—	—
Y4	—	VSS	—	—	—	—	—	—	—	—	—
Y5	—	VSS	—	—	—	—	—	—	—	—	—
Y6	—	VSS	—	—	—	—	—	—	—	—	—
Y7	—	VSS	—	—	—	—	—	—	—	—	—
Y8	—	VSS	—	—	—	—	—	—	—	—	—
Y9	—	VSS	—	—	—	—	—	—	—	—	—
Y10	—	VSS	—	—	—	—	—	—	—	—	—
Y11	—	VSS	—	—	—	—	—	—	—	—	—
Y12	—	VDD33	—	—	—	—	—	—	—	—	—
Y13	—	VDD33	—	—	—	—	—	—	—	—	—
Y14	—	VDD1833_5	—	—	—	—	—	—	—	—	—
Y15	—	VDD1833_5	—	—	—	—	—	—	—	—	—

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (18 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
Y16	—	VSS	—	—	—	—	—	—	—	—	—
Y17	—	USB_USVDD18	—	—	—	—	—	—	—	—	—
Y18	—	USB_USVDD33	—	—	—	—	—	—	—	—	—
Y19	—	PCIE_VDD08A_L1	—	—	—	—	—	—	—	—	—
Y20	—	PCIE_VDD18A_L1	—	—	—	—	—	—	—	—	—
Y21	—	PCIE_VDD18A_L0	—	—	—	—	—	—	—	—	—
Y22	—	VSS	—	—	—	—	—	—	—	—	—
Y23	—	VSS	—	—	—	—	—	—	—	—	—
Y24	—	VSS	—	—	—	—	—	—	—	—	—
Y25	—	VSS	—	—	—	—	—	—	—	—	—
Y26	—	VSS	—	—	—	—	—	—	—	—	—
Y27	—	VSS	—	—	—	—	—	—	—	—	—
AA1	—	—	—	—	—	—	—	—	DDR_DQB8	—	—
AA2	—	VSS	—	—	—	—	—	—	—	—	—
AA3	—	—	—	—	—	—	—	—	DDR_DQB15	—	—
AA4	—	VSS	—	—	—	—	—	—	—	—	—
AA5	—	—	—	—	—	—	—	—	DDR_DQSB_T1	—	—
AA6	—	VSS	—	—	—	—	—	—	—	—	—
AA7	VDD1833_4	IRQ11	P06_5	—	GTETRGC	IIC_SDA1 / XSPI0_IO7	—	—	—	—	HDSL05_SEL1
AA8	VDD1833_4	IRQ3	P05_1	—	—	XSPI0_CKP	—	—	—	—	DUEI06 / HDSL04_SMPL
AA9	VDD1833_4	IRQ4	P05_2	—	—	IIC_SCL2 / XSPI0_CKN	—	—	—	—	TST_OUT06 / HDSL04_CLK1
AA10	VDD33	IRQ2	P05_0	—	MTIOC6C / MTIOC0B / GTIOC03_4B	IIC_SDA1	—	—	—	—	ENCIFDI03 / RXDE03 / HDSL04_LINK
AA11	—	VSS	—	—	—	—	—	—	—	—	—
AA12	VDD33	—	P04_2	—	MTIOC7C / GTIOC03_1B / CMTW0_TOC1	—	—	—	—	—	DUEI05 / HDSL03_MISO1
AA13	VDD1833_5	IRQ9	P02_2	—	MTIOC6A / MTIOC1A / GTIOC01_4A	ETH3_CRS / IIC_SDA2 / XSPI1_IO6	MCLK22	—	USB_VBUSEN	—	ENCIFDO01 / TXDE01 / HDSL01_MISO2
AA14	—	VSS	—	—	—	—	—	—	—	—	—
AA15	—	VSS	—	—	—	—	—	—	—	—	—
AA16	—	USB_USDVDD	—	—	—	—	—	—	—	—	—
AA17	—	USB_USVDD18	—	—	—	—	—	—	—	—	—
AA18	—	USB_USVDD33	—	—	—	—	—	—	—	—	—
AA19	—	PCIE_VDD08A_L1	—	—	—	—	—	—	—	—	—
AA20	—	PCIE_VDD18A_L1	—	—	—	—	—	—	—	—	—
AA21	—	PCIE_VDD18A_L0	—	—	—	—	—	—	—	—	—
AA22	—	AVSS_ADC0	—	—	—	—	—	—	—	—	—
AA23	—	AVDD_ADC0	—	—	—	—	—	—	—	—	—

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (19 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
AA24	—	AVSSIO_ADC0	—	—	—	—	—	—	—	—	—
AA25	—	AVSSIO_ADC0	—	—	—	—	—	—	—	—	—
AA26	—	—	—	—	—	—	—	—	AN002	—	—
AA27	—	—	—	—	—	—	—	—	AN000	—	—
AB1	—	—	—	—	—	—	—	—	DDR_DQB14	—	—
AB2	—	—	—	—	—	—	—	—	DDR_DQB9	—	—
AB3	—	—	—	—	—	—	—	—	DDR_DMIB1	—	—
AB4	—	—	—	—	—	—	—	—	DDR_DQB10	—	—
AB5	—	—	—	—	—	—	—	—	DDR_DQSB_C1	—	—
AB6	VDD33	BSCANP	—	—	—	—	—	—	—	—	—
AB7	VDD1833_4	IRQ9	P06_3	—	GTETRGA	IIC_SDA0 / XSPI0_IO5	—	—	—	—	TST_OUT09 / HDSL05_SMPL
AB8	VDD1833_4	IRQ10	P06_4	—	GTETRGB	IIC_SCL1 / XSPI0_IO6	—	—	—	—	SI09# / HDSL05_CLK1
AB9	VDD1833_4	—	P07_7	—	—	IIC_SCL0 / XSPI0_WP0#	MCLK10	—	—	—	ENCIFDO05 / TXDE05 / HDSL06_CLK1
AB10	VDD33	IRQ14	P03_4	D12	MTCLKB / MTIOC8D / GTIOC02_3B / GTADSM09_1 / CMTW1_TOC1 / RTCAT1HZ	IIC_SDA1	—	—	—	—	ENCIFOE02 / DEE02 / HDSL02_MISO2
AB11	VDD33	IRQ13	P03_3	D11	MTCLKA / MTIOC8C / GTIOC02_3A / GTADSM09_0 / CMTW1_TIC1	IIC_SCL1	—	—	—	—	ENCIFCK02 / SCKE02 / HDSL02_SEL2
AB12	VDD33	—	P03_7	—	MTIOC6B / MTIOC1B / GTIOC03_0A / CMTW0_TIC0	—	—	—	—	—	DUEI04 / HDSL03_SMPL
AB13	VDD1833_5	IRQ8	P02_1	—	MTCLKD / MTIOC0D / GTIOC01_3B	ETH3_RXER / IIC_SCL2 / XSPI1_IO5	MDAT21	—	—	—	ENCIFOE01 / DEE01 / HDSL01_SEL2
AB14	VDD1833_5	—	P01_3	—	MTIOC6D / MTIC5U / GTIOC01_0B / GTIOC04_0B	XSPI1_DS	—	—	—	—	TST_OUT02 / HDSL01_SMPL
AB15	VDD33	—	P00_5	—	MTIOC4D / MTIOC8C / GTIOC00_2B	—	—	—	USB_VBUSEN	—	SI01# / HDSL00_MOSI1
AB16	—	USB_USDVDD	—	—	—	—	—	—	—	—	—
AB17	—	VSS	—	—	—	—	—	—	—	—	—
AB18	—	VSS	—	—	—	—	—	—	—	—	—
AB19	—	VSS	—	—	—	—	—	—	—	—	—
AB20	—	VSS	—	—	—	—	—	—	—	—	—
AB21	—	PCIE_VDD18A_CMN	—	—	—	—	—	—	—	—	—
AB22	—	AVSS_ADC1	—	—	—	—	—	—	—	—	—
AB23	—	AVDD_ADC1	—	—	—	—	—	—	—	—	—
AB24	—	AVDDIO_ADC0	—	—	—	—	—	—	—	—	—
AB25	—	AVDDREF_ADC0	—	—	—	—	—	—	—	—	—
AB26	—	—	—	—	—	—	—	—	AN001	—	—
AB27	—	—	—	—	—	—	—	—	AN003	—	—

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (20 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
AC1	—	—	—	—	—	—	—	—	DDR_DQB12	—	—
AC2	—	VSS	—	—	—	—	—	—	—	—	—
AC3	—	—	—	—	—	—	—	—	DDR_DQB13	—	—
AC4	—	—	—	—	—	—	—	—	DDR_DQB11	—	—
AC5	—	VSS	—	—	—	—	—	—	—	—	—
AC6	VDD1833_4	—	P06_1	—	—	XSPI0_IO3	—	—	—	—	SI08# / HDSL04_MOSI2
AC7	VDD1833_4	—	P07_6	—	—	IIC_SDA2 / XSPI0_ECS1#	MDAT02	—	—	—	ENCIFOE05 / DEE05 / HDSL06_SMPL
AC8	VDD1833_4	—	P07_4	—	—	IIC_SDA1 / XSPI0_INT1#	MDAT01	—	—	—	ENCIFDI04 / ENCIFDI12 / RXDE04 / RXDE08 / HDSL05_MOSI2
AC9	VDD33	IRQ12	P03_2	D10	MTIOC4D / MTIOC1A / GTIOC02_2B / GTADSM08_1 / CMTW1_TOC0	—	—	—	—	—	ENCIFDI02 / RXDE02 / HDSL02_CLK2
AC10	VDD33	—	P03_1	D9	MTIOC4B / MTIOC1B / GTIOC02_2A / GTADSM08_0 / CMTW1_TIC0	—	—	—	—	—	ENCIFDO02 / TXDE02 / HDSL02_MOSI1
AC11	VDD33	IRQ1	P04_7	—	MTIOC6A / MTIOC0A / GTIOC03_4A	IIC_SCL1	—	—	—	—	ENCIFDO03 / TXDE03 / HDSL03_MOSI2
AC12	VDD33	IRQ15	P03_5	—	MTIOC3A / MTIC5W / GTIOC02_4A	IIC_SCL2	—	—	—	—	ENCIFDO02 / TXDE02 / HDSL02_MOSI2
AC13	VDD1833_5	—	P01_7	—	MTIOC7D / MTIOC0B / GTIOC01_2B / GTIOC04_2B	XSPI1_IO3	—	—	—	—	SI03# / HDSL01_MOSI1
AC14	VDD1833_5	IRQ7	P02_0	—	MTCLKC / MTIOC0C / GTIOC01_3A	ETH3_TXER / IIC_SDA1 / XSPI1_IO4	MCLK21	—	—	—	ENCIFCK01 / SCKE01 / HDSL01_CLK2
AC15	VDD33	IRQ1	P00_2	D2	MTIOC4A / GTIOC00_1A	ETH3_CRS	—	—	ADTRG0# / USB_EXICEN	—	SI00# / HDSL00_CLK1
AC16	—	VSS	—	—	—	—	—	—	—	—	—
AC17	—	—	—	—	—	—	—	—	USB_TXRTUNE	—	—
AC18	—	VSS	—	—	—	—	—	—	—	—	—
AC19	—	—	—	—	—	—	—	—	PCIE_REFCLK_N1	—	—
AC20	—	—	—	—	—	—	—	—	PCIE_REFCLK_P0	—	—
AC21	—	PCIE_VDD18A_CMN	—	—	—	—	—	—	—	—	—
AC22	—	AVDD_ADC2	—	—	—	—	—	—	—	—	—
AC23	—	AVDDIO_ADC2	—	—	—	—	—	—	—	—	—
AC24	—	AVDDIO_ADC1	—	—	—	—	—	—	—	—	—
AC25	—	AVDDREF_ADC1	—	—	—	—	—	—	—	—	—
AC26	—	—	—	—	—	—	—	—	AN103	—	—
AC27	—	—	—	—	—	—	—	—	AN100	—	—
AD1	VDD33	MDX	—	—	—	—	—	—	—	—	—

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (21 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
AD2	VDD33	SEI / CKIO	P08_6	—	GTIOC08_3A / GTETRGSB	IIC_SDA1 / SD1_IOVS	MDAT02 / MCLK11	—	—	—	DUEI11 / HDSL06_MOSI2
AD3	VDD33	TCK	P08_3	—	—	—	—	—	—	—	SI10# / HDSL06_CLK2
AD4	VDD33	RES#	—	—	—	—	—	—	—	—	—
AD5	VDD33	TDI	P08_2	—	—	—	—	—	—	—	TST_OUT10 / HDSL06_MOSI1
AD6	VDD1833_4	—	P05_6	—	—	XSPI0_IO0	—	—	—	—	SI07# / HDSL04_CLK2
AD7	VDD1833_4	—	P08_0	—	RTCAT1HZ	IIC_SDA0 / XSPI0_WP1#	MDAT10	—	—	MBX_HINT#	ENCIFDI05 / RXDE05 / HDSL06_SEL1
AD8	—	VSS	—	—	—	—	—	—	—	—	—
AD9	VDD33	—	P02_6	D6	MTIOC3D / MTIOC8B / GTIOC02_0B / GTADSM06_1 / CMTW0_TOC0	SD0_IOVS	MDAT00	—	—	—	HDSL02_CLK1 / POUTB
AD10	VDD33	IRQ0	P04_6	—	MTCLKD / MTIOC0D / GTIOC03_3B / CMTW1_TOC1	IIC_SDA0	—	—	ADTRG2#	MBX_HINT#	ENCIFOE03 / DEE03 / HDSL03_MISO2
AD11	VDD33	—	P02_5	D5	MTIOC3B / MTIOC8A / GTIOC02_0A / GTADSM06_0 / CMTW0_TIC0	IIC_SCL0 / SD0_PWEN	MCLK00	—	—	—	HDSL02_SMPL / POUTA
AD12	—	VSS	—	—	—	—	—	—	—	—	—
AD13	VDD1833_5	—	P01_6	—	MTIOC7B / MTIOC0A / GTIOC01_2A / GTIOC04_2A	XSPI1_IO2	—	—	—	—	TST_OUT03 / HDSL01_MISO1
AD14	VDD1833_5	—	P01_2	—	MTIOC6B / MTIOC8B / GTIOC01_0A / GTIOC04_0A	XSPI1_CS1#	—	—	—	—	DUEI02 / HDSL01_LINK
AD15	VDD33	IRQ0	P00_1	D1	MTIOC3D / GTIOC00_0B	ETH3_RXER	—	—	USB_OVRCUR	—	TST_OUT00 / HDSL00_SMPL
AD16	—	—	—	—	—	—	—	—	USB_VUBUSIN	—	—
AD17	—	—	—	—	—	—	—	—	USB_OTG_ID	—	—
AD18	—	VSS	—	—	—	—	—	—	—	—	—
AD19	—	—	—	—	—	—	—	—	PCIE_REFCLK_P1	—	—
AD20	—	—	—	—	—	—	—	—	PCIE_REFCLK_N0	—	—
AD21	—	VSS	—	—	—	—	—	—	—	—	—
AD22	—	AVSS_ADC2	—	—	—	—	—	—	—	—	—
AD23	—	AVSSIO_ADC2	—	—	—	—	—	—	—	—	—
AD24	—	AVSSIO_ADC1	—	—	—	—	—	—	—	—	—
AD25	—	AVSSIO_ADC1	—	—	—	—	—	—	—	—	—
AD26	—	—	—	—	—	—	—	—	AN102	—	—
AD27	—	—	—	—	—	—	—	—	AN101	—	—
AE1	VDD33	—	P09_0	—	—	IIC_SDA2	MCLK12	—	—	—	SI11# / HDSL07_SMPL
AE2	VDD33	—	P09_2	—	—	—	MCLK20	—	—	—	TST_OUT12 / HDSL07_SEL1

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (22 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
AE3	VDD33	IRQ8 / RSTOUT#	P08_5	—	GTETRGSA	IIC_SCL1 / SD1_PWEN	MCLK02	—	—	—	HDSL06_MISO2
AE4	VDD33	TDO	P08_4	—	—	—	—	—	—	—	HDSL06_SEL2
AE5	VDD1833_4	—	P06_0	—	—	XSPI0_IO2	—	—	—	—	TST_OUT08 / HDSL04_MISO2
AE6	VDD1833_4	IRQ8	P06_2	—	—	IIC_SCL0 / XSPI0_IO4	—	—	—	—	DUEI09 / HDSL05_LINK
AE7	VDD1833_4	—	P05_7	—	—	XSPI0_IO1	—	—	—	—	DUEI08 / HDSL04_SEL2
AE8	VDD1833_4	—	P07_3	—	POE11#	IIC_SCL1 / XSPI0_INT0#	MCLK01	—	—	—	ENCIFDO04 / ENCIFDO12 / TXDE04 / TXDE08 / HDSL05_MISO2
AE9	VDD33	—	P02_7	D7	MTIOC4A / MTIC5U / GTIOC02_1A / GTADSM07_0 / CMTW0_TIC1	—	MCLK01	—	—	—	HDSL02_SEL1 / POUTZ
AE10	VDD33	—	P03_0	D8	MTIOC4C / MTIC5V / GTIOC02_1B / GTADSM07_1 / CMTW0_TOC1	—	MDAT01	—	—	—	HDSL02_MISO1
AE11	VDD33	—	P04_0	—	MTIOC6D / GTIOC03_0B / CMTW0_TOC0	—	—	—	—	—	TST_OUT04 / HDSL03_CLK1
AE12	VDD1833_5	IRQ11	P02_4	—	POE0#	IIC_SDA0	MDAT20	—	USB_EXICEN	MBX_HINT#	HDSL02_LINK
AE13	VDD1833_5	—	P01_5	—	MTIOC7C / MTIC5W / GTIOC01_1B / GTIOC04_1B	XSPI1_IO1	—	—	—	—	DUEI03 / HDSL01_SEL1
AE14	VDD33	IRQ5	P00_7	—	MTCLKB / MTIOC1B / GTIOC00_3B	IIC_SDA0	—	—	USB_EXICEN	—	ENCIFOE00 / ENCIFOE04 / DEE00 / DEE04 / HDSL00_SEL2
AE15	VDD33	SEI	P00_0	D0	MTIOC3B / GTIOC00_0A	ETH3_TXER	—	—	USB_VBUSEN	—	DUEI00 / HDSL00_LINK
AE16	—	VSS	—	—	—	—	—	—	—	—	—
AE17	—	VSS	—	—	—	—	—	—	—	—	—
AE18	—	VSS	—	—	—	—	—	—	—	—	—
AE19	—	VSS	—	—	—	—	—	—	—	—	—
AE20	—	VSS	—	—	—	—	—	—	—	—	—
AE21	—	VSS	—	—	—	—	—	—	—	—	—
AE22	—	VSS	—	—	—	—	—	—	—	—	—
AE23	—	VSS	—	—	—	—	—	—	—	—	—
AE24	—	VSS	—	—	—	—	—	—	—	—	—
AE25	—	AVSSIO_ADC2	—	—	—	—	—	—	—	—	—
AE26	—	—	—	—	—	—	—	—	AN202	—	—
AE27	—	—	—	—	—	—	—	—	AN201	—	—
AF1	VDD33	—	P09_3	—	—	—	MDAT20	—	—	—	SI12# / HDSL07_MISO1
AF2	VDD33	IRQ0	P08_7	A0	GTIOC08_3B	IIC_SCL2 / IIC_SCL1	MDAT11	—	—	—	TST_OUT11 / HDSL07_LINK
AF3	VDD33	TRST#	—	—	—	—	—	—	—	—	—

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (23 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
AF4	VDD1833_4	IRQ14	P07_1	—	POE8#	IIC_SCL0 / XSPI0_RSTO0#	MCLK00	—	—	—	ENCIFCK04 / ENCIFCK12 / SCKE04 / SCKE08 / HDSL05_CLK2
AF5	VDD1833_4	—	P05_5	—	—	XSPI0_DS	—	—	—	—	TST_OUT07 / HDSL04_MOSI1
AF6	—	VSS	—	—	—	—	—	—	—	—	—
AF7	VDD1833_4	IRQ12	P06_7	—	POE4# / GTETRGD	GMAC1_MDC / IIC_SCL2	—	—	—	—	HDSL05_MISO1
AF8	VDD1833_4	IRQ13	P07_0	—	—	GMAC1_MDIO / IIC_SDA2 / XSPI0_RESET1#	—	—	—	—	HDSL05_MOSI1
AF9	VDD33	SEI	P04_5	—	MTCLKC / MTIOC0C / GTIOC03_3A / CMTW1_TIC1	IIC_SCL0	—	—	ADTRG1#	—	ENCIFCK03 / SCKE03 / HDSL03_SEL2
AF10	—	VSS	—	—	—	—	—	—	—	—	—
AF11	VDD33	—	P04_1	—	MTIOC7A / GTIOC03_1A / CMTW0_TIC1	—	—	—	—	—	SI04# / HDSL03_SEL1
AF12	VDD1833_5	IRQ10	P02_3	—	MTIOC6C / MTIOC1B / GTIOC01_4B	ETH3_COL / IIC_SCL0 / IIC_SCL2 / XSPI1_IO7	MDAT22	—	USB_OVRCUR	—	ENCIFDI01 / RXDE01 / HDSL01_MOSI2
AF13	VDD1833_5	IRQ6	P01_0	—	MTIOC3A / MTIOC1A / GTIOC00_4A / GTIOC00_2B	IIC_SCL1 / XSPI1_CKP	—	—	—	—	ENCIFDO00 / ENCIFDO04 / TXDE00 / TXDE04 / HDSL00_MISO2
AF14	—	VSS	—	—	—	—	—	—	—	—	—
AF15	VDD33	IRQ3	P00_4	D4	MTIOC4B / GTIOC00_2A	—	—	—	ADTRG2#	—	TST_OUT01 / HDSL00_MISO1
AF16	—	VSS	—	—	—	—	—	—	—	—	—
AF17	—	—	—	—	—	—	—	—	USB_QDP	—	—
AF18	—	VSS	—	—	—	—	—	—	—	—	—
AF19	—	—	—	—	—	—	—	—	PCIE_RXDN_L1	—	—
AF20	—	—	—	—	—	—	—	—	PCIE_RXDN_L0	—	—
AF21	—	VSS	—	—	—	—	—	—	—	—	—
AF22	—	—	—	—	—	—	—	—	PCIE_TXDN_L1	—	—
AF23	—	—	—	—	—	—	—	—	PCIE_TXDN_L0	—	—
AF24	—	VSS	—	—	—	—	—	—	—	—	—
AF25	—	AVDDREF_ADC2	—	—	—	—	—	—	—	—	—
AF26	—	—	—	—	—	—	—	—	AN204	—	—
AF27	—	—	—	—	—	—	—	—	AN200	—	—
AG1	—	VSS	—	—	—	—	—	—	—	—	—
AG2	VDD33	—	P09_1	—	—	—	MDAT12	—	—	—	DUEI12 / HDSL07_CLK1
AG3	VDD33	TMS	P08_1	—	—	—	—	—	—	—	DUEI10 / HDSL06_MISO1
AG4	VDD1833_4	IRQ5	P05_3	—	—	XSPI0_CS0#	—	—	—	—	SI06# / HDSL04_SEL1
AG5	VDD1833_4	IRQ15	P07_2	—	POE10#	IIC_SDA0 / XSPI0_RSTO1#	MDAT00	—	—	—	ENCIFOE04 / ENCIFOE12 / DEE04 / DEE08 / HDSL05_SEL2

Table 1.20 List of pins and pin functions (RZ/T2H 729-pin FCBGA) (24 of 24)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
AG6	VDD1833_4	—	P07_5	—	—	IIC_SCL2 / XSPI0_ECS0#	MCLK02	—	—	—	ENCIFCK05 / SCKE05 / HDSL06_LINK
AG7	VDD1833_4	IRQ6	P05_4	—	—	IIC_SDA2 / XSPI0_CS1#	—	—	—	—	DUEI07 / HDSL04_MISO1
AG8	VDD1833_4	MDD	P06_6	—	—	XSPI0_RESET0#	—	—	—	—	—
AG9	VDD33	—	P03_6	—	MTIOC3C / MTIOC1A / GTIOC02_4B	IIC_SDA2	—	—	—	—	ENCIFDI02 / RXDE02 / HDSL03_LINK
AG10	VDD33	—	P04_3	—	MTIOC7B / GTIOC03_2A / CMTW1_TIC0	—	—	—	—	—	TST_OUT05 / HDSL03_MOSI1
AG11	VDD33	—	P04_4	—	MTIOC7D / GTIOC03_2B / CMTW1_TOC0	—	—	—	ADTRG0#	—	SI05# / HDSL03_CLK2
AG12	VDD1833_5	—	P01_1	—	MTIOC3C / MTIOC8A / GTIOC00_4B	XSPI1_CS0#	MCLK20	—	—	—	ENCIFDI00 / ENCIFDI04 / RXDE00 / RXDE04 / HDSL00_MOSI2
AG13	VDD1833_5	—	P01_4	—	MTIOC7A / MTIC5V / GTIOC01_1A / GTIOC04_1A	XSPI1_IO0	—	—	—	—	SI02# / HDSL01_CLK1
AG14	VDD33	IRQ4	P00_6	—	MTCLKA / MTIOC8D / GTIOC00_3A	IIC_SCL0	—	—	USB_OVRCUR	—	ENCIFCK00 / ENCIFCK04 / SCKE00 / SCKE04 / HDSL00_CLK2
AG15	VDD33	IRQ2	P00_3	D3	MTIOC4C / GTIOC00_1B	ETH3_COL	—	—	ADTRG1#	—	DUEI01 / HDSL00_SEL1
AG16	—	VSS	—	—	—	—	—	—	—	—	—
AG17	—	—	—	—	—	—	—	—	USB_QDM	—	—
AG18	—	VSS	—	—	—	—	—	—	—	—	—
AG19	—	—	—	—	—	—	—	—	PCIE_RXDP_L1	—	—
AG20	—	—	—	—	—	—	—	—	PCIE_RXDP_L0	—	—
AG21	—	VSS	—	—	—	—	—	—	—	—	—
AG22	—	—	—	—	—	—	—	—	PCIE_TXDP_L1	—	—
AG23	—	—	—	—	—	—	—	—	PCIE_TXDP_L0	—	—
AG24	—	VSS	—	—	—	—	—	—	—	—	—
AG25	—	—	—	—	—	—	—	—	AN205	—	—
AG26	—	—	—	—	—	—	—	—	AN203	—	—
AG27	—	VSS	—	—	—	—	—	—	—	—	—

1.7 RZ/N2H FCBGA 576 Pin Assignments

Figure 1.3 shows the pin arrangement. Table 1.21 shows the lists of pins and the pin functions.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
A	VSS	P09_4	P10_6	P10_7	P11_0	P13_4	P13_7	P14_0	P14_7	P14_6	P17_1	P18_5	P18_3	P20_5	P20_1	P21_2	P21_3	P22_4	P24_5	P25_0	P25_2	P24_6	P27_3	VSS	A
B	P09_5	P10_1	P10_2	P10_5	VSS	P12_7	P13_0	P12_4	P14_4	VSS	P17_2	P18_1	P18_4	P20_0	VSS	P20_4	P22_2	P22_5	VSS	P25_1	P24_7	P26_5	P27_1	P27_2	B
C	P09_6	P10_0	P10_3	P09_7	P10_4	P13_1	P13_5	P12_0	P14_1	P14_3	P17_0	P18_2	P18_6	P21_6	P21_7	P21_1	P22_1	P22_6	P25_6	P26_2	P25_3	P26_6	P27_5	P27_6	C
D	DDR_DQA12	DDR_DQA9	VSS	DDR_DQA13	VSS	P12_6	P12_5	VSS	P14_5	P16_7	P17_3	P17_6	VSS	P21_5	P20_7	P20_6	VSS	P22_7	P26_3	P25_5	VSS	P27_0	P27_4	P30_4	D
E	VSS	DDR_DQA8	DDR_DQA11	DDR_DQA14	DDR_DQSA_C1	P13_6	P12_1	P12_2	P14_2	P16_5	P16_6	P17_7	P17_4	P20_3	P21_0	P21_4	P22_3	P26_1	P25_4	P25_7	P26_4	P26_7	P29_1	P30_2	E
F	DDR_DQA15	VSS	DDR_DQA10	DDR_DMIA1	DDR_DQSA_T1	DDR_VDDQ	P12_3	P13_2	P13_3	VSS	P17_5	P18_0	P18_7	P20_2	VSS	P22_0	P23_0	P26_0	VSS	P30_0	P29_6	P29_4	VSS	P30_7	F
G	DDR_DQA6	DDR_DQA5	VSS	VSS	VSS	DDR_VDDQ	VDD33	VSS	VDDP_18_33	VDD33	VDD1833_6	VDD1833_7	VDD1833_0	VDD1833_0	VDDP_18_1	VSS_PLL0	VSS	VSS	P31_0	P30_5	P29_5	P29_3	P30_1	P31_1	G
H	DDR_DQA4	VSS	DDR_DMIA0	DDR_DQA7	DDR_DQSA_C0	DDR_VDDQ	VDD33	VSS_PLL2	DVDD08A_TSU	VDDP_18_33	VDDP_18_7	VDD18_PLL3	VSS_PLL3	VDDP_18_0	VDDP_18_33	VDD18_PLL0	VSS	VSS	VDD1833_1	P29_2	VSS	P29_7	P30_3	P30_6	H
J	DDR_DQA2	DDR_DQA0	DDR_DQA1	DDR_DQA3	DDR_DQSA_T0	VSS	VDD33	VDD18_PLL2	AVDD18A_TSU	VDDP_18_6	VDD08	VDD08_PLL3	VSS	VDD08	VSS	VDD08_PLL0	VDD33	VSS	VDD1833_1	P31_6	P31_4	P31_3	P31_5	P31_2	J
K	VSS	DDR_CAA1	VSS	VSS	VSS	DDR_CKEA1	VSS	VDD08_PLL2	VDD08	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VDDP_18_2	VDD1833_2	VDD1833_2	P34_1	P34_5	P34_2	P34_4	P34_6	K
L	DDR_CKA_C	DDR_CAA3	DDR_CAA4	DDR_CSA0	DDR_CSA1	DDR_CKEA0	VSS	VSS	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VDDP_18_33	VSS	VSS	P33_2	P34_3	P34_0	P33_7	VSS	L
M	DDR_CKA_T	VSS	DDR_CAA2	VSS	DDR_CAA5	VSS	DDR_VDDQ	DDR_VAA	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VDD33	VSS	VSS	P33_3	P33_6	VSS	VSS	EXTCLKIN	M
N	VSS	DDR_CKEB1	DDR_CAB0	DDR_CAA0	DDR_VDDQ	VSS	DDR_RESET_N	DDR_ATEST	VSS	VDD08	VSS	VDD08	VSS	VDD08	VSS	VDD08	VDDP_18_3	VDD1833_3	VDD1833_3	P33_4	P33_5	XTALSEL	XTAL	EXTAL	N
P	DDR_CKB_T	DDR_CKEB0	VSS	VSS	DDR_VDDQ	VSS	DDR_DTEST	DDR_ZN	VDD08	VDD08	VSS	VDD08	VDD18_PLL4	VDD08	VSS	VSS	VDD33_X	VDDP_18_X	OTPVDD08	OTPVDD18	VSS	VSS	VSS	VSS	P
R	DDR_CKB_C	DDR_CAB1	DDR_VDDQ	DDR_CAB2	DDR_CAB5	DDR_CSB0	VSS	VSS	VDD08	VDD08	VSS	VSS	VSS_PLL4	PCIE_VDD08A_L1	PCIE_VDD08A_L0	VSS	VSS	VSS	VSS	VSS	AVSSIO_ADC0	AVDDREF_ADC0	AN002	AN000	R
T	DDR_CAB3	VSS	DDR_CAB4	DDR_DQB0	VSS	DDR_CSB1	VSS	VSS	VSS	VSS_PLL1	VDD18_PLL1	VDD08_PLL1	VDD08_PLL4	PCIE_VDD08A_L1	PCIE_VDD08A_L0	PCIE_VDD18A_L1	PCIE_VDD18A_L0	VSS	VSS	AVDD_ADC0	AVSSIO_ADC0	AVDDIO_ADC0	AN003	AN001	T
U	DDR_DQB2	DDR_DQB1	VSS	DDR_DQB3	DDR_DQSB_T0	VSS	VSS	VSS	VSS	VDDP_18_33	VDD1833_4	VSS	VDDP_18_5	VSS	USB_USVDD18	PCIE_VDD18A_L1	PCIE_VDD18A_L0	VSS	VSS	AVSS_ADC0	AVDDIO_ADC1	AVDDREF_ADC1	AN100	AN103	U
V	VSS	DDR_DQB4	DDR_DQB7	DDR_DMIB0	DDR_DQSB_C0	VSS	TRST#	VDD33	VDD33	VDDP_18_4	VDD1833_4	VDDP_18_33	VDD1833_5	VSS	USB_USVDD18	USB_USVDD33	USB_USVDD33	VSS	VSS	AVSS_ADC1	AVSSIO_ADC1	AVSSIO_ADC1	AN102	AN101	V
W	DDR_DQB6	DDR_VDDQ	DDR_DQB5	VSS	VSS	VSS	P06_5	P06_3	P03_4	VSS	P01_0	P00_2	VDD1833_5	USB_USDVDD	USB_USDVDD	VSS	VSS	PCIE_VDD18A_CMN	VSS	AVDD_ADC1	AVSSIO_ADC2	AVSSIO_ADC2	AN206	AN210	W
Y	DDR_DQB8	VSS	DDR_DQB15	VSS	DDR_DQSB_T1	VSS	P06_2	P06_4	P03_1	P02_3	P02_4	P01_2	VDD33	USB_VUBUSIN	USB_TXRTUNE	PCIE_REFCLK_N1	PCIE_REFCLK_P0	PCIE_VDD18A_CMN	VSS	AVDD_ADC2	AVDDIO_ADC2	AN209	AN208	AN202	Y
AA	DDR_DQB14	DDR_DQB9	DDR_DMIB1	DDR_DQB10	DDR_DQSB_C1	BSCANP	P05_1	VSS	P03_3	P02_1	P02_2	P02_0	P00_1	USB_OTG_ID	VSS	PCIE_REFCLK_P1	PCIE_REFCLK_N0	VSS	VSS	AVSS_ADC2	AVDDREF_ADC2	AN213	AN201	AN200	AA
AB	DDR_DQB12	VSS	DDR_DQB13	DDR_DQB11	VSS	P05_2	P06_1	P05_4	P03_2	P02_6	P01_1	P01_7	P00_0	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	AN203	AVSSIO_ADC2	AN212	AB
AC	MDX	P08_5	P08_4	RES#	P08_3	P06_0	P05_7	P06_7	P02_7	VSS	P01_5	P01_3	P00_4	USB_QDP	VSS	PCIE_RXDN_L1	PCIE_RXDN_L0	VSS	PCIE_TXDN_L1	PCIE_TXDN_L0	VSS	AN211	AN204	AN207	AC
AD	VSS	P08_6	P08_1	P08_2	P05_6	P05_5	P06_6	P05_3	P03_0	P02_5	P01_6	P01_4	P00_3	USB_QDM	VSS	PCIE_RXDP_L1	PCIE_RXDP_L0	VSS	PCIE_TXDP_L1	PCIE_TXDP_L0	VSS	AN205	AN214	AVSSIO_ADC2	AD
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	

Figure 1.3 Pin arrangement (RZ/N2H 576-pin FCBGA) (top view)

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (1 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
A1	—	VSS	—	—	—	—	—	—	—	—	—
A2	VDD33	—	P09_4	D13	MTIOC6B / GTIOC04_0A / GTIOC10_0A	—	MCLK70	DISP_CLK	—	—	DUEI13 / HDSL07_MOSI1
A3	VDD33	IRQ0	P10_6	A3	MTIOC0B / GTIOC05_0A	DE0	MCLK21	DISP_DATAR6	—	—	HDSL08_MOSI1 / POUTA

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (2 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
A4	VDD33	IRQ9	P10_7	A4	MTIC5U / GTIOC05_0B / GTIOC00_3A	SCK1	MDAT21	DISP_DATAR7	—	—	HDSL08_CLK2 / POUTB
A5	VDD33	IRQ13	P11_0	A5	GTIOC00_3B	ESC_RESETOUT# / RXD1/SCL1/MISO1	MCLK22	DISP_DATAG0	—	—	HDSL08_SEL2 / POUTZ
A6	VDD1833_6	—	P13_4	D28	GTIOC03_3B	RXD3/SCL3/MISO3 / SPI_SSL31	MCLK40	—	—	—	ENCIFCK13 / SCKE09 / HDSL10_SEL2
A7	VDD1833_6	IRQ14	P13_7	D31	GTIOC06_4A / GTIOC04_3B	CTS3#	MDAT41	—	—	—	ENCIFDI13 / RXDE09 / HDSL11_LINK
A8	VDD1833_6	IRQ5	P14_0	A0	GTIOC06_4B	ETHSW_PTPOUT2 / ESC_SYNC0 / DE3	MCLK42	—	—	—	HDSL11_SMPL
A9	VDD33	IRQ9	P14_7	—	POE11# / GTIOC09_3A / CMTW0_TOC1	ESC_I2CDATA / IIC_SDA0 / SD0_IOVS	MCLK32	—	—	—	SI02# / HDSL11_MISO2
A10	VDD33	IRQ8	P14_6	—	POE10# / GTIOC06_2B / GTIOC09_2B / CMTW0_TIC1	ESC_I2CCLK / DE4 / IIC_SCL0 / SD0_PWEN	—	DISP_DATAG6	—	—	TST_OUT02 / HDSL11_SEL2
A11	VDD1833_7	IRQ13	P17_1	—	GTIOC03_2A	SD1_DATA2	—	—	—	—	DUEI06 / HDSL13_CLK2
A12	VDD33	IRQ2	P18_5	A13	GTIOC07_3B / GTADSM04_1	SS1#/CTS1#/ RTS1# / CANTX1	MDAT11	DISP_DATAB6	—	—	ENCIFOE13 / ENCIFOE14 / DEE09 / DEE10 / HDSL14_MISO2
A13	VDD33	IRQ0	P18_3	A11	GTADSM03_1 / RTCAT1HZ	ETH1_COL / GMAC1_MDIO / RXD1/SCL1/MISO1 / CANTX0 / SD1_IOVS	MDAT10	DISP_DATAB4	—	—	HDSL14_CLK2
A14	VDD1833_0	—	P20_5	—	—	ETH0_TXEN	—	—	—	—	DUEI11 / HDSL00_LINK
A15	VDD1833_0	MDV	P20_1	—	—	ETH0_TXD0	—	—	—	—	—
A16	VDD1833_0	—	P21_2	—	—	ETH0_RXD3 / CANTXDP0	—	—	—	—	SI12# / HDSL00_MOSI1
A17	VDD1833_0	—	P21_3	—	—	ETH0_RXDV	—	—	—	—	DUEI13 / HDSL00_CLK2
A18	VDD33	IRQ6	P22_4	A21	GTETRGD	ETH0_COL / SS5#/ CTS5#/RTS5# / CANTXDP0	—	—	—	—	TST_OUT14 / HDSL01_MOSI1
A19	VDD1833_1	—	P24_5	—	—	ETH1_TXCLK	—	—	—	—	HDSL03_CLK1
A20	VDD1833_1	MD2	P25_0	—	—	ETH1_TXD2 / CANRXDP0	—	—	—	—	—
A21	VDD1833_1	MDW1	P25_2	—	—	ETH1_TXEN	—	—	—	—	—
A22	VDD1833_1	MD0	P24_6	—	—	ETH1_TXD0	—	—	—	—	—
A23	VDD33	—	P27_3	—	MTIOC2A / GTIOC08_3A / GTIOC02_1A	GMAC1_PTPTRG1 / SCK0 / CANRXDP1 / SPI_MOSI0	—	—	—	HSPI_IO1	ENCIFCK14 / SCKE10 / HDSL04_MISO2
A24	—	VSS	—	—	—	—	—	—	—	—	—
B1	VDD33	—	P09_5	D14	MTIOC6D / GTIOC04_0B / GTIOC10_0B	—	MDAT70	DISP_HSYNC	—	—	TST_OUT13 / HDSL07_CLK2
B2	VDD33	IRQ7	P10_1	WAIT#	MTIOC7D / GTIOC04_2B / GTIOC10_2B	SCK0	MDAT72	DISP_DATAR1	—	—	SI14# / HDSL08_LINK
B3	VDD33	IRQ1	P10_2	CS0#	MTCLKC / MTIOC2A / GTIOC04_3A / GTIOC10_3A	RXD0/SCL0/MISO0	MCLK10 / MCLK00	DISP_DATAR2	—	—	ENCIFCK04 / SCKE04 / HDSL08_SMPL
B4	VDD33	—	P10_5	A2	MTIOC1B / MTIOC0A / GTIOC04_4B	CTS0#	MDAT11 / MDAT01	DISP_DATAR5	—	—	ENCIFDI04 / RXDE04 / HDSL08_MISO1
B5	—	VSS	—	—	—	—	—	—	—	—	—

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (3 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
B6	VDD1833_6	IRQ2	P12_7	D23	GTIOC05_4B / CMTW1_TOC1	CTS2# / SD0_DATA5	MDAT10	—	—	—	ENCIFDI05 / RXDE05 / HDSL10_CLK1
B7	VDD1833_6	—	P13_0	D24	GTIOC02_3A	DE2 / SPI_RSPCK3 / SD0_DATA6	MCLK00	—	—	—	ENCIFCK12 / ENCIFCK03 / SCKE08 / SCKE03 / HDSL10_SEL1
B8	VDD1833_6	IRQ1	P12_4	D20	GTIOC05_3A / CMTW1_TIC0	RXD2/SCL2/MISO2 / SD0_DATA2	MCLK02	—	—	—	ENCIFCK05 / SCKE05 / HDSL09_MOSI2
B9	VDD33	—	P14_4	DACK	POE4# / GTIOC06_1B / GTIOC09_1B / GTIOC06_3A / CMTW0_TIC0	ESC_IRQ / SS4#/ CTS4#/RTS4# / SD1_WP	—	DISP_DATAG4	—	MBX_HINT#	ENCIFDO00 / TXDE00 / HDSL11_MOSI1
B10	—	VSS	—	—	—	—	—	—	—	—	—
B11	VDD1833_7	IRQ14	P17_2	—	GTIOC03_2B	SD1_DATA3	—	—	—	—	TST_OUT06 / HDSL13_SEL2
B12	VDD33	IRQ15	P18_1	A9	GTADSM02_1 / GTIOC07_3A	ESC_LEDERR / CTS0# / CANTXDP0 / SD1_IOVS	—	DISP_DATAB2	—	—	SI08# / HDSL14_MISO1
B13	VDD33	IRQ1	P18_4	A12	GTIOC07_3A / GTADSM04_0	ESC_LEDSTER / TXD1/SDA1/MOSI1 / CANRX1	MCLK11	DISP_DATAB5	—	—	ENCIFCK13 / ENCIFCK14 / SCKE09 / SCKE10 / HDSL14_SEL2
B14	VDD1833_0	—	P20_0	—	—	ETH0_TXCLK	—	—	—	—	HDSL15_MOSI2
B15	—	VSS	—	—	—	—	—	—	—	—	—
B16	VDD1833_0	—	P20_4	—	—	ETH0_TXD3 / CANTX0	—	—	—	—	—
B17	VDD33	—	P22_2	A23	GTETRGB	ETH0_RXER / RXD5/SCL5/MISO5 / CANRX0	—	—	—	—	HDSL01_SEL1
B18	VDD33	IRQ7	P22_5	A20	GTETRGSA	GMAC0_PTPTRG0 / ESC_LATCH0 / CTS5# / CANRX1 / SD0_CD	—	—	—	—	SI14# / HDSL01_CLK2
B19	—	VSS	—	—	—	—	—	—	—	—	—
B20	VDD1833_1	MDW0	P25_1	—	—	ETH1_TXD3 / CANTXDP0	—	—	—	—	—
B21	VDD1833_1	MD1	P24_7	—	—	ETH1_TXD1	—	—	—	—	—
B22	VDD1833_1	IRQ12	P26_5	—	—	CANTX0	—	—	—	—	ENCIFCK01 / SCKE01 / HDSL04_CLK1
B23	VDD33	IRQ2	P27_1	—	GTIOC02_0A	ETH1_COL / CANRX1 / SPI_SSL03	—	—	—	HSPI_CS#	HDSL04_CLK2
B24	VDD33	IRQ3	P27_2	—	GTIOC02_0B	GMAC1_PTPTRG0 / ESC_LEDERR / CANTX1 / SPI_RSPCK0	—	—	—	HSPI_IO0	HDSL04_SEL2
C1	VDD33	—	P09_6	D15	MTIOC7A / GTIOC04_1A / GTIOC10_1A	—	MCLK71	DISP_VSYNC	—	—	SI13# / HDSL07_SEL2
C2	VDD33	IRQ4	P10_0	WE1#	MTIOC7B / GTIOC04_2A / GTIOC10_2A	—	MCLK72	DISP_DATAR0	—	—	TST_OUT14 / HDSL07_MOSI2
C3	VDD33	IRQ2	P10_3	RD#	MTCLKD / MTIOC2B / GTIOC04_3B / GTIOC10_3B	TXD0/SDA0/MOSI0	MDAT10 / MDAT00	DISP_DATAR3	—	—	ENCIFOE04 / DEE04 / HDSL08_CLK1
C4	VDD33	—	P09_7	WE0#	MTIOC7C / GTIOC04_1B / GTIOC10_1B	—	MDAT71	DISP_DE	—	—	DUEI14 / HDSL07_MISO2

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (4 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
C5	VDD33	IRQ3	P10_4	A1	MTIOC1A / GTIOC04_4A	SS0#/CTS0#/RTS0#	MCLK11 / MCLK01	DISP_DATAR4	—	—	ENCIFDO04 / TXDE04 / HDSL08_SEL1
C6	VDD1833_6	—	P13_1	D25	GTIOC02_3B	SPI_MOSI3 / SD0_DATA7	MDAT00	—	—	—	ENCIFOE12 / ENCIFOE03 / DEE08 / DEE03 / HDSL10_MISO1
C7	VDD1833_6	IRQ4	P13_5	D29	GTIOC06_3A	TXD3/SDA3/MOSI3 / SPI_SSL32	MDAT40	—	—	—	ENCIFOE13 / DEE09 / HDSL10_MISO2
C8	VDD1833_6	—	P12_0	D16	MTIC5V / GTIOC05_1A / CMTW0_TIC0	CANRX1 / SD0_CLK	—	—	—	—	DUEI01 / HDSL09_MOSI1
C9	VDD33	—	P14_1	RD/WR#	GTIOC06_0A / GTIOC09_0A / GTIOC05_3A / RTCAT1HZ	SCK4 / SD0_CD	MDAT42	DISP_DATAG1	—	—	DUEI02 / HDSL11_CLK1
C10	VDD33	IRQ6	P14_3	DREQ	POE0# / GTIOC06_1A / GTIOC09_1A	ESC_LINKACT2 / TXD4/SDA4/MOSI4 / SD1_CD	—	DISP_DATAG3	—	—	ENCIFOE00 / DEE00 / HDSL11_MISO1
C11	VDD1833_7	IRQ12	P17_0	—	GTIOC03_1B	SD1_DATA1	—	—	—	—	SI05# / HDSL13_MOSI1
C12	VDD33	SEI	P18_2	A10	GTADSM03_0 / GTIOC07_3B	ETH1_CRS / GMAC1_MDC / SCK1 / CANRX0 / SD1_PWEN	MCLK10	DISP_DATAB3	—	—	HDSL14_MOSI1
C13	VDD33	IRQ3	P18_6	A14	GTIOC07_4A / GTADSM05_0	CTS1# / CANRXDP1	MCLK12	DISP_DATAB7	—	—	ENCIFDO13 / ENCIFDO14 / TXDE09 / TXDE10 / HDSL14_MOSI2
C14	VDD1833_0	—	P21_6	—	—	ETHSW_PHYLINK0 / ESC_PHYLINK0 / CANRXDP1	—	—	—	—	HDSL00_MOSI2
C15	VDD1833_0	ETH0_REFCLK / RMII0_REFCLK	P21_7	—	—	CANTXDP1	—	—	—	—	HDSL01_LINK
C16	VDD1833_0	—	P21_1	—	—	ETH0_RXD2 / CANRXDP0	—	—	—	—	TST_OUT12 / HDSL00_MISO1
C17	VDD33	—	P22_1	—	GTETRGA	ETH0_TXER / TXD5/ SDA5/MOSI5 / CANTX0	—	—	—	—	HDSL01_CLK1
C18	VDD33	IRQ8	P22_6	A19	GTETRGSB	GMAC0_PTPTRG1 / ESC_LATCH1 / DE5 / CANTX1 / SD0_WP	—	—	—	—	DUEI15 / HDSL01_SEL2
C19	VDD1833_1	—	P25_6	—	—	ETH1_RXD2 / CANRX1	—	—	—	—	DUEI04 / HDSL03_CLK2
C20	VDD1833_1	—	P26_2	—	—	GMAC1_MDIO / ETHSW_MDIO / ESC_MDIO / CANTXDP1	—	—	—	—	HDSL04_LINK
C21	VDD1833_1	—	P25_3	—	—	ETH1_RXCLK	—	—	—	—	DUEI03 / HDSL03_SEL1
C22	VDD33	SEI	P26_6	CS2#	—	ETH1_TXER / ESC_RESETOUT# / CANRX0	—	—	—	—	ENCIFOE01 / DEE01 / HDSL04_SEL1
C23	VDD33	—	P27_5	—	MTIOC1A / GTIOC08_4A / GTIOC02_2A	TXD0/SDA0/MOSI0 / SPI_SSL00	—	—	—	HSPI_IO3	ENCIFDO14 / TXDE10 / HDSL05_LINK
C24	VDD33	—	P27_6	—	MTIOC1B / GTIOC08_4B / GTIOC02_2B	—	—	—	—	HSPI_CK	ENCIFDI14 / RXDE10 / HDSL05_SMPL
D1	—	—	—	—	—	—	—	—	DDR_DQA12	—	—
D2	—	—	—	—	—	—	—	—	DDR_DQA9	—	—
D3	—	VSS	—	—	—	—	—	—	—	—	—

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (5 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
D4	—	—	—	—	—	—	—	—	DDR_DQA13	—	—
D5	—	VSS	—	—	—	—	—	—	—	—	—
D6	VDD1833_6	—	P12_6	D22	GTIOC05_4A / GTIOC01_3B / CMTW1_TIC1	SS2#/CTS2#/ RTS2# / SD0_DATA4	MCLK10	—	—	—	ENCIFDO05 / TXDE05 / HDSL10_SMPL
D7	VDD1833_6	—	P12_5	D21	GTIOC05_3B / GTIOC01_3A / CMTW1_TOC 0	TXD2/SDA2/MOSI2 / SD0_DATA3	MDAT02	—	—	—	ENCIFOE05 / DEE05 / HDSL10_LINK
D8	—	VSS	—	—	—	—	—	—	—	—	—
D9	VDD33	—	P14_5	TEND	POE8# / GTIOC06_2A / GTIOC09_2A / GTIOC06_3B / CMTW0_TOC 0	ESC_RESETOUT# / CTS4#	—	DISP_DATAG5	—	—	ENCIFDI00 / RXDE00 / HDSL11_CLK2
D10	VDD1833_7	—	P16_7	—	GTIOC03_1A	SD1_DATA0	—	—	—	—	TST_OUT05 / HDSL13_MISO1
D11	VDD1833_7	IRQ15	P17_3	—	GTETRGA	—	—	—	—	—	SI06# / HDSL13_MISO2
D12	VDD33	—	P17_6	WE2#	GTADSM01_0 / GTETRGD / CMTW1_TIC1	ETHSW_PTPOUT0 / ESC_SYNC0 / RXD0/SCL0/MISO0 / SD1_PWEN	—	DISP_DATAG7	—	—	SI07# / HDSL14_SMPL
D13	—	VSS	—	—	—	—	—	—	—	—	—
D14	VDD1833_0	—	P21_5	—	—	GMAC0_MDIO / ETHSW_MDIO / ESC_MDIO / CANTX1	—	—	—	—	SI13# / HDSL00_MISO2
D15	VDD1833_0	—	P20_7	—	—	ETH0_RXD0	—	—	—	—	SI11# / HDSL00_CLK1
D16	VDD1833_0	—	P20_6	—	—	ETH0_RXCLK	—	—	—	—	TST_OUT11 / HDSL00_SMPL
D17	—	VSS	—	—	—	—	—	—	—	—	—
D18	VDD33	IRQ9	P22_7	A18	GTIOC06_0A	ETH1_CRS / ETHSW_TDMAOUT 2 / ESC_LINKACT0 / CANRXDP1	—	—	—	—	TST_OUT15 / HDSL01_MISO2
D19	VDD1833_1	—	P26_3	—	—	ETHSW_PHYLINK1 / ESC_PHYLINK1	—	—	—	—	HDSL04_SMPL
D20	VDD1833_1	—	P25_5	—	—	ETH1_RXD1	—	—	—	—	SI03# / HDSL03_MOSI1
D21	—	VSS	—	—	—	—	—	—	—	—	—
D22	VDD33	IRQ1	P27_0	CS5#	—	ETH1_CRS / CANTXDP0 / SPI_SSL02	—	—	—	HSPI_INT#	ENCIFDI01 / RXDE01 / HDSL04_MOSI1
D23	VDD33	—	P27_4	—	MTIOC2B / GTIOC08_3B / GTIOC02_1B	RXD0/SCL0/MISO0 / CANTXDP1 / SPI_MISO0	—	—	—	HSPI_IO2	ENCIFOE14 / DEE10 / HDSL04_MOSI2
D24	VDD1833_2	—	P30_4	—	GTIOC09_5B	ETH2_RXDV	—	—	—	—	ENCIFDI11 / RXDE11 / HDSL07_SEL1
E1	—	VSS	—	—	—	—	—	—	—	—	—
E2	—	—	—	—	—	—	—	—	DDR_DQA8	—	—
E3	—	—	—	—	—	—	—	—	DDR_DQA11	—	—
E4	—	—	—	—	—	—	—	—	DDR_DQA14	—	—
E5	—	—	—	—	—	—	—	—	DDR_DQSA_C 1	—	—
E6	VDD1833_6	—	P13_6	D30	GTIOC06_3B / GTIOC04_3A	SS3#/CTS3#/ RTS3# / SPI_SSL23	MCLK41	—	—	—	ENCIFDO13 / TXDE09 / HDSL10_MOSI2

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (6 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
E7	VDD1833_6	—	P12_1	D17	MTIC5W / GTIOC05_1B / CMTW0_TOC0	CANTX1 / SD0_CMD	—	—	—	—	TST_OUT01 / HDSL09_CLK2
E8	VDD1833_6	—	P12_2	D18	GTIOC05_2A / CMTW0_TIC1	CANRXDP1 / SD0_DATA0	—	—	—	—	SI01# / HDSL09_SEL2
E9	VDD33	—	P14_2	BS#	GTIOC06_0B / GTIOC09_0B / GTIOC05_3B	RXD4/SCL4/MISO4 / SD0_WP	—	DISP_DATAG2	—	—	ENCIFCK00 / SCKE00 / HDSL11_SEL1
E10	VDD1833_7	—	P16_5	—	GTIOC03_0A	SD1_CLK	—	—	—	—	SI04# / HDSL13_CLK1
E11	VDD1833_7	—	P16_6	—	GTIOC03_0B	SD1_CMD	—	—	—	—	DUEI05 / HDSL13_SEL1
E12	VDD33	—	P17_7	WE3#/AH#	GTADSM01_1 / CMTW1_TOC1	ETHSW_PTPOUT1 / ESC_SYNC1 / TXD0/SDA0/MOSI0 / SD1_IOVS	—	DISP_DATAB0	—	—	DUEI08 / HDSL14_CLK1
E13	VDD33	—	P17_4	A6 / DREQ	GTADSM00_0 / GTETRGB / CMTW1_TIC0	DE0 / CANRX0 / SD1_CD	—	—	—	—	DUEI07 / HDSL13_MOSI2
E14	VDD1833_0	—	P20_3	—	—	ETH0_TXD2 / CANRX0	—	—	—	—	—
E15	VDD1833_0	—	P21_0	—	—	ETH0_RXD1	—	—	—	—	DUEI12 / HDSL00_SEL1
E16	VDD1833_0	—	P21_4	—	—	GMAC0_MDC / ETHSW_MDC / ESC_MDC / CANRX1	—	—	—	—	TST_OUT13 / HDSL00_SEL2
E17	VDD33	IRQ5	P22_3	A22	GTETRGC	ETH0_CRS / SCK5 / CANRXDP0	—	—	—	—	DUEI14 / HDSL01_MISO1
E18	VDD1833_1	—	P26_1	—	—	GMAC1_MDC / ETHSW_MDC / ESC_MDC / CANRXDP1	—	—	—	—	HDSL03_MOSI2
E19	VDD1833_1	—	P25_4	—	—	ETH1_RXD0	—	—	—	—	TST_OUT03 / HDSL03_MISO1
E20	VDD1833_1	—	P25_7	—	—	ETH1_RXD3 / CANTX1	—	—	—	—	TST_OUT04 / HDSL03_SEL2
E21	VDD1833_1	ETH1_REFCLK / RMII1_REFCLK	P26_4	—	—	—	—	—	—	—	—
E22	VDD33	IRQ0	P26_7	CS3#	—	ETH1_RXER / ESC_LEDSTER / CANRXDP0 / SPI_SSL01	—	—	—	—	ENCIFDO01 / TXDE01 / HDSL04_MISO1
E23	VDD1833_2	—	P29_1	—	GTIOC09_0A	ETH2_TXCLK	—	—	—	—	ENCIFCK09 / SCKE09 / HDSL06_CLK1
E24	VDD1833_2	IRQ10	P30_2	—	GTIOC09_4B	ETH2_RXD2 / SPI_MOSI2	—	—	—	—	ENCIFOE11 / DEE11 / HDSL07_SMPL
F1	—	—	—	—	—	—	—	—	DDR_DQA15	—	—
F2	—	VSS	—	—	—	—	—	—	—	—	—
F3	—	—	—	—	—	—	—	—	DDR_DQA10	—	—
F4	—	—	—	—	—	—	—	—	DDR_DMIA1	—	—
F5	—	—	—	—	—	—	—	—	DDR_DQSA_T1	—	—
F6	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
F7	VDD1833_6	—	P12_3	D19	GTIOC05_2B / CMTW0_TOC1	SCK2 / CANTXDP1 / SD0_DATA1	—	—	—	—	HDSL09_MISO2

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (7 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
F8	VDD1833_6	IRQ3	P13_2	D26	—	SPI_MISO3 / SD0_RST#	MCLK01	—	—	—	ENCIFDO12 / ENCIFDO03 / TXDE08 / TXDE03 / HDSL10_MOSI1
F9	VDD1833_6	—	P13_3	D27	GTIOC03_3A	SCK3 / SPI_SSL30	MDAT01	—	—	—	ENCIFDI12 / ENCIFDI03 / RXDE08 / RXDE03 / HDSL10_CLK2
F10	—	VSS	—	—	—	—	—	—	—	—	—
F11	VDD33	—	P17_5	A7 / DACK	GTADSM00_1 / GTETRGC / CMTW1_TOC0	SCK0 / CANTX0 / SD1_WP	—	—	—	—	TST_OUT07 / HDSL14_LINK
F12	VDD33	IRQ7	P18_0	A8 / TEND	GTADSM02_0	ESC_LEDRUN / SS0#/CTS0#/ RTS0# / CANRXDP0 / SD1_PWEN	—	DISP_DATAB1	—	—	TST_OUT08 / HDSL14_SEL1
F13	VDD33	IRQ4	P18_7	A15	GTIOC07_4B / GTADSM05_1	ETHSW_PTPOUT3 / ESC_SYNC1 / DE1 / CANTXDP1	MDAT12	—	—	—	ENCIFDI13 / ENCIFDI14 / RXDE09 / RXDE10 / HDSL15_LINK
F14	VDD1833_0	—	P20_2	—	—	ETH0_TXD1	—	—	—	—	—
F15	—	VSS	—	—	—	—	—	—	—	—	—
F16	VDD1833_0	IRQ11	P22_0	—	—	—	—	—	—	—	HDSL01_SMPL
F17	VDD33	IRQ10	P23_0	A17	GTIOC06_0B	ETH1_COL / ETHSW_TDMAOUT3 / ESC_LINKACT1 / CANTXDP1	—	—	—	—	SI15# / HDSL01_MOSI2
F18	VDD1833_1	—	P26_0	—	—	ETH1_RXDV	—	—	—	—	SI04# / HDSL03_MISO2
F19	—	VSS	—	—	—	—	—	—	—	—	—
F20	VDD1833_2	—	P30_0	—	GTIOC09_3B	ETH2_RXD0	—	—	—	—	ENCIFDI10 / RXDE10 / HDSL06_MOSI2
F21	VDD1833_2	—	P29_6	—	GTIOC09_2B	ETH2_TXEN / SPI_SSL22	—	—	—	—	ENCIFOE10 / DEE10 / HDSL06_SEL2
F22	VDD1833_2	IRQ8	P29_4	—	GTIOC09_1B	ETH2_TXD2 / SPI_SSL20	—	—	—	—	ENCIFDI09 / RXDE09 / HDSL06_MOSI1
F23	—	VSS	—	—	—	—	—	—	—	—	—
F24	VDD1833_2	IRQ14	P30_7	—	—	ETHSW_PHYLINK2 / ESC_PHYLINK2 / SPI_MISO3 / SD1_IOVS	MCLK30	—	—	—	SI07# / HDSL07_CLK2
G1	—	—	—	—	—	—	—	—	DDR_DQA6	—	—
G2	—	—	—	—	—	—	—	—	DDR_DQA5	—	—
G3	—	VSS	—	—	—	—	—	—	—	—	—
G4	—	VSS	—	—	—	—	—	—	—	—	—
G5	—	VSS	—	—	—	—	—	—	—	—	—
G6	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
G7	—	VDD33	—	—	—	—	—	—	—	—	—
G8	—	VSS	—	—	—	—	—	—	—	—	—
G9	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
G10	—	VDD33	—	—	—	—	—	—	—	—	—
G11	—	VDD1833_6	—	—	—	—	—	—	—	—	—
G12	—	VDD1833_7	—	—	—	—	—	—	—	—	—
G13	—	VDD1833_0	—	—	—	—	—	—	—	—	—

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (8 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
G14	—	VDD1833_0	—	—	—	—	—	—	—	—	—
G15	—	VDDP_18_1	—	—	—	—	—	—	—	—	—
G16	—	VSS_PLL0	—	—	—	—	—	—	—	—	—
G17	—	VSS	—	—	—	—	—	—	—	—	—
G18	—	VSS	—	—	—	—	—	—	—	—	—
G19	VDD1833_2	ETH2_REFCLK / RMII2_REFCLK	P31_0	—	GTETRGSA	SPI_SSL30	—	—	—	—	HDSL07_SEL2
G20	VDD1833_2	—	P30_5	—	GTIOC09_6A	GMAC2_MDC / ETHSW_MDC / ESC_MDC / SPI_RSPCK3	—	—	—	—	DUEI07 / HDSL07_MISO1
G21	VDD1833_2	IRQ9	P29_5	—	GTIOC09_2A	ETH2_TXD3 / SPI_SSL21	—	—	—	—	ENCIFCK10 / SCKE10 / HDSL06_CLK2
G22	VDD1833_2	—	P29_3	—	GTIOC09_1A	ETH2_TXD1	—	—	—	—	ENCIFDO09 / TXDE09 / HDSL06_MISO1
G23	VDD1833_2	—	P30_1	—	GTIOC09_4A	ETH2_RXD1	—	—	—	—	ENCIFCK11 / SCKE11 / HDSL07_LINK
G24	VDD1833_2	IRQ13	P31_1	—	GTETRGSB	ETH2_RXER / SPI_SSL31	—	—	—	—	HDSL07_MISO2
H1	—	—	—	—	—	—	—	—	DDR_DQA4	—	—
H2	—	VSS	—	—	—	—	—	—	—	—	—
H3	—	—	—	—	—	—	—	—	DDR_DMIA0	—	—
H4	—	—	—	—	—	—	—	—	DDR_DQA7	—	—
H5	—	—	—	—	—	—	—	—	DDR_DQSA_C0	—	—
H6	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
H7	—	VDD33	—	—	—	—	—	—	—	—	—
H8	—	VSS_PLL2	—	—	—	—	—	—	—	—	—
H9	—	DVDD08A_TSU	—	—	—	—	—	—	—	—	—
H10	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
H11	—	VDDP_18_7	—	—	—	—	—	—	—	—	—
H12	—	VDD18_PLL3	—	—	—	—	—	—	—	—	—
H13	—	VSS_PLL3	—	—	—	—	—	—	—	—	—
H14	—	VDDP_18_0	—	—	—	—	—	—	—	—	—
H15	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
H16	—	VDD18_PLL0	—	—	—	—	—	—	—	—	—
H17	—	VSS	—	—	—	—	—	—	—	—	—
H18	—	VSS	—	—	—	—	—	—	—	—	—
H19	—	VDD1833_1	—	—	—	—	—	—	—	—	—
H20	VDD1833_2	—	P29_2	—	GTIOC09_0B	ETH2_TXD0	—	—	—	—	ENCIFOE09 / DEE09 / HDSL06_SEL1
H21	—	VSS	—	—	—	—	—	—	—	—	—
H22	VDD1833_2	—	P29_7	—	GTIOC09_3A	ETH2_RXCLK / SPI_SSL23	—	—	—	—	ENCIFDO10 / TXDE10 / HDSL06_MISO2
H23	VDD1833_2	IRQ11	P30_3	—	GTIOC09_5A	ETH2_RXD3 / SPI_MISO2	—	—	—	—	ENCIFDO11 / TXDE11 / HDSL07_CLK1

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (9 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
H24	VDD1833_2	—	P30_6	—	GTIOC09_6B	GMAC2_MDIO / ETHSW_MDIO / ESC_MDIO / SPI_MOSI3	—	—	—	—	TST_OUT07 / HDSL07_MOSI1
J1	—	—	—	—	—	—	—	—	DDR_DQA2	—	—
J2	—	—	—	—	—	—	—	—	DDR_DQA0	—	—
J3	—	—	—	—	—	—	—	—	DDR_DQA1	—	—
J4	—	—	—	—	—	—	—	—	DDR_DQA3	—	—
J5	—	—	—	—	—	—	—	—	DDR_DQSA_T 0	—	—
J6	—	VSS	—	—	—	—	—	—	—	—	—
J7	—	VDD33	—	—	—	—	—	—	—	—	—
J8	—	VDD18_PLL2	—	—	—	—	—	—	—	—	—
J9	—	AVDD18A_TS U	—	—	—	—	—	—	—	—	—
J10	—	VDDP_18_6	—	—	—	—	—	—	—	—	—
J11	—	VDD08	—	—	—	—	—	—	—	—	—
J12	—	VDD08_PLL3	—	—	—	—	—	—	—	—	—
J13	—	VSS	—	—	—	—	—	—	—	—	—
J14	—	VDD08	—	—	—	—	—	—	—	—	—
J15	—	VSS	—	—	—	—	—	—	—	—	—
J16	—	VDD08_PLL0	—	—	—	—	—	—	—	—	—
J17	—	VDD33	—	—	—	—	—	—	—	—	—
J18	—	VSS	—	—	—	—	—	—	—	—	—
J19	—	VDD1833_1	—	—	—	—	—	—	—	—	—
J20	VDD33	—	P31_6	A16 / TEND	POE11#	GMAC2_PTPTRG0 / ETHSW_TDMAOUT 0 / ESC_LEDRUN / SPI_MISO0	MDAT32	—	—	—	ENCIFCK15 / ENCIFCK01 / SCKE11 / SCKE01 / HDSL08_SEL1
J21	VDD33	—	P31_4	DREQ	POE8#	ETH2_CRS / ETHSW_PTPOUT2 / ESC_SYNC0 / SPI_RSPCK0 / SPI_SSL30	MCLK81 / MDAT31	—	—	HSPI_IO6	ENCIFDO09 / TXDE09 / HDSL08_SMPL / POUTB
J22	VDD33	—	P31_3	—	POE4#	ETH2_RXER / ETHSW_TDMAOUT 1 / ESC_LEDERR / SPI_SSL33	MDAT80 / MCLK31	—	—	HSPI_IO5	ENCIFOE09 / DEE09 / HDSL08_LINK
J23	VDD33	—	P31_5	DACK	POE10#	ETH2_COL / ETHSW_PTPOUT3 / ESC_SYNC1 / SPI_MOSI0 / SPI_SSL31	MDAT81 / MCLK32	—	—	HSPI_IO7	ENCIFDI09 / RXDE09 / HDSL08_CLK1 / POUTZ
J24	VDD33	—	P31_2	—	POE0#	ETH2_TXER / SPI_SSL32	MCLK80 / MDAT30	—	—	HSPI_IO4	ENCIFCK09 / SCKE09 / HDSL07_MOSI2 / POUTA
K1	—	VSS	—	—	—	—	—	—	—	—	—
K2	—	—	—	—	—	—	—	—	DDR_CAA1	—	—
K3	—	VSS	—	—	—	—	—	—	—	—	—
K4	—	VSS	—	—	—	—	—	—	—	—	—
K5	—	VSS	—	—	—	—	—	—	—	—	—
K6	—	—	—	—	—	—	—	—	DDR_CKEA1	—	—
K7	—	VSS	—	—	—	—	—	—	—	—	—
K8	—	VDD08_PLL2	—	—	—	—	—	—	—	—	—
K9	—	VDD08	—	—	—	—	—	—	—	—	—
K10	—	VDD08	—	—	—	—	—	—	—	—	—

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (10 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
K11	—	VSS	—	—	—	—	—	—	—	—	—
K12	—	VDD08	—	—	—	—	—	—	—	—	—
K13	—	VSS	—	—	—	—	—	—	—	—	—
K14	—	VDD08	—	—	—	—	—	—	—	—	—
K15	—	VSS	—	—	—	—	—	—	—	—	—
K16	—	VDD08	—	—	—	—	—	—	—	—	—
K17	—	VDDP_18_2	—	—	—	—	—	—	—	—	—
K18	—	VDD1833_2	—	—	—	—	—	—	—	—	—
K19	—	VDD1833_2	—	—	—	—	—	—	—	—	—
K20	VDD1833_3	—	P34_1	A23	GTADSM03_1 / GTIOC03_0B	ETH3_RXD0 / SPI_MISO2	—	—	—	—	ENCIFDI06 / RXDE06 / HDSL10_CLK1
K21	VDD1833_3	—	P34_5	CS3#	GTADSM05_1 / GTIOC03_2B	ETH3_RXDV / ESC_I2CCLK / TXD3/SDA3/MOSI3 / IIC_SCL1 / SPI_SSL23	—	—	ADTRG1#	—	ENCIFDI07 / RXDE07 / HDSL10_CLK2
K22	VDD1833_3	—	P34_2	A24	GTADSM04_0 / GTIOC03_1A	ETH3_RXD1 / SPI_SSL20	—	—	—	—	ENCIFCK07 / SCKE07 / HDSL10_SEL1
K23	VDD1833_3	—	P34_4	CS2#	GTADSM05_0 / GTIOC03_2A	ETH3_RXD3 / RXD3/ SCL3/MISO3 / SPI_SSL22 / SD1_IOVS	—	—	ADTRG0#	—	ENCIFDO07 / TXDE07 / HDSL10_MOSI1
K24	VDD1833_3	ETH3_REFCLK / RMII3_REFCLK	P34_6	CS5#	—	ETH1_RXER / ESC_I2CDATA / IIC_SDA1 / SPI_RSPCK3	—	—	ADTRG2#	—	DUEI08 / HDSL10_SEL2
L1	—	—	—	—	—	—	—	—	DDR_CKA_C	—	—
L2	—	—	—	—	—	—	—	—	DDR_CAA3	—	—
L3	—	—	—	—	—	—	—	—	DDR_CAA4	—	—
L4	—	—	—	—	—	—	—	—	DDR_CSA0	—	—
L5	—	—	—	—	—	—	—	—	DDR_CSA1	—	—
L6	—	—	—	—	—	—	—	—	DDR_CKEA0	—	—
L7	—	VSS	—	—	—	—	—	—	—	—	—
L8	—	VSS	—	—	—	—	—	—	—	—	—
L9	—	VSS	—	—	—	—	—	—	—	—	—
L10	—	VDD08	—	—	—	—	—	—	—	—	—
L11	—	VSS	—	—	—	—	—	—	—	—	—
L12	—	VDD08	—	—	—	—	—	—	—	—	—
L13	—	VSS	—	—	—	—	—	—	—	—	—
L14	—	VDD08	—	—	—	—	—	—	—	—	—
L15	—	VSS	—	—	—	—	—	—	—	—	—
L16	—	VDD08	—	—	—	—	—	—	—	—	—
L17	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
L18	—	VSS	—	—	—	—	—	—	—	—	—
L19	—	VSS	—	—	—	—	—	—	—	—	—
L20	VDD1833_3	—	P33_2	A16	GTADSM00_0	ETH3_TXCLK / SCK1 / SPI_RSPCK1 / SPI_SSL30	MCLK50	—	—	—	ENCIFCK01 / SCKE01 / HDSL09_MOSI1
L21	VDD1833_3	—	P34_3	A25	GTADSM04_1 / GTIOC03_1B	ETH3_RXD2 / SPI_SSL21 / SD1_PWEN	—	—	—	—	ENCIFOE07 / DEE07 / HDSL10_MISO1
L22	VDD1833_3	—	P34_0	A22	GTADSM03_0 / GTIOC03_0A	ETH3_RXCLK / SPI_MOSI2	—	—	—	—	ENCIFDO06 / TXDE06 / HDSL10_SMPL

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (11 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
L23	VDD1833_3	—	P33_7	A21	GTADSM02_1	ETH3_TXEN / SPI_RSPCK2	MDAT52	—	—	—	ENCIFOE06 / DEE06 / HDSL10_LINK
L24	—	VSS	—	—	—	—	—	—	—	—	—
M1	—	—	—	—	—	—	—	—	DDR_CKA_T	—	—
M2	—	VSS	—	—	—	—	—	—	—	—	—
M3	—	—	—	—	—	—	—	—	DDR_CAA2	—	—
M4	—	VSS	—	—	—	—	—	—	—	—	—
M5	—	—	—	—	—	—	—	—	DDR_CAA5	—	—
M6	—	VSS	—	—	—	—	—	—	—	—	—
M7	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
M8	—	DDR_VAA	—	—	—	—	—	—	—	—	—
M9	—	VSS	—	—	—	—	—	—	—	—	—
M10	—	VDD08	—	—	—	—	—	—	—	—	—
M11	—	VSS	—	—	—	—	—	—	—	—	—
M12	—	VDD08	—	—	—	—	—	—	—	—	—
M13	—	VSS	—	—	—	—	—	—	—	—	—
M14	—	VDD08	—	—	—	—	—	—	—	—	—
M15	—	VSS	—	—	—	—	—	—	—	—	—
M16	—	VDD08	—	—	—	—	—	—	—	—	—
M17	—	VDD33	—	—	—	—	—	—	—	—	—
M18	—	VSS	—	—	—	—	—	—	—	—	—
M19	—	VSS	—	—	—	—	—	—	—	—	—
M20	VDD1833_3	IRQ12	P33_3	A17	GTADSM00_1	ETH3_TXD0 / RXD1/ SCL1/MISO1 / SPI_MOSI1 / SPI_RSPCK0	MDAT50	—	PCIE_RSTOUT0B	—	ENCIFOE01 / DEE01 / HDSL09_CLK2
M21	VDD1833_3	IRQ15	P33_6	A20	GTADSM02_0	ETH3_TXD3 / TXD2/ SDA2/MOSI2 / SPI_SSL11 / SPI_SSL00	MCLK52	—	—	—	ENCIFCK06 / SCKE06 / HDSL09_MOSI2
M22	—	VSS	—	—	—	—	—	—	—	—	—
M23	—	VSS	—	—	—	—	—	—	—	—	—
M24	VDD33_X	EXTCLKIN	—	—	—	—	—	—	—	—	—
N1	—	VSS	—	—	—	—	—	—	—	—	—
N2	—	—	—	—	—	—	—	—	DDR_CKEB1	—	—
N3	—	—	—	—	—	—	—	—	DDR_CAB0	—	—
N4	—	—	—	—	—	—	—	—	DDR_CAA0	—	—
N5	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
N6	—	VSS	—	—	—	—	—	—	—	—	—
N7	—	—	—	—	—	—	—	—	DDR_RESET_N	—	—
N8	—	—	—	—	—	—	—	—	DDR_ATEST	—	—
N9	—	VSS	—	—	—	—	—	—	—	—	—
N10	—	VDD08	—	—	—	—	—	—	—	—	—
N11	—	VSS	—	—	—	—	—	—	—	—	—
N12	—	VDD08	—	—	—	—	—	—	—	—	—
N13	—	VSS	—	—	—	—	—	—	—	—	—
N14	—	VDD08	—	—	—	—	—	—	—	—	—
N15	—	VSS	—	—	—	—	—	—	—	—	—
N16	—	VDD08	—	—	—	—	—	—	—	—	—

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (12 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
N17	—	VDDP_18_3	—	—	—	—	—	—	—	—	—
N18	—	VDD1833_3	—	—	—	—	—	—	—	—	—
N19	—	VDD1833_3	—	—	—	—	—	—	—	—	—
N20	VDD1833_3	IRQ13	P33_4	A18	GTADSM01_0	ETH3_TXD1 / TXD1/ SDA1/MOSI1 / SPI_MISO1 / SPI_MOSI0	MCLK51	—	PCIE_RSTOUT1B	—	ENCIFDO01 / TXDE01 / HDSL09_SEL2
N21	VDD1833_3	IRQ14	P33_5	A19	GTADSM01_1	ETH3_TXD2 / RXD2/ SCL2/MISO2 / SPI_SSL10 / SPI_MISO0	MDAT51	—	—	—	ENCIFDI01 / RXDE01 / HDSL09_MISO2
N22	VDD33_X	XTALSEL	—	—	—	—	—	—	—	—	—
N23	—	XTAL	—	—	—	—	—	—	—	—	—
N24	—	EXTAL	—	—	—	—	—	—	—	—	—
P1	—	—	—	—	—	—	—	—	DDR_CKB_T	—	—
P2	—	—	—	—	—	—	—	—	DDR_CKEB0	—	—
P3	—	VSS	—	—	—	—	—	—	—	—	—
P4	—	VSS	—	—	—	—	—	—	—	—	—
P5	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
P6	—	VSS	—	—	—	—	—	—	—	—	—
P7	—	—	—	—	—	—	—	—	DDR_DTEST	—	—
P8	—	—	—	—	—	—	—	—	DDR_ZN	—	—
P9	—	VDD08	—	—	—	—	—	—	—	—	—
P10	—	VDD08	—	—	—	—	—	—	—	—	—
P11	—	VSS	—	—	—	—	—	—	—	—	—
P12	—	VDD08	—	—	—	—	—	—	—	—	—
P13	—	VDD18_PLL4	—	—	—	—	—	—	—	—	—
P14	—	VDD08	—	—	—	—	—	—	—	—	—
P15	—	VSS	—	—	—	—	—	—	—	—	—
P16	—	VSS	—	—	—	—	—	—	—	—	—
P17	—	VDD33_X	—	—	—	—	—	—	—	—	—
P18	—	VDDP_18_X	—	—	—	—	—	—	—	—	—
P19	—	OTPVDD08	—	—	—	—	—	—	—	—	—
P20	—	OTPVDD18	—	—	—	—	—	—	—	—	—
P21	—	VSS	—	—	—	—	—	—	—	—	—
P22	—	VSS	—	—	—	—	—	—	—	—	—
P23	—	VSS	—	—	—	—	—	—	—	—	—
P24	—	VSS	—	—	—	—	—	—	—	—	—
R1	—	—	—	—	—	—	—	—	DDR_CKB_C	—	—
R2	—	—	—	—	—	—	—	—	DDR_CAB1	—	—
R3	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
R4	—	—	—	—	—	—	—	—	DDR_CAB2	—	—
R5	—	—	—	—	—	—	—	—	DDR_CAB5	—	—
R6	—	—	—	—	—	—	—	—	DDR_CSB0	—	—
R7	—	VSS	—	—	—	—	—	—	—	—	—
R8	—	VSS	—	—	—	—	—	—	—	—	—
R9	—	VDD08	—	—	—	—	—	—	—	—	—
R10	—	VDD08	—	—	—	—	—	—	—	—	—
R11	—	VSS	—	—	—	—	—	—	—	—	—
R12	—	VSS	—	—	—	—	—	—	—	—	—

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (13 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
R13	—	VSS_PLL4	—	—	—	—	—	—	—	—	—
R14	—	PCIE_VDD08A_L1	—	—	—	—	—	—	—	—	—
R15	—	PCIE_VDD08A_L0	—	—	—	—	—	—	—	—	—
R16	—	VSS	—	—	—	—	—	—	—	—	—
R17	—	VSS	—	—	—	—	—	—	—	—	—
R18	—	VSS	—	—	—	—	—	—	—	—	—
R19	—	VSS	—	—	—	—	—	—	—	—	—
R20	—	VSS	—	—	—	—	—	—	—	—	—
R21	—	AVSSIO_ADC0	—	—	—	—	—	—	—	—	—
R22	—	AVDDREF_ADC0	—	—	—	—	—	—	—	—	—
R23	—	—	—	—	—	—	—	—	AN002	—	—
R24	—	—	—	—	—	—	—	—	AN000	—	—
T1	—	—	—	—	—	—	—	—	DDR_CAB3	—	—
T2	—	VSS	—	—	—	—	—	—	—	—	—
T3	—	—	—	—	—	—	—	—	DDR_CAB4	—	—
T4	—	—	—	—	—	—	—	—	DDR_DQB0	—	—
T5	—	VSS	—	—	—	—	—	—	—	—	—
T6	—	—	—	—	—	—	—	—	DDR_CSB1	—	—
T7	—	VSS	—	—	—	—	—	—	—	—	—
T8	—	VSS	—	—	—	—	—	—	—	—	—
T9	—	VSS	—	—	—	—	—	—	—	—	—
T10	—	VSS_PLL1	—	—	—	—	—	—	—	—	—
T11	—	VDD18_PLL1	—	—	—	—	—	—	—	—	—
T12	—	VDD08_PLL1	—	—	—	—	—	—	—	—	—
T13	—	VDD08_PLL4	—	—	—	—	—	—	—	—	—
T14	—	PCIE_VDD08A_L1	—	—	—	—	—	—	—	—	—
T15	—	PCIE_VDD08A_L0	—	—	—	—	—	—	—	—	—
T16	—	PCIE_VDD18A_L1	—	—	—	—	—	—	—	—	—
T17	—	PCIE_VDD18A_L0	—	—	—	—	—	—	—	—	—
T18	—	VSS	—	—	—	—	—	—	—	—	—
T19	—	VSS	—	—	—	—	—	—	—	—	—
T20	—	AVDD_ADC0	—	—	—	—	—	—	—	—	—
T21	—	AVSSIO_ADC0	—	—	—	—	—	—	—	—	—
T22	—	AVDDIO_ADC0	—	—	—	—	—	—	—	—	—
T23	—	—	—	—	—	—	—	—	AN003	—	—
T24	—	—	—	—	—	—	—	—	AN001	—	—
U1	—	—	—	—	—	—	—	—	DDR_DQB2	—	—
U2	—	—	—	—	—	—	—	—	DDR_DQB1	—	—
U3	—	VSS	—	—	—	—	—	—	—	—	—
U4	—	—	—	—	—	—	—	—	DDR_DQB3	—	—
U5	—	—	—	—	—	—	—	—	DDR_DQSB_T0	—	—

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (14 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
U6	—	VSS	—	—	—	—	—	—	—	—	—
U7	—	VSS	—	—	—	—	—	—	—	—	—
U8	—	VSS	—	—	—	—	—	—	—	—	—
U9	—	VSS	—	—	—	—	—	—	—	—	—
U10	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
U11	—	VDD1833_4	—	—	—	—	—	—	—	—	—
U12	—	VSS	—	—	—	—	—	—	—	—	—
U13	—	VDDP_18_5	—	—	—	—	—	—	—	—	—
U14	—	VSS	—	—	—	—	—	—	—	—	—
U15	—	USB_USVDD18	—	—	—	—	—	—	—	—	—
U16	—	PCIE_VDD18A_L1	—	—	—	—	—	—	—	—	—
U17	—	PCIE_VDD18A_L0	—	—	—	—	—	—	—	—	—
U18	—	VSS	—	—	—	—	—	—	—	—	—
U19	—	VSS	—	—	—	—	—	—	—	—	—
U20	—	AVSS_ADC0	—	—	—	—	—	—	—	—	—
U21	—	AVDDIO_ADC1	—	—	—	—	—	—	—	—	—
U22	—	AVDDREF_ADC1	—	—	—	—	—	—	—	—	—
U23	—	—	—	—	—	—	—	—	AN100	—	—
U24	—	—	—	—	—	—	—	—	AN103	—	—
V1	—	VSS	—	—	—	—	—	—	—	—	—
V2	—	—	—	—	—	—	—	—	DDR_DQB4	—	—
V3	—	—	—	—	—	—	—	—	DDR_DQB7	—	—
V4	—	—	—	—	—	—	—	—	DDR_DMIB0	—	—
V5	—	—	—	—	—	—	—	—	DDR_DQSB_C0	—	—
V6	—	VSS	—	—	—	—	—	—	—	—	—
V7	VDD33	TRST#	—	—	—	—	—	—	—	—	—
V8	—	VDD33	—	—	—	—	—	—	—	—	—
V9	—	VDD33	—	—	—	—	—	—	—	—	—
V10	—	VDDP_18_4	—	—	—	—	—	—	—	—	—
V11	—	VDD1833_4	—	—	—	—	—	—	—	—	—
V12	—	VDDP_18_33	—	—	—	—	—	—	—	—	—
V13	—	VDD1833_5	—	—	—	—	—	—	—	—	—
V14	—	VSS	—	—	—	—	—	—	—	—	—
V15	—	USB_USVDD18	—	—	—	—	—	—	—	—	—
V16	—	USB_USVDD33	—	—	—	—	—	—	—	—	—
V17	—	USB_USVDD33	—	—	—	—	—	—	—	—	—
V18	—	VSS	—	—	—	—	—	—	—	—	—
V19	—	VSS	—	—	—	—	—	—	—	—	—
V20	—	AVSS_ADC1	—	—	—	—	—	—	—	—	—
V21	—	AVSSIO_ADC1	—	—	—	—	—	—	—	—	—
V22	—	AVSSIO_ADC1	—	—	—	—	—	—	—	—	—

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (15 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
V23	—	—	—	—	—	—	—	—	AN102	—	—
V24	—	—	—	—	—	—	—	—	AN101	—	—
W1	—	—	—	—	—	—	—	—	DDR_DQB6	—	—
W2	—	DDR_VDDQ	—	—	—	—	—	—	—	—	—
W3	—	—	—	—	—	—	—	—	DDR_DQB5	—	—
W4	—	VSS	—	—	—	—	—	—	—	—	—
W5	—	VSS	—	—	—	—	—	—	—	—	—
W6	—	VSS	—	—	—	—	—	—	—	—	—
W7	VDD1833_4	IRQ11	P06_5	—	GTETRGC	IIC_SDA1 / XSPI0_IO7	—	—	—	—	HDSL05_SEL1
W8	VDD1833_4	IRQ9	P06_3	—	GTETRGA	IIC_SDA0 / XSPI0_IO5	—	—	—	—	TST_OUT09 / HDSL05_SMPL
W9	VDD33	IRQ14	P03_4	D12	MTCLKB / MTIOC8D / GTIOC02_3B / GTADSM09_1 / CMTW1_TOC1 / RTCAT1HZ	IIC_SDA1	—	—	—	—	ENCIFOE02 / DEE02 / HDSL02_MISO2
W10	—	VSS	—	—	—	—	—	—	—	—	—
W11	VDD1833_5	IRQ6	P01_0	—	MTIOC3A / MTIOC1A / GTIOC00_4A / GTIOC00_2B	IIC_SCL1 / XSPI1_CKP	—	—	—	—	ENCIFDO00 / ENCIFDO04 / TXDE00 / TXDE04 / HDSL00_MISO2
W12	VDD33	IRQ1	P00_2	D2	MTIOC4A / GTIOC00_1A	ETH3_CRS	—	—	ADTRG0# / USB_EXICEN	—	SI00# / HDSL00_CLK1
W13	—	VDD1833_5	—	—	—	—	—	—	—	—	—
W14	—	USB_USDVDD	—	—	—	—	—	—	—	—	—
W15	—	USB_USDVDD	—	—	—	—	—	—	—	—	—
W16	—	VSS	—	—	—	—	—	—	—	—	—
W17	—	VSS	—	—	—	—	—	—	—	—	—
W18	—	PCIE_VDD18A_CMN	—	—	—	—	—	—	—	—	—
W19	—	VSS	—	—	—	—	—	—	—	—	—
W20	—	AVDD_ADC1	—	—	—	—	—	—	—	—	—
W21	—	AVSSIO_ADC2	—	—	—	—	—	—	—	—	—
W22	—	AVSSIO_ADC2	—	—	—	—	—	—	—	—	—
W23	—	—	—	—	—	—	—	—	AN206	—	—
W24	—	—	—	—	—	—	—	—	AN210	—	—
Y1	—	—	—	—	—	—	—	—	DDR_DQB8	—	—
Y2	—	VSS	—	—	—	—	—	—	—	—	—
Y3	—	—	—	—	—	—	—	—	DDR_DQB15	—	—
Y4	—	VSS	—	—	—	—	—	—	—	—	—
Y5	—	—	—	—	—	—	—	—	DDR_DQSB_T1	—	—
Y6	—	VSS	—	—	—	—	—	—	—	—	—
Y7	VDD1833_4	IRQ8	P06_2	—	—	IIC_SCL0 / XSPI0_IO4	—	—	—	—	DUEI09 / HDSL05_LINK
Y8	VDD1833_4	IRQ10	P06_4	—	GTETRGB	IIC_SCL1 / XSPI0_IO6	—	—	—	—	SI09# / HDSL05_CLK1

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (16 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
Y9	VDD33	—	P03_1	D9	MTIOC4B / MTIOC1B / GTIOC02_2A / GTADSM08_0 / CMTW1_TIC0	—	—	—	—	—	ENCIFDO02 / TXDE02 / HDSL02_MOSI1
Y10	VDD1833_5	IRQ10	P02_3	—	MTIOC6C / MTIOC1B / GTIOC01_4B	ETH3_COL / IIC_SCL0 / IIC_SCL2 / XSPI1_IO7	MDAT22	—	USB_OVRCUR	—	ENCIFDI01 / RXDE01 / HDSL01_MOSI2
Y11	VDD1833_5	IRQ11	P02_4	—	POE0#	IIC_SDA0	MDAT20	—	USB_EXICEN	MBX_HINT#	HDSL02_LINK
Y12	VDD1833_5	—	P01_2	—	MTIOC6B / MTIOC8B / GTIOC01_0A / GTIOC04_0A	XSPI1_CS1#	—	—	—	—	DUEI02 / HDSL01_LINK
Y13	—	VDD33	—	—	—	—	—	—	—	—	—
Y14	—	—	—	—	—	—	—	—	USB_VUBUSIN	—	—
Y15	—	—	—	—	—	—	—	—	USB_TXRTUNE	—	—
Y16	—	—	—	—	—	—	—	—	PCIE_REFCLK_N1	—	—
Y17	—	—	—	—	—	—	—	—	PCIE_REFCLK_P0	—	—
Y18	—	PCIE_VDD18A_CMN	—	—	—	—	—	—	—	—	—
Y19	—	VSS	—	—	—	—	—	—	—	—	—
Y20	—	AVDD_ADC2	—	—	—	—	—	—	—	—	—
Y21	—	AVDDIO_ADC2	—	—	—	—	—	—	—	—	—
Y22	—	—	—	—	—	—	—	—	AN209	—	—
Y23	—	—	—	—	—	—	—	—	AN208	—	—
Y24	—	—	—	—	—	—	—	—	AN202	—	—
AA1	—	—	—	—	—	—	—	—	DDR_DQB14	—	—
AA2	—	—	—	—	—	—	—	—	DDR_DQB9	—	—
AA3	—	—	—	—	—	—	—	—	DDR_DMIB1	—	—
AA4	—	—	—	—	—	—	—	—	DDR_DQB10	—	—
AA5	—	—	—	—	—	—	—	—	DDR_DQSB_C1	—	—
AA6	VDD33	BSCANP	—	—	—	—	—	—	—	—	—
AA7	VDD1833_4	IRQ3	P05_1	—	—	XSPI0_CKP	—	—	—	—	DUEI06 / HDSL04_SMPL
AA8	—	VSS	—	—	—	—	—	—	—	—	—
AA9	VDD33	IRQ13	P03_3	D11	MTCLKA / MTIOC8C / GTIOC02_3A / GTADSM09_0 / CMTW1_TIC1	IIC_SCL1	—	—	—	—	ENCIFCK02 / SCKE02 / HDSL02_SEL2
AA10	VDD1833_5	IRQ8	P02_1	—	MTCLKD / MTIOC0D / GTIOC01_3B	ETH3_RXER / IIC_SCL2 / XSPI1_IO5	MDAT21	—	—	—	ENCIFOE01 / DEE01 / HDSL01_SEL2
AA11	VDD1833_5	IRQ9	P02_2	—	MTIOC6A / MTIOC1A / GTIOC01_4A	ETH3_CRS / IIC_SDA2 / XSPI1_IO6	MCLK22	—	USB_VBUSEN	—	ENCIFDO01 / TXDE01 / HDSL01_MISO2
AA12	VDD1833_5	IRQ7	P02_0	—	MTCLKC / MTIOC0C / GTIOC01_3A	ETH3_TXER / IIC_SDA1 / XSPI1_IO4	MCLK21	—	—	—	ENCIFCK01 / SCKE01 / HDSL01_CLK2
AA13	VDD33	IRQ0	P00_1	D1	MTIOC3D / GTIOC00_0B	ETH3_RXER	—	—	USB_OVRCUR	—	TST_OUT00 / HDSL00_SMPL
AA14	—	—	—	—	—	—	—	—	USB_OTG_ID	—	—
AA15	—	VSS	—	—	—	—	—	—	—	—	—

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (17 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
AA16	—	—	—	—	—	—	—	—	PCIE_REFCLK_P1	—	—
AA17	—	—	—	—	—	—	—	—	PCIE_REFCLK_N0	—	—
AA18	—	VSS	—	—	—	—	—	—	—	—	—
AA19	—	VSS	—	—	—	—	—	—	—	—	—
AA20	—	AVSS_ADC2	—	—	—	—	—	—	—	—	—
AA21	—	AVDDREF_ADC2	—	—	—	—	—	—	—	—	—
AA22	—	—	—	—	—	—	—	—	AN213	—	—
AA23	—	—	—	—	—	—	—	—	AN201	—	—
AA24	—	—	—	—	—	—	—	—	AN200	—	—
AB1	—	—	—	—	—	—	—	—	DDR_DQB12	—	—
AB2	—	VSS	—	—	—	—	—	—	—	—	—
AB3	—	—	—	—	—	—	—	—	DDR_DQB13	—	—
AB4	—	—	—	—	—	—	—	—	DDR_DQB11	—	—
AB5	—	VSS	—	—	—	—	—	—	—	—	—
AB6	VDD1833_4	IRQ4	P05_2	—	—	IIC_SCL2 / XSPI0_CKN	—	—	—	—	TST_OUT06 / HDSL04_CLK1
AB7	VDD1833_4	—	P06_1	—	—	XSPI0_IO3	—	—	—	—	SI08# / HDSL04_MOSI2
AB8	VDD1833_4	IRQ6	P05_4	—	—	IIC_SDA2 / XSPI0_CS1#	—	—	—	—	DUEI07 / HDSL04_MISO1
AB9	VDD33	IRQ12	P03_2	D10	MTIOC4D / MTIOC1A / GTIOC02_2B / GTADSM08_1 / CMTW1_TOC0	—	—	—	—	—	ENCIFDI02 / RXDE02 / HDSL02_CLK2
AB10	VDD33	—	P02_6	D6	MTIOC3D / MTIOC8B / GTIOC02_0B / GTADSM06_1 / CMTW0_TOC0	SD0_IOVS	MDAT00	—	—	—	HDSL02_CLK1 / POUTB
AB11	VDD1833_5	—	P01_1	—	MTIOC3C / MTIOC8A / GTIOC00_4B	XSPI1_CS0#	MCLK20	—	—	—	ENCIFDI00 / ENCIFDI04 / RXDE00 / RXDE04 / HDSL00_MOSI2
AB12	VDD1833_5	—	P01_7	—	MTIOC7D / MTIOC0B / GTIOC01_2B / GTIOC04_2B	XSPI1_IO3	—	—	—	—	SI03# / HDSL01_MOSI1
AB13	VDD33	SEI	P00_0	D0	MTIOC3B / GTIOC00_0A	ETH3_TXER	—	—	USB_VBUSEN	—	DUEI00 / HDSL00_LINK
AB14	—	VSS	—	—	—	—	—	—	—	—	—
AB15	—	VSS	—	—	—	—	—	—	—	—	—
AB16	—	VSS	—	—	—	—	—	—	—	—	—
AB17	—	VSS	—	—	—	—	—	—	—	—	—
AB18	—	VSS	—	—	—	—	—	—	—	—	—
AB19	—	VSS	—	—	—	—	—	—	—	—	—
AB20	—	VSS	—	—	—	—	—	—	—	—	—
AB21	—	VSS	—	—	—	—	—	—	—	—	—
AB22	—	—	—	—	—	—	—	—	AN203	—	—
AB23	—	AVSSIO_ADC2	—	—	—	—	—	—	—	—	—
AB24	—	—	—	—	—	—	—	—	AN212	—	—

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (18 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
AC1	VDD33	MDX	—	—	—	—	—	—	—	—	—
AC2	VDD33	IRQ8 / RSTOUT#	P08_5	—	GTETRGSA	IIC_SCL1 / SD1_PWEN	MCLK02	—	—	—	HDSL06_MISO2
AC3	VDD33	TDO	P08_4	—	—	—	—	—	—	—	HDSL06_SEL2
AC4	VDD33	RES#	—	—	—	—	—	—	—	—	—
AC5	VDD33	TCK	P08_3	—	—	—	—	—	—	—	SI10# / HDSL06_CLK2
AC6	VDD1833_4	—	P06_0	—	—	XSPI0_IO2	—	—	—	—	TST_OUT08 / HDSL04_MISO2
AC7	VDD1833_4	—	P05_7	—	—	XSPI0_IO1	—	—	—	—	DUEI08 / HDSL04_SEL2
AC8	VDD1833_4	IRQ12	P06_7	—	POE4# / GTETRGD	GMAC1_MDC / IIC_SCL2	—	—	—	—	HDSL05_MISO1
AC9	VDD33	—	P02_7	D7	MTIOC4A / MTIC5U / GTIOC02_1A / GTADSM07_0 / CMTW0_TIC1	—	MCLK01	—	—	—	HDSL02_SEL1 / POUTZ
AC10	—	VSS	—	—	—	—	—	—	—	—	—
AC11	VDD1833_5	—	P01_5	—	MTIOC7C / MTIC5W / GTIOC01_1B / GTIOC04_1B	XSPI1_IO1	—	—	—	—	DUEI03 / HDSL01_SEL1
AC12	VDD1833_5	—	P01_3	—	MTIOC6D / MTIC5U / GTIOC01_0B / GTIOC04_0B	XSPI1_DS	—	—	—	—	TST_OUT02 / HDSL01_SMPL
AC13	VDD33	IRQ3	P00_4	D4	MTIOC4B / GTIOC00_2A	—	—	—	ADTRG2#	—	TST_OUT01 / HDSL00_MISO1
AC14	—	—	—	—	—	—	—	—	USB_QDP	—	—
AC15	—	VSS	—	—	—	—	—	—	—	—	—
AC16	—	—	—	—	—	—	—	—	PCIE_RXDN_L1	—	—
AC17	—	—	—	—	—	—	—	—	PCIE_RXDN_L0	—	—
AC18	—	VSS	—	—	—	—	—	—	—	—	—
AC19	—	—	—	—	—	—	—	—	PCIE_TXDN_L1	—	—
AC20	—	—	—	—	—	—	—	—	PCIE_TXDN_L0	—	—
AC21	—	VSS	—	—	—	—	—	—	—	—	—
AC22	—	—	—	—	—	—	—	—	AN211	—	—
AC23	—	—	—	—	—	—	—	—	AN204	—	—
AC24	—	—	—	—	—	—	—	—	AN207	—	—
AD1	—	VSS	—	—	—	—	—	—	—	—	—
AD2	VDD33	SEI / CKIO	P08_6	—	GTIOC08_3A / GTETRGSB	IIC_SDA1 / SD1_IOVS	MDAT02 / MCLK11	—	—	—	DUEI11 / HDSL06_MOSI2
AD3	VDD33	TMS	P08_1	—	—	—	—	—	—	—	DUEI10 / HDSL06_MISO1
AD4	VDD33	TDI	P08_2	—	—	—	—	—	—	—	TST_OUT10 / HDSL06_MOSI1
AD5	VDD1833_4	—	P05_6	—	—	XSPI0_IO0	—	—	—	—	SI07# / HDSL04_CLK2
AD6	VDD1833_4	—	P05_5	—	—	XSPI0_DS	—	—	—	—	TST_OUT07 / HDSL04_MOSI1
AD7	VDD1833_4	MDD	P06_6	—	—	XSPI0_RESET0#	—	—	—	—	—
AD8	VDD1833_4	IRQ5	P05_3	—	—	XSPI0_CS0#	—	—	—	—	SI06# / HDSL04_SEL1

Table 1.21 List of pins and pin functions (RZ/N2H 576-pin FCBGA) (19 of 19)

Pin number	IO Port power domain	Power supply clock, system control, Interrupt	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI, SDHI)	DSMIF	LCDC	Analog / High speed (DDRSS, ADC, USB, PCIE)	Host Interface (SHOST, MBXSEM)	Encoder I/F (ENCIF, SCIE, ENDAT, HDSL, ENCOUT)
AD9	VDD33	—	P03_0	D8	MTIOC4C / MTIC5V / GTIOC02_1B / GTADSM07_1 / CMTW0_TOC1	—	MDAT01	—	—	—	HDSL02_MISO1
AD10	VDD33	—	P02_5	D5	MTIOC3B / MTIOC8A / GTIOC02_0A / GTADSM06_0 / CMTW0_TIC0	IIC_SCL0 / SD0_PWEN	MCLK00	—	—	—	HDSL02_SMPL / POUTA
AD11	VDD1833_5	—	P01_6	—	MTIOC7B / MTIOC0A / GTIOC01_2A / GTIOC04_2A	XSPI1_IO2	—	—	—	—	TST_OUT03 / HDSL01_MISO1
AD12	VDD1833_5	—	P01_4	—	MTIOC7A / MTIC5V / GTIOC01_1A / GTIOC04_1A	XSPI1_IO0	—	—	—	—	SI02# / HDSL01_CLK1
AD13	VDD33	IRQ2	P00_3	D3	MTIOC4C / GTIOC00_1B	ETH3_COL	—	—	ADTRG1#	—	DUEI01 / HDSL00_SEL1
AD14	—	—	—	—	—	—	—	—	USB_QDM	—	—
AD15	—	VSS	—	—	—	—	—	—	—	—	—
AD16	—	—	—	—	—	—	—	—	PCIE_RXDP_L1	—	—
AD17	—	—	—	—	—	—	—	—	PCIE_RXDP_L0	—	—
AD18	—	VSS	—	—	—	—	—	—	—	—	—
AD19	—	—	—	—	—	—	—	—	PCIE_TXDP_L1	—	—
AD20	—	—	—	—	—	—	—	—	PCIE_TXDP_L0	—	—
AD21	—	VSS	—	—	—	—	—	—	—	—	—
AD22	—	—	—	—	—	—	—	—	AN205	—	—
AD23	—	—	—	—	—	—	—	—	AN214	—	—
AD24	—	AVSSIO_ADC2	—	—	—	—	—	—	—	—	—

2. Cortex-A55

2.1 Overview

The LSI includes Arm® Cortex®-A55 Core and DynamIQ™ Shared Unit.

Table 2.1 shows the specifications of the Cortex-A55.

Table 2.1 Specifications of Cortex-A55

Item		Specification
Cluster	Core	Arm® Cortex-A55 Core r2p0 (ARMv8-A), Quad cores
	DSU	Arm® DynamIQ™ Shared Unit (r4p1)
Clock frequency	Core	600 / 1200 MHz
	DSU	500 / 1000 MHz
Address space		32 GB
Cache	L1 instruction cache	32 KB (with parity) per core
	L1 data cache	32 KB (with ECC) per core
	L2 cache	0 KB
	L3 cache	1024 KB (with ECC)
Instruction set		Armv8.2-A A64 instruction sets (AArch64)
Data arrangement		Little endian
System bus interface		• 128-bit ACE master interface • 64-bit AXI master peripheral port (0x00_8000_0000 to 0x00_902F_FFFF)
Security		• Arm® TrustZone® technology • Cryptographic Extension (optional)
Data engine unit		The Advanced SIMD and scalar floating-point instructions in the A64 instruction set including Dot Product instruction
Debug		• Reliability, Availability, and Serviceability (RAS) Extension • Performance Monitoring Unit (PMU) • Embedded Trace Macrocell (ETM) for instruction trace
Interrupt controller		GIC-600

For details, refer to the following documents supplied by Arm.

- Arm® Cortex®-A55 Core Technical Reference Manual
- Arm® Cortex®-A55 Core Advanced SIMD and Floating-point Support Technical Reference Manual
- Arm® Cortex®-A55 Core Cryptographic Extension Technical Reference Manual
- Arm® DynamIQ™ Shared Unit Technical Reference Manual
- Arm® CoreLink™ GIC-600 Generic Interrupt Controller Technical Reference Manual
- Arm® Architecture Reference Manual Armv8, for the Armv8-A architecture profile

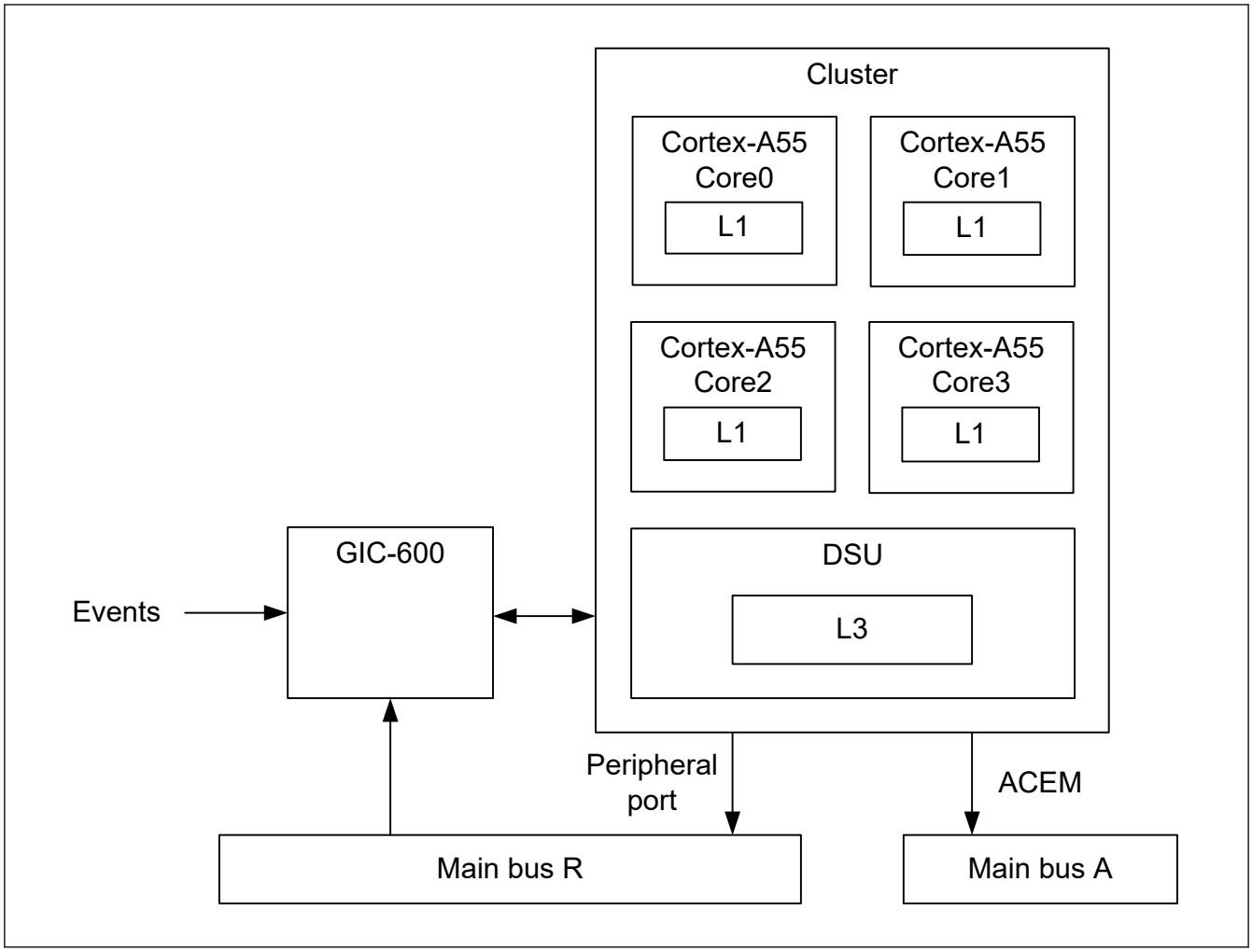


Figure 2.1 Block diagram of Cortex-A55

2.2 Configuration Information

Table 2.2 and Table 2.3 lists the configuration options for Cortex-A55 Core and DSU of this LSI.

Table 2.2 Core Configuration options

Feature	Setting value
L1 instruction cache size	32 KB
L1 data cache size	32 KB
L2 cache	Not included
L2 cache size	—
ECC or parity core cache protection	Included
Advanced SIMD and floating-point support	Included
Cryptographic Extension	Included (Security support product only)
CoreSight Embedded Logic Analyzer (ELA)	Not included
CoreSight ELA RAM address size	—
Page Based Hardware Attributes (PBHA) support	Not included

Table 2.3 DSU Configuration options (1 of 2)

Feature	Setting value
NUM_BIG_CORES	0

Table 2.3 DSU Configuration options (2 of 2)

Feature	Setting value
NUM_LITTLE_CORES	4
NUM_OTHER_CORES	0
ACE	TRUE
MASTER_DATA_WIDTH	128
ACP	FALSE
PERIPH_PORT	TRUE
SCU_CACHE_PROTECTION	TRUE
L3_CACHE	TRUE
L3_CACHE_SIZE	1024 KB
NUM_SLICES	1
L3_DATA_WR_LATENCY	1
L3_DATA_RD_LATENCY	2
ASYNC_BRIDGE	TRUE
LEGACY_V7_DEBUG_MAP	FALSE
ELA	FALSE
L3_PBHA	FALSE

2.3 Restrictions on CPU

For details on restrictions on Cortex-A55 implemented in this LSI, refer to the information provided at the website of Arm.

2.4 Register Map

Table 2.4 Cortex-A55 auxiliary register map

Address	Register symbol	Register name	Write protection
0x8129_5020 + 0x08 × n	RVBALn	CA55 Core n Reset Vector Address Low Configuration Register (n = 0 to 3)	PRCRS.PRC3
0x8129_5024 + 0x08 × n	RVBAHn	CA55 Core n Reset Vector Address High Configuration Register (n = 0 to 3)	PRCRS.PRC3

Table 2.5 Cortex-A55 auxiliary related system control register

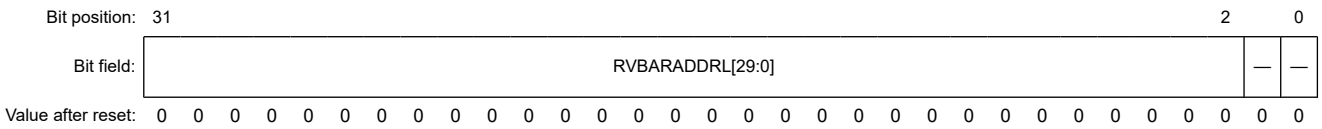
Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
RVBALn, RVBAHn	—	—	SLVACCCTL6.CA55_SL
GIC-600	—	—	SLVACCCTL7.GIC600_SL

2.5 Register Descriptions

2.5.1 RVBALn : CA55 Core n Reset Vector Address Low Configuration Register (n = 0 to 3)

Base address: CA55 = 0x8129_5000

Offset address: 0x20 + 0x08 × n



Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
31:2	RVBARADDRL[29:0]	Cortex-A55 Core n Reset Vector Address Low Set 30 bits from bit 31 to bit 2 of the 40-bit address. Lower 2 bits from bit 1 and bit 0 of the 40-bit address is set 00b automatically.	R/W

The RVBALn register is used to set reset vector address of Cortex-A55. In case of Cortex-A55 boot mode, 0x101FD000 is set to RVBAL0 automatically.

2.5.2 RVBAHn : CA55 Core n Reset Vector Address High Configuration Register (n = 0 to 3)

Base address: CA55 = 0x8129_5000

Offset address: 0x24 + 0x08 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RVBARADDRH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RVBARADDRH[7:0]	Cortex-A55 Core n Reset Vector Address High Set 8 bits from bit 39 to bit 32 of the 40-bit address. Bit [7:3] should be set to 0.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The RVBAHn register is used to set reset vector address of Cortex-A55.

2.6 Usage Notes

2.6.1 How to Start up Cortex-A55

After a RES# pin reset, a system software reset, or an error reset occurs, Cortex-A55 Core 0 to Core 3 and WDTs for Cortex-A55 remain in the reset state and module stop state in case of Cortex-R52 boot mode. Start-up Cortex-A55 by other CPU according to the below sequence.

1. Store the instruction code to the reset vector address you want to specify.
2. Start up the PLL0. For detail, see section 7.5. PLL Circuit.
3. Release Core n from the module stop state by resetting corresponding bit (bit 2 to bit 5) in MSTPCRN register. DSU is always out of the module stop state.
4. Set the reset vector address in RVBALn and RVBAHn registers.
5. Release DSU and Core n from the reset state by setting 0x00000000 to SWR55C and corresponding SWR55n register.

In case of Cortex-A55 boot mode, Cortex-A55 Core 0 starts up automatically by performing the above sequence in boot ROM. Cortex-A55 Core 1 to Core 3 remain in the reset state and module stop state. Start up sequence is same as the above, but it is not needed to operate PLL0 and DSU repeatedly.

Note that P-Channels and Q-Channels are not used in this LSI.

2.6.2 Generic Timer

Cortex-A55 is provided system counter value from Global System Counter for Generic Timer. For detail, see section 3.6.3. Global System Counter (TSG-CNT).

3. Cortex-R52

3.1 Overview

The LSI includes two Cortex®-R52 CPUs. The revision of module is r1p4-00refl0.

Table 3.1 shows the specifications of the CPU.

Table 3.1 CPU specifications

Item		Description	
		CPU0	CPU1
Cortex-R52	Minimum instruction execution time	One clock per instruction	
	Address space	4 GB	
	Instruction cache size	16 KB (with ECC)	16 KB (with ECC)
	Data cache size	16 KB (with ECC)	16 KB (with ECC)
	Tightly coupled memory (TCM) size	ATCM: 512 KB (with ECC) BTCM: 64 KB (with ECC)	ATCM: 512 KB (with ECC) BTCM: 64 KB (with ECC)
	Instruction set	Armv8-R architecture supporting Thumb®/Thumb-2	
	Data arrangement	Instruction: Little-endian Data: Little-endian	
	Memory protection	2-stage memory protection unit (MPU)	
	FPU	• Supports addition, subtraction, multiplication, division, product-sum operation, and square-root operation in single-precision and double-precision • 64-bit single word register: 32 registers • Can also be used as sixteen 128-bit double word registers	
	NEON	• The advanced SIMD supporting integer or single precision results	

For details, see the following Arm documents:

- Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile
- Arm® Architecture Reference Manual Supplement Armv8, for the Armv8-R architecture profile
- Arm® Cortex®-R52 Processor Technical Reference Manual

3.2 Configuration Information

Table 3.2 lists the configuration options for the Cortex-R52 CPU of this LSI.

Table 3.2 Configuration options (1 of 2)

Feature	Setting value	
	CPU0	CPU1
Number of cores	1	1
Lock-step	No redundant logic included	No redundant logic included
EL1-controlled MPU	24	24
EL2-controlled MPU	24	24
AXI bus protection	No bus protection included	No bus protection included
RAM protection	Included for all RAMs	Included for all RAMs
Number of interrupts (SPIs) into the interrupt controller	544	544
AXIS ID bits	14	14
Advanced SIMD and floating-point capabilities for each core	Supported	Supported

Table 3.2 Configuration options (2 of 2)

Feature	Setting value	
	CPU0	CPU1
Size of each of the three TCMs on each core	ATCM: 512 KB BTCM: 64 KB CTCM: 0 KB	ATCM: 512 KB BTCM: 64 KB CTCM: 0 KB
TCM wait states for each core	ATCM: 0 wait state (500 MHz) / 1 wait state (1000 MHz)*1 BTCM: 0 wait state	ATCM: 0 wait state (500 MHz) / 1 wait state (1000 MHz)*1 BTCM: 0 wait state
Instruction cache size for each core	16 KB	16 KB
Data cache size for each core	16 KB	16 KB
External device interfaces to GIC	0	0
Flash ECC scheme	Not available	Not available
Vector table base address out of reset (after running boot ROM)	0x0000_0000 (Start address of ATCM in CPU0)	0x0000_0000 (Start address of ATCM in CPU1)
Base address of the AXIS interface	0x2000_0000	0x2100_0000
Base address of the LLPP	0x9000_0000	0x9000_0000
Base address of the GIC registers	0x9400_0000	0x9C00_0000
Cluster ID	0x0 at affinity level 1 0x1 at affinity level 2	0x0 at affinity level 1 0x2 at affinity level 2

Note 1. It depends on the MDW0/1 pin setting at release of reset. If 1000 MHz operation is required, 1 wait state is mandatory.

3.3 Restrictions on the CPU

For details on restrictions of the Cortex-R52 CPU implemented in this LSI, see the information provided on the Arm website.

3.4 Register Map

Table 3.3 Cortex-R52 auxiliary register map

Address	Register symbol	Register name	Write protection
0x8129_5008	CPU0WAIT	CPU0 ATCM Wait State Register	—
0x8129_500C	CPU1WAIT	CPU1 ATCM Wait State Register	—
0x8129_5010	CPU1HALT	CPU1 HALT Control Register	PRCRS.PRC3
0x8C20_0000	CNTCR	Global System Counter Control Register	—
0x8C20_0004	CNTSR	Global System Counter Status Register	—
0x8C20_0008	CNTCVL	Global System Counter Current Value Lower Register	—
0x8C20_000C	CNTCVU	Global System Counter Current Value Upper Register	—
0x8C20_0020	CNTFID0	Global System Counter Base Frequency ID Register	—

Table 3.4 Cortex-R52 auxiliary register related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
CPU0WAIT	—	—	SLVACCCTL6.CR520_SL
CPU1WAIT, CPU1HALT	—	—	SLVACCCTL6.CR521_SL
Global System Counter related registers	—	—	SLVACCCTL7.CSDBGAPB_SL

3.5 Register Descriptions

3.5.1 CPU0WAIT : CPU0 ATCM Wait State Register

Base address: TCMAW = 0x8129_5000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPU0 WAIT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x*1

Bit	Symbol	Function	R/W
0	CPU0WAIT	ATCM Wait state 0: 0-wait state (MDW0 = 0) 1: 1-wait state (MDW0 = 1)	R
31:1	—	These bits are read as 0.	R

Note 1. It depends on the state of MDW pin latched at release of RES# pin on power on.

The CPU0WAIT is a register that indicates ATCM access wait in Cortex-R52 CPU0. It is set by the state of the MDW0 pin latched at release of RES# pin on power on.

3.5.2 CPU1WAIT : CPU1 ATCM Wait State Register

Base address: TCMAW = 0x8129_5000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPU1 WAIT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	CPU1WAIT	ATCM Wait state 0: 0-wait state (MDW1 = 0) 1: 1-wait state (MDW1 = 1)	R
31:1	—	These bits are read as 0.	R

The CPU1WAIT is a register that indicates ATCM access wait in Cortex-R52 CPU1. It is set by the state of the MDW1 pin latched at release of RES# pin on power on.

3.5.3 CPU1HALT : CPU1 HALT Control Register

Base address: TCMAW = 0x8129_5000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPU1 HALT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	CPU1HALT	Instruction fetch after reset release 0: Start instruction fetch 1: Stop instruction fetch	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The CPU1HALT register is used to stop instruction fetch after Cortex-R52 CPU1 software reset is released. After storing the instruction code to ATCM of Cortex-R52 CPU1 via AXIS interface, write 0 to CPU1HALT bit to start instruction fetch. This register is reset in case of cold reset, system software reset, and error reset. This register is not reset in case of other CPU software resets.

3.5.4 CNTCR : Global System Counter Control Register

Base address: GSC = 0x8C20_0000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HDBG	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

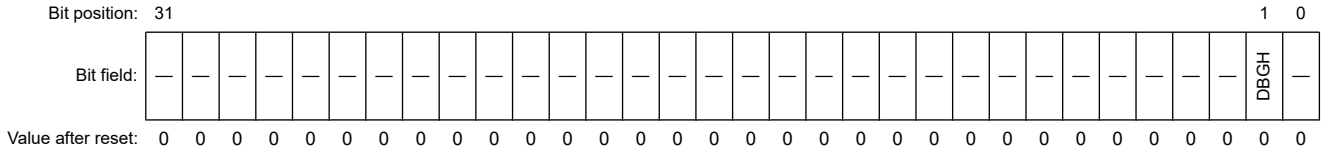
Bit	Symbol	Function	R/W
0	EN	Counter Enable 0: The counter is disabled and not incrementing 1: The counter is enabled and is incrementing	R/W
1	HDBG	Halt on Debug HLTDBG is internal signal and it is driven high when CTI trigger output[4] (TSG-CNT HALTREQ) is asserted. It is driven low when CTI trigger output[5] (TSG-CNT RESTARTREQ) is asserted. 0: Do not halt on debug, HLTDBG signal into the counter has no effect 1: Halt on debug, when HLTDBG is driven high, the count value is held static	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The CNTCR register controls the global system counter (TSG-CNT) increments.

3.5.5 CNTSR : Global System Counter Status Register

Base address: GSC = 0x8C20_0000

Offset address: 0x04



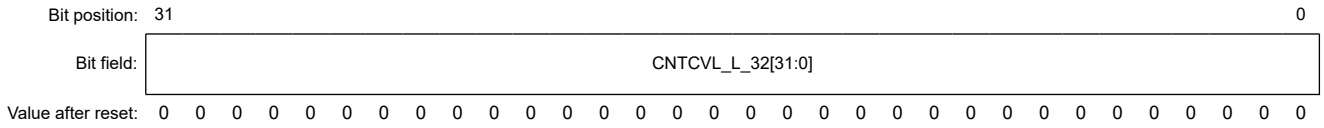
Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	DBGH	Debug Halted 0: Not halt on debug 1: Halt on debug	R
31:2	—	These bits are read as 0.	R

The CNTSR register indicates the global system counter (TSG-CNT) status.

3.5.6 CNTCVL : Global System Counter Current Value Lower Register

Base address: GSC = 0x8C20_0000

Offset address: 0x08



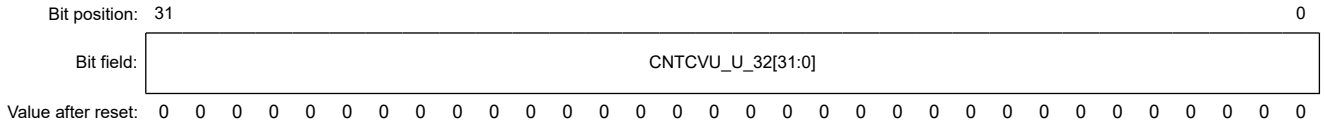
Bit	Symbol	Function	R/W
31:0	CNTCVL_L_32[31:0]	Current value of the counter, lower 32 bits To change the current value, write the lower 32 bits of the new value to this register before writing the upper 32 bits to CNTCVU. The counter value is not changed until the CNTCVU register is written to.	R/W

The CNTCVL register reads or writes the lower 32 bits of the current global system counter (TSG-CNT) value. Before writing to this register, clear the CNTCR.EN bit.

3.5.7 CNTCVU : Global System Counter Current Value Upper Register

Base address: GSC = 0x8C20_0000

Offset address: 0x0C



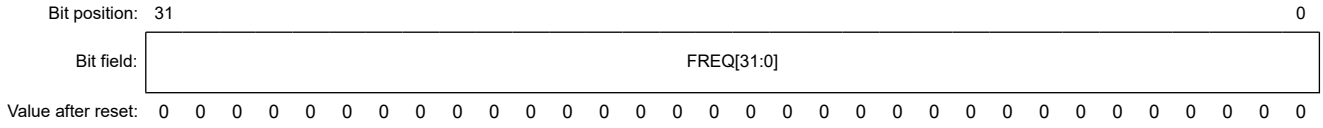
Bit	Symbol	Function	R/W
31:0	CNTCVU_U_32[31:0]	Current value of the counter, upper 32 bits To change the current value, write the lower 32 bits of the new value to CNTCVL before writing the upper 32 bits to this register. The 64-bit value is updated with the value from both writes when this register is written to.	R/W

The CNTCVU register reads or writes the upper 32 bits of the current global system counter (TSG-CNT) value. Before writing to this register, clear the CNTCR.EN bit.

3.5.8 CNTFID0 : Global System Counter Base Frequency ID Register

Base address: GSC = 0x8C20_0000

Offset address: 0x20



Bit	Symbol	Function	R/W
31:0	FREQ[31:0]	Frequency in number of ticks per second Clock for generic timer counter is the main oscillator clock (25 MHz). Write 0x017D7840 for 25 MHz clock.	R/W

The CNTFID0 register specifies the clock frequency of the global system counter (TSG-CNT) in ticks per second.

3.6 Usage Notes

3.6.1 How to Start Up Cortex-R52 CPU0

After a RES# pin reset, a system software reset, or an error reset occurs, Cortex-R52 CPU0 starts up automatically in boot ROM in case of Cortex-R52 boot mode.

In case of Cortex-A55 boot mode, Cortex-R52 CPU0 remains WFI state. Module stop and reset is released. Start-up Cortex-R52 CPU0 by other CPU according to the below sequence.

1. Set filter unit 0 of TZC-400-8 to allow access to TCM of CPU0.
2. Store the instruction code to start address of the ATCM of CPU0 via AXIS interface (0x2000_0000). Since privileged access is only permitted to access AXIS interface, transaction must be privileged.
3. Reset CPU0 and release the reset state by setting 0x4321A502 to SWRCPU0 register.

3.6.2 How to Start Up Cortex-R52 CPU1

After a RES# pin reset, a system software reset, or an error reset occurs, Cortex-R52 CPU1 and WDT for CPU1 remain in the reset state and module stop state. Start-up Cortex-R52 CPU1 by other CPU according to the below sequence.

1. Set ATCM of CPU1 via AXIS interface (0x2100_000) to Device attribute.
2. Release from the module stop state by resetting MSTPCRN01 bit in MSTPCRN register.
3. Release from the reset state by setting 0x00000000 to SWRCPU1 register.
4. Release from the slave stop state by controlling AXIS1_REQ and AXIS1_ACK bits in SSTPCR7 register.
5. Set filter unit 1 of TZC-400-8 to allow access to TCM of CPU1.
6. Store the instruction code to start address of the ATCM of CPU1 via AXIS interface (0x2100_0000) with 64-bit aligned 64-bit access by STRD instruction. Since privileged access is only permitted to access AXIS interface after reset release, transaction must be privileged. It is recommended to initialize the remaining area of ATCM with the same access manner if it exists since ECC is enabled by default.
7. Start instruction fetch by writing 0 to CPU1HALT bit in CPU1HALT register.

3.6.3 Global System Counter (TSG-CNT)

The global system counter (TSG-CNT) is a 64-bit counter that provides system counter value to generic timers of Cortex-A55, Cortex-R52 CPU0 and CPU1. Before using generic timer, the global system counter related registers must be set. Clock for global system counter is the main oscillator clock (25 MHz). Therefore, the Global System Counter Base Frequency ID register (CNTFID0) must be set to 0x017D7840.

TSG-CNT is implemented in CoreSight. For details, see section 10, Debugging Interface.

3.6.4 Caution of Interrupt Service Routine

In case of level interrupt, it is needed to clear interrupt factor in the interrupt service routine (ISR) generally. It may take some time to complete it due to write data buffering in the CPU and internal bus. If interrupt handler exits before completion of clearing interrupt factor at the peripheral, interrupt handler starts again. And if clearing interrupt factor is completed before GIC register is read in the interrupt handler, INTID of 1023 (0x3FF) may be read since no pending interrupt is available at this moment. To avoid this situation, make sure that interrupt factor is cleared definitely by reading related register before exiting ISR.

4. Operating Modes

4.1 Overview

The LSI is intended for booting up from an external flash memory. Seven boot modes are available for the types of flash memory that supports seven different operating modes. In each of the boot modes, the user program stored in the corresponding external flash memory is booted up and runs. Secure boot mode (in which a user program is protected by encryption) can also be selected for the products that support security functions.

4.2 Types and Selection of Operating Modes

One of the seven types of operating modes can be selected, depending on the method of connecting to an external flash memory. An operating mode is selected based on the input levels of the mode setting pins (MD2, MD1, and MD0) at the time pin reset (except software reset) is released.

Table 4.1 describes the relationship between the input levels of the mode setting pins (MD2, MD1, and MD0) at the time reset is released and the selected operating mode. Value of the pins (MD2 to MD0) are latched to the registers when reset is released. For details on individual operating modes, see section 4.6. Operating Mode Descriptions.

Table 4.1 Selection of operating mode for each combination of levels of mode setting pins

Mode setting pins			
MD2	MD1	MD0	Operating mode
Low	Low	Low	xSPI0 boot mode (x1 boot serial flash) Boots a program from x1 boot serial flash memory connected to the xSPI0 CS0 space. Supporting voltage: 3.3 V or 1.8 V
Low	Low	High	xSPI0 boot mode (x8 boot serial flash) Boots a program from x8 boot serial flash memory such as HyperFlash memory connected to the xSPI0 CS0 space. Supporting voltage: 3.3 V or 1.8 V
Low	High	Low	xSPI1 boot mode (x1 boot serial flash) Boots a program from x1 boot serial flash memory connected to the xSPI1 CS0 space. Supporting voltage: 3.3 V or 1.8 V
Low	High	High	eSD boot mode Boots a program from eSD connected to SDHI1. Supporting voltage: 3.3 V
High	Low	Low	eMMC boot mode Boots a program from eMMC connected to SDHI0. Supporting voltage: 3.3 V or 1.8 V
High	Low	High	SCI (UART) boot mode Boots a program from host PC through UART communication connected to the SCI0. For flash writer use.
High	High	Low	USB boot mode Boots a program from host PC through USB. For flash writer use.
High	High	High	Reserved (setting prohibited)

IO voltage of boot peripheral which supports 3.3 V or 1.8 V is selected based on the input levels of the voltage setting pin (MDV). Table 4.2 shows the selection of IO voltage of boot peripheral which supports 3.3 V or 1.8 V. Value of the pin (MDV) is latched to the register when reset is released. The target operating mode is xSPI0 boot mode (x1, x8), xSPI1 boot mode (x1) and eMMC boot mode. For eSD boot mode, SCI boot mode and USB boot mode, MDV setting has no influence.

Table 4.2 Selection of IO voltage of boot peripheral which supports 3.3 V or 1.8 V

MDV	IO voltage of boot peripheral
Low	1.8 V
High	3.3 V

Wait cycle for ATCM in Cortex-R52 is selectable from 0 wait or 1 wait only during reset. So ATCM wait cycle setting pins (MDW0 for CPU0 and MDW1 for CPU1) are used to capture the value. The values of these pins are latched to the register when reset is released as well as MDn and MDV.

Table 4.3 Selection of ATCM wait cycle

MDW0/1	ATCM wait cycle
Low	0 wait. Valid in case CPU operating frequency is 500 MHz.
High	1 wait. Valid in case CPU operating frequency is 500 MHz or 1000 MHz. When using 1000 MHz, select this setting.

JTAG authentication by hash is handled by boot ROM. Boot ROM must identify whether JTAG is connected at boot sequence. JTAG authentication by hash enabling pin (MDD) is used to notify the CPU of whether JTAG is connected securely. The value of this pin is latched to the register when reset is released as well as other mode setting pins. This pin is valid for products that support security functions. For non-security support products, value of this pin is not used.

Table 4.4 Selection of JTAG authentication by Hash

MDD	JTAG mode
Low	Normal mode JTAG authentication by Hash is disabled
High	JTAG authentication by Hash mode

These setting pins (MD2 to MD0, MDV, MDW0/1, and MDD) can be shared with other peripheral function pins as strap option. The I/O voltage of these pins is voltage of the corresponding IO power domain (VDD1833_n).

4.3 Hardware Used in Individual Operating Modes

Table 4.5 describes hardware used in individual operating modes. Pins to be used indicates pins required to execute each operating mode. At boot, the initial values defined for each operating mode are automatically set to these pins by hardware.

Table 4.5 Hardware used in individual operating modes

Operating mode	Peripheral module	Pins to be used
xSPI0 boot mode (x1 boot Serial flash)	Expanded serial peripheral interface (xSPI0)	XSPI0_CKP XSPI0_CS0# XSPI0_IO0, XSPI0_IO1
xSPI0 boot mode (x8 boot Serial flash)	Expanded serial peripheral interface (xSPI0)	XSPI0_CKP, XSPI0_CKN XSPI0_CS0#, XSPI0_DS XSPI0_IO0 to XSPI0_IO7 XSPI0_RESET0#
xSPI1 boot mode (x1 boot Serial flash)	Expanded serial peripheral interface (xSPI1)	XSPI1_CKP XSPI1_CS0# XSPI1_IO0, XSPI1_IO1
eSD boot mode	SD/eMMC host interface (SDHI1)	SD1_CLK, SD1_CMD, SD1_DATA0 to SD1_DATA3
eMMC boot mode	SD/eMMC host interface (SDHI0)	SD0_CLK, SD0_CMD, SD0_DATA0 to SD0_DATA7, SD0_RST#
SCI (UART) boot mode	Serial communication interface (SCI)	TXD0, RXD0
USB boot mode	USB 2.0 HS function module	USB_QDP USB_QDM

Note: For relationship between pin function and I/O port on boot mode, see section 17.7.6. Pin Assignment of each Boot Mode.

4.4 Register Map

Table 4.6 Operating Mode register map

Address	Register symbol	Register name	Write protection
0x8029_4100	MD_MON	Operating Mode Monitor Register	—

Table 4.7 Operating Mode related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
MD_MON	—	—	Fixed to 0

4.5 Register Descriptions

4.5.1 MD_MON : Operating Mode Monitor Register

Base address: MD_NS = 0x8029_4100

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MDVMON	—	MD2MON	MD1MON	MD0MON	—	—	—	MDP	—	—	MDW1MON	MDW0MON	—	—	—	MDDMON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1	0	*1	*1	*1	0	0	0	*2	0	0	*1	*1	0	0	0	*1

Bit	Symbol	Function	R/W
0	MDDMON	MDD status flag This bit holds a value that indicates the level on the external MDD pin latched when reset is released.	R
3:1	—	These bits are read as 0.	R
4	MDW0MON	MDW0 status flag This bit holds a value that indicates the level on the external MDW0 pin latched when reset is released.	R
5	MDW1MON	MDW1 status flag This bit holds a value that indicates the level on the external MDW1 pin latched when reset is released.	R
7:6	—	These bits are read as 0.	R
8	MDP	Package type The bit indicates the package type of this device. 0: RZ/T2H 729pin FCBGA 1: RZ/N2H 576pin FCBGA	R
11:9	—	These bits are read as 0.	R
12	MD0MON	MD0 pin status flag This bit holds a value that indicates the level on the external MD0 pin latched when reset is released.	R
13	MD1MON	MD1 pin status flag This bit holds a value that indicates the level on the external MD1 pin latched when reset is released.	R
14	MD2MON	MD2 pin status flag This bit holds a value that indicates the level on the external MD2 pin latched when reset is released.	R
15	—	This bit is read as 0.	R
16	MDVMON	MDV status flag This bit holds a value that indicates the level on the external MDV pin latched when reset is released.	R
31:17	—	These bits are read as 0.	R

Note 1. The value depends on whether the signal is being externally pulled up or down, and the value at the time of latching on deassertion of the reset signal.

Note 2. The value depends on the package type.

The MD_MON register is for checking the mode (externally pulled up or down) of the MDD, MDW0/1, MD0 to MD2, and MDV pins by the CPU.

This register latches the values at the same time as deassertion of the signal on the external reset pin (RES#). Afterwards, it holds the values until the signal is asserted again. For reset sources and the timing of release from the reset state, see section 6, Reset.

4.6 Operating Mode Descriptions

4.6.1 Boot Function

After reset is released on this LSI, the boot function executes the boot processing on Cortex-R52 CPU0 as described below. The boot processing can extract a loader program that was stored in an external memory in advance by the user, to either the internal tightly coupled memory (TCM) area or System SRAM (SYSRAM) dependent on the second boot CPU, and hand over the processing to the loader program at the start address of that program. The loader program processes the second boot. Second boot CPU can be selected from Cortex-R52 CPU0 or Cortex-A55 Core 0.

(1) Setting the boot peripheral (xSPI or SDHI) specified by the mode setting pins (MD2, MD1, and MD0)

(2) Loading parameters for the loader from an external memory, and checking CHECK_SUM value

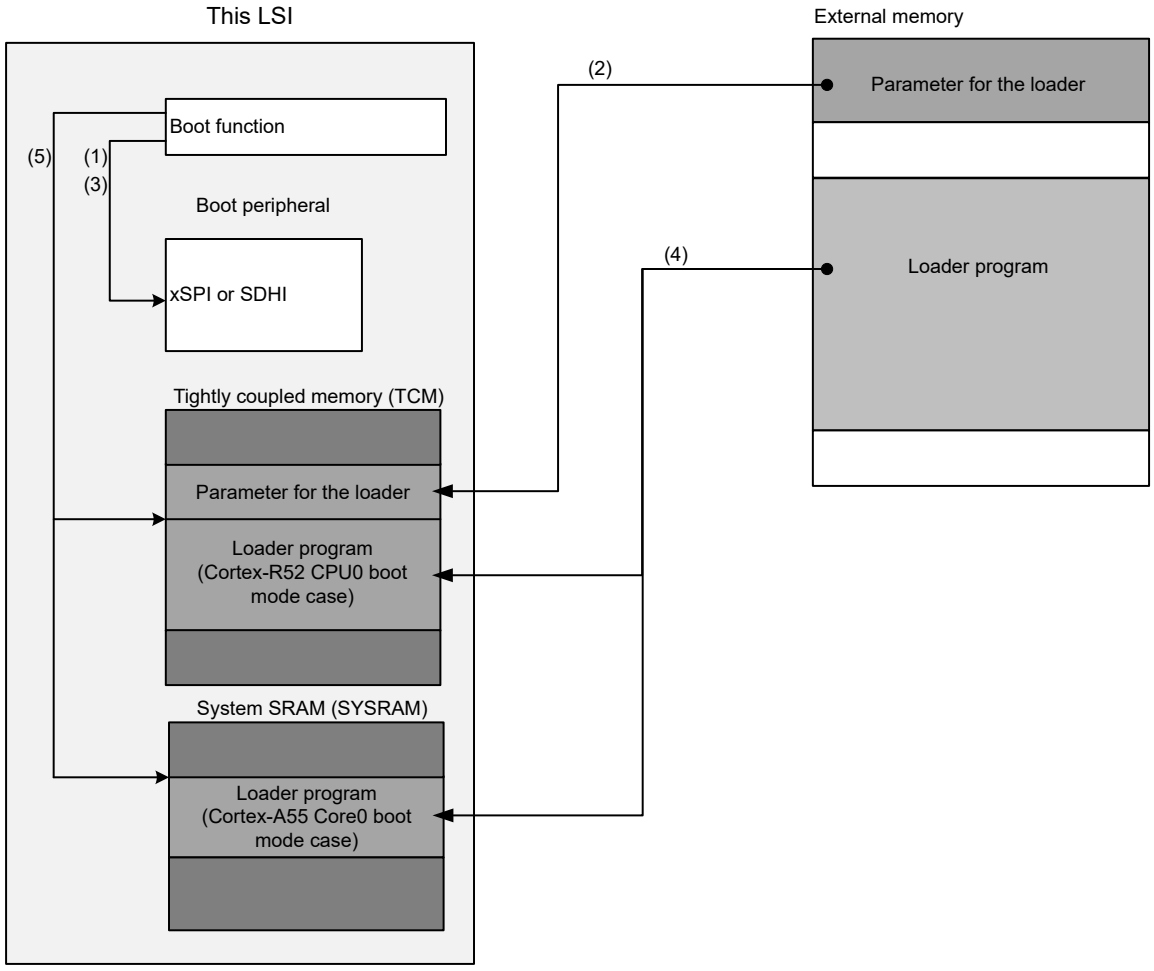
(3) Setting for speeding up the boot peripheral (xSPI or SDHI) by using parameters for the loader

(4) Loading the loader program from an external memory

(5) When Cortex-A55 is selected as second boot CPU, reset of Cortex-A55 Core0 is released and Cortex-R52 CPU0 transitioned to the WFI state instead. Then, second boot CPU (Cortex-R52 CPU0 or Cortex-A55 Core0) branches off to the start address of the loader program extracted to the tightly coupled memory (TCM) for Cortex-R52 CPU0 boot or System SRAM (SYSRAM) for Cortex-A55 Core0 boot.

Parameters for the loader can have configuration information that is suited for the user system, such as loader program information, cache setting for speeding up the boot processing, and boot peripheral (xSPI or SDHI) settings. Parameters for the loader must be stored in an external memory in advance by the user.

Figure 4.1 shows the operating overview of boot processing.



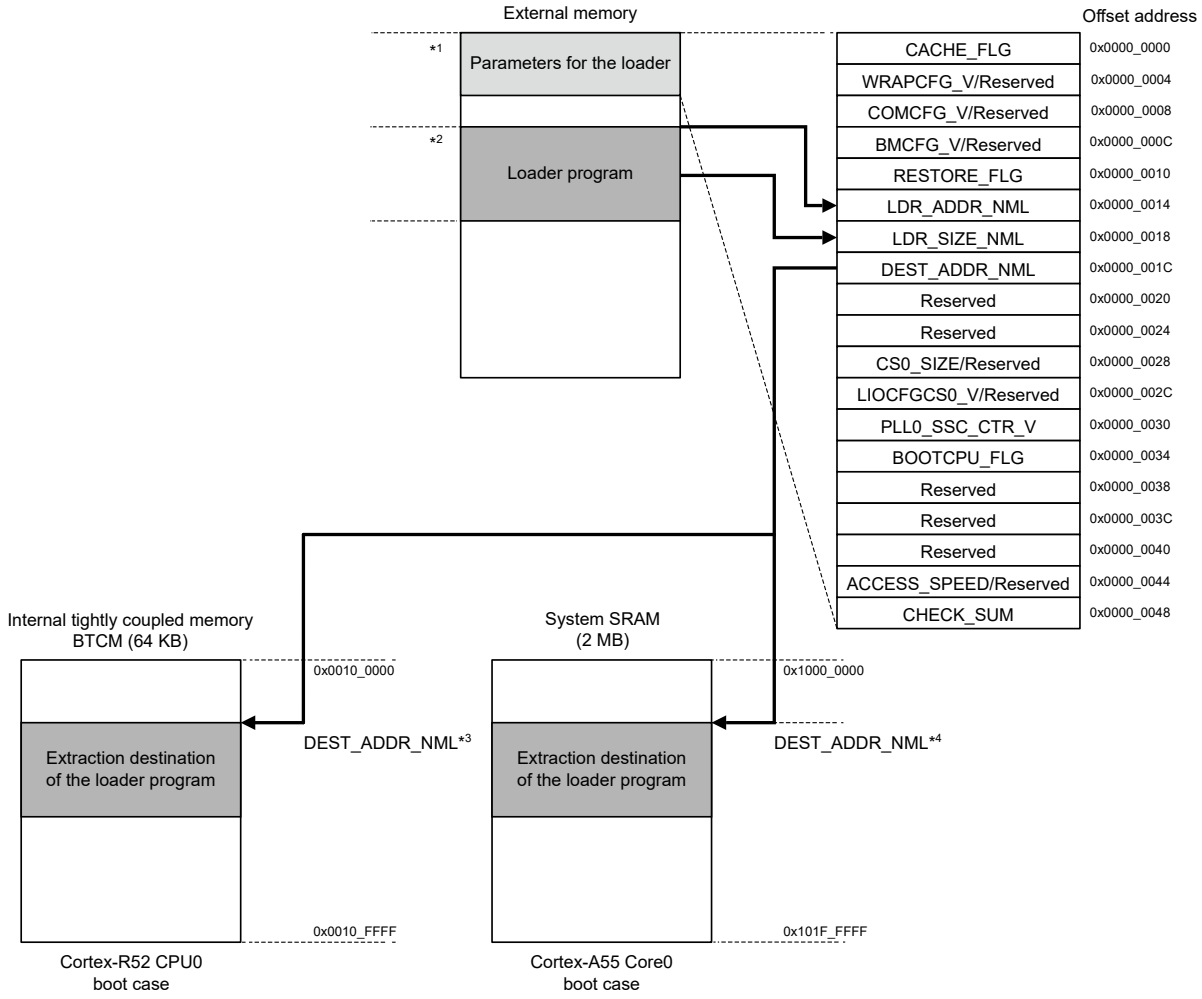
Note: (1) Set the boot peripheral (xSPI or SDHI) specified by the operation mode.
(2) Transfer the parameters for the loader from an external memory.
(3) Configure the boot peripheral on the parameters for the loader.
(4) Transfer the loader program from an external memory.
(5) When Cortex-A55 is selected as second boot CPU, release the reset of Cortex-A55 Core0 and transition Cortex-R52 CPU0 to the WFI state. Then, hand over the second boot CPU processing to the loader program in the TCM area (Cortex-R52 CPU0 boot mode case) or SYSRAM area (Cortex-A55 Core0 boot Mode case).

Figure 4.1 Operating overview of boot processing

4.6.2 Parameters for the Loader

The parameters for the loader are setting parameters for boot processing, which are loaded from an external memory during boot processing and used by the boot function. The parameters for the loader specify information, such as the cache settings during boot processing in individual operating modes, setting of the boot peripheral (xSPI or SDHI) used for communication with an external memory, and the size of the loader program.

Figure 4.2 shows memory assignment of the loader program and parameters for the loader.



Note 1. The parameters for the loader must be stored in the following address range:

xSPI boot mode: xSPI0 or xSPI1 space (0x4000_0000 and higher or 0x5000_0000 and higher)

eSD boot mode: Sector 1 to 7 for multiplexing

eMMC boot mode: Sector 1

Note 2. The loader program can be stored in any address for xSPI boot mode and in 512 byte align address for eSD/eMMC boot mode.

For details, see section 4.6.5. eSD Boot Mode and section 4.6.6. eMMC Boot Mode.

The program size (LDR_SIZE_NML) must be multiples of 512 bytes. Maximum size is 52 KB for BTCM and 2036 KB for SYSRAM.

Note 3. Specify the extraction destination area of the loader program so that the area falls in the range from 0x0010_2000 to 0x0010_EFFF for Cortex-R52 CPU0 boot case.

Note 4. Specify the extraction destination area of the loader program so that the area falls in the range from 0x1000_0000 to 0x101F_CFFF for Cortex-A55 Core0 boot case.

Figure 4.2 Memory assignment of the loader program and parameters for the loader

Table 4.8 describes parameter information for the loader in xSPI boot mode. Table 4.9 describes parameter information for the loader in eSD/eMMC boot mode.

Table 4.8 Parameter information for the loader in xSPI boot mode (1 of 3)

Offset address	Parameter name	Description
0x0000_0000	CACHE_FLG	Selects whether to enable the I1 cache and D1 cache of Cortex-R52 at boot processing (for speeding up). 0x0000_0001: Enables the I1 and D1 caches. Setting values other than above: Disables the I1 and D1 caches.

Table 4.8 Parameter information for the loader in xSPI boot mode (2 of 3)

Offset address	Parameter name	Description
0x0000_0004	WRAPCFG_V	Setting value of the xSPI Wrapper Configuration register (WRAPCFG). This parameter value is set to the DSSFTCS0[4:0] bits in the WRAPCFG register during the setting for speeding up of xSPI in (3) of section 4.6.1. Boot Function *1. The bit 13 of WRAPCFG_V is used as the flag. In case of WRAPCFG_V[13] = 0: This parameter value is ignored and kept the reset value. In case of WRAPCFG_V[13] = 1: This parameter value is valid and set to the WRAPCFG register. The DSSFTCS1[4:0] bits in the WRAPCFG register remain the reset value.
0x0000_0008	COMCFG_V	Setting value of the xSPI Common Configuration register (COMCFG). This parameter value is set to the COMCFG register during the setting for speeding up of xSPI in (3) of section 4.6.1. Boot Function *1.
0x0000_000C	BMCFG_V	Setting value of the xSPI Bridge Map Configuration register (BMCFG). This parameter value is set to the PREEN bit in the BMCFG register during the setting for speeding up of xSPI in (3) of section 4.6.1. Boot Function *1. The other bits in the BMCFG register remain the reset value.
0x0000_0010	RESTORE_FLG	Selects whether to change the xSPI setting back to the initial value after the boot processing finishes. 0x2236_0679: Changes the xSPI setting value back to the initial value after the boot processing finishes. Setting values other than above: Retains the xSPI setting value used during boot processing.*2
0x0000_0014	LDR_ADDR_NML	Sets the start address of the loader program stored in the external memory.
0x0000_0018	LDR_SIZE_NML	Specifies the size of the loader program. Note that the program size must be multiples of 512 bytes. Maximum size is 52 KB for BTCM in Cortex-R52 CPU0 boot case and 2036 KB for SYSRAM in Cortex-A55 Core0 boot case.*3
0x0000_001C	DEST_ADDR_NML	Specifies the start address of the tightly coupled memory (BTCM) in Cortex-R52 CPU0 boot case or System SRAM (SYSRAM) in Cortex-A55 Core0 boot case that is used as the extraction destination of the loader program. Specify the extraction destination area of the loader program so that the area falls in the range from 0x0010_2000 to 0x0010_EFFF in BTCM for Cortex-R52 CPU0 boot case or from 0x1000_0000 to 0x101F_CFFF in SYSRAM for Cortex-A55 Core0 boot case. The address should be 4-byte aligned. If loader program is Thumb, bit0 should be flagged to 1.
0x0000_0020	Reserved	Should be 0x0000_0000
0x0000_0024	Reserved	Should be 0x0000_0000
0x0000_0028	CS0_SIZE	Sets the size of the device attached to xSPI CS0 area. Value added base address (0x4000_0000 for xSPI0 or 0x5000_0000 for xSPI1) to this parameter value is set to the CS0ENDAD register during the setting for speeding up of xSPI in (3) of section 4.6.1. Boot Function*1. Value should be less than 0x0800_0000.
0x0000_002C	LIOCFGCS0_V	Setting value of the xSPI link I/O Configuration Register CS0 (LIOCFGCS0). This parameter value is set to the LIOCFGCS0 register during the setting for speeding up of xSPI in (3) of section 4.6.1. Boot Function*1. This parameter is effective on only bit 31 to bit 16 of LIOCFGCS0 register. Bit 11 to bit 0 are fixed each boot mode (x1 or x8). The bit 14 of LIOCFGCS0_V is used as the flag. In case of LIOCFGCS0_V[14] = 0: This parameter value is ignored and kept the reset value. In case of LIOCFGCS0_V[14] = 1: This parameter value is valid and set to the LIOCFGCS0 register.
0x0000_0030	PLL0_SSC_CTR_V	Setting value of the PLL0 SSC Control Register (PLL0_SSC_CTR). This parameter value is set to the PLL0_SSC_CTR register during (3) of section 4.6.1. Boot Function when the second boot CPU is selected Cortex-A55 Core0.
0x0000_0034	BOOTCPU_FLG	Selects the second boot CPU. 0x0000_0000: Cortex-R52 CPU0 0x0000_0001: Cortex-A55 Core0 Setting values other than above: Prohibited
0x0000_0038	Reserved	Should be 0x0000_0000
0x0000_003C	Reserved	Should be 0x0000_0000

Table 4.8 Parameter information for the loader in xSPI boot mode (3 of 3)

Offset address	Parameter name	Description
0x0000_0040	Reserved	Should be 0x0000_0000
0x0000_0044	ACCESS_SPEED	This parameter is used to adjust access speed for the external memory. For xSPI boot mode using xSPI0, this value is set to FSELXSPI0[2:0] bits and DIVSELXSPI0 bit of SCKCR register. For example, Bit 6: DIVSELXSPI0 = 0 The base clock to generate serial clock is 800 MHz. Bits [2:0]: FSELXSPI0[2:0] = 100b. The clock frequency is 50.0 MHz. Other bit: Is not used to set SCKCR register. For xSPI boot mode using xSPI1, set bit 6 and bits [2:0] of this parameter as well as xSPI0 to set FSELXSPI1[2:0] bits and DIVSELXSPI1 bit of SCKCR register. Bit 14 and bits [10:8] of this parameter are ignored.
0x0000_0048	CHECK_SUM	CHECK_SUM value of the parameters for the loader This parameter specifies the sum of the upper-order 16 bits and the lower-order 16 bits of the parameters (in unsigned long (32-bit) format) in the range of the offset addresses 0x0000 to 0x0044.

Note 1. For details on the WRAPCFG, COMCFG, BMCFG, CS0ENDAD, and LIOCFGCS0 registers, see section 37, Expanded Serial Peripheral Interface (xSPI) (n = 0, 1).
Note 2. For details about the setting status of the individual peripheral modules after the boot processing finishes, see section 4.6.4. xSPI Boot Mode.
Note 3. The loader program must be in the range from 0x4000_004C to 0x40FF_FE00 (xSPI0) or 0x5000_004C to 0x50FF_FE00 (xSPI1) in the external address space.

Table 4.9 Parameter information for the loader in eSD or eMMC boot mode (1 of 2)

Offset address	Parameter name	Description
0x0000_0000	CACHE_FLG	Selects whether to enable the I1 cache and D1 cache of Cortex-R52 at boot processing (for speeding up). 0x0000_0001: Enables the I1 and D1 caches. Setting values other than above: Disables the I1 and D1 caches.
0x0000_0004	Reserved	Should be 0x0000_0000
0x0000_0008	Reserved	Should be 0x0000_0000
0x0000_000C	Reserved	Should be 0x0000_0000
0x0000_0010	RESTORE_FLG	Selects whether to change the SDHI setting back to the initial value after the boot processing finishes. 0x2236_0679: Changes the SDHI setting value back to the initial value after the boot processing finishes. Setting values other than above: Retains the SDHI setting value used during boot processing*1.
0x0000_0014	LDR_ADDR_NML	Sets the start address of the loader program stored in the external memory.
0x0000_0018	LDR_SIZE_NML	Specifies the size of the loader program. Note that the program size must be multiples of 512 bytes. Maximum size is 52 KB for BTCM in Cortex-R52 CPU0 boot case and 2036 KB for SYSRAM in Cortex-A55 Core0 boot case*2.
0x0000_001C	DEST_ADDR_NML	Specifies the start address of the tightly coupled memory (BTCM) or System SRAM (SYSRAM) that is used as the extraction destination of the loader program. Specify the extraction destination area of the loader program so that the area falls in the range from 0x0010_2000 to 0x0010_EFFF in BTCM for Cortex-R52 CPU0 boot case or from 0x1000_0000 to 0x101F_CFFF in SYSRAM for Cortex-A55 Core0 boot case.
0x0000_0020	Reserved	Should be 0x0000_0000
0x0000_0024	Reserved	Should be 0x0000_0000
0x0000_0028	Reserved	Should be 0x0000_0000
0x0000_002C	Reserved	Should be 0x0000_0000
0x0000_0030	PLL0_SSC_CTR_V	Setting value of the PLL0 SSC Control Register (PLL0_SSC_CTR). This parameter value is set to the PLL0_SSC_CTR register during (3) of section 4.6.1. Boot Function when the second boot CPU is selected Cortex-A55 Core0.

Table 4.9 Parameter information for the loader in eSD or eMMC boot mode (2 of 2)

Offset address	Parameter name	Description
0x0000_0034	BOOTCPU_FLG	Selects the second boot CPU. 0x0000_0000: Cortex-R52 CPU0 0x0000_0000: Cortex-A55 Core0 Setting values other than above: Prohibited
0x0000_0038	Reserved	Should be 0x0000_0000
0x0000_003C	Reserved	Should be 0x0000_0000
0x0000_0040	Reserved	Should be 0x0000_0000
0x0000_0044	Reserved	Should be 0x0000_0000
0x0000_0048	CHECK_SUM	CHECK_SUM value of the parameters for the loader This parameter specifies the sum of the upper-order 16 bits and the lower-order 16 bits of the parameters (in unsigned long (32-bit) format) in the range of the offset addresses 0x0000 to 0x0044.

Note 1. For details about the setting status of the individual peripheral modules after the boot processing finishes, see section 4.6.5. eSD Boot Mode and section 4.6.6. eMMC Boot Mode.

Note 2. The loader program must be placed at sector 8 or after for eSD boot mode and at sector 2 or after for eMMC boot mode. For details, see section 4.6.5. eSD Boot Mode and section 4.6.6. eMMC Boot Mode.

4.6.3 Loader Program

The loader program is a user program that is transferred from an external memory to the internal tightly coupled memory (TCM) or System SRAM (SYSRAM) by the boot function and starts its processing after the boot processing finishes. The loader program can execute such processing that suits the user system, for example, extracting a user application program from an external memory to the internal TCM area or System SRAM (SYSRAM) and executing it at high speed.

Set the loader program so that the following conditions are satisfied:

Program size (LDR_SIZE_NML)	Multiples of 512 bytes. Maximum size is 52 KB for BTCM in Cortex-R52 CPU0 boot case and 2036 KB for SYSRAM in Cortex-A55 Core0 boot case.
Storage address in the external memory (LDR_ADDR_NML) in xSPI0 boot mode	Address range from 0x4000_004C to 0x40FF_FE00
Storage address in the external memory (LDR_ADDR_NML) in xSPI1 boot mode	Address range from 0x5000_004C to 0x50FF_FE00
Storage address in the eSD memory (LDR_ADDR_NML) in eSD boot mode	Sector 8 or after. (Address should be 512byte aligned)
Storage address in the eMMC memory (LDR_ADDR_NML) in eMMC boot mode	Sector 2 or after. (Address should be 512byte aligned)

These setting values must be stored in an external memory as parameters for the loader. For details, see section 4.6.2. Parameters for the Loader.

4.6.4 xSPI Boot Mode

4.6.4.1 x1 Boot Serial Flash

In xSPI boot mode with x1 boot serial flash, this LSI boots a program from an external serial flash memory connected to the expanded Serial Peripheral Interface (xSPI0 or xSPI1) space CS0 with single, dual, quad, or octal data signals. In this mode, used protocol is 1S-1S-1S regardless of number of data signals of the flash. The xSPI controller is set to the mode of reading the external address space, and the XSPI0_CKP, XSPI0_CS0#, and XSPI0_IO[1:0] pins are enabled for MD[2:0] = 000b. XSPI1_CKP, XSPI1_CS0#, and XSPI1_IO[1:0] pins are enabled for MD[2:0] = 010b.

After the reset is released, the LSI executes the boot processing on Cortex-R52 CPU0. The program stored in a serial flash memory connected to the expanded SPI space is extracted to the internal tightly coupled memory (TCM) or System SRAM (SYSRAM), and then the processing is executed.

Figure 4.3 shows the connection diagram of this LSI with a serial flash memory.

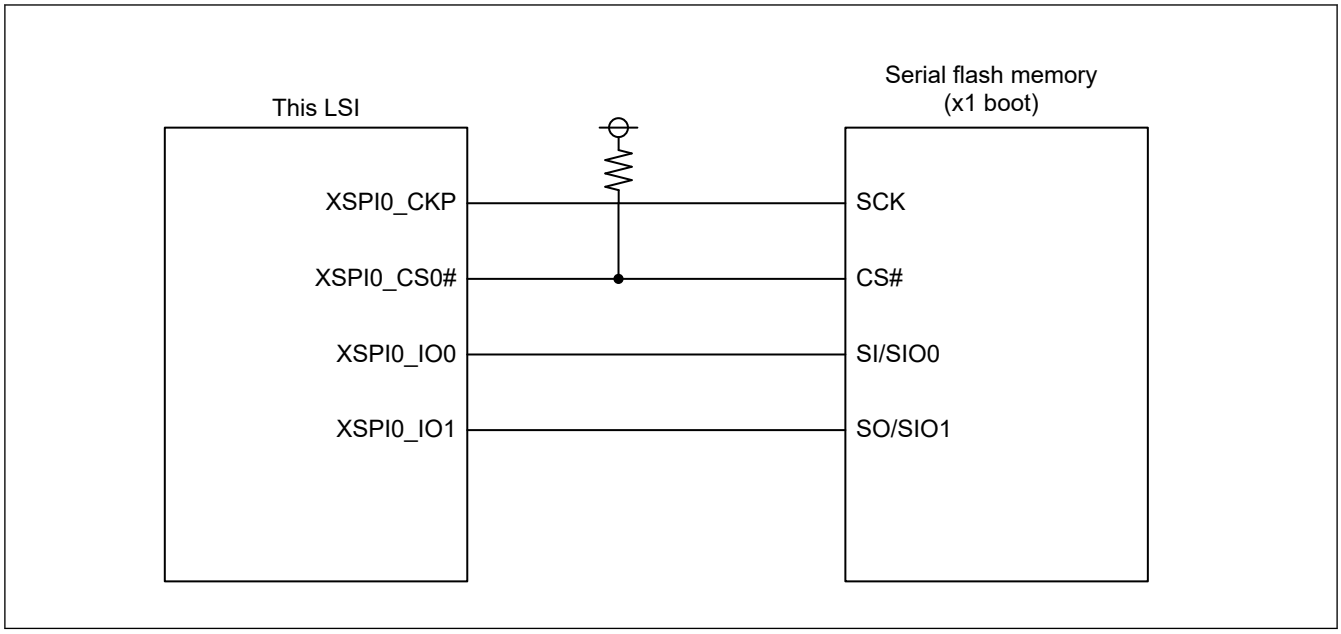


Figure 4.3 Connection diagram of this LSI with x1 boot serial flash memory

4.6.4.2 x8 Boot Serial Flash

In xSPI boot mode with x8 boot serial flash, this LSI boots a program from an external x8 serial flash memory such as HyperFlash and xSPI 8D-8D-8D profile 2.0 compliant device connected to the expanded Serial Peripheral Interface (xSPI0) space CS0 with octal data signals. In this mode, the xSPI0 controller is set to the mode of reading the external address space, and the XSPI0_CKP, XSPI0_CKN, XSPI0_CS0#, XSPI0_IO[7:0], XSPI0_DS, and XSPI0_RESET0# pins are enabled.

After the reset is released, the LSI executes the boot processing on Cortex-R52 CPU0. The program stored in a serial flash memory connected to the expanded SPI space is extracted to the internal tightly coupled memory (TCM) or System SRAM (SYSRAM), and then the processing is executed.

Figure 4.4 shows the connection diagram of this LSI with a serial flash memory.

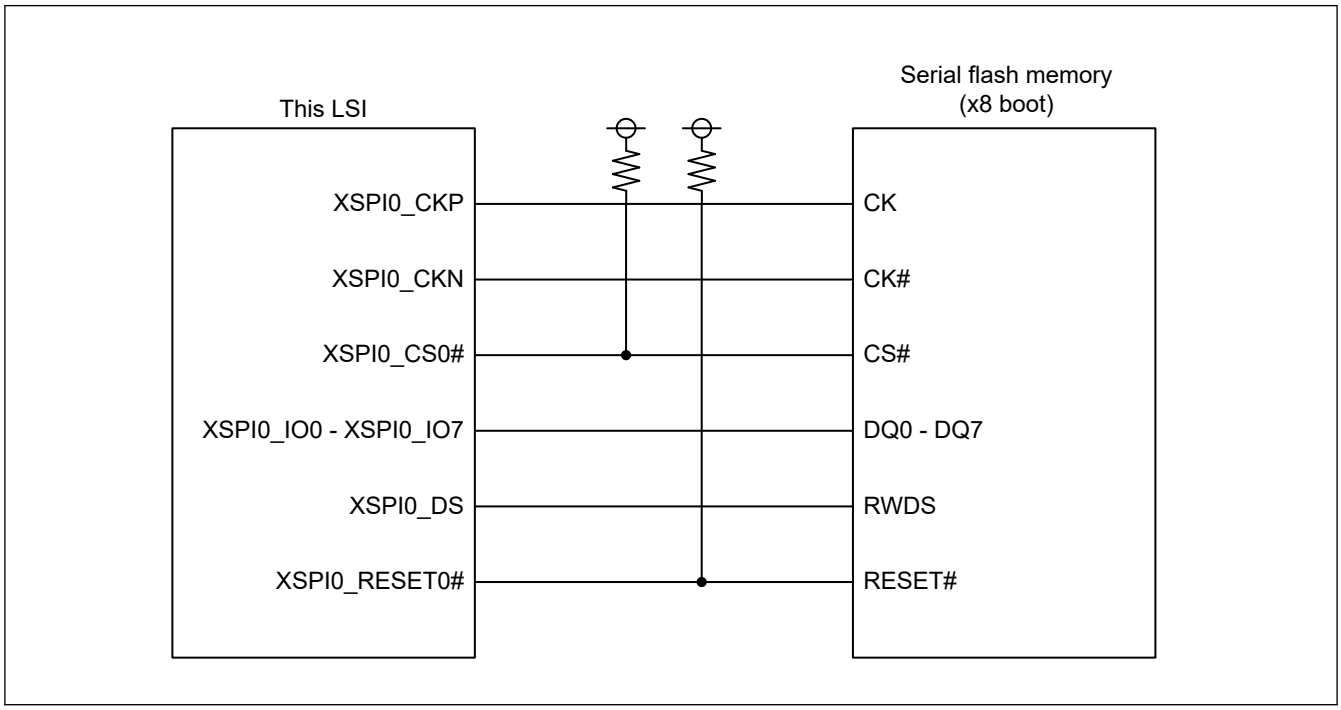


Figure 4.4 Connection diagram of this LSI with x8 boot serial flash memory

4.6.4.3 Operation Settings in xSPI Boot Mode

Immediately after the boot processing starts in xSPI boot mode after the reset is released, the LSI operates with the following initial setting values and executes processing until it transfers parameters for the loader.

Items	x1 boot	x8 boot
CPU clock (CPU0CLK)	500 MHz	
xSPI bit rate (XSPI_CLKn)	12.5 MHz*1	
Supported command	Read (0x03)	Read (CA[47:45] = 101b)
Command output	1 byte	6 bytes
Address output	3 bytes	
Dummy cycle (Read latency cycle)	0	10 Latency cycle increments from CA[23:16]
Initial data width	1 bit	8 bits
Protocol	1S-1S-1S	8D-8D-8D profile 2.0
Driving edge	Falling edge of XSPIn_CKP	Both edges of XSPI0_CKP
Sampling edge	Falling edge of XSPIn_CKP	Both edges of XSPI0_DS with shift*2
Flash reset	Software (0x66 + 0x99)	Hardware by XSPI0_RESET0#

Note 1. Serial clock frequency is configurable in parameter.
Note 2. DS shift size is configurable in parameter.

Table 4.10 describes the setting values of the related peripheral modules and registers at the time the xSPI boot mode finishes.

Table 4.10 Setting register values of the related peripheral modules when xSPI boot mode finishes (1 of 2)

Peripheral module	Register	When xSPI is initialized (RESTORE_FLG = 0x22360679)		When xSPI is not initialized (RESTORE_FLG ≠ 0x22360679)	
		x1	x8	x1	x8
xSPI	WRAPCFG	0x00000000 (initial value)	0x00000000 (initial value)	Setting value of WRAPCFG_V (only DSSFTCS0[4:0] bits) If flag bit is not set, 0x00000000.	Setting value of WRAPCFG_V (only DSSFTCS0[4:0] bits) If flag bit is not set, 0x00001F00.
	COMCFG	0x00000000 (initial value)	0x00000000 (initial value)	Setting value of COMCFG_V	Setting value of COMCFG_V
	BMCFG	0x00000000 (initial value)	0x00000000 (initial value)	Setting value of BMCFG_V (only PREEN bit)	Setting value of BMCFG_V (only PREEN bit)
	CMCFG0CS0	0x00000008	0x0000000E	0x00000008	0x0000000E
	CMCFG1CS0	0x00000300	0x000AA000	0x00000300	0x000AA000
	LIOCFGCS0	Setting value of LIOCFGCS0_V (bit 31 to bit 16) 0x0000 (bit 15 to bit 0) If flag bit is not set, 0x00070000.	Setting value of LIOCFGCS0_V (bit 31 to bit 16) 0x0000 (bit 15 to bit 0) If flag bit is not set, 0x000707FF.	Setting value of LIOCFGCS0_V (bit 31 to bit 16) 0x0000 (bit 15 to bit 0) If flag bit is not set, 0x00070000.	Setting value of LIOCFGCS0_V (bit 31 to bit 16) 0x0000 (bit 15 to bit 0) If flag bit is not set, 0x000707FF.
	BMCTL0	0x00000001	0x00000001	0x00000001	0x00000001
	CS0ENDAD	Setting value of xSPI base (0x4000_0000 for xSPI0, 0x5000_0000 for xSPI1) + CS0_SIZE	Setting value of xSPI base (0x4000_0000 for xSPI0, 0x5000_0000 for xSPI1) + CS0_SIZE	Setting value of xSPI base (0x4000_0000 for xSPI0, 0x5000_0000 for xSPI1) + CS0_SIZE	Setting value of xSPI base (0x4000_0000 for xSPI0, 0x5000_0000 for xSPI1) + CS0_SIZE

Table 4.10 Setting register values of the related peripheral modules when xSPI boot mode finishes (2 of 2)

Peripheral module	Register	When xSPI is initialized (RESTORE_FLG = 0x22360679)		When xSPI is not initialized (RESTORE_FLG ≠ 0x22360679)	
		x1	x8	x1	x8
Clock	SCKCR	0x00730606 (initial value)	0x00730606 (initial value)	Setting value of ACCESS_SPEED (only bit 6, bit 2 to bit 0 for xSPI0 and bit 14, bit 10 to bit 8 for xSPI1)	Setting value of ACCESS_SPEED (only bit 6, bit 2 to bit 0)
	PMSEL*2	0x00002601	0x00002601	0x00002601	0x00002601
	PLL0MON*2	0x00000001	0x00000001	0x00000001	0x00000001
	PLL0EN*2	0x00000001	0x00000001	0x00000001	0x00000001
	PLL0_SSC_CTR*2	Setting value of PLL0_SSC_CTR_V	Setting value of PLL0_SSC_CTR_V	Setting value of PLL0_SSC_CTR_V	Setting value of PLL0_SSC_CTR_V
Reset	MRCTLA	xSPI0: 0x00000020 xSPI1: 0x00000010	xSPI0: 0x00000020	xSPI0: 0x00000020 xSPI1: 0x00000010	xSPI0: 0x00000020
	SWR55C*2	0x00000000	0x00000000	0x00000000	0x00000000
	SWR550*2	0x00000000	0x00000000	0x00000000	0x00000000
Low power consumption	MSTPCRA	xSPI0: 0x0FFF1F21 xSPI1: 0x0FFF1F11	xSPI0: 0x0FFF1F21	xSPI0: 0x0FFF1F21 xSPI1: 0x0FFF1F11	xSPI0: 0x0FFF1F21
	MSTPCRN*2	0x0000003A	0x0000003A	0x0000003A	0x0000003A
I/O ports*1	PMCm	xSPI0: 0xCA (m = 05) xSPI1: 0x33 (m = 01)	xSPI0: 0xEE (m = 05), 0x7F (m = 06)	xSPI0: 0xCA (m = 05) xSPI1: 0x33 (m = 01)	xSPI0: 0xEE (m = 05), 0x7F (m = 06)
	PFCm	xSPI0: 0x1C1C0000_1C001C00 (m = 05) xSPI1: 0x00001C1C_00001C1C (m = 01)	xSPI0: 0x1C1C1C00_1C1C1C00 (m = 05), 0x001C1C1C_1C1C1C1C (m = 06)	xSPI0: 0x1C1C0000_1C001C00 (m = 05) xSPI1: 0x00001C1C_00001C1C (m = 01)	xSPI0: 0x1C1C1C00_1C1C1C00 (m = 05), 0x001C1C1C_1C1C1C1C (m = 06)
	DRCTLm	xSPI0: (3.3 V) 0x32320000_32003200 (m = 05) (1.8 V) 0x22220000_22002200 (m = 05) xSPI1: (3.3 V) 0x00003232_00003232 (m = 01) (1.8 V) 0x00002222_00002222 (m = 01)	xSPI0: (3.3 V) 0x32323200_32323200 (m = 05) 0x00003232_32323200 (m = 06) (1.8 V) 0x22222200_22222200 (m = 05) 0x00002222_22222200 (m = 06)	xSPI0: (3.3 V) 0x32320000_32003200 (m = 05) (1.8 V) 0x22220000_22002200 (m = 05) xSPI1: (3.3 V) 0x00003232_00003232 (m = 01) (1.8 V) 0x00002222_00002222 (m = 01)	xSPI0: (3.3 V) 0x32323200_32323200 (m = 05) 0x00003232_32323200 (m = 06) (1.8 V) 0x22222200_22222200 (m = 05) 0x00002222_22222200 (m = 06)
TZC-400	GATE_KEEPER	0x000F000F (TZC-400-5) 0x00030003 (TZC-400-7)	0x000F000F (TZC-400-5) 0x00030003 (TZC-400-7)	0x000F000F (TZC-400-5) 0x00030003 (TZC-400-7)	0x000F000F (TZC-400-5) 0x00030003 (TZC-400-7)
	SPECULATION_CTRL	0x00000003 (TZC-400-5, TZC-400-7)	0x00000003 (TZC-400-5, TZC-400-7)	0x00000003 (TZC-400-5, TZC-400-7)	0x00000003 (TZC-400-5, TZC-400-7)
	REGION_ATTRIBUTES_0	0xC000000F (TZC-400-5) 0x00000003 (TZC-400-7)	0xC000000F (TZC-400-5) 0x00000003 (TZC-400-7)	0xC000000F (TZC-400-5) 0x00000003 (TZC-400-7)	0xC000000F (TZC-400-5) 0x00000003 (TZC-400-7)
	REGION_ID_ACCESS_0	0x000F000F (TZC-400-5, TZC-400-7)	0x000F000F (TZC-400-5, TZC-400-7)	0x000F000F (TZC-400-5, TZC-400-7)	0x000F000F (TZC-400-5, TZC-400-7)
Cortex-A55	RVBAL0*2	0x101FD000	0x101FD000	0x101FD000	0x101FD000

Note 1. For details, see section 17.7.6.1. Pin Assignment for xSPI Boot Mode.
Note 2. In case the second boot CPU is Cortex-A55 Core0.

4.6.5 eSD Boot Mode

In eSD boot mode, this LSI boots a program from an external eSD device with 4-bit data bus. The operating voltage is set to 3.3V and not changed during boot process. This LSI supports the embedded SD (eSD) defined in the SD Specification Part 1 eSD Addendum (Version 2.10). In this mode, the SD1_CLK, SD1_CMD, and SD1_DATA[3:0] pins are enabled.

After the reset is released, the LSI executes the boot processing on Cortex-R52 CPU0. The eSD device connected to the SDHI1 interface is mounted and issued a read command to obtain the loader parameter from sector 1 of the selected partition. After obtaining the loader parameter, loader program stored in the embedded SD is extracted to the internal tightly coupled memory (TCM) or System SRAM (SYSRAM) dependent on the selected second boot CPU, and then the processing is executed. In this mode, PLL2 is automatically activated with disabled SSC.

Figure 4.5 shows the connection diagram of this LSI with embedded SD (eSD) device.

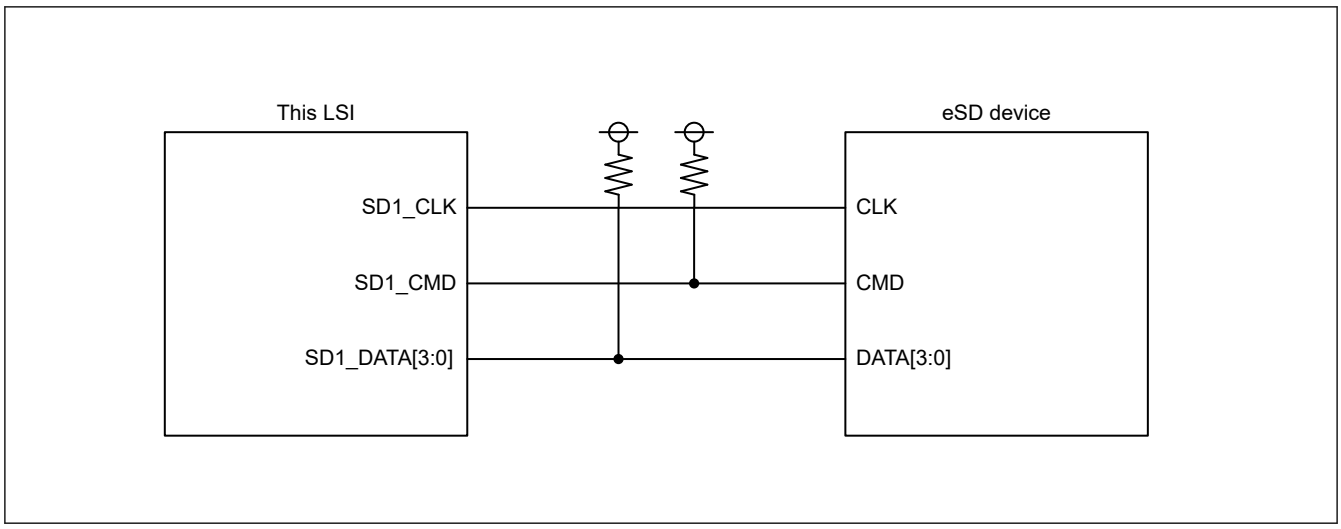


Figure 4.5 Connection diagram of this LSI with eSD device

4.6.5.1 Allocation of the Loader Parameter and Loader Program

(1) Allocation in eSD V2.0 (Single Partition)

Figure 4.6 shows the allocation of the loader parameter and loader programs in the eSD device conforming to the eSD V2.0 standard. To prevent read-disturb errors, up to seven loader parameter and loader programs can be multiplexed and written to the eSD device.

The loader parameters are allocated to sectors 1 to 7 in physical partition #0 of the eSD device. If a communication error occurs during data transfer, read access is retried up to three times per area. If the retry processing is repeated three times during the transfer of the loader parameter in a single area, another loader parameter is transferred from the next multiplexed area.

The loader programs are allocated to address specified in the loader parameter (sector 8 or after). If a communication error occurs during data transfer, read access is retried up to three times per area. If the retry processing is repeated three times during the transfer of the loader parameter in a single area, another loader program is transferred from the next multiplexed area.

Note that even when the loader parameter is transferred from a multiplexed (reserved) area, the transfer of the loader program always begins from the area for loader program #0.

Figure 4.7 shows the structure of the loader parameter in a sector. First 76 bytes defined in Table 4.9 are valid and the rest is not used.

(2) Allocation in eSD V2.1 (Multi Partitions)

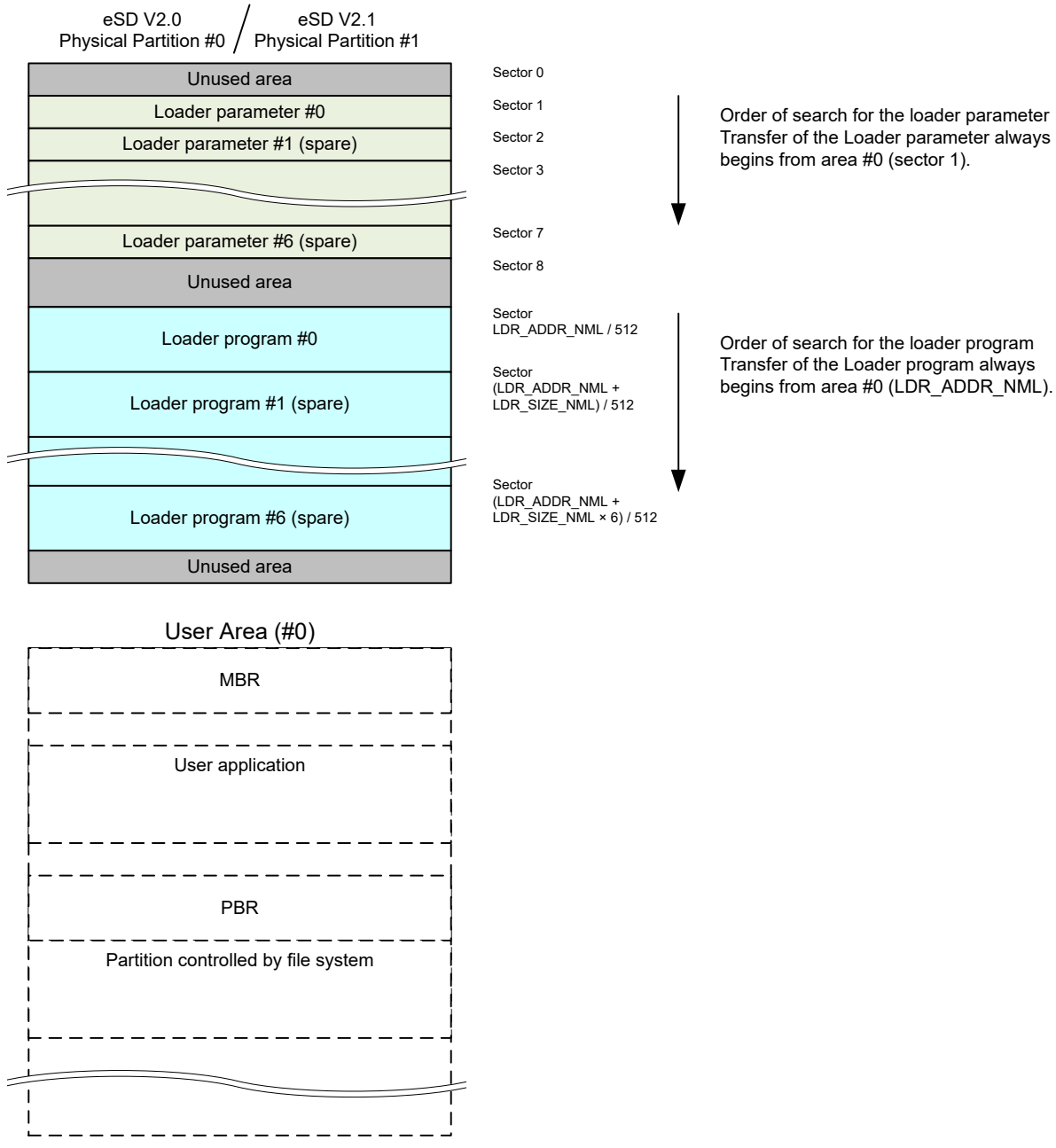
Basically, same as eSD V2.0 standard except physical partition number to be transferred from. In case of eSD V2.1, loader parameter and loader program are transferred from physical partition #1 of the eSD device.

(3) How to Distinguish between eSD V2.0 (Single Partition) and V2.1 (Multi Partitions)

This boot program first assumes that an eSD device supporting multi partitions is connected and begins loader parameter transfer from physical partition #1. If an error regarding a command for multi partitions shown in the Table 4.11 occurs, the boot program assumes that an eSD device with a single partition is connected and transfers the loader program from physical partition #0.

Table 4.11 Errors regarding commands for multi partitions

No.	Issued Command	Command Function	Error
1	CMD43 (SELCT_PARTITION)	Selects physical partition #1.	Multi partitions are not supported.
2	CMD45 (QUERY_PARTITION)	Obtains the physical partition ID.	The specified partition does not exist.



Note: Write the same data to loader parameter areas #0 to #6.
Write the same data to loader program areas #0 to #6.
The start sector location of the loader program #n is (LDR_ADDR_NML + LDR_SIZE_NML × n) / 512.

Figure 4.6 Allocation of loader parameter and loader program in the eSD device

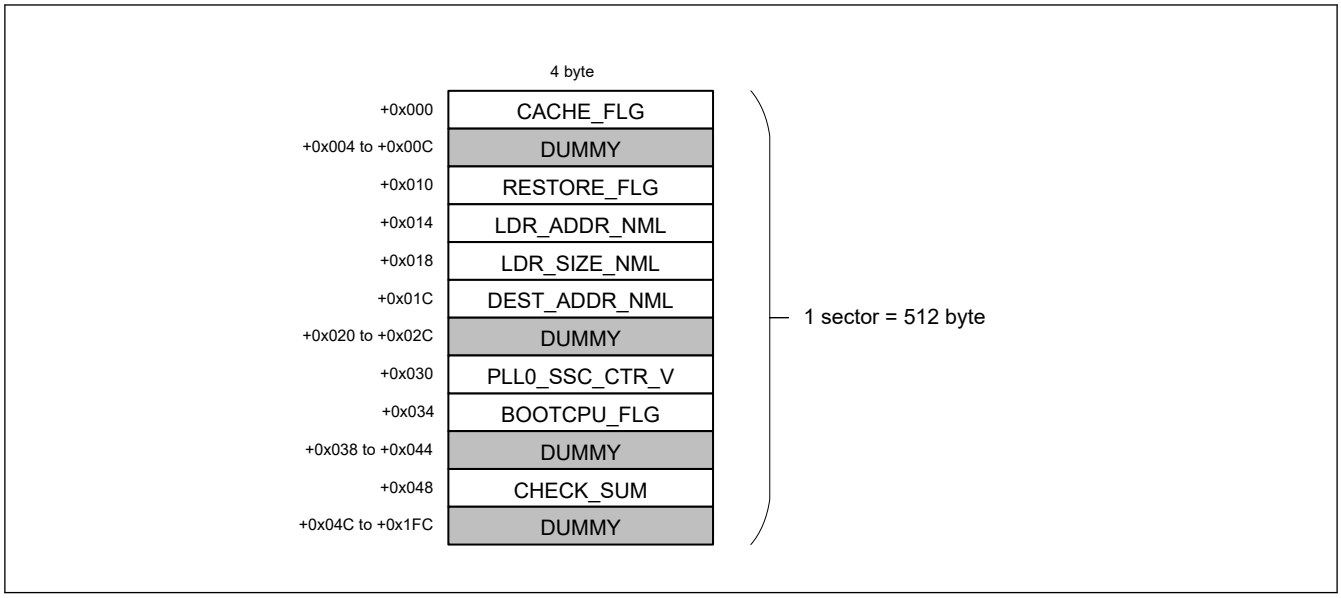


Figure 4.7 Structure of the loader parameter area

4.6.5.2 Operation Settings in eSD Boot Mode

Table 4.12 describes the setting values of the related peripheral modules and registers at the time when eSD boot mode finishes.

Table 4.12 Setting register values of the related peripheral modules when eSD boot mode finishes (1 of 2)

Peripheral module	Register	When SDHI is initialized (RESTORE_FLG = 0x22360679)	When SDHI is not initialized (RESTORE_FLG ≠ 0x22360679)
SDHI1	SD_CMD	0x00000000	Undefined
	SD_ARG0	0x00000000	Undefined
	SD_ARG1	0x00000000	Undefined
	SD_STOP	0x00000000	Undefined
	SD_SECCNT	0x00000000	Undefined
	SD_RSP10, SD_RSP1, SD_RSP32, SD_RSP3, SD_RSP54, SD_RSP5, SD_RSP76, SD_RSP7	0x00000000	Undefined
	SD_INFO1	0x00000400	Undefined
	SD_CLK_CTRL	0x00000020	0x00000102
	SD_OPTION	0x000040EE	0x000040AE
	SDIO_INFO1	0x00000000	Undefined
	HOST_MODE	0x00000000	0x00000101
Clock	SCKCR	0x00730606 (initial value)	0x00730606 (initial value)
	PMSEL*2	0x00002601	0x00002601
	PLL0MON*2	0x00000001	0x00000001
	PLL0EN*2	0x00000001	0x00000001
	PLL0_SSC_CTR*2	Setting value of PLL0_SSC_CTR_V	Setting value of PLL0_SSC_CTR_V
	PLL2MON	0x00000001	0x00000001
	PLL2EN	0x00000001	0x00000001
Reset	SWR55C*2	0x00000000	0x00000000
	SWR550*2	0x00000000	0x00000000

Table 4.12 Setting register values of the related peripheral modules when eSD boot mode finishes (2 of 2)

Peripheral module	Register	When SDHI is initialized (RESTORE_FLG = 0x22360679)	When SDHI is not initialized (RESTORE_FLG ≠ 0x22360679)
Low power consumption	MSTPCRM	0x00001111	0x00001111
	MSTPCRN*2	0x0000003A	0x0000003A
I/O ports*1	PMCm	0xE0 (m = 16), 0x07 (m = 17)	0xE0 (m = 16), 0x07 (m = 17)
	PFCm	0x29292900_00000000 (m = 16), 0x00000000_00292929 (m = 17)	0x29292900_00000000 (m = 16), 0x00000000_00292929 (m = 17)
	DRCTLm	0x21212200_00000000 (m = 16), 0x00000000_00212121 (m = 17)	0x21212200_00000000 (m = 16), 0x00000000_00212121 (m = 17)
TZC-400	GATE_KEEPER	0x000F000F (TZC-400-5)	0x000F000F (TZC-400-5)
	SPECULATION_CTRL	0x00000003 (TZC-400-5)	0x00000003 (TZC-400-5)
	REGION_ATTRIBUTES_0	0xC000000F (TZC-400-5)	0xC000000F (TZC-400-5)
	REGION_ID_ACCESS_0	0x000F000F (TZC-400-5)	0x000F000F (TZC-400-5)
Cortex-A55	RVBAL0*2	0x101FD000	0x101FD000

Note 1. For details, see section 17.7.6.2. Pin Assignment for eSD/eMMC Boot Mode
Note 2. In case the second boot CPU is Cortex-A55 Core0.

4.6.6 eMMC Boot Mode

In eMMC boot mode, this LSI boots a program from an external eMMC device with 8-bit data bus. The operating voltage is fixed to 1.8 V or 3.3 V and MDV pin must be set to Low in case of 1.8 V and High in case of 3.3 V. This LSI supports the eMMC that operates in the boot operation mode prescribed in JEDEC STANDARD JESD84 A44 (MMCA 4.4). In this mode, the SD0_CLK, SD0_CMD, SD0_DATA[7:0], and SD0_RST# pins are enabled.

After the reset is released, the LSI executes the boot processing on Cortex-R52 CPU0. Boot process starts alternative boot operation mode of the eMMC device, and read access begins from sector 0 of the partition selected by the [179] field (PARTITION_CONFIG) of the extended CSD (EXT_CSD) register in the eMMC device.

After dummy read from sector 0, the loader parameter is read from sector 1. According to the loader parameter, loader program stored in the eMMC device is extracted to the internal tightly coupled memory (TCM) or System SRAM (SYSRAM) dependent on the selected second boot CPU, and then the processing is executed. In this mode, PLL2 is automatically activated with disabled SSC.

Figure 4.8 shows the connection diagram of this LSI with eMMC device.

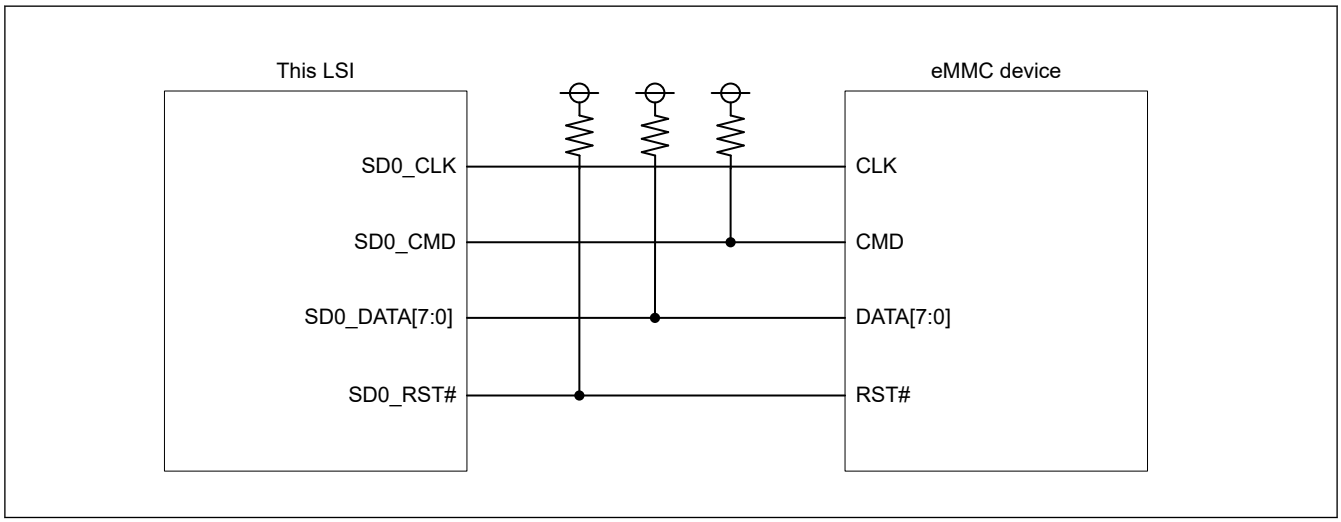


Figure 4.8 Connection diagram of this LSI with eMMC device

4.6.6.1 Allocation of the Loader Parameter and Loader Program

Figure 4.9 shows the allocation of the loader program in the eMMC device conforming to the MMCA 4.4 standard.

The loader parameters are allocated to sectors 1 in boot partition 1 or 2 of the eMMC device. A dummy read from sector 0 is done, and then loader parameter is read from sector 1.

The loader program is allocated to the address specified in the loader parameter (sector 2 or after).

The structure of the loader parameter in a sector for eMMC boot mode is same for eSD boot mode shown in Figure 4.7. First 76 bytes defined in Table 4.9 are valid and the rest is not used.

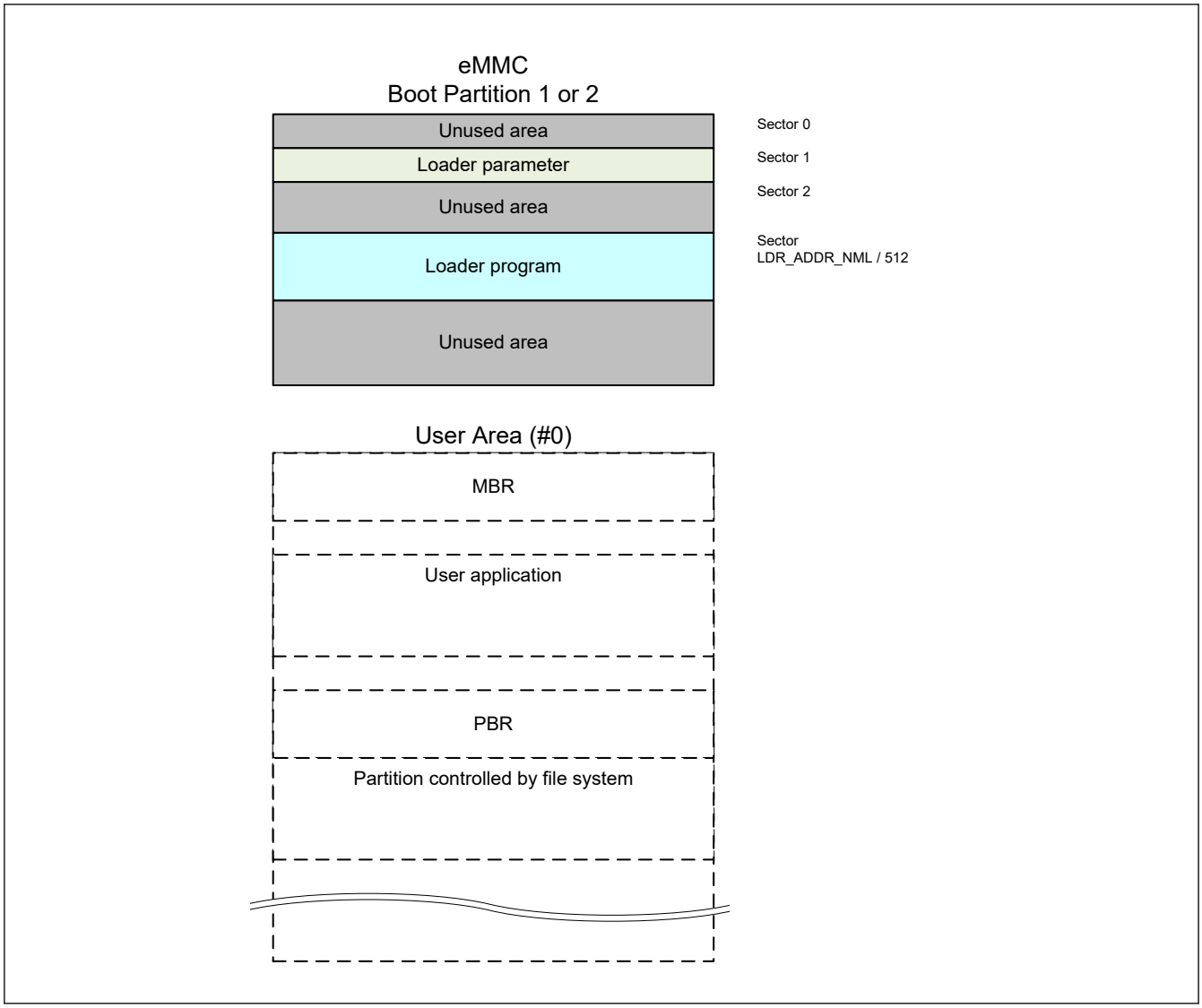


Figure 4.9 Allocation of loader parameter and loader program in the eMMC Device

In the eMMC device, the loader parameter and loader program are read from the partition specified by the BOOT_PARTITION_ENABLE bits in the PARTITION_CONFIG (ECSD[179]) field of the extended CSD register. Write the loader parameter and loader program to the partition selected by this register.

Boot ROM uses the alternative boot operation and sets up the registers in SDHI0 so that the SDHI operates with an 8-bit bus width when booting from the eMMC device. On the eMMC device side, the data bus width in the alternative boot operation is determined by the setting of the BOOT_BUS_WIDTH[177] field in the extended CSD register. To boot up from the eMMC device correctly, set up the extended CSD register in the eMMC device when writing a program to the eMMC device.

Boot Ack is not detected in this LSI. The output of Boot Ack from the eMMC device can be specified in the PARTITION_CONFIG[179] field. Be sure to stop the output of Boot Ack by using this field.

Table 4.13 shows the setting of the extended CSD register.

Table 4.13 Setting of the Extended CSD Register

No.	Field name	Bit name	Setting
1	PARTITION_CONFIG (ECSD[179])	BOOT_PARTITION_ENABLE bits [5:3]	0x0: Device not boot enabled (default) 0x1: Boot partition 1 enabled for boot 0x2: Boot partition 2 enabled for boot 0x7: User area enabled for boot
		BOOT_ACK bit 6	Set to 0 (The eMMC device does not output a boot acknowledge in the alternative boot operation.)
2	BOOT_BUS_WIDTH (ECSD[177])	BOOT_BUS_WIDTH bits [1:0]	Set to 10b (Bus width is set to 8-bit in alternative boot operation.)

It is recommended to allocate partitions as follows.

- Store the loader parameter and loader program in boot partition 1.
- Store the application program in boot partition 2.
- Allocate the user area to a partition controlled by the file system.

4.6.6.2 Operation Settings in eMMC Boot Mode

Table 4.14 describes the setting values of the related peripheral modules and registers at the time when eMMC boot mode finishes.

Table 4.14 Setting register values of the related peripheral modules when eMMC boot mode finishes (1 of 2)

Peripheral module	Register	When SDHI is initialized (RESTORE_FLG = 0x22360679)	When SDHI is not initialized (RESTORE_FLG ≠ 0x22360679)
SDHI0	SD_SECCNT	0x00000000	0xFFFFFFFF
	SD_INFO1	0x00000400	0x00000701
	SD_CLK_CTRL	0x00000020	0x00000102
	HOST_MODE	0x00000000	0x00000101
	SCC_TMPPORT	(3.3V) 0x00000010 (1.8V) 0x00000001	(3.3V) 0x00000010 (1.8V) 0x00000001
Clock	SCKCR	0x00730606 (initial value)	0x00730606 (initial value)
	PMSEL*2	0x00002601	0x00002601
	PLL0MON*2	0x00000001	0x00000001
	PLL0EN*2	0x00000001	0x00000001
	PLL0_SSC_CTR*2	Setting value of PLL0_SSC_CTR_V	Setting value of PLL0_SSC_CTR_V
	PLL2MON	0x00000001	0x00000001
	PLL2EN	0x00000001	0x00000001
Reset	SWR55C*2	0x00000000	0x00000000
	SWR550*2	0x00000000	0x00000000
Low power consumption	MSTPCRM	0x00002111	0x00002111
	MSTPCRN*2	0x0000003A	0x0000003A
I/O ports*1	PMCm	0xFF (m = 12), 0x07 (m = 13)	0xFF (m = 12), 0x07 (m = 13)
	PFCm	0x29292929_29292929 (m = 12), 0x00000000_00292929 (m = 13)	0x29292929_29292929 (m = 12), 0x00000000_00292929 (m = 13)
	DRCTLm	(3.3V) 0x22222222_22222222 (m = 12), 0x00000000_00222222 (m = 13) (1.8V) 0x22222222_22222223 (m = 12), 0x00000000_00222222 (m = 13)	(3.3V) 0x22222222_22222222 (m = 12), 0x00000000_00222222 (m = 13) (1.8V) 0x22222222_22222223 (m = 12), 0x00000000_00222222 (m = 13)

Table 4.14 Setting register values of the related peripheral modules when eMMC boot mode finishes (2 of 2)

Peripheral module	Register	When SDHI is initialized (RESTORE_FLG = 0x22360679)	When SDHI is not initialized (RESTORE_FLG ≠ 0x22360679)
TZC-400	GATE_KEEPER	0x000F000F (TZC-400-5)	0x000F000F (TZC-400-5)
	SPECULATION_CTRL	0x00000003 (TZC-400-5)	0x00000003 (TZC-400-5)
	REGION_ATTRIBUTES_0	0xC000000F (TZC-400-5)	0xC000000F (TZC-400-5)
	REGION_ID_ACCESS_0	0x000F000F (TZC-400-5)	0x000F000F (TZC-400-5)
Cortex-A55	RVBAL0*2	0x101FD000	0x101FD000

Note 1. For details, see section 17.7.6.2. Pin Assignment for eSD/eMMC Boot Mode
Note 2. In case the second boot CPU is Cortex-A55 Core0.

4.6.7 SCI (UART) Boot Mode

The SCI (UART) Boot mode supports download and executes the loader program using SCI0 (UART) interface, which can be directly executed in internal memory. When the LSI transmits the start message to the PC by serial communication, the PC starts to transmit the loader program and it is transferred to the internal tightly coupled memory (BTCM) or System SRAM (SYSRAM). After completion of the transmission, the processing of the loader program is executed.

Figure 4.10 shows the connection diagram of this LSI with a PC.

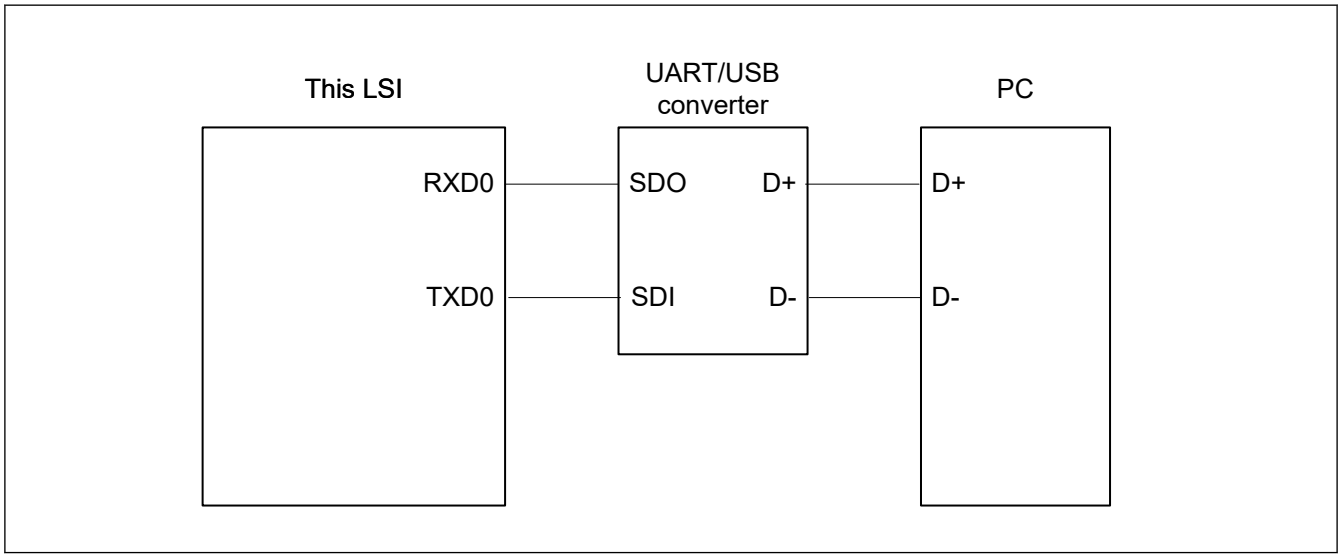


Figure 4.10 Connection diagram of this LSI with a PC

4.6.7.1 Operation Setting in SCI Boot Mode

After the reset is released, if the boot processing starts in SCI boot mode, the following setting values are used:

- CPU clock (CPU0CLK): 500 MHz
- SCI mode: Asynchronous mode
- Start bit: 1 bit
- Parity bit: None
- Data: 8 bits
- Stop bit: 1 bit
- Baud rate: 115200 bps

4.6.7.2 Boot Procedure

This boot procedure assumes using terminal emulator.

Table 4.15 SCI boot procedure

Steps	What to do
1	PC will receive "SCI Download mode (Normal SCI boot)" message after power-on-reset is released and LSI initialization is done on SCI boot mode.
2	The LSI waits to receive "HDR NM" data of Motorola S-record format, S0 record type, from the PC.
3	After received the data, the LSI will send out the following message to the PC. "--- Load Program to RAM ----------"
4	The LSI can receive a loader program by Motorola S-record format, S3 record type, from the PC. The received data will be changed to binary data and forwarded to internal memory.
5	When SCI boot program receives destination address by Motorola S-record format, S7 record type, that is a notification from the PC to terminate the transfer.
6	The LSI will send out the following message to the PC. "--- Start Boot Program on RAM ----------"
7	An end process is executed and the loader program is transferred to BTCM or SYSRAM.

Note: See section 4.6.9. Motorola S-record Format for SCI/USB Boot about S-record format.

When an error happens, an error code is returned and the boot program is aborted. The error condition is below.

- SCI communication error
- Receive Motorola S-record format except S0, S3, and S7 record
- Check sum error of S-record format
- Addressing: Outside of BTCM or SYSRAM area
- Loader program size: Outside of BTCM or SYSRAM capacity

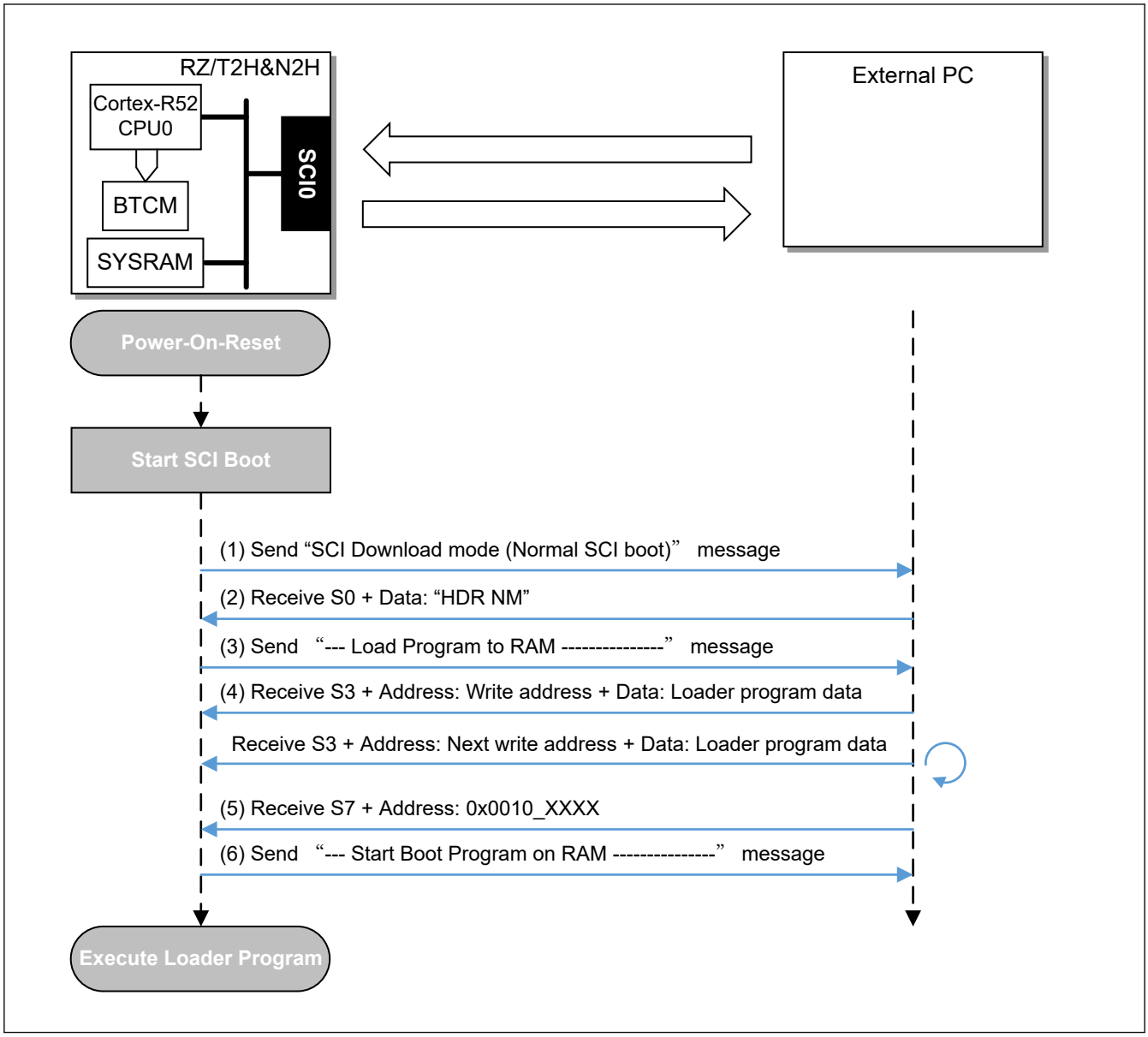


Figure 4.11 SCI boot sequence

4.6.8 USB Boot Mode

The USB Boot mode supports download and executes the loader program using the USB 2.0 interface, which can be directly executed in internal memory. When the LSI receives the Open request from PC, the LSI transmits the start message to the PC. The PC starts to transmit the loader program and it is transferred to the internal tightly coupled memory (BTCM) or System SRAM (SYSRAM). After completion of the transmission, the processing of the loader program is executed.

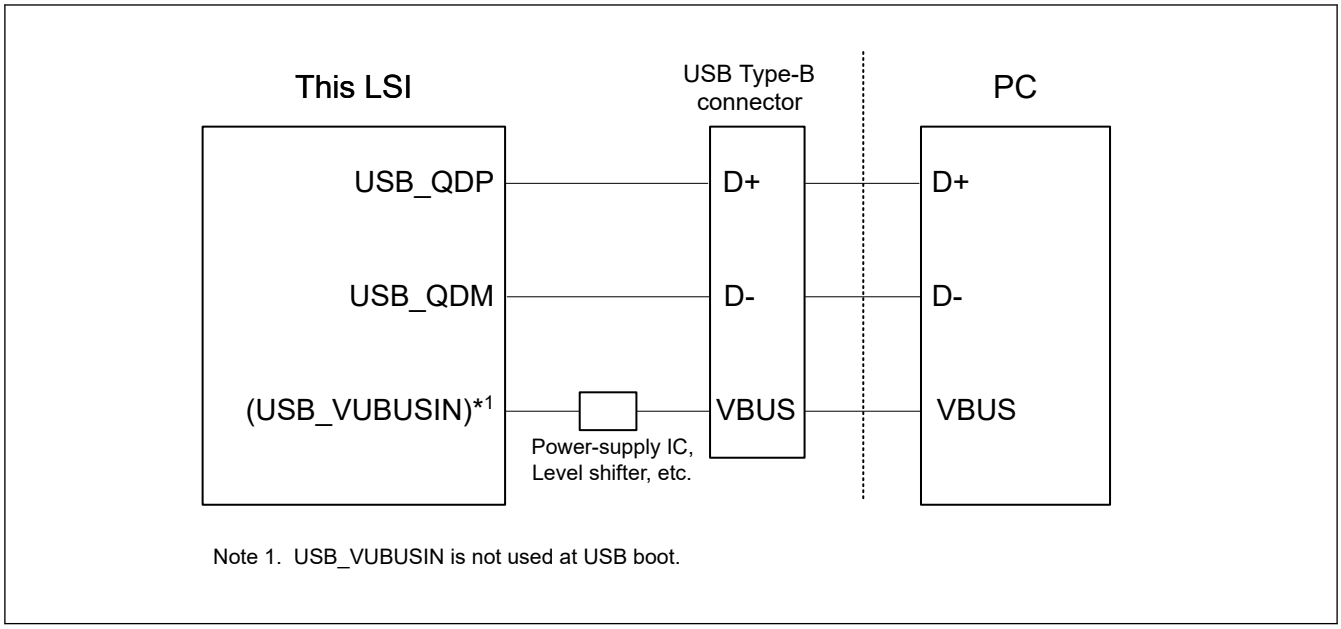


Figure 4.12 Connection diagram of this LSI with a PC (USB)

4.6.8.1 Operation Setting in USB Boot Mode

After the reset is released, if the boot processing starts in USB boot mode, the following setting values are used:

- CPU clock (CPU0CLK): 500 MHz
- USB mode: High speed, function mode
- USB function: USB Communication Device Class (CDC)

Note: After USB cable is connected with the LSI, USB cable should not be disconnected until the loader program is executed.

4.6.8.2 Boot Procedure

This boot procedure assumes using terminal emulator.

Table 4.16 USB boot procedure

Steps	What to do
1	After power-on-reset is released, LSI initialization and USB enumeration are performed on USB boot mode. Then, the LSI waits to receive "OPEN" data of Motorola S-record format, S0 record type, from the PC.
2	PC will receive "USB Download mode (Normal USB boot)" message.
3	The LSI waits to receive "HDR NM" data of Motorola S-record format, S0 record type, from the PC.
4	After received the data, the LSI will send out the following message to the PC. "--- Load Program to RAM ----------"
5	The LSI can receive a loader program by Motorola S-record format, S3 record type, from the PC. The received data will be changed to binary data and forwarded to internal memory.
6	When USB boot program receives destination address by Motorola S-record format, S7 record type, that is a notification from the PC to terminate the transfer.
7	The LSI will send out the following message to the PC. "--- Start Boot Program on RAM ----------"
8	An end process is executed and the loader program is transferred to BTCM or SYSRAM.

Note: See section 4.6.9. Motorola S-record Format for SCI/USB Boot about S-record format.

When an error happens, an error code is returned and the boot program is aborted. The error condition is below.

- USB communication error
- Receive Motorola S-record format except S0, S3, and S7 record

- Check sum error of S-record format
- Addressing: Outside of BTCM or SYSRAM area
- Loader program size: Outside of BTCM or SYSRAM capacity

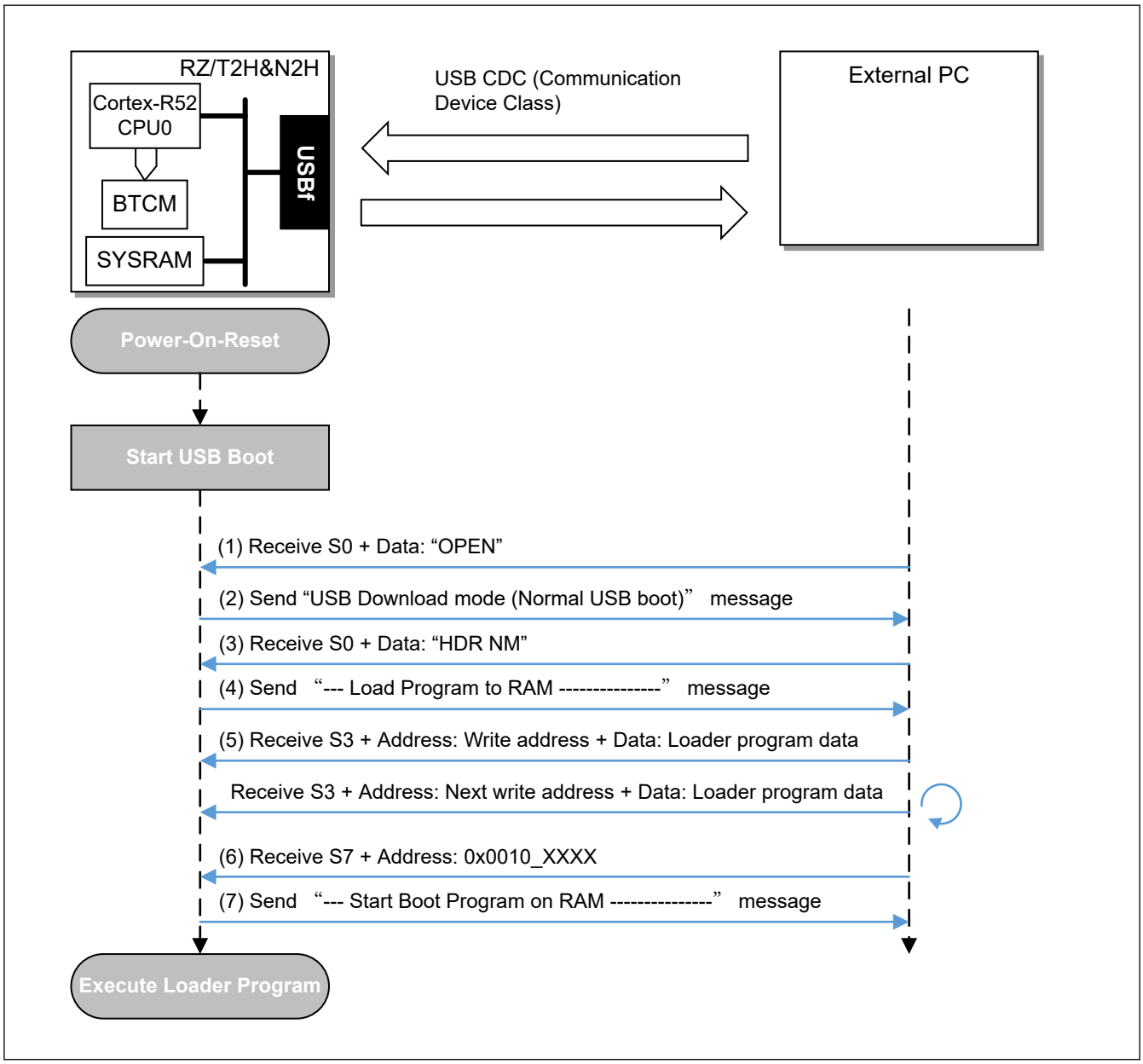


Figure 4.13 USB boot sequence

4.6.9 Motorola S-record Format for SCI/USB Boot

SCI/USB boot uses S0, S3, and S7 record type only. The record field is shown in Figure 4.14.

S-record Format

S-Type	Byte Count	Address	Data	Check sum
S0	2 byte	2 byte	2 byte × n (n = 1, 2, ⋯)	2 byte

S-Type	Byte Count	Address	Data	Check sum
S3	2 byte	4 byte	2 byte × n (n = 1, 2, ⋯)	2 byte

S-Type	Byte Count	Address	Check sum
S7	2 byte	4 byte	2 byte

Figure 4.14 S-record format for SCI/USB boot

4.6.10 SCI/USB Authentication

SCI/USB authentication mechanism checks access permission for SCI (UART) boot mode and USB boot mode.

In order to enable connection, a pass result of the authentication mechanism is required if authentication is required.

Authentication level is selectable from 3 states by setting SCI/USB Boot Authentication Mode area (AUTHMODES) in OTP:

- No authentication
- Authentication
- Permanent prohibition

When authentication is selected, 128-bit authentication ID must be written to SCI/USB Boot Authentication Plaintext ID area (SIDP) in OTP in advance.

If authentication is selected, this LSI requests to input 128-bit authentication ID from SCI/USB at the beginning of the boot. If authentication fails, then no connection is permitted and boot fails. RES# pin reset is required to retry boot.

If permanent prohibition is selected, this LSI prohibits connection by SCI/USB boot mode permanently and boot fails.

Note: In non-security support product, SCI/USB authentication is performed by plaintext comparison. It is the simple method but also known that it is comparatively weak from the sniffing point of view. It is recommended to use plaintext authentication in the limited area only where the risk of sniffing is low. In general, it is strongly recommended to use security support product which provides more secure authentication.

4.6.11 SCI Authentication Boot Procedure with Plaintext ID

This boot procedure assumes using terminal emulator. This authentication is supported for non-security product.

Table 4.17 SCI authentication boot procedure with plaintext ID (1 of 2)

Steps	What to do
1	PC will receive "SCI Download mode (SCI authentication boot (plaintext))" message after power-on-reset is released and LSI initialization is done on SCI boot mode.
2	The LSI waits to receive "HDR PL" data of Motorola S-record format, S0 record type, from the PC.

Table 4.17 SCI authentication boot procedure with plaintext ID (2 of 2)

Steps	What to do
3	The LSI waits to receive 16-byte password, authentication ID, filled in data area of Motorola S-record format, S0 record type, from the PC. Until SCI boot program receive this record type, the LSI discards other record types.
4	The SCI boot program forwards the password to a register and verifies password against authentication ID which is set into OTP. If the password check passed, SCI boot program starts to receive a loader program. If failed, an error code is returned and the boot program is aborted.
5	The LSI will send out the following message to the PC. "--- Load Program to RAM ----------"
6	The LSI can receive a loader program by Motorola S-record format, S3 record type, from the PC, and the received data will be changed to binary data, forward to internal memory.
7	When SCI boot program receives Motorola S-record format, S7 record type, that notice to terminate from the PC.
8	The LSI will send out the following message to the PC. "--- Start Boot Program on RAM ----------"
9	Executes an end process and the loader program transferred into BTCM.

When an error happens, an error code is returned and the boot program is aborted. The error condition is below.

- SCI communication error
- Receive Motorola S-record format except S0, S3 and S7 record
- Check sum error of S-record format
- Addressing: Outside of BTCM or SYSRAM area
- Loader program size: Outside of BTCM or SYSRAM capacity
- Authentication is failed

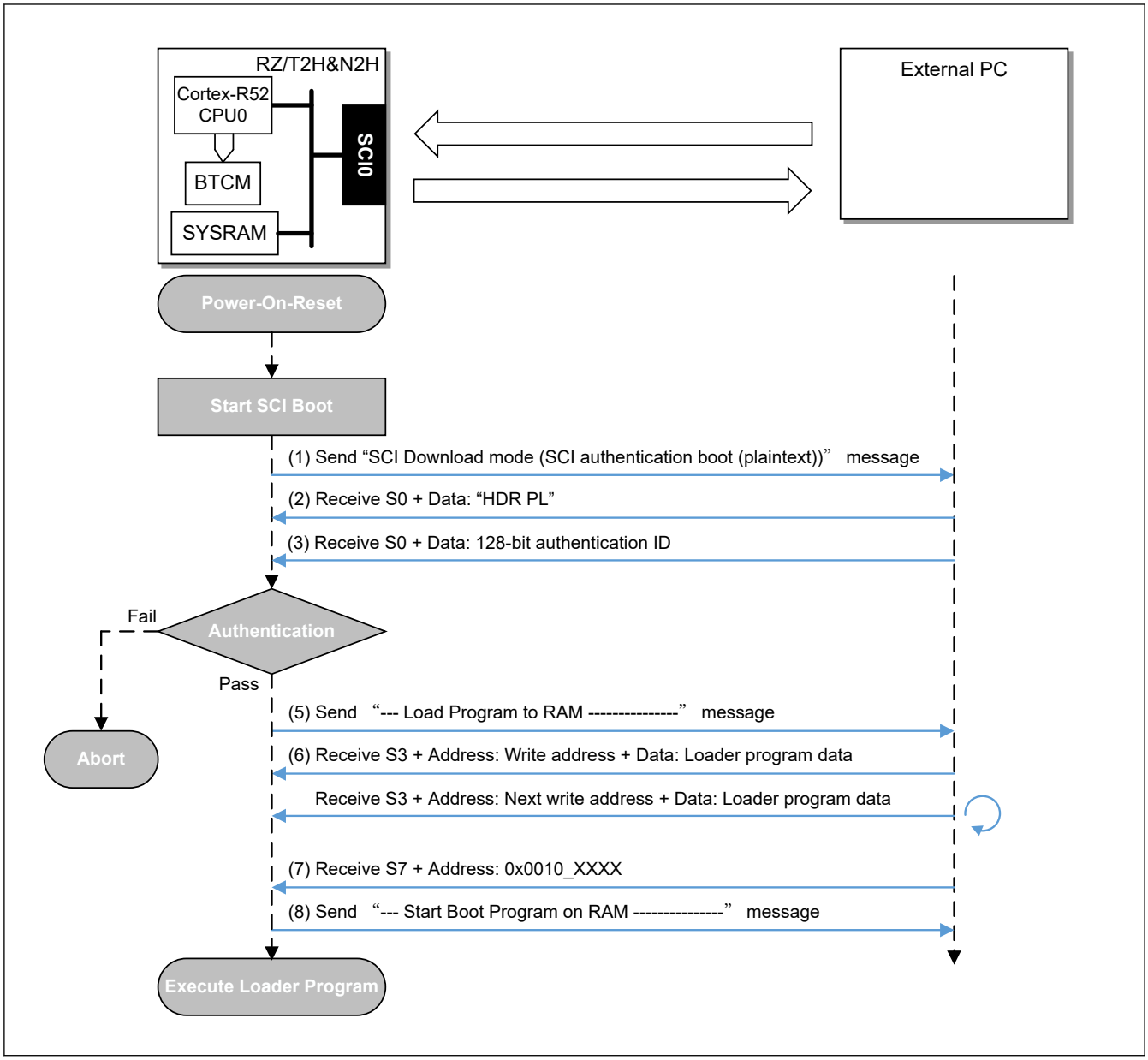


Figure 4.15 SCI authentication boot (plaintext) sequence

4.6.12 USB Authentication Boot Procedure with Plaintext ID

This boot procedure assumes using terminal emulator. This authentication is supported for non-security product.

Table 4.18 USB authentication boot procedure with plaintext ID (1 of 2)

Steps	What to do
1	After power-on-reset is released, LSI initialization and USB enumeration are performed on USB boot mode. Then, the LSI waits to receive "OPEN" data of Motorola S-record format, S0 record type, from the PC.
2	PC will receive "USB Download mode (USB authentication boot (plaintext))" message.
3	The LSI waits to receive "HDR PL" data of Motorola S-record format, S0 record type, from the PC.
4	The LSI waits to receive 16-byte password, authentication ID, filled in data area of Motorola S-record format, S0 record type, from the PC. Until USB boot program receive this record type, the LSI discards other record types.

Table 4.18 USB authentication boot procedure with plaintext ID (2 of 2)

Steps	What to do
5	The USB boot program forwards the password to a register and verifies password against authentication ID which is set into OTP. If the password check passed, USB boot program starts to receive a loader program. If failed, an error code is returned and the boot program is aborted.
6	The LSI will send out the following message to the PC. "--- Load Program to RAM ----------"
7	The LSI can receive a loader program by Motorola S-record format, S3 record type, from the PC, and the received data will be changed to binary data, forward to internal memory.
8	When USB boot program receives Motorola S-record format, S7 record type, that notice to terminate from the PC.
9	The LSI will send out the following message to the PC. "--- Start Boot Program on RAM ----------"
10	Executes an end process and the loader program transferred into BTCM.

When an error happens, an error code is returned and the boot program is aborted. The error condition is below.

- USB communication error
- Receive Motorola S-record format except S0, S3 and S7 record
- Check sum error of S-record format
- Addressing: Outside of BTCM or SYSRAM area
- Loader program size: Outside of BTCM or SYSRAM capacity
- Authentication is failed

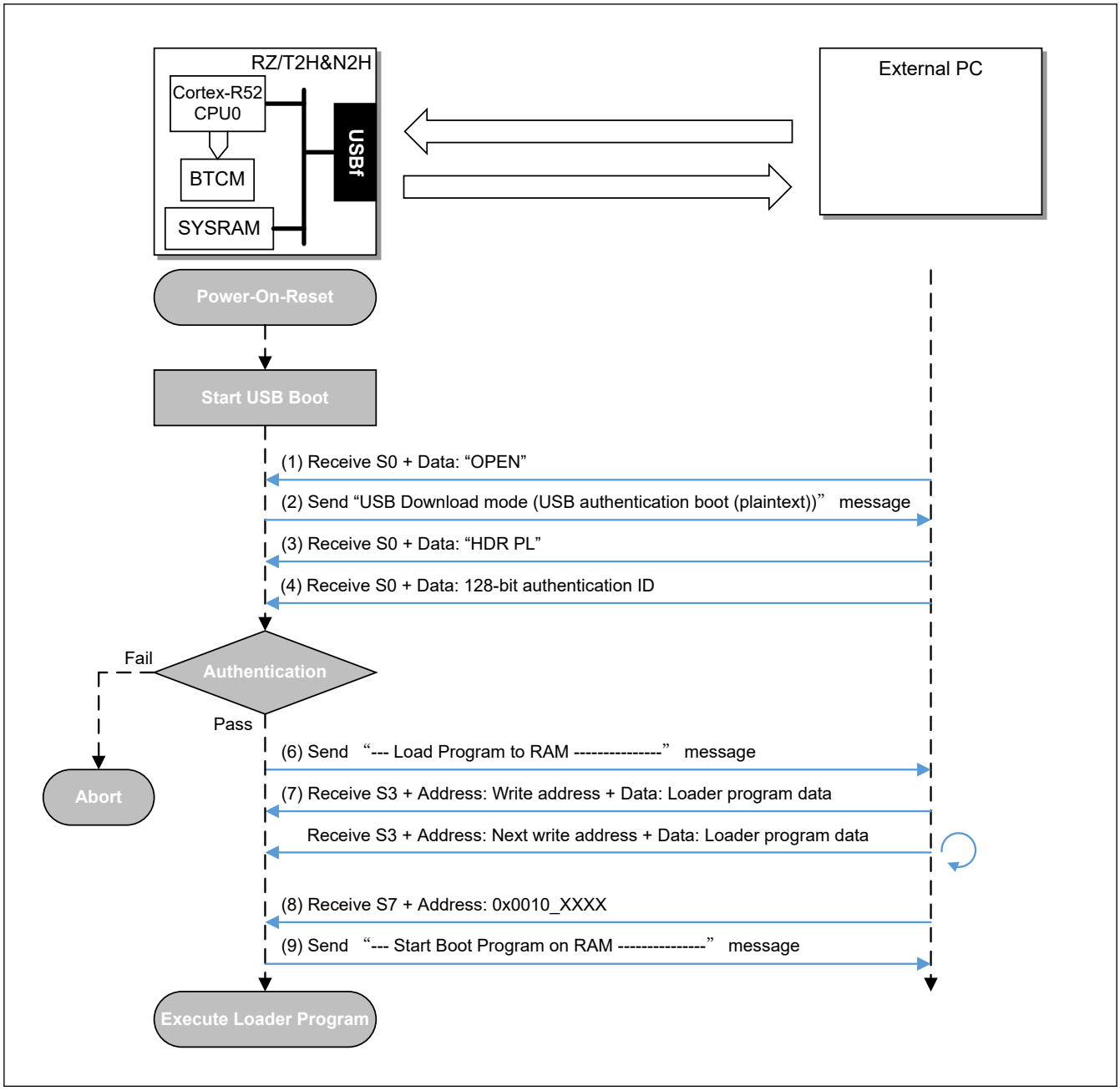


Figure 4.16 USB authentication boot (plaintext) sequence

4.6.13 Usage Note

4.6.13.1 Separated I/O domain

The LSI supports 1.8 V or 3.3 V operation voltage of separated I/O domains for Ethernet interface, xSPI, and SDHI.

Table 4.19 shows the list of separated I/O domain. Since voltage tolerance of I/O buffer is automatically controlled by detecting the applied voltage (1.8 V or 3.3 V) to VDD1833_n, indication of applied voltage by external pin is not needed. However, applied voltage of the I/O domain used as boot peripheral must be indicated by MDV pin so that boot ROM can optimize drivability of I/O buffer according to the information.

Table 4.19 List of separated I/O domain

No. of I/O domain	Power supply pin	Feature	Note
0	VDD1833_0	Ethernet port 0 (ETH0_xxx)	Applied voltage depends on type of interface each port. • RGMII: 1.8 V only • MII/RMII: 3.3 V only
1	VDD1833_1	Ethernet port 1 (ETH1_xxx)	
2	VDD1833_2	Ethernet port 2 (ETH2_xxx)	
3	VDD1833_3	Ethernet port 3 (ETH3_xxx)	
4	VDD1833_4	xSPI unit 0 (XSPI0_xxx)	Applied voltage is selectable from 1.8 V or 3.3 V at power-on each unit.
5	VDD1833_5	xSPI unit 1 (XSPI1_xxx)	
6	VDD1833_6	SDHI unit 0 (SD0_xxx[*1]) SD or eMMC interface support	Applied voltage is selectable from 1.8 V or 3.3 V at power-on in case of eMMC interface. Change of applied voltage is supported according to the operating mode in case of SD interface.
7	VDD1833_7	SDHI unit 1 (SD1_xxx[*1]) SD interface support	Change of applied voltage is supported according to the operating mode.

Note 1. Except SDn_CD, SDn_WP, SDn_PWEN, SDn_IOVS (n = 0, 1). I/O voltage of these pins are fixed to 3.3 V.

4.6.13.2 Reset after eSD Boot

The eSD boot assumes start from 3.3 V operation. When system is reset after changing operating IO voltage from 3.3 V to 1.8 V, perform power cycle (1.8V → Off → 3.3 V) at the reset to initialize the SD device.

5. Address Space

5.1 Address Space

This LSI has a 32-GB address space ranging from 0x0_0000_0000 to 0x7_FFFF_FFFF (35-bit address) for Cortex-A55, PCIE, and CoreSight, and a 4-GB address space ranging from 0x0000_0000 to 0xFFFF_FFFF (32-bit address) for Cortex-R52 and other masters. Figure 5.1, Figure 5.2, and Figure 5.3 show the memory maps of the bus masters. Accessible areas will differ depending on the operating mode and the states of control bits.

In over 4-GB address space, PCIE and DDR areas are available. 32-bit address master can access to these areas via mirror area. Mirror area can be set from any 256 MB for PCIE area and 512 MB for DDR area each bus master. 64-bit address master can access to these areas via over 4-GB address space. In addition, except CoreSight ETR, 64-bit address master can also access to these areas via mirror area in 32-bit address space. For details, see section 13.4.5. Address Expander. Note that upper 3-bit may be omitted for the register address up to 4-GB in this manual.

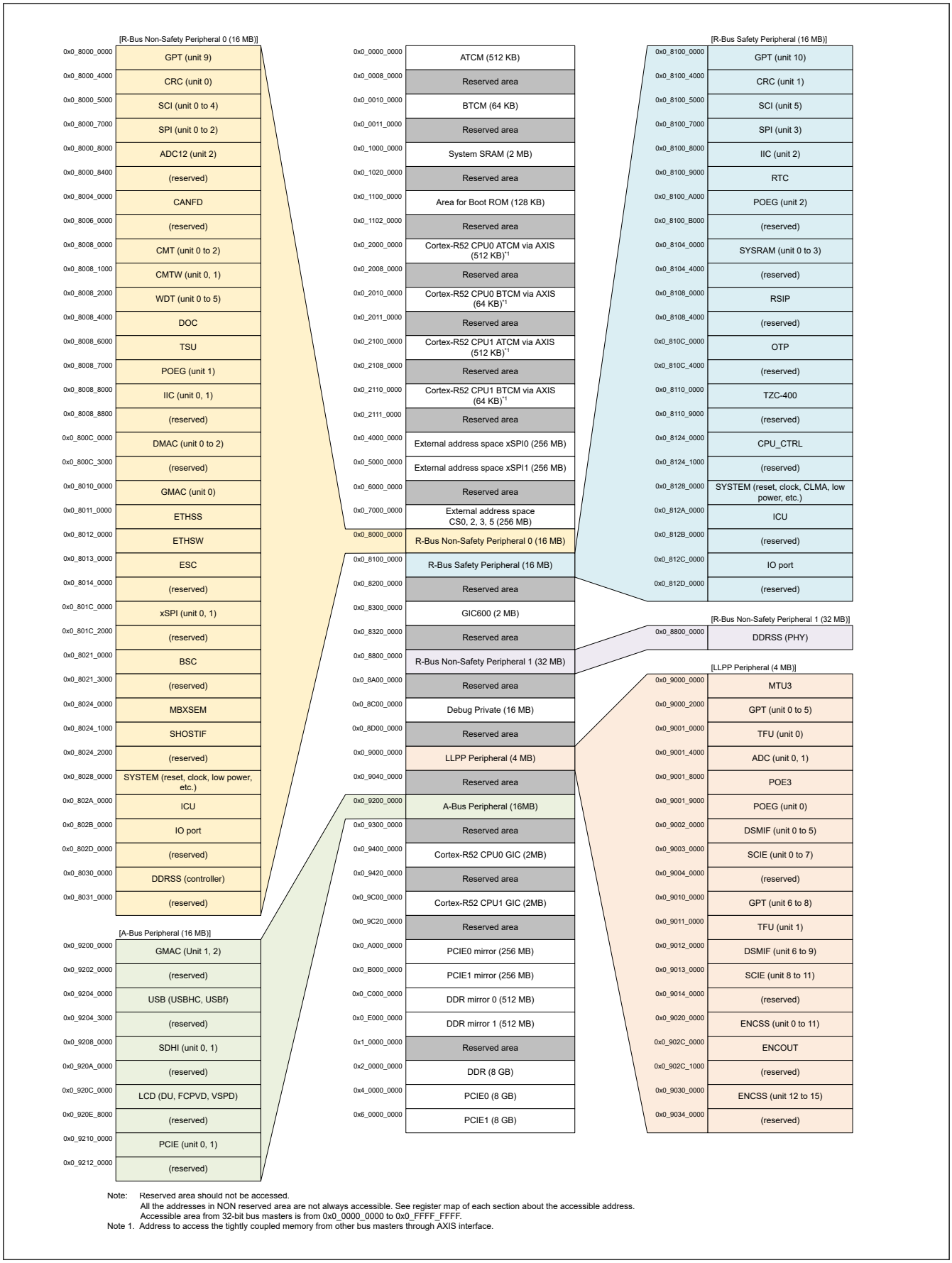
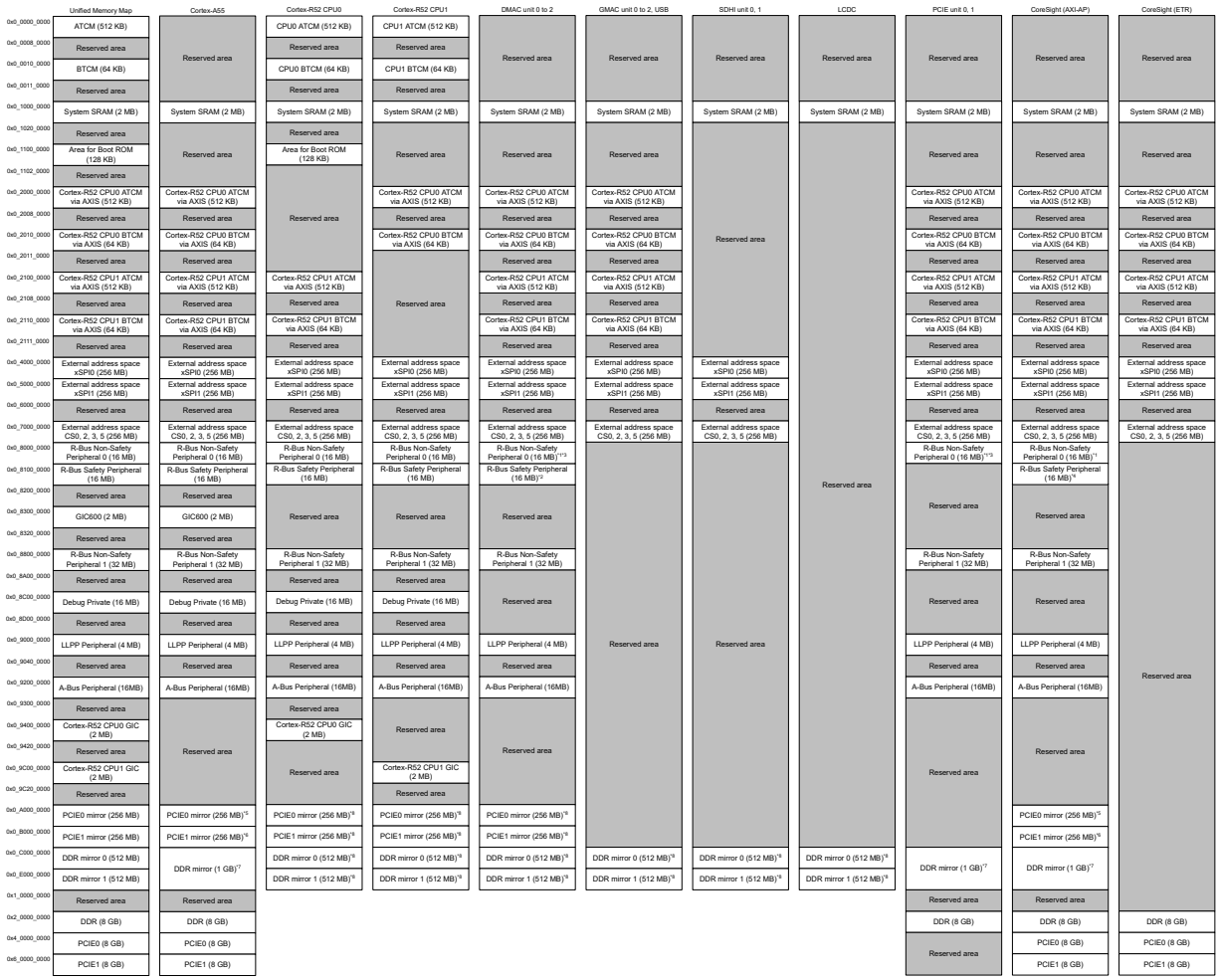


Figure 5.1 Unified memory map



Note 1. A part of IO port (0x802B000 to 0x802BFFFF) is not accessible.
Note 2. RSIP register only.
Note 3. DMAC, xSPI, BSC, MBXSEM, SHOSTIF, SYSTEM, ICU, DDRSS registers are not accessible.
Note 4. SYSTEM and TZC400 registers only.
Note 5. Mirror of 0x4_0000_0000 to 0x4_0FFF_FFFF
Note 6. Mirror of 0x6_0000_0000 to 0x6_0FFF_FFFF
Note 7. Mirror of 0x2_0000_0000 to 0x2_3FFF_FFFF
Note 8. Mirror area is configurable. For details, see section 13.4.5. Address Expander.

Figure 5.2 Memory map each bus master

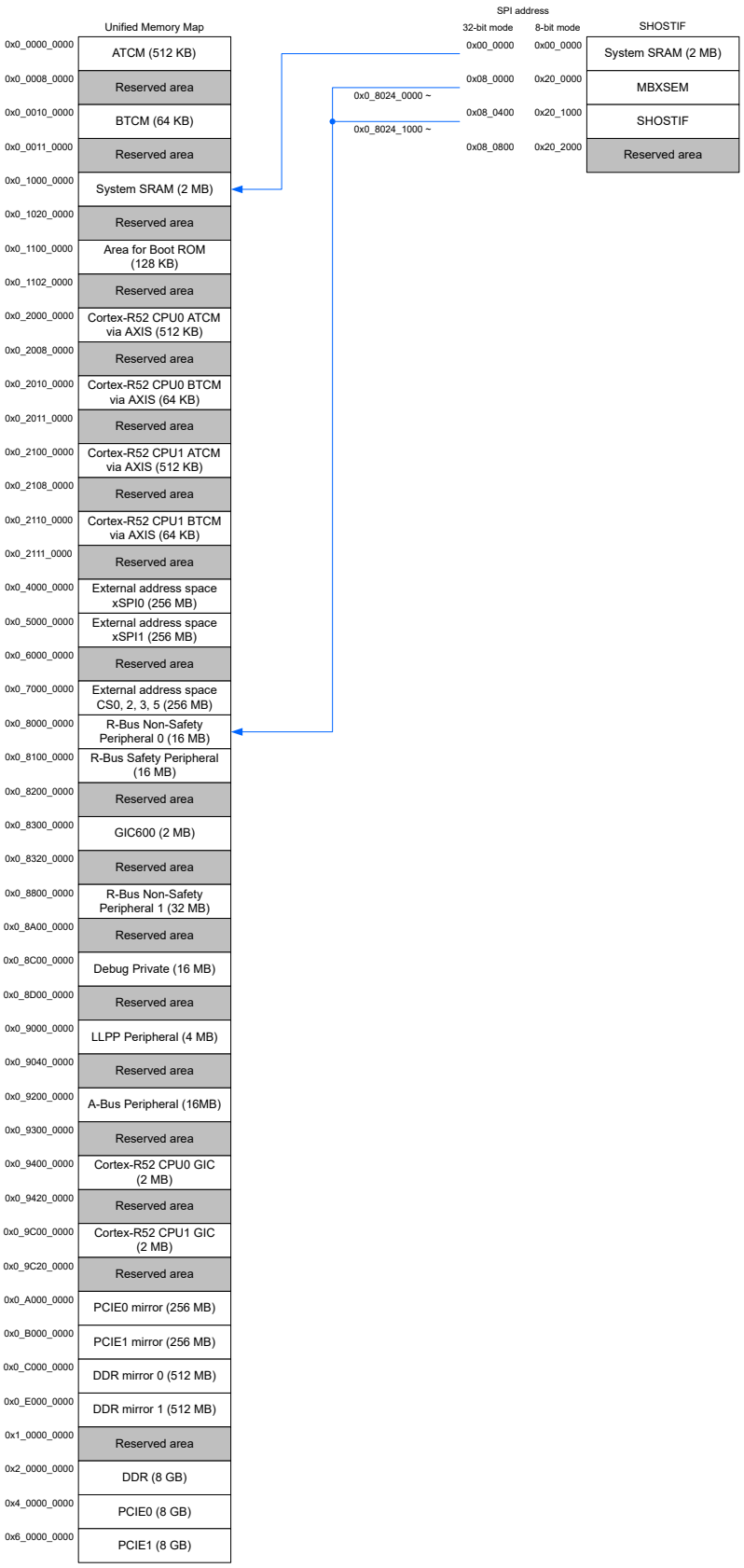


Figure 5.3 Memory map for external host interface

6. Reset

6.1 Overview

Available reset types are RES# pin reset, system software reset, Cortex-A55 Cluster, Core0/1/2/3 software reset, Cortex-R52 CPU0/1 software reset, and error reset. Table 6.1 lists the reset names and sources.

Table 6.1 Reset names and sources

Reset name	Source
RES# pin reset	The low level is applied to the RES# pin
System software reset	SWRSYS register setting
Cortex-A55 Cluster software reset	SWR55C register setting
Cortex-A55 Core0 software reset	SWR550 register setting (release only once after power-on)
Cortex-A55 Core1 software reset	SWR551 register setting (release only once after power-on)
Cortex-A55 Core2 software reset	SWR552 register setting (release only once after power-on)
Cortex-A55 Core3 software reset	SWR553 register setting (release only once after power-on)
Cortex-R52 CPU0 software reset	SWRCPU0 register setting
Cortex-R52 CPU1 software reset	SWRCPU1 register setting
Error reset	Reset request from the ICU

The internal states and pins are initialized by reset.

Table 6.2 lists the reset targets to be initialized for each reset type. For details on reset control during debugging, see section 10.3.3. Reset Configuration and the Method of Connecting with the Emulator.

Table 6.2 Targets to be initialized for each reset type (✓: to be initialized, —: no change) (1 of 2)

Reset target		Reset source										
		RES# pin reset (cold)	RES# pin reset (warm)	System software reset	Cortex-A55 cluster software reset	Cortex-A55 Core0 software reset	Cortex-A55 Core1 software reset	Cortex-A55 Core2 software reset	Cortex-A55 Core3 software reset	Cortex-R52 CPU0 software reset	Cortex-R52 CPU1 software reset	Error reset
RES# pin reset flag (RSTSR0.TRF)		—	—	✓	—	—	—	—	—	—	—	✓
System software reset detect flag (RSTSR0.SWRSF)		✓	✓	—	—	—	—	—	—	—	—	✓
Cortex-A55 software reset detect flag	Cluster (RSTSR0.SWR55C)	✓	✓	✓	—	—	—	—	—	—	—	✓
	Core0 (RSTSR0.SWR550)	✓	✓	✓	—	—	—	—	—	—	—	✓
	Core1 (RSTSR0.SWR551)	✓	✓	✓	—	—	—	—	—	—	—	✓
	Core2 (RSTSR0.SWR552)	✓	✓	✓	—	—	—	—	—	—	—	✓
	Core3 (RSTSR0.SWR553)	✓	✓	✓	—	—	—	—	—	—	—	✓

Table 6.2 Targets to be initialized for each reset type (✓: to be initialized, —: no change) (2 of 2)

Reset target		Reset source										
		RES# pin reset (cold)	RES# pin reset (warm)	System software reset	Cortex-A55 cluster software reset	Cortex-A55 Core0 software reset	Cortex-A55 Core1 software reset	Cortex-A55 Core2 software reset	Cortex-A55 Core3 software reset	Cortex-R52 CPU0 software reset	Cortex-R52 CPU1 software reset	Error reset
Cortex-R52 software reset detect flag	CPU0 (RSTSR0.SWR0F)	✓	✓	✓	—	—	—	—	—	—	—	✓
	CPU1 (RSTSR0.SWR1F)	✓	✓	✓	—	—	—	—	—	—	—	✓
Error reset detect flag (RSTSR0.ERRF) Error event capture registers		✓	✓	✓	—	—	—	—	—	—	—	—
Pin state		✓	✓	✓	—	—	—	—	—	—	—	✓
Operating mode		✓*1	✓*1	—*2	—	—	—	—	—	—	—	—*2
Cortex-A55 and WDT	Cluster / WDTA3-0	✓	✓	✓	✓	—	—	—	—	—	—	✓
	Core0 / WDTA0	✓	✓	✓	✓	✓	—	—	—	—	—	✓
	Core1 / WDTA1	✓	✓	✓	✓	—	✓	—	—	—	—	✓
	Core2 / WDTA2	✓	✓	✓	✓	—	—	✓	—	—	—	✓
	Core3 / WDTA3	✓	✓	✓	✓	—	—	—	✓	—	—	✓
Cortex-R52 and WDT	CPU0 / WDTR0	✓	✓	✓	—	—	—	—	—	✓	—	✓
	CPU1 / WDTR1	✓	✓	✓	—	—	—	—	—	—	✓	✓
GIC600 for Cortex-A55		✓	✓	✓	✓	—	—	—	—	—	—	✓
GIC for Cortex-R52 CPU0/CPU1		✓	✓	✓	—	—	—	—	—	—	—	✓
System SRAM, TCM, cache		✓	✓	✓	—	—	—	—	—	—	—	✓
Registers other than the above, and internal state		✓	✓	✓	—	—	—	—	—	—	—	✓
RSTOUT# pin output		✓ (low)*3	✓ (low)*3	✓ (low)*3	—	—	—	—	—	—	—	✓ (low)*3

Note 1. An operating mode is selected according to the input level of the mode setting pin (MD2, MD1, and MD0) when the pin reset state (RES# and TRST# pins are both low) is released. For details on the operating mode, see section 4.2. Types and Selection of Operating Modes.

Note 2. The operating mode is not initialized in response to this type of reset and the LSI is in the operating mode that was selected following the previous release from the reset state applied by the low level on the RES#. For details, see section 4, Operating Modes.

Note 3. For the low output period, see section 6.4.9. Reset Output Pin (RSTOUT#).

Table 6.3 lists the input and output pins related to the reset.

Table 6.3 Input and output pins related to reset

Pin name	I/O	Function
RES#	Input	Reset pin. Use this pin to reset the entire LSI except the debugging circuit and the TAP (Test Access Port). Since the power-on reset circuit is not incorporated in this LSI, a reset circuit must be implemented outside this LSI. For an example of configuration of the external reset circuit, see section 10.3.3. Reset Configuration and the Method of Connecting with the Emulator.
TRST#	Input	Test reset pin. Use this pin to reset the TAP. If you design a board which enables an emulator, set the TRST# pin to low level during the same period as the RES# pin at the time of power-on. The TRST# pin should also be controllable independently. When unused, this pin must be set to low level or connected to the same signal as the RES# pin. For details, see section 10.3.3. Reset Configuration and the Method of Connecting with the Emulator.
RSTOUT#	Output	Reset output pin. This pin outputs low-level signal on occurrence of a reset. For details, see section 6.4.9. Reset Output Pin (RSTOUT#). This pin can be used for resetting the external device.

Note: For details on resetting the debugging circuit, see section 10, Debugging Interface.

6.2 Register Map

Table 6.4 Reset register map

Address	Register symbol	Resister name	Write protection
0x8028_0200	RSTSR0	Reset Status Register 0	PRCRN.PRC1
0x8128_0210	SWRSYS	System Software Reset Register	PRCRS.PRC1
0x8128_0220	SWRCPU0	Cortex-R52 CPU0 Software Reset Register	PRCRS.PRC1
0x8128_0224	SWRCPU1	Cortex-R52 CPU1 Software Reset Register	PRCRS.PRC1
0x8128_0228	SWR55C	Cortex-A55 Cluster Software Reset Register	PRCRS.PRC1
0x8128_022C	SWR550	Cortex-A55 Core0 Software Reset Register	PRCRS.PRC1
0x8128_0230	SWR551	Cortex-A55 Core1 Software Reset Register	PRCRS.PRC1
0x8128_0234	SWR552	Cortex-A55 Core2 Software Reset Register	PRCRS.PRC1
0x8128_0238	SWR553	Cortex-A55 Core3 Software Reset Register	PRCRS.PRC1
0x8128_023C	SWR55ARC	Cortex-A55 Software Reset Automatic Release Control Register	PRCRS.PRC1
0x8028_0240	MRCTLA	Module Reset Control Register A	PRCRN.PRC1
0x8028_0250	MRCTLE	Module Reset Control Register E	PRCRN.PRC1
0x8128_0260	MRCTLI	Module Reset Control Register I	PRCRN.PRC1
0x8028_0270	MRCTLM	Module Reset Control Register M	PRCRN.PRC1

Table 6.5 Reset related system control register

Target	Module reset control register	Module stop control register	Slave access control register
All the registers	—	—	SL.VACCCTL8.SYSCTRL_SL

6.3 Register Descriptions

The registers related to the clock generation circuit can be write-protected. To write to the registers, specify bit 1 of the Protect Register (PRCRN and PRCRS) to cancel the write protection. For details, see section 11, Register Write Protection Function.

6.3.1 RSTSR0 : Reset Status Register 0

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SWR553	SWR552	SWR551	SWR550	SWR55C	SWR1F	SWR0F	SWRSF	ERRF	TRF	—
Value after reset:	0	0	0	0	0	x*1	x*1	x*1	x*1	x*1	x*1	x*1	x*1	x*1	x*1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	TRF	RES# Pin Reset Detect Flag This bit indicates the detection of RES# pin reset. 0: RES# pin reset not detected 1: RES# pin reset detected	R/W*2
2	ERRF	Error Reset Detect Flag This bit indicates the detection of error reset. 0: Error reset not detected 1: Error reset detected	R/W*2
3	SWRSF	System Software Reset Detect Flag This bit indicates the detection of system software reset. 0: System software reset not detected 1: System software reset detected	R/W*2
4	SWR0F	Cortex-R52 CPU0 Software Reset Detect Flag This bit indicates the detection of Cortex-R52 CPU0 software reset. 0: Cortex-R52 CPU0 software reset not detected 1: Cortex-R52 CPU0 software reset detected	R/W*2
5	SWR1F	Cortex-R52 CPU1 Software Reset Detect Flag This bit indicates the detection of Cortex-R52 CPU1 software reset. 0: Cortex-R52 CPU1 software reset not detected 1: Cortex-R52 CPU1 software reset detected	R/W*2
6	SWR55C	Cortex-A55 Cluster Software Reset Detect Flag This bit indicates the detection of Cortex-A55 Cluster software reset. 0: Cortex-A55 Cluster software reset no detected 1: Cortex-A55 Cluster software reset detected	R/W*2
7	SWR550	Cortex-A55 Core0 Software Reset Detect Flag This bit indicates the detection of Cortex-A55 Core0 software reset. 0: Cortex-A55 Core0 software reset no detected 1: Cortex-A55 Core0 software reset detected	R/W*2
8	SWR551	Cortex-A55 Core1 Software Reset Detect Flag This bit indicates the detection of Cortex-A55 Core1 software reset. 0: Cortex-A55 Core1 software reset no detected 1: Cortex-A55 Core1 software reset detected	R/W*2
9	SWR552	Cortex-A55 Core2 Software Reset Detect Flag This bit indicates the detection of Cortex-A55 Core2 software reset. 0: Cortex-A55 Core2 software reset no detected 1: Cortex-A55 Core2 software reset detected	R/W*2
10	SWR553	Cortex-A55 Core3 Software Reset Detect Flag This bit indicates the detection of Cortex-A55 Core3 software reset. 0: Cortex-A55 Core3 software reset no detected 1: Cortex-A55 Core3 software reset detected	R/W*2
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source. The following table lists the sources for the setting (1) and clearing (0) of each bit.

Bit	Setting conditions (0 to 1)	Clearing conditions (1 to 0)
TRF	When the low level is input to the RES# pin.	• When system software reset or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
ERRF	When an error predefined as a reset source by ICU setting occurs	• When the low level is input to the RES# pin or system software reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
SWRSF	When system software reset occurs	• When the low level is input to the RES# pin or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
SWR0F	When Cortex-R52 CPU0 software reset occurs	• When the low level is input to the RES# pin, system software reset or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
SWR1F	When Cortex-R52 CPU1 software reset occurs	• When the low level is input to the RES# pin, system software reset or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
SWR55C	When Cortex-A55 Cluster software reset occurs	• When the low level is input to the RES# pin, system software reset or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
SWR550	When Cortex-A55 Core0 software reset occurs	• When the low level is input to the RES# pin, system software reset or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
SWR551	When Cortex-A55 Core1 software reset occurs	• When the low level is input to the RES# pin, system software reset or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
SWR552	When Cortex-A55 Core2 software reset occurs	• When the low level is input to the RES# pin, system software reset or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
SWR553	When Cortex-A55 Core3 software reset occurs	• When the low level is input to the RES# pin, system software reset or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read

Note 2. Only 0 can be written to clear the flag.

The RSTSR0 register specifies reset generation sources.

6.3.2 SWRSYS : System Software Reset Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x210

Bit position:	31																												0
Bit field:	SWR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SWR[31:0]	System Software Reset 1. When 0x4321A501 is written, a system software reset occurs. After about 93 µs (Typ.) from the reset occurrence, it automatically releases the reset, starts the system in the selected boot mode in the operating mode pins, and the program is restarted. 2. Other than the above written value, nothing is changed.	W

The SWRSYS register controls system software reset.

6.3.3 SWRCPU0 : Cortex-R52 CPU0 Software Reset Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x220

Bit position:	31																												0
Bit field:	SWR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SWR[31:0]	Cortex-R52 CPU0 Software Reset 1. When 0x4321A501 is written, Cortex-R52 CPU0 software reset occurs. Cortex-R52 CPU0 and WDTR0 for Cortex-R52 CPU0 are in reset state. Reset is not released automatically. Write 0x00000000 to release the reset. 2. When 0x4321A502 is written, Cortex-R52 CPU0 software reset occurs. Cortex-R52 CPU0 and WDTR0 for Cortex-R52 CPU0 are in reset state. After around 1.02 µs passes from reset occurrence, reset is released automatically. Prohibit to write 0x00000000 to this register before above time passes. 3. Other than the above written value, nothing is changed. These bits are read as 0x00000001 during reset, and as 0x00000000 when reset is released. Be careful that after writing 0x4321A501 or 0x4321A502 to this register, transfer Cortex-R52 CPU0 to WFI state manually. After confirming Cortex-R52 CPU0 is in WFI state by hardware, Cortex-R52 CPU0 and WDTR0 for Cortex-R52 CPU0 are reset. It is prohibited to write 0x4321A501 after writing 0x4321A502 and 0x4321A502 after writing 0x4321A501.	R/W

The SWRCPU0 register controls the Cortex-R52 CPU0 software reset.

6.3.4 SWRCPU1 : Cortex-R52 CPU1 Software Reset Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x224

Bit position:	31																												0
Bit field:	SWR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
31:0	SWR[31:0]	Cortex-R52 CPU1 Software Reset 1. When 0x4321A501 is written, Cortex-R52 CPU1 software reset occurs. Cortex-R52 CPU1 and WDTR1 for Cortex-R52 CPU1 are in reset state. Reset is not released automatically. Write 0x00000000 to release the reset. 2. When 0x4321A502 is written, Cortex-R52 CPU1 software reset occurs. Cortex-R52 CPU1 and WDTR1 for Cortex-R52 CPU1 are in reset state. After around 1.02 µs passes from reset occurrence, reset is released automatically. Prohibit to write 0x00000000 to this register before above time passes. 3. Other than the above written value, nothing is changed. These bits are read as 0x00000001 during reset, and as 0x00000000 when reset is released. Be careful that after writing 0x4321A501 or 0x4321A502 to this register, transfer Cortex-R52 CPU1 to WFI state manually. After confirming Cortex-R52 CPU1 is in WFI state, Cortex-R52 CPU1 and WDTR1 for Cortex-R52 CPU1 are reset. It is prohibited to write 0x4321A501 after writing 0x4321A502 and 0x4321A502 after writing 0x4321A501.	R/W

The SWRCPU1 register controls Cortex-R52 CPU1 software reset.

6.3.5 SWR55C : Cortex-A55 Cluster Software Reset Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x228

Bit position:	31																												0
Bit field:	SWR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
31:0	SWR[31:0]	Cortex-A55 Cluster Software Reset 1. When 0x4321A501 is written, Cortex-A55 Cluster software reset occurs. Cortex-A55 Cluster and WDTA3-0 for Cortex-A55 are in reset state. Reset is not released automatically. Write 0x00000000 to release the reset. 2. When 0x4321A502 is written, Cortex-A55 Cluster software reset occurs. Cortex-A55 Cluster and WDTA3-0 for Cortex-A55 are in reset state. After around 1.02 µs passes from reset occurrence, reset is released automatically. Prohibit to write 0x00000000 to this register before above time passes. 3. Other than the above written value, nothing is changed. These bits are read as 0x00000001 during reset, and as 0x00000000 when reset is released. Be careful that after writing 0x4321A501 or 0x4321A502 to this register, transfer Cortex-A55 Cluster to WFI state manually. After confirming Cortex-A55 Cluster is in WFI state by hardware, Cortex-A55 Cluster and WDTA3-0 for Cortex-A55 are reset. It is prohibited to write 0x4321A501 after writing 0x4321A502 and 0x4321A502 after writing 0x4321A501.	R/W

The SWR55C register controls the Cortex-A55 Cluster software reset.

6.3.6 SWR550 : Cortex-A55 Core0 Software Reset Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x22C

Bit position:	31																												0
Bit field:	SWR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
31:0	SWR[31:0]	Cortex-A55 Core0 Software Reset When reset is released after power-on, write 0x00000000. These bits are read as 0x00000001 during reset, and as 0x00000000 when reset is released.	R/W

The SWR550 register controls the Cortex-A55 Core0 software reset after power-on.

6.3.7 SWR551 : Cortex-A55 Core1 Software Reset Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x230

Bit position:	31																												0
Bit field:	SWR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
31:0	SWR[31:0]	Cortex-A55 Core1 Software Reset When reset is released after power-on, write 0x00000000. These bits are read as 0x00000001 during reset, and as 0x00000000 when reset is released.	R/W

The SWR551 register controls the Cortex-A55 Core1 software reset after power-on.

6.3.8 SWR552 : Cortex-A55 Core2 Software Reset Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x234

Bit position:	31																												0
Bit field:	SWR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
31:0	SWR[31:0]	Cortex-A55 Core2 Software Reset When reset is released after power-on, write 0x00000000. These bits are read as 0x00000001 during reset, and as 0x00000000 when reset is released.	R/W

The SWR552 register controls the Cortex-A55 Core2 software reset after power-on.

6.3.9 SWR553 : Cortex-A55 Core3 Software Reset Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x238

Bit position:	31																												0
Bit field:	SWR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
31:0	SWR[31:0]	Cortex-A55 Core3 Software Reset When reset is released after power-on, write 0x00000000. These bits are read as 0x00000001 during reset, and as 0x00000000 when reset is released.	R/W

The SWR553 register controls the Cortex-A55 Core3 software reset after power-on.

6.3.10 SWR55ARC : Cortex-A55 Software Reset Automatic Release Control Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x23C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ARC553	ARC552	ARC551	ARC550
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	ARC550	Select automatic reset release for Cortex-A55 Core0 when Cortex-A55 Cluster software reset occurs with automatic reset release. 0: Reset is not released. 1: Reset is released automatically.	R/W

Bit	Symbol	Function	R/W
1	ARC551	Select automatic reset release for Cortex-A55 Core1 when Cortex-A55 Cluster software reset occurs with automatic reset release. 0: Reset is not released. 1: Reset is released automatically.	R/W
2	ARC552	Select automatic reset release for Cortex-A55 Core2 when Cortex-A55 Cluster software reset occurs with automatic reset release. 0: Reset is not released. 1: Reset is released automatically.	R/W
3	ARC553	Select automatic reset release for Cortex-A55 Core3 when Cortex-A55 Cluster software reset occurs with automatic reset release. 0: Reset is not released. 1: Reset is released automatically.	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The SWR55ARC register determines which Cortex-A55 Cores are released from reset state automatically when Cortex-A55 Cluster software reset occurs with automatic reset release.

6.3.11 MRCTLA : Module Reset Control Register A

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x240

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	MRCTLA05	MRCTLA04	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	MRCTLA04	xSPI Unit 0 Reset Control This bit controls the reset of the xSPI unit 0. 0: xSPI Unit 0 is released from reset 1: xSPI Unit 0 is in the reset state	R/W
5	MRCTLA05	xSPI Unit 1 Reset Control This bit controls the reset of xSPI unit 1. 0: xSPI Unit 1 is released from reset 1: xSPI Unit 1 is in the reset state	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The MRCTLA register controls the reset of xSPI. This register is available in Non Safety region.

6.3.12 MRCTLE : Module Reset Control Register E

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x250

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MRCTLE19	MRCTLE18	MRCTLE17	MRCTLE16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	MRCTLE06	MRCTLE05	MRCTLE04	MRCTLE03	MRCTLE02	MRCTLE01	MRCTLE00
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MRCTLE00	GMAC Unit 0 (PCLKH clock domain) Reset Control This bit controls the reset of GMAC Unit 0 (PCLKH clock domain). 0: GMAC Unit 0 (PCLKH clock domain) is released from reset 1: GMAC Unit 0 (PCLKH clock domain) is in the reset state	R/W
1	MRCTLE01	GMAC Unit 0 (PCLKM clock domain) Reset Control This bit controls the reset of GMAC Unit 0 (PCLKM clock domain). 0: GMAC Unit 0 (PCLKM clock domain) is released from reset 1: GMAC Unit 0 (PCLKM clock domain) is in the reset state	R/W
2	MRCTLE02	ETHSW Reset Control This bit controls the reset of ETHSW. 0: ETHSW is released from reset 1: ETHSW is in the reset state	R/W
3	MRCTLE03	ESC (Bus clock domain) Reset Control This bit controls the reset of ESC (Bus clock domain). 0: ESC (Bus clock domain) is released from reset 1: ESC (Bus clock domain) is in the reset state	R/W
4	MRCTLE04	ESC (IP clock domain) Reset Control This bit controls the reset of ESC (IP clock domain). 0: ESC (IP clock domain) is released from reset 1: ESC (IP clock domain) is in the reset state	R/W
5	MRCTLE05	Ethernet Subsystem Register Reset Control This bit controls the reset of Ethernet Subsystem register. 0: Ethernet Subsystem register is released from reset 1: Ethernet Subsystem register is in the reset state	R/W
6	MRCTLE06	MII Converter Reset Control This bit controls the reset of MII converter. 0: MII converter is released from reset 1: MII converter is in the reset state	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
16	MRCTLE16	GMAC Unit 1 (PCLKAH clock domain) Reset Control This bit controls the reset of GMAC Unit 1 (PCLKAH clock domain). 0: GMAC Unit 1 (PCLKAH clock domain) is released from reset 1: GMAC Unit 1 (PCLKAH clock domain) is in the reset state	R/W
17	MRCTLE17	GMAC Unit 1 (PCLKAM clock domain) Reset Control This bit controls the reset of GMAC Unit 1 (PCLKAM clock domain). 0: GMAC Unit 1 (PCLKAM clock domain) is released from reset 1: GMAC Unit 1 (PCLKAM clock domain) is in the reset state	R/W
18	MRCTLE18	GMAC Unit 2 (PCLKAH clock domain) Reset Control This bit controls the reset of GMAC Unit 2 (PCLKAH clock domain). 0: GMAC Unit 1 (PCLKAH clock domain) is released from reset 1: GMAC Unit 1 (PCLKAH clock domain) is in the reset state	R/W

Bit	Symbol	Function	R/W
19	MRCTLE19	GMAC Unit 2 (PCLKAM clock domain) Reset Control This bit controls the reset of GMAC Unit 2 (PCLKAM clock domain). 0: GMAC Unit 2 (PCLKAM clock domain) is released from reset 1: GMAC Unit 2 (PCLKAM clock domain) is in the reset state	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The MRCTLE register controls the reset of peripheral modules related to Ethernet.

6.3.13 MRCTLI : Module Reset Control Register I

Base address: SYSC_S = 0x8128_0000

Offset address: 0x260

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MRCTLI03	MRCTLI02	MRCTLI01	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	MRCTLI01	SHOSTIF (Master bus clock domain) Reset Control 0: SHOSTIF (Master bus clock domain) is released from reset 1: SHOSTIF (Master bus clock domain) is in the reset state	R/W
2	MRCTLI02	SHOSTIF (Slave bus clock domain) Reset Control 0: SHOSTIF (Slave bus clock domain) is released from reset 1: SHOSTIF (Slave bus clock domain) is in the reset state	R/W
3	MRCTLI03	SHOSTIF (IP clock domain) Reset Control 0: SHOSTIF (IP clock domain) is released from reset 1: SHOSTIF (IP clock domain) is in the reset state	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The MRCTLI register controls reset of SHOSTIF. This register is available in Safety region.

6.3.14 MRCTLM : Module Reset Control Register M

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x270

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	MRCTLM25	MRCTLM24	MRCTLM23	MRCTLM22	MRCTLM21	MRCTLM20	MRCTLM19	MRCTLM18	MRCTLM17	MRCTLM16
Value after reset:	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MRCTLM8	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	MRCTLM8	PCIE Reset Control 0: PCIE is released from reset 1: PCIE is in the reset state	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	MRCTLM16	DDRSS (rst_n) Reset Control 0: DDRSS (rst_n) is released from reset 1: DDRSS (rst_n) is in the reset state	R/W
17	MRCTLM17	DDRSS (PwrOkIn) Reset Control 0: DDRSS (PwrOkIn) is released from reset 1: DDRSS (PwrOkIn) is in the reset state	R/W
18	MRCTLM18	DDRSS (Reset) Reset Control 0: DDRSS (Reset) is released from reset 1: DDRSS (Reset) is in the reset state	R/W
19	MRCTLM19	DDRSS (axi0_ARESETn) Reset Control 0: DDRSS (axi0_ARESETn) is released from reset 1: DDRSS (axi0_ARESETn) is in the reset state	R/W
20	MRCTLM20	DDRSS (axi1_ARESETn) Reset Control 0: DDRSS (axi1_ARESETn) is released from reset 1: DDRSS (axi1_ARESETn) is in the reset state	R/W
21	MRCTLM21	DDRSS (axi2_ARESETn) Reset Control 0: DDRSS (axi2_ARESETn) is released from reset 1: DDRSS (axi2_ARESETn) is in the reset state	R/W
22	MRCTLM22	DDRSS (axi3_ARESETn) Reset Control 0: DDRSS (axi3_ARESETn) is released from reset 1: DDRSS (axi3_ARESETn) is in the reset state	R/W
23	MRCTLM23	DDRSS (axi4_ARESETn) Reset Control 0: DDRSS (axi4_ARESETn) is released from reset 1: DDRSS (axi4_ARESETn) is in the reset state	R/W
24	MRCTLM24	DDRSS (MC_PRESETn) Reset Control 0: DDRSS (MC_PRESETn) is released from reset 1: DDRSS (MC_PRESETn) is in the reset state	R/W
25	MRCTLM25	DDRSS (PHY_PRESETn) Reset Control 0: DDRSS (PHY_PRESETn) is released from reset 1: DDRSS (PHY_PRESETn) is in the reset state	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The MRCTLM register controls reset of PCIE and DDRSS. This register is available in Non-Safety region. Set MRCTLM[25:16] bits for DDRSS Reset Control to the same value.

6.4 Operation

6.4.1 RES# Pin Reset

This reset occurs when a signal arrives at the RES# pin from the externally connected reset circuit. When the signal becomes low at the RES# pin, all on-going processes are aborted and the LSI enters the reset state. To successfully reset the LSI, the RES# pin should be held at the low level for the specified time after power-on. For details on the reset configuration, see section 10.3.3. Reset Configuration and the Method of Connecting with the Emulator.

After the reset is released, Cortex-R52 CPU0 starts boot processing. When a RES# pin reset occurs, the RSTSR0.TRF flag is set to 1.

6.4.2 System Software Reset

System software reset occurs when 0x4321A501 is written to the SWRSYS register. When system software reset occurs, all on-going processes are aborted and the LSI enters the reset state. After the reset is released, CPU0 starts boot processing.

When system software reset occurs, the RSTSR0.SWRSF flag is set to 1.

6.4.3 Cortex-R52 CPU0/1 Software Reset

These software reset apply to target CPU core and its related WDT. It occurs when 0x4321A501 or 0x4321A502 is written to the target CPU's software reset register and target CPU core is transferred to WFI state by WFI instruction. Do not write the same software reset register again until target CPU core is transferred to WFI state and the software reset register is changed to 0x00000001.

When 0x4321A501 is written, reset is not released automatically. To release the reset state of target CPU core software reset, write 0x00000000 to the software reset register from other CPU after confirming the read value of the software reset register is 0x00000001.

When 0x4321A502 is written, reset is released automatically. Do not write 0x00000000 to the software reset register.

When Cortex-R52 CPU0/1 software reset occurs, corresponding flag in RSTSR0 register is set to 1.

6.4.4 Cortex-A55 Core0/1/2/3 Software Reset

These software reset apply to target CPU core and its related WDT.

SWR55n register is only used to release Cortes-A55 Core n from the reset state after power-on. Reset of the Cortex-A55 Core n and its related WDT are released when 0x00000000 is written to the SWR55n register.

To reset the Cortex-A55 Core n and release it from the reset state again automatically, set the RR bit in the RMR_EL3 register in target Cortex-A55 Core n and transfer the core to WFI state by WFI instruction.

It is not supported to remain the reset state again once reset is released after power-on.

6.4.5 Cortex-A55 Cluster Software Reset

Cortex-A55 Cluster software reset applies to Cortex-A55 Cluster and all WDT for Cortex-A55. It occurs when 0x4321A501 or 0x4321A502 is written to the SWR55C register and all the Cortex-A55 Cores are transferred to WFI state by WFI instruction. Do not write the SWR55C register again until all the Cortex-A55 Cores are transferred to WFI state and SWR55C register is changed to 0x00000001.

When 0x4321A501 is written, reset is not released automatically. To release the reset state of each Cortex-A55 Core software reset, write 0x00000000 to the corresponding software reset register from non-reset CPU after confirming the read value of SWR55C is 0x00000001.

When 0x4321A502 is written, reset is released automatically for the cores which are specified in the SWR55ARC register. Do not write 0x00000000 to the SWR55C register.

When Cortex-A55 Cluster software reset occurs, the RSTSR0.SWR55C flag is set to 1.

6.4.6 Error Reset

This reset is generated in response to a reset request from ICU. ICU receives critical errors, such as oscillation stop detection, from individual modules in the LSI, and generates reset requests corresponding to respective errors. For details on the ICU operation, see section 12, Interrupt Controller (ICU).

When error reset occurs, all on-going processes are aborted and the LSI enters the reset state. After the reset is released, Cortex-R52 CPU0 starts boot processing. When the error reset occurs, the RSTSR0.ERRF flag is set to 1.

6.4.7 Reset Release from RES# Pin Reset, System Software Reset or Error Reset

When a RES# pin reset, system software reset, or error reset occurs, CPU cores other than Cortex-R52 CPU0 is in reset state initially in case the second boot CPU is Cortex-R52 CPU0, and CPU cores other than Cortex-A55 Core0 is in reset state initially in case the second boot CPU is Cortex-A55 Core0.

To release the reset state, write 0x00000000 to the target CPU's software reset register from non-reset CPU.

Note that some other settings like vector table setting and storing the code are needed before releasing reset. See section 2, Cortex-A55 and section 3, Cortex-R52.

6.4.8 Determination of Reset Generation Source

Reading the RSTSR0 register determines which reset source was used for reset execution. Figure 6.1 shows an example of the flow to identify a reset generation source.

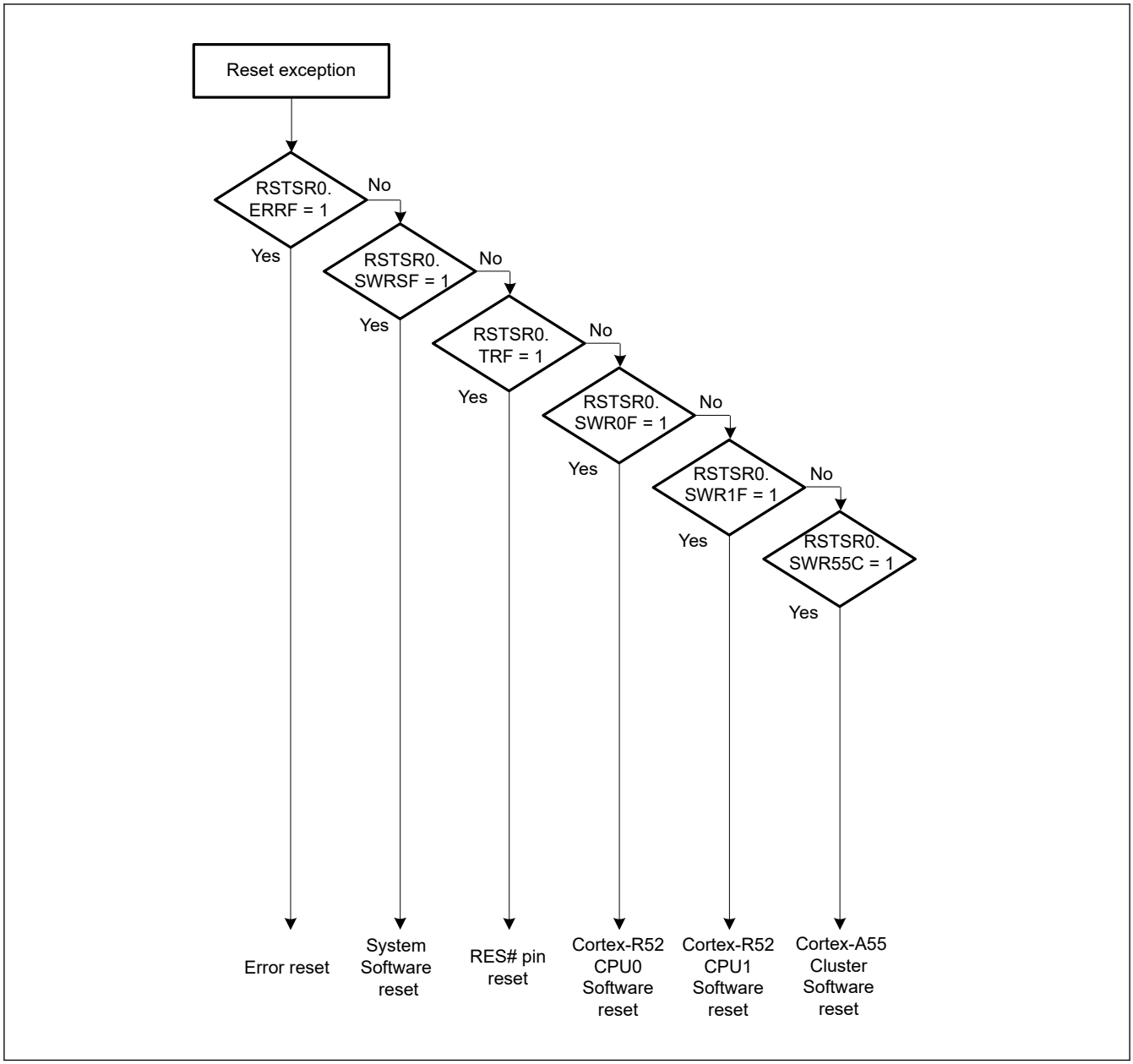


Figure 6.1 Example flow to identify a reset generation source

6.4.9 Reset Output Pin (RSTOUT#)

The reset output pin (RSTOUT#) outputs the low level on occurrence of a reset when the input level of the RES# pin is low. It also outputs the low level on occurrence of a system software reset or an error reset. If the RES# pin remains low for a specified time period and then changes to high, the reset output remains low for 450 µs (Typ.) and then changes to high. Similarly, the reset output remains low for 450 µs (Typ.) following a system software reset or an error reset, and then changes to high.

6.4.10 Noise Cancellation for Reset Input

Noise cancellation using analog delay is applied to the RES# and TRST# pins. This can eliminate noises within 100 ns (Min.).

6.5 Usage Note

6.5.1 Notes on Module Reset Control Register Operation

1. To secure processing after release from a module reset, dummy read the same register at least seven times except RTC and LCDC after writing to initiate release from the module reset, and only then proceed with the subsequent processing. For RTC, dummy read the same register at least 300 times and for LCDC, at least 100 times.
2. When module is reset once and released again, make sure that the target bit of module reset control register is set to 1 by reading the register before releasing from a module reset. Then release from a module reset.
3. When controlling the module resetting of modules that can initiate DMA transfer or that otherwise have operations related to those of other blocks, confirm that DMA transfer initiated by such modules and other points of interlinked operation has been completed before releasing modules from or placing modules in the module reset state. For set products in which such target modules are not used, design of the control of the module reset is on the assumption that the modules are initially stopped. Therefore, the absence of unexpected operations in states other than the above is not guaranteed. For example, operation after releasing modules from or placing modules in the module reset state while a module is running is not guaranteed.
4. In case of AXI-type slave module, the access to the slave I/F needs to be stopped by slave stop function before changing module reset state. For details, see section 13.4.3. Slave Stop Function.

6.5.2 Connection of Reset Output Pin (RSTOUT#)

The low level is output for a specified period of time on RSTOUT# after deassertion of the signal on the RES# pin. This means that RSTOUT# should not be directly connected as a reset signal for a flash memory to be used in booting the LSI. Otherwise, release of the LSI itself from the reset state may precede that for the flash memory.

However, using RSTOUT# as the reset signal for external device is still possible as long as the timing requirement imposed by the output on the RSTOUT# pin is satisfied.

RSTOUT# is shared function for port P08_5 but it is set by default after RES# pin reset, system software reset, and error reset. Even if function other than RSTOUT# is assigned, low level output from RSTOUT# is always effective when RES# pin reset, system software reset, or error reset occurs.

For details, see section 6.4.9. Reset Output Pin (RSTOUT#).

7. Clock Generation Circuit

7.1 Overview

The LSI incorporates a Clock Generation Circuit (CGC).

Table 7.1 and Table 7.2 show the specifications of the clock generation circuit. Figure 7.1 shows the block diagram and Table 7.3 shows the pin configuration of the clock generation circuit.

Table 7.1 Specifications of Clock Generation Circuit (1 of 2)

Item	Description
Main clock oscillator	Resonator frequency: 25 MHz
	External clock input frequency: 25 MHz
	Connectable resonator or additional circuit: Ceramic resonator, crystal resonator (XTALSEL = 1) Connection pin: EXTAL, XTAL
	External clock input instead of resonator (XTALSEL = 0) Connection pin: EXTCLKIN
	Oscillation stop detection function: When the CLMA6 detects an abnormality in oscillation of the main clock, the system clock source is switched to LOCO and PLL output clocks are switched to LOCO or free-running clock of PLL, and MTU3 and GPT pins are placed in the high-impedance state.
PLL0 circuit (for Cortex-A55 CPU clock)	Input clock source: Main clock oscillator
	Input frequency: 25 MHz
	Frequency multiplication ratio: 48 multiplication
	Output clock frequency of the PLL0 circuit: 1200 MHz
	SSC (Spread Spectrum Clock) / Non-SSC selectable
	Oscillation abnormality detection function: When the CLMA0 detects an abnormality in oscillation of the PLL0 with MOSC, the PLL0 output clock is switched to MOSC, and MTU3 and GPT pins are placed in the high-impedance state.
PLL1 circuit (for Cortex-R52 CPU, main bus R and peripheral clock)	Input clock source: Main clock oscillator
	Input frequency: 25 MHz
	Frequency multiplication ratio: 40 multiplication
	Output clock frequency of the PLL1 circuit: 1000 MHz
	Non-SSC fixed
	Oscillation abnormality detection function: When the CLMA1 detects an abnormality in oscillation of the PLL1 with MOSC, the PLL1 output clock is switched to MOSC, and MTU3 and GPT pins are placed in the high-impedance state.
PLL2 circuit (for DDRSS and SDHI clock)	Input clock source: Main clock oscillator
	Input frequency: 25 MHz
	Frequency multiplication ratio: 32 multiplication
	Output clock frequency of the PLL2 circuit: 800 MHz
	SSC/Non-SSC selectable
	Oscillation abnormality detection function: When the CLMA2 detects an abnormality in oscillation of the PLL2 with MOSC, the PLL2 output clock is switched to MOSC.
PLL3 circuit (for LCDC clock)	Input clock source: Dividing clock (1/50) of PLL4
	Input frequency: 48 MHz
	Frequency multiplication ratio: Configurable
	Output clock frequency of the PLL3 circuit: 25 MHz to 430 MHz
	Non-SSC fixed
	Oscillation abnormality detection function: When the CLMA3 detects an abnormality in oscillation of the PLL3 with MOSC, the PLL3 output clock is switched to MOSC.

Table 7.1 Specifications of Clock Generation Circuit (2 of 2)

Item	Description
PLL4 circuit (for main bus A and peripheral clock)	Input clock source: Main clock oscillator
	Input frequency: 25 MHz
	Frequency multiplication ratio: 96 multiplication
	Output clock frequency of the PLL4 circuit: 2400 MHz
	Non-SSC fixed
	Oscillation abnormality detection function: When the CLMA4 detects an abnormality in oscillation of the PLL4 with MOSC, the PLL4 output clock is switched to MOSC, and MTU3 and GPT pins are placed in the high-impedance state.
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 1 MHz +/- 20%
	Oscillation abnormality detection function: The CLMA2 can detect an abnormality in oscillation of the LOCO.
External clock input for JTAG (TCK)	Input frequency: 50 MHz (max.)

Table 7.2 Specifications of Clock Generation Circuit (internal clock) (1 of 3)

Item	Clock source	Supplied to	Frequency	Reset value
Cortex-A55 Core0 clock (CA55C0CLK)	Dividing clock of PLL0 (1/1 or 1/2)	Cortex-A55 Core0	600, 1200 MHz	600 MHz
Cortex-A55 Core1 clock (CA55C1CLK)	Dividing clock of PLL0 (1/1 or 1/2)	Cortex-A55 Core1	600, 1200 MHz	600 MHz
Cortex-A55 Core2 clock (CA55C2CLK)	Dividing clock of PLL0 (1/1 or 1/2)	Cortex-A55 Core2	600, 1200 MHz	600 MHz
Cortex-A55 Core3 clock (CA55C3CLK)	Dividing clock of PLL0 (1/1 or 1/2)	Cortex-A55 Core3	600, 1200 MHz	600 MHz
Cortex-A55 DSU clock (CA55SCLK)	Dividing clock of PLL1 (1/1 or 1/2)	Cortex-A55 DSU	500, 1000 MHz	500 MHz
Cortex-R52 CPU0 clock (CR52C0CLK)	Dividing clock of PLL1 (1/1 or 1/2)	Cortex-R52 CPU0	500, 1000 MHz	500 MHz
Cortex-R52 CPU1 clock (CR52C1CLK)	Dividing clock of PLL1 (1/1 or 1/2)	Cortex-R52 CPU1	500, 1000 MHz	500 MHz
Peripheral module clock AH (PCLKAH)	Dividing clock of PLL4 (1/6)	Main bus A, PCIE, LCDC (clk_a), GMAC unit 1, 2 (ACLK), DDRSS (A0, A1, A4)	400 MHz	400 MHz
Peripheral module clock AM (PCLKAM)	Dividing clock of PLL4 (1/12)	USB, GMAC unit 1, 2 (HCLK), SDHI (ACLK, IMCLK)	200 MHz (PCLKAH/2)	200 MHz
Peripheral module clock AL (PCLKAL)	Dividing clock of PLL4 (1/24)	LCDC (clk_p)	100 MHz (PCLKAH/4)	100 MHz
SDHI clock (SDHI_clkhs)	PLL2 clock	SDHI (clk_hs)	800 MHz	800 MHz
DDR controller DFI clock (DFICLK)	PLL2 clock	DDRSS	800 MHz	800 MHz
LCDC clock (LCDC_clkd)	Dividing clock of PLL3 (1/2, 1/4, 1/6, 1/8, 1/10, 1/12, 1/14, 1/16, 1/18, 1/20, 1/22, 1/24, 1/26, 1/28, 1/30, 1/32)	LCDC (clk_d)	5 MHz to 100 MHz	12.5 MHz
Peripheral module clock H (PCLKH)	Dividing clock of PLL1 (1/4)	Main bus R, SYSRAM, LLPP, Peripheral module (MTU3, GPT (LLPP, bus), TFU, DSMIF (bus), POE3, POEG(LLPP), ADC12 unit 0, 1 (bus)), xSPI (bus), DMAC, MPU, GMAC unit 0 (ACLK), SHOSTIF (pclk, mhclk), ELC, RSIP, DDRSS (R2, R3), MBXSEM	250 MHz	250 MHz

Table 7.2 Specifications of Clock Generation Circuit (internal clock) (2 of 3)

Item	Clock source	Supplied to	Frequency	Reset value
Peripheral module clock M (PCLKM)	Dividing clock of PLL1 (1/8)	Peripheral module (CRC, SCI (bus), SPI (bus), GPT (Non Safety, Safety), CANFD (bus)), ETHSW(bus), GMAC unit 0 (HCLK), OTP, Ethernet SS Reg, GPIO, CLMA, ADC12 unit 2 (bus), ICU	125 MHz (PCLKH/2)	125 MHz
Peripheral module clock L (PCLKL)	Dividing clock of PLL1 (1/16)	Peripheral module (POEG (Non Safety, Safety), IIC, DOC,CMT, CMTW, WDT, RTC, TSU), ADC12 (conversion), DDRSS (PHY_reg)	62.5 MHz (PCLKH/4)	62.5 MHz
Peripheral module clock for GPT in LLPP (PCLKGPTL)	Dividing clock of PLL1 (1/2)	GPT (LLPP, operation)	500 MHz (PCLKH × 2)	500 MHz
Peripheral module clock for DSMIF (PCLKDSMIFn)	Dividing clock of PLL4 (1/6) and PLL1 (1/4)	DSMIF (core)	250 MHz (PCLKH), 400 MHz (Asynchronous)	250 MHz
Peripheral module clock for SCI/SCIE (PCLKSCIn/ PCLKSCIEm) (n = 0 to 5, m = 0 to 11)	Dividing clock of PLL4 (1/24, 1/25, 1/30, 1/32) and PLL1 (1/8)	SCIn/SCIEm (operation)	75, 80, 96, 100 MHz (Asynchronous), 125 MHz (PCLKM)	75 MHz
Peripheral module clock for SPI (PCLKSPIn) (n = 0 to 3)	Dividing clock of PLL4 (1/24, 1/25, 1/30, 1/32) and PLL1 (1/8)	SPIn (operation)	75, 80, 96, 100 MHz (Asynchronous), 125 MHz (PCLKM)	75 MHz
Peripheral module clock for CANFD (PCLKCAN)	Dividing clock of PLL4 (1/30, 1/60)	CANFD (serial)	40, 80 MHz	40 MHz
Peripheral module clock for SHOSTIF (PCLKSHOST)	Dividing clock of PLL4 (1/6)	SHOSTIF (ssi_clk)	400 MHz	400 MHz
External bus clock (BSC_CLK, CKIO)	Dividing clock of PLL1 (1/8, 1/12, 1/16, 1/20, 1/24, 1/28, 1/32)	External bus	125, 83.3, 62.5, 50, 41.7, 35.7, 31.25 MHz*1	50 MHz
Peripheral module clock for RTC (PCLKRTC)	Dividing clock of MOSC (1/128)	RTC (operation)	195.3 kHz	195.3 kHz
USB clock (USB_CLK)	Dividing clock of PLL4 (1/48)	USB	50 MHz	50 MHz
xSPIn serial clock (XSPI_CLKn) (n = 0, 1)	Dividing clock of PLL4 (1/18, 1/24, 1/32, 1/48, 1/64, 1/96, 1/192)	xSPIn (operation)	133, 100, 75, 50, 37.5, 25, 12.5 MHz*2	12.5 MHz
Ethernet clock A (ETCLKA)	Dividing clock of PLL1 (1/5)	ETHSW	200 MHz	200 MHz
Ethernet clock B (ETCLKB)	Dividing clock of PLL1 (1/8)	ETHSW, GMAC, RGMII/RMII Converter	125 MHz	125 MHz
Ethernet clock C (ETCLKC)	Dividing clock of PLL1 (1/10)	ESC	100 MHz	100 MHz
Ethernet clock D (ETCLKD)	Dividing clock of PLL1 (1/20)	RGMII/RMII Converter	50 MHz	50 MHz
Ethernet clock E (ETCLKE)	Dividing clock of PLL1 (1/40)	ESC, RGMII/RMII Converter	25 MHz	25 MHz
CLMAn sampling clock (CLMAMCLKA) (n = 0, 1, 2, 4)	Main clock oscillator divided by 2	CLMAn (n = 0, 1, 2, 4)	12.5 MHz	12.5 MHz
CLMA3 sampling clock (CLMAMCLKB)	Main clock oscillator divided by 16	CLMA3	1.56 MHz	1.56 MHz
CLMA5 sampling clock (CLMAMCLKC)	Main clock oscillator divided by 256	CLMA5	97.6 kHz	97.6 kHz

Table 7.2 Specifications of Clock Generation Circuit (internal clock) (3 of 3)

Item	Clock source	Supplied to	Frequency	Reset value
CLMA0 monitor clock (CLMAPLCLK0)	Dividing clock for PLL0 (1/16)	CLMA0	75 MHz	75 MHz
CLMA1 monitor clock (CLMAPLCLK1)	Dividing clock for PLL1 (1/16)	CLMA1	62.5 MHz	62.5 MHz
CLMA2 monitor clock (CLMAPLCLK2)	Dividing clock for PLL2 (1/16)	CLMA2	50 MHz	50 MHz
CLMA3 monitor clock (CLMAPLCLK3)	Dividing clock for PLL3 (1/4)	CLMA3	6.25 to 107.5 MHz	6.25 MHz
CLMA4 monitor clock (CLMAPLCLK4)	Dividing clock for PLL4 (1/32)	CLMA4	75 MHz	75 MHz
CLMA5 monitor clock (CLMALCLK)	LOCO output clock	CLMA5	1 MHz	1 MHz
CLMA6 monitor clock (CLMAMCLK)	Main clock oscillator divided by 2	CLMA6	12.5 MHz	12.5 MHz
JTAG clock (JTAGTCK)	TCK	JTAG	Up to 50 MHz	—
Serial Host I/F clock (SHIFCLK)	HSPI_CK	SHOSTIF	Up to 40 MHz	—
EnDat Clock (PCLKENDAT)	Dividing clock of PLL4 (1/24)	ENDAT	100 MHz	100 MHz
BiSS-C Clock (PCLKBISS)	Dividing clock of PLL4 (1/30)	BISS	80 MHz	80 MHz
HIPERFACE DSL Clock (PCLKHDSL)	Dividing clock of PLL1 (1/32)	HDSL	75 MHz	75 MHz
A-format Clock (PCLKAFMT)	Dividing clock of PLL1 (1/30)	AFMT	80 MHz	80 MHz
ENCOUT clock (PCLKENCO)	Dividing clock of PLL1 (1/30, 1/120)	ENCOUT	20, 80 MHz	20 MHz
Ethernet PHY reference clock (ETHn_REFCLK) (n = 0 to 3)	Dividing clock of PLL1 (1/40), Main clock oscillator	— (External Ethernet PHY)	25 MHz	25 MHz
Ethernet PHY reference clock (RMIIn_REFCLK) (n = 0 to 3)	Dividing clock of PLL1 (1/20)	— (External Ethernet PHY)	50 MHz	50 MHz

Note 1. Max operating frequency depends on operating conditions (C, $T_j$min). For details, see section 58.5.3. Bus Timing.

Note 2. Max operating frequency depends on operating conditions (Voltage, SDR/DDR). For details, see section 58.5.5.12. xSPI Timing.

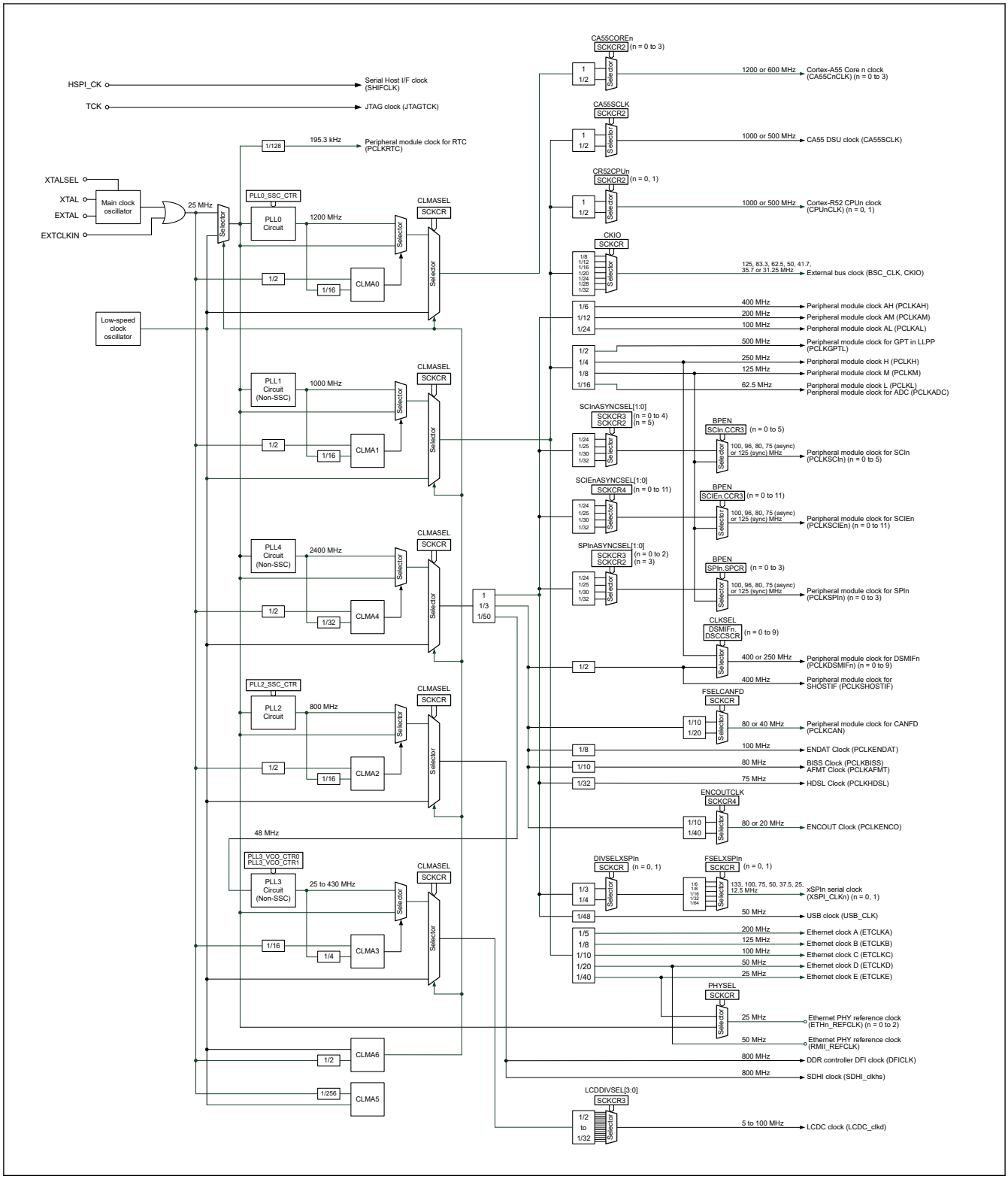


Figure 7.1 Block diagram of clock generation circuit

Table 7.3 Pin configuration of clock generation circuit (1 of 2)

Pin name	I/O	Description
XTAL	Output	These pins are used to connect a crystal resonator. XTALSEL pin must be driven high. When using an external clock signal, EXTAL pin must be driven low. XTAL pin must never be driven or loaded by anything other than crystal resonator. For details, see section 7.4.2. External Clock Input.
EXTAL	Input	

Table 7.3 Pin configuration of clock generation circuit (2 of 2)

Pin name	I/O	Description
EXTCLKIN	Input	This pin is used to input an external clock signal. XTALSEL pin must be driven low. When using a crystal resonator, this pin must be driven low.
XTALSEL	Input	Main clock source select pin (low: EXTCLKIN, high: XTAL/EXTAL)
TCK	Input	This pin is used to input the clock for the JTAG.
CKIO	Output	This pin is used to supply the external bus clock (CKIO) to external devices.
ETH0_REFCLK	Output	This pin is used to supply 25 MHz reference clock to the external Ethernet PHY.
ETH1_REFCLK	Output	This pin is used to supply 25 MHz reference clock to the external Ethernet PHY.
ETH2_REFCLK	Output	This pin is used to supply 25 MHz reference clock to the external Ethernet PHY.
ETH3_REFCLK	Output	This pin is used to supply 25 MHz reference clock to the external Ethernet PHY.
RMII0_REFCLK	Output	This pin is used to supply 50 MHz reference clock to the external Ethernet PHY.
RMII1_REFCLK	Output	This pin is used to supply 50 MHz reference clock to the external Ethernet PHY.
RMII2_REFCLK	Output	This pin is used to supply 50 MHz reference clock to the external Ethernet PHY.
RMII3_REFCLK	Output	This pin is used to supply 50 MHz reference clock to the external Ethernet PHY.

7.2 Register Map

Table 7.4 Clock Generation Circuit register map

Address	Register symbol	Register name	Write protection
0x8028_0000	SCKCR	System Clock Control Register	PRCRN.PRC0
0x8128_0004	SCKCR2	System Clock Control Register 2	PRCRS.PRC0
0x8028_0008	SCKCR3	System Clock Control Register 3	PRCRN.PRC0
0x8028_000C	SCKCR4	System Clock Control Register 4	PRCRN.PRC0
0x8128_0010	PMSEL	PLL/Main OSC clock Select Register	PRCRS.PRC0
0x8128_0020	PLL0MON	PLL0 Monitor Register	—
0x8128_0030	PLL0EN	PLL0 Enable Register	PRCRS.PRC0
0x8128_0034	PLL0_SSC_CTR	PLL0 SSC Control Register	PRCRS.PRC0
0x8128_0040	PLL1MON	PLL1 Monitor Register	—
0x8128_0070	LOCOCR	Low-Speed On-Chip Oscillator Control Register	PRCRS.PRC0
0x8128_0080	HIZCTRLEN	High-Impedance Control Enable Register	—
0x8128_0090	PLL2MON	PLL2 Monitor Register	—
0x8128_00A0	PLL2EN	PLL2 Enable Register	PRCRS.PRC0
0x8128_00AC	PLL2_SSC_CTR	PLL2 SSC Control Register	PRCRS.PRC0
0x8128_00B0	PLL3MON	PLL3 Monitor Register	—
0x8128_00C0	PLL3EN	PLL3 Enable Register	PRCRS.PRC0
0x8128_00C4	PLL3_VCO_CTR0	PLL3 VCO Control Register 0	PRCRS.PRC0
0x8128_00C8	PLL3_VCO_CTR1	PLL3 VCO Control Register 1	PRCRS.PRC0
0x8128_00D0	PLL4MON	PLL4 Monitor Register	—

Table 7.5 Clock Generation Circuit related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
All the registers	—	—	SLVACCCTL8.SYSCTRL_SL

7.3 Register Descriptions

The registers related to the clock generation circuit can be write-protected. To write to the registers, specify bit 0 of the Protect Register (PRCRN and PRCRS) to cancel the write protection. For details, see section 11, Register Write Protection Function.

7.3.1 SCKCR : System Clock Control Register

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	CLMASEL	PHYSEL	FSELCAND	—	CKIO[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	DIVSELXSPI1	—	—	—	FSELXSPI1[2:0]			—	DIVSELXSPI0	—	—	—	FSELXSPI0[2:0]		
Value after reset:	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
2:0	FSELXSPI0[2:0]	Set the frequency of the clock provided to xSPI Unit 0 in combination with bit 6 (DIVSELXSPI0). The combination is shown below. Both DIVSELXSPI0 and FSELXPI0 can be changed at the same time. Be careful that the set frequency is output after the intermediate frequency is output. Max operating frequency depends on operating conditions (Voltage, SDR/DDR). For details, see section 58.5.5.12. xSPI Timing.	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	DIVSELXSPI0	Select the base clock to generate serial clock for xSPI Unit 0 0: 800 MHz 1: 600 MHz	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	FSELXSPI1[2:0]	Set the frequency of the clock provided to xSPI Unit 1 in combination with bit 14 (DIVSELXSPI1). Same as bits [2:0] for xSPI Unit 0.	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	DIVSELXSPI1	Select the base clock to generate serial clock for xSPI Unit 1 0: 800 MHz 1: 600 MHz	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

FSELXSPI0[2:0]	DIVSELXSPI0 = 0	DIVSELXSPI0 = 1
000b	Setting prohibited	
001b	Setting prohibited	
010b	133.3 MHz	Setting prohibited
011b	100.0 MHz	75.0 MHz
100b	50.0 MHz	37.5 MHz
101b	25.0 MHz	Setting prohibited
110b	12.5 MHz	Setting prohibited
111b	Setting prohibited	

Bit	Symbol	Function	R/W
18:16	CKIO[2:0]	Set the frequency of the external bus clock (CKIO) and the clock supplied to BSC. Max operating frequency depends on operating conditions (C, $T_j$min). For details, see section 58.5.3. Bus Timing. **CKIO[2:0] / Frequency** 000b: 125 MHz 001b: 83.3 MHz 010b: 62.5 MHz 011b: 50.0 MHz 100b: 41.7 MHz 101b: 35.7 MHz 110b: 31.25 MHz 111b: Setting prohibited	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	FSELCANFD	Select the frequency of the clock supplied to CANFD 0: 80.0 MHz 1: 40.0 MHz	R/W
21	PHYSEL	Select the Ethernet PHY reference clock output (ETHn_REFCLK, n = 0 to 3) 0: PLL1 divider clock 1: Main clock oscillator	R/W
22	CLMASEL	Select alternative clock when main clock abnormal oscillation is detected in CLMA6 0: LOCO 1: PLL	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

The SCKCR register is used to select the frequency and source for each of the xSPI interface clock (XSPI_CLKn, n = 0, 1), external bus clock (CKIO), CANFD clock (PCLKCAN), Ethernet PHY reference clock (ETHn_REFCLK, n = 0 to 3), and alternative clock when the main clock abnormal oscillation is detected in CLMA6.

7.3.2 SCKCR2 : System Clock Control Register 2

Base address: SYSC_S = 0x8128_0000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SCI5ASYNCSEL[1:0]		SPI3ASYNCSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CA55SCLK	CA55CORE3	CA55CORE2	CA55CORE1	CA55CORE0	—	—	—	—	CR52CPU1[1:0]		CR52CPU0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CR52CPU0[1:0]	Set the frequency of the clock provided to Coretex-R52 CPU0 0 0: 500.0 MHz 0 1: 1000.0 MHz 1 0: Setting prohibited 1 1: Setting prohibited	R/W

Bit	Symbol	Function	R/W
3:2	CR52CPU1[1:0]	Set the frequency of the clock provided to Coretex-R52 CPU1 0 0: 500.0 MHz 0 1: 1000.0 MHz 1 0: Setting prohibited 1 1: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	CA55CORE0	Set the frequency of the clock provided to Coretex-A55 Core0 0: 600.0 MHz 1: 1200.0 MHz	R/W
9	CA55CORE1	Set the frequency of the clock provided to Coretex-A55 Core1 0: 600.0 MHz 1: 1200.0 MHz	R/W
10	CA55CORE2	Set the frequency of the clock provided to Coretex-A55 Core2 0: 600.0 MHz 1: 1200.0 MHz	R/W
11	CA55CORE3	Set the frequency of the clock provided to Coretex-A55 Core3 0: 600.0 MHz 1: 1200.0 MHz	R/W
12	CA55SCLK	Set the frequency of the clock provided to Coretex-A55 DSU 0: 500.0 MHz 1: 1000.0 MHz	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
17:16	SPI3ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SPI3 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
19:18	SCI5ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCI5 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The SCKCR2 register is used to select the frequency of the clock supplied to the Cortex-R52 CPU0 and CPU1, Cortex-A55 Core0 to Core3 and so on.

7.3.3 SCKCR3 : System Clock Control Register 3

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	LCDCDIVSEL[3:0]				—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCI4ASYNCSEL[1:0]		SCI3ASYNCSEL[1:0]		SCI2ASYNCSEL[1:0]		SCI1ASYNCSEL[1:0]		SCI0ASYNCSEL[1:0]		SPI2ASYNCSEL[1:0]		SPI1ASYNCSEL[1:0]		SPI0ASYNCSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SPI0ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SPI0 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
3:2	SPI1ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SPI1 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
5:4	SPI2ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SPI2 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
7:6	SCI0ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCI0 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
9:8	SCI1ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCI1 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
11:10	SCI2ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCI2 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
13:12	SCI3ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCI3 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
15:14	SCI4ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCI4 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
19:16	—	These bits are read as 0. The write value should be 0.	R/W
23:20	LCDCDIVSEL[3:0]	Select the external divider of PLL3 output clock 0x0: 1/2 0x1: 1/4 0x2: 1/6 0x3: 1/8 0x4: 1/10 0x5: 1/12 0x6: 1/14 0x7: 1/16 0x8: 1/18 0x9: 1/20 0xA: 1/22 0xB: 1/24 0xC: 1/26 0xD: 1/28 0xE: 1/30 0xF: 1/32	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The SCKCR3 register is used to select the frequency of the clock supplied to the SPI and SCI, and external divider of the PLL3 output clock.

7.3.4 SCKCR4 : System Clock Control Register 4

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ENCOUTCLK	SCIE11ASYNCSEL[1:0]		SCIE10ASYNCSEL[1:0]		SCIE9ASYNCSEL[1:0]		SCIE8ASYNCSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCIE7ASYNCSEL[1:0]		SCIE6ASYNCSEL[1:0]		SCIE5ASYNCSEL[1:0]		SCIE4ASYNCSEL[1:0]		SCIE3ASYNCSEL[1:0]		SCIE2ASYNCSEL[1:0]		SCIE1ASYNCSEL[1:0]		SCIE0ASYNCSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SCIE0ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE0 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
3:2	SCIE1ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE1 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
5:4	SCIE2ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE2 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
7:6	SCIE3ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE3 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
9:8	SCIE4ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE4 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
11:10	SCIE5ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE5 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
13:12	SCIE6ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE6 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
15:14	SCIE7ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE7 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W

Bit	Symbol	Function	R/W
17:16	SCIE8ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE8 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
19:18	SCIE9ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE9 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
21:20	SCIE10ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE10 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
23:22	SCIE11ASYNCSEL[1:0]	Select clock frequency when asynchronous serial clock is selected in SCIE11 0 0: 75.0 MHz 0 1: 80.0 MHz 1 0: 96.0 MHz 1 1: 100.0 MHz	R/W
24	ENCOUTCLK	Select the frequency of the clock provided to ENCOUT 0: 20.0 MHz 1: 80.0 MHz	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The SCKCR4 register is used to select the frequency of the clock supplied to the SCIE and ENCOUT.

7.3.5 PMSEL : PLL/Main OSC clock Select Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	PMSEL4MON	PMSEL3MON	PMSEL2MON	PMSEL1MON	PMSEL0MON	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	—	These bits are read as 0. The write value should be 0.	R/W
9	PMSEL0MON	PLL0 domain clock source monitor 0: External clock 1: PLL0 output clock	R
10	PMSEL1MON	PLL1 domain clock source monitor 0: External clock 1: PLL1 output clock	R
11	PMSEL2MON	PLL2 domain clock source monitor 0: External clock 1: PLL2 output clock	R
12	PMSEL3MON	PLL3 domain clock source monitor 0: External clock 1: PLL3 output clock	R

Bit	Symbol	Function	R/W
13	PMSEL4MON	PLL4 domain clock source monitor 0: External clock 1: PLL4 output clock	R
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The PMSEL register is used to select and monitor clock source of PLL clock domain.

7.3.6 PLL0MON : PLL0 Monitor Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0MON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PLL0MON	PLL0 Lock State Monitor This bit monitors whether the PLL0 is locked or not. 0: PLL0 is not locked 1: PLL0 is locked	R
31:1	—	These bits are read as 0.	R

The PLL0MON register is used to monitor the locking state of PLL0.

7.3.7 PLL0EN : PLL0 Enable Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PLL0EN	PLL0 Enable 0: PLL0 is in standby state 1: PLL0 is released from standby state	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The PLL0EN register is used to control PLL0 standby state.

7.3.8 PLL0_SSC_CTR : PLL0 SSC Control Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PLL0MFR								—	—	PLL0MRR					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0SSCEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PLL0SSCEN	PLL0 SSC Enable 0: SSC is disabled 1: SSC is enabled	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
21:16	PLL0MRR	PLL0 SSC Modulation Rate	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	PLL0MFR	PLL0 SSC Modulation Frequency	R/W

The PLL0_SSC_CTR register is used to control SSC (Spread Spectrum Clock) function of PLL0. Before changing this register, make sure that PLL0 is in standby state by PLL0EN register.

7.3.9 PLL1MON : PLL1 Monitor Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1MON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PLL1MON	PLL1 Lock State Monitor This bit monitors whether the PLL1 is locked or not. 0: PLL1 is not locked 1: PLL1 is locked	R
31:1	—	These bits are read as 0.	R

The PLL1MON register is used to monitor the locking state of PLL1.

7.3.10 LOCOCR : Low-Speed On-Chip Oscillator Control Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x70

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LCSTP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LCSTP	LOCO Stop Specify whether to run or stop the low-speed on-chip oscillator (LOCO). 0: Stop 1: Run	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: 1. When setting the LOCOCR to make the LOCO run again after it has been stopped, start using the LOCO clock after the LOCO oscillation stabilization time (t_{LOCOWT}) has elapsed.

2. Processing to stop oscillation after the bit has been set to stop the LOCO also requires a certain amount of time. Therefore, the following restrictions apply to making the LOCO run and stopping it:
   - When setting the bit to make the LOCO run again after it has been stopped, ensure that the period over which the LOCO has been stopped is at least the LOCO oscillation stabilization time (t_{LOCOWT}).
   - To set the bit to stop the LOCO, ensure that the LOCO oscillation is stable at the time.
3. LOCO may generate glitch when it is stopped after operation. Therefore, stop CLMA5 and CLMA6 which use LOCO clock, and transit to the module stop state by MSTPCRG register before LOCO is stopped.

The LOCOCR register is used to control operation of a low-speed on-chip oscillator (LOCO).

7.3.11 HIZCTRLEN : High-Impedance Control Enable Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	CLMA4MASK	CLMA3MASK	CLMA2MASK	CLMA1MASK	CLMA0MASK	CLMA6MASK
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1

Bit	Symbol	Function	R/W
0	CLMA6MASK	CLMA6 error mask to POE3 and POEG 0: CLMA6 error is not transferred to POE3 and POEG. 1: CLMA6 error is transferred to POE3 and POEG.	R/W
1	CLMA0MASK	CLMA0 error mask to POE3 and POEG 0: CLMA0 error is not transferred to POE3 and POEG. 1: CLMA0 error is transferred to POE3 and POEG.	R/W
2	CLMA1MASK	CLMA1 error mask to POE3 and POEG 0: CLMA1 error is not transferred to POE3 and POEG. 1: CLMA1 error is transferred to POE3 and POEG.	R/W

Bit	Symbol	Function	R/W
3	CLMA2MASK	CLMA2 error mask to POE3 and POEG 0: CLMA2 error is not transferred to POE3 and POEG. 1: CLMA2 error is transferred to POE3 and POEG.	R/W
4	CLMA3MASK	CLMA3 error mask to POE3 and POEG 0: CLMA3 error is not transferred to POE3 and POEG. 1: CLMA3 error is transferred to POE3 and POEG.	R/W
5	CLMA4MASK	CLMA4 error mask to POE3 and POEG 0: CLMA4 error is not transferred to POE3 and POEG. 1: CLMA4 error is transferred to POE3 and POEG.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The HIZCTRLEN register is used to transfer CLMA error signals to POE3 and POEG.

7.3.12 PLL2MON : PLL2 Monitor Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x90

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2MON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PLL2MON	PLL2 Lock State Monitor This bit monitors whether the PLL2 is locked or not. 0: PLL2 is not locked 1: PLL2 is locked	R
31:1	—	These bits are read as 0.	R

The PLL2MON register is used to monitor the locking state of PLL2.

7.3.13 PLL2EN : PLL2 Enable Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0xA0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PLL2EN	PLL2 Enable 0: PLL2 is in standby state 1: PLL2 is released from standby state	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The PLL2EN register is used to control PLL2 standby state.

7.3.14 PLL2_SSC_CTR : PLL2 SSC Control Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0xAC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PLL2MFR								—	—	PLL2MRR					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2SSCEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PLL2SSCEN	PLL2 SSC Enable 0: SSC is disabled 1: SSC is enabled	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
21:16	PLL2MRR	PLL2 SSC Modulation Rate	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	PLL2MFR	PLL2 SSC Modulation Frequency	R/W

The PLL2_SSC_CTR register is used to control SSC function of PLL2. Before changing this register, make sure that PLL2 is in standby state by PLL2EN register.

7.3.15 PLL3MON : PLL3 Monitor Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0xB0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL3MON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PLL3MON	PLL3 Lock State Monitor This bit monitors whether the PLL3 is locked or not. 0: PLL3 is not locked 1: PLL3 is locked	R

Bit	Symbol	Function	R/W
31:1	—	These bits are read as 0.	R

The PLL3MON register is used to monitor the locking state of PLL3.

7.3.16 PLL3EN : PLL3 Enable Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0xC0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL3EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PLL3EN	PLL3 Enable 0: PLL3 is in standby state 1: PLL3 is released from standby state	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The PLL3EN register is used to control PLL3 standby state.

7.3.17 PLL3_VCO_CTR0 : PLL3 VCO Control Register 0

Base address: SYSC_S = 0x8128_0000

Offset address: 0xC4

Bit position:	31										21						16									9						0
Bit field:	—	—	—	—	—	—	—	—	—	—	PLL3P[5:0]						—	—	—	—	—	—	PLL3M[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0

Bit	Symbol	Function	R/W
9:0	PLL3M[9:0]	PLL3 M-Divider Setting	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
21:16	PLL3P[5:0]	PLL3 P-Divider Setting	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

The PLL3_VCO_CTR0 register is used to control divider of PLL3. Before changing this register, make sure that PLL3 is in standby state by PLL3EN register.

7.3.18 PLL3_VCO_CTR1 : PLL3 VCO Control Register 1

Base address: SYSC_S = 0x8128_0000

Offset address: 0xC8

Bit position:	31															16													2	0
Bit field:	PLL3K[15:0]																—	—	—	—	—	—	—	—	—	—	—	—	PLL3S[2:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
2:0	PLL3S[2:0]	PLL3 S-Divider Setting	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
31:16	PLL3K[15:0]	PLL3 Delta-Sigma Modulator Setting	R/W

The PLL3_VCO_CTR1 register is used to control divider of PLL3. Before changing this register, make sure that PLL3 is in standby state by PLL3EN register.

7.3.19 PLL4MON : PLL4 Monitor Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0xD0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL4MON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PLL4MON	PLL4 Lock State Monitor This bit monitors whether the PLL4 is locked or not. 0: PLL4 is not locked 1: PLL4 is locked	R
31:1	—	These bits are read as 0.	R

The PLL4MON register is used to monitor the locking state of PLL4.

7.4 Selecting Input to Main Clock Oscillator

The clock signal is supplied to the main clock oscillator by connecting a resonator or by the input of an external clock signal to EXTCLKIN. The clock input mode is set based on the status of the XTAL, EXTAL, EXTCLKIN, and XTALSEL pins. For details about the settings, see Table 7.6.

Table 7.6 Clock input mode selection

Clock input mode	XTAL	EXTAL	EXTCLKIN	XTALSEL
Resonator connection	Resonator connected	Resonator connected	VSS via a resistor	VDD33 via a resistor
External clock input	Open	VSS via a resistor	Clock input	VSS via a resistor

7.4.1 Connecting a Crystal Resonator

Figure 7.2 shows an example of connecting a crystal resonator. Make sure to connect EXTCLKIN to VSS via a resister and XTALSEL to VDD33 via a resistor.

A damping resistor Rd should be added, if necessary. Because the resistor values vary depending on the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer.

When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency of the resonator for the main clock oscillator described in Table 7.7.

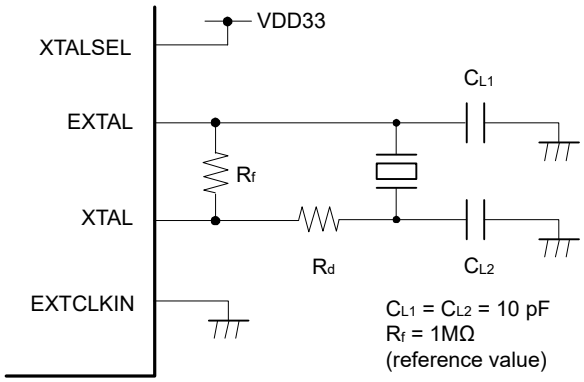


Figure 7.2 Example of crystal resonator connection

Table 7.7 Damping resistance (reference values)

Frequency (MHz)	25
Rd (Ω)	0

Figure 7.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in Table 7.8.

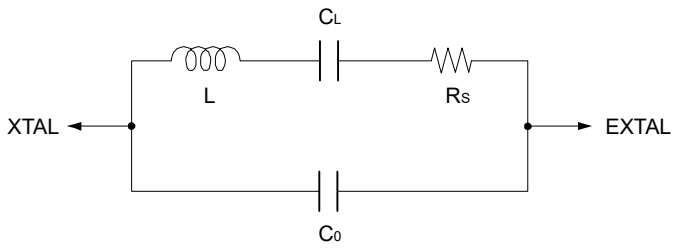


Figure 7.3 Equivalent circuit of crystal resonator

Table 7.8 Crystal resonator characteristics (reference values)

Frequency (MHz)	25
RS_{max} (Ω)	150

7.4.2 External Clock Input

Figure 7.4 shows an example connection of external clock input. Connect EXTAL to VSS via a resistor and leave XTAL open. Additionally, connect XTALSEL to VSS via a resistor.

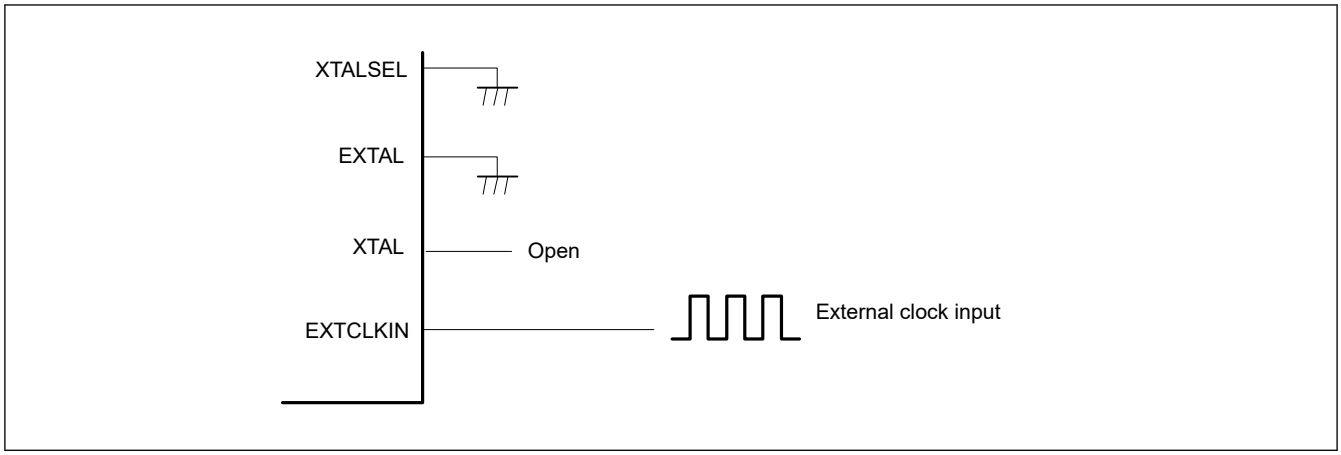


Figure 7.4 Example of external clock connection

7.5 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator. The LSI incorporates five PLL circuits (PLL0 to PLL4). In the initial state after power on, PLL1 and PLL4 are in normal mode, and PLL0, PLL2, and PLL3 are in standby mode. Start the PLL from standby state if related peripherals are used. Note that PLL0 and/or PLL2 are changed to normal mode by boot ROM automatically depending on the selected boot mode. SSC (Spread Spectrum Clock) function is available for PLL0 and PLL2 to reduce the EMI. Only down spread is supported. PLL output clock frequency must be set for PLL3. Table 7.10 shows the PLL start flow. Table 7.11 and Table 7.12 show PLL settings and constraints.

Table 7.9 PLL feature

PLL	Target	Reference clock	Output clock	Initial state	SSC function
PLL0	Cortex-A55 CPU clock	25 MHz (Main clock oscillator)	1200 MHz	Standby*1	Configurable
PLL1	Cortex-R52 CPU, main bus R and peripheral clock	25 MHz (Main clock oscillator)	1000 MHz	Normal	Non-SSC fixed
PLL2	DDRSS and SDHI clock	25 MHz (Main clock oscillator)	800 MHz	Standby*2	Configurable
PLL3	LCDC clock	48 MHz (Dividing clock of PLL4)	Configurable	Standby	Non-SSC fixed
PLL4	Main bus A and peripheral clock	25 MHz (Main clock oscillator)	2400 MHz	Normal	Non-SSC fixed

Note 1. When Cortex-A55 is the second boot CPU, Boot ROM sets PLL0 to normal mode.
Note 2. When eSD boot mode or eMMC boot mode is selected, Boot ROM sets PLL2 to normal mode.

Table 7.10 PLL start flow

No.	Process	PLL0, PLL2	PLL3
1	Confirmation of standby mode	• Confirm PLL is in standby mode by PLLnEN register.	
2	SSC setting	• Set PLLnMFR, PLLnMRR and PLLnSSCEN bits in PLLn_SSC_CTR register. See Table 7.11.	— (not supported)
3	PLL output clock frequency setting	— (not supported)	• Set PLL3P and PLL3M bits in PLL3_VCO_CTR0 register and PLL3S and PLL3K in PLL3_VCO_CTR1 register. See Table 7.12.
4	Transition to normal mode	• Set 1 to PLLnEN bit in PLLnEN register. • Poll PLLnMON bit in PLLnMON register and confirm it is set to 1, which indicates PLL is locked.	
5	Clock source change of PLL clock domain	• Poll PMSELnMON bit in PMSEL register and confirm it is set to 1, which indicates clock source change is completed.	
6	Module stop release	• Release reset of the CPU and peripheral to be used by resetting MSTPCRx register.	

Table 7.11 PLL0 and PLL2 SSC setting and constraints

Parameter	Symbol	Formula	Constraint		Unit
			Min	Max	
Modulation frequency control	mfr (PLLnMFR[7:0])	—	0x0	0xFF	—
Modulation rate control	mrr (PLLnMRR[5:0])	—	0x1	0x3F	—
—	—	mfr × mrr	0x0	0x200	—
Modulation frequency	M_{MF}	25 MHz / mfr / 2^5	30	150	kHz
Modulation rate (pk-pk)	M_{MR}	mfr × mrr / m / 2^6 × 100 (m = 96 for PLL0, 64 for PLL2)	0	5	%

Table 7.12 PLL3 setting and constraints

Parameter	Symbol	Formula	Constraint		Unit
			Min	Max	
P-divider	p (PLL3P[5:0])	—	0x1	0x3F	—
M-divider	m (PLL3M[9:0])	—	0x040	0x3FF	—
S-divider	s (PLL3S[2:0])	—	0x0	0x6	—
Delta-Sigma Modulator	k (PLL3K[15:0])	—	0x0000	0x7FFF	—
Reference frequency	F_{FREF}	48 MHz / p	6	25	MHz
VCO frequency	F_{FVCO}	(m + k / 65536) × 48 MHz / p)	1600	3200	MHz
PLL output frequency	F_{FOUT}	((m + k / 65536) × (48 MHz / p)) / 2^s	25	430	MHz

7.6 Low-Speed On-Chip Oscillator (LOCO)

The LSI incorporates low-speed on-chip oscillator (LOCO). It is used as sampling clock of CLMA6 and alternative clock of main clock and PLL clock when oscillation abnormality is detected.

After a reset is released, the LOCO is disabled. To enable this function, set the LOCO stop bit in the Low-Speed On-Chip Oscillator Control Register (LOCOCR.LCSTP) to 1.

7.7 Oscillation Abnormality Detection Function for the Main Clock

7.7.1 Main Clock Oscillation Abnormality Detection and Operation after Detection

The LSI incorporates a clock monitor circuit (CLMA6) that enables the detection of abnormal oscillation by the main clock. After a reset is released, the oscillation abnormality detection function is disabled. To enable this function, set the clock monitor enable bit in CLMA Control Register 0 (CLMA6.CTL0.CLME) to 1. For details, see section 8, Clock Monitor Circuit (CLMA).

An oscillation abnormality interrupt can be generated when an oscillation abnormality is detected. Low-speed on-chip oscillator (LOCO) clock or PLL free run clock is supplied as the system clock source instead of the main clock and PLL clock. LOCO clock or PLL free run clock is selectable by the select alternative clock bit in the System Clock Control Register (SCKCR.CLMASEL). When PLL free run clock is selected, clock frequency is not guaranteed.

In addition, the MTU3 and GPT output can be forcedly driven to the high-impedance on the detection. For details, see:

- section 18, Multi-Function Timer Pulse Unit 3 (MTU3)
- section 19, Port Output Enable 3 (POE3)
- section 20, General PWM Timer (GPT)
- section 21, Port Output Enable for GPT (POEG)

Note: If CLMA6 detects abnormal oscillation, CLMA0 to CLMA4 also generate abnormal oscillation detected interrupt when both CLMA6 and CLMA0 to CLMA4 are enabled.

7.7.2 Main Clock Oscillation Abnormality Detection Interrupt

When abnormal oscillation is detected, an abnormal oscillation detected interrupt is generated by the CLMA6 circuit, and it is transferred to the Port Output Enable 3 (POE3) block and Port Output Enable for GPT (POEG) if transmission is enabled in HIZCTRLEN register. On accepting the notification that the oscillation is to be stopped, the POE3 sets the OSTST high-impedance flag in the Input Level Control/Status Register 6 (ICSR6.OSTSTF) to 1 and the POEG also sets the Oscillation Stop Detection flags (POEGGn.OSTPF (n = A to D)) in the POEG Group n Setting Registers to 1.

The abnormal oscillation detected interrupt is connected to the Interrupt Controller Unit (ICU) as one of the error events of peripheral error interrupt so that error interrupt output and error reset are usable. When error interrupt output is used, the interrupt by all error events is disabled in the initial state after release from the reset state. To use oscillation stop detection interrupt (error event), release mask of the event in PRIERR_RSTMSK0 register when the ICU detects an error event. For details, see section 12, Interrupt Controller (ICU).

Note: If an abnormality in oscillation of the main clock occurs, oscillation of the PLL0 to PLL4 clock source also becomes abnormal, which may result in incorrect operation of the CPU. The POE3 and POEG set the high-impedance flag regardless of the state of the CPU, and the MTU3 and GPT outputs are driven to the high-impedance state.

7.8 PLL Oscillation Abnormality Detection Function

The PLL oscillation abnormality detection function detects an abnormality in oscillation of the PLL0 to PLL4 by using the Clock Monitor Circuits (CLMA0 to CLMA4) to monitor the frequency. It then supplies the main clock instead of the PLL0 to PLL4 clock. For details about the CLMA, see section 8, Clock Monitor Circuit (CLMA).

When abnormal oscillation is detected, an abnormal oscillation detected interrupt is generated by the CLMA0 to CLMA4 circuit, and it is transferred to POE3 and POEG if transmission is enabled in HIZCTRLEN register. On accepting the notification that the oscillation is to be stopped, the POE3 sets the OSTST high-impedance flag in the Input Level Control/Status Register 6 (ICSR6.OSTSTF) to 1 and the POEG also sets the Oscillation Stop Detection flags (POEGGn.OSTPF (n = A to D)) in the POEG Group n Setting Registers to 1.

The abnormal oscillation detected interrupt is connected to the Interrupt Controller Unit (ICU) as one of the error events of peripheral error interrupt so that error interrupt output and error reset are usable. When error interrupt output is used, the interrupt by all error events is disabled in the initial state after release from the reset state. To use oscillation stop detection interrupt (error event), release mask of the event in PRIERR_RSTMSK0 register when the ICU detects an error event. For details, see section 12, Interrupt Controller (ICU) .

7.9 Low-Speed On-Chip Oscillator Oscillation Abnormality Detection Function

The clock monitor circuit (CLMA5) can be used to detect an abnormality in oscillation of the low-speed on-chip oscillator. For details about the CLMA, see section 8, Clock Monitor Circuit (CLMA).

7.10 Internal Clock

Clock sources of internal clock signals are the main clock, LOCO clock, PLL clock, and the external clock for JTAG. Multiple internal clocks listed in Table 7.13 are produced from these sources.

Table 7.13 Overview of Clock Generation Circuit Specifications (Internal Clock) (1 of 4)

Clock name	Description
Cortex-A55 Core0 clock (CA55C0CLK)	The Cortex-A55 Core0 clock (CA55C0CLK) is used as the operating clock of the Cortex-A55 Core0. The Cortex-A55 Core0 operating frequency is specified by the SCKCR2.CA55CORE0 bit. Frequency of Cortex-A55 Core0 is selectable from 600 MHz or 1200 MHz.
Cortex-A55 Core1 clock (CA55C1CLK)	The Cortex-A55 Core1 clock (CA55C1CLK) is used as the operating clock of the Cortex-A55 Core1. The Cortex-A55 Core1 operating frequency is specified by the SCKCR2.CA55CORE1 bit. Frequency of Cortex-A55 Core1 is selectable from 600 MHz or 1200 MHz.
Cortex-A55 Core2 clock (CA55C2CLK)	The Cortex-A55 Core2 clock (CA55C2CLK) is used as the operating clock of the Cortex-A55 Core2. The Cortex-A55 Core2 operating frequency is specified by the SCKCR2.CA55CORE2 bit. Frequency of Cortex-A55 Core2 is selectable from 600 MHz or 1200 MHz.
Cortex-A55 Core3 clock (CA55C3CLK)	The Cortex-A55 Core3 clock (CA55C3CLK) is used as the operating clock of the Cortex-A55 Core3. The Cortex-A55 Core3 operating frequency is specified by the SCKCR2.CA55CORE3 bit. Frequency of Cortex-A55 Core3 is selectable from 600 MHz or 1200 MHz.

Table 7.13 Overview of Clock Generation Circuit Specifications (Internal Clock) (2 of 4)

Clock name	Description
Cortex-A55 DSU clock (CA55SCLK)	The Cortex-A55 DSU clock (CA55SCLK) is used as the operating clock of the Cortex-A55 DSU. The Cortex-A55 DSU operating frequency is specified by the SCKCR2.CA55SCLK bit. Frequency of Cortex-A55 DSU is selectable from 500 MHz or 1000 MHz.
Cortex-R52 CPU0 clock (CR52C0CLK)	The Cortex-R52 CPU0 clock (CR52C0CLK) is used as the operating clock of the Cortex-R52 CPU0. The Cortex-R52 CPU0 operating frequency is specified by the SCKCR2.CR52CPU0 bits. Frequency of Cortex-R52 CPU0 is selectable from 500 MHz or 1000 MHz.
Cortex-R52 CPU1 clock (CR52C1CLK)	The Cortex-R52 CPU1 clock (CR52C1CLK) is used as the operating clock of the Cortex-R52 CPU1. The Cortex-R52 CPU1 operating frequency is specified by the SCKCR2.CR52CPU1 bits. Frequency of Cortex-R52 CPU1 is selectable from 500 MHz or 1000 MHz.
Peripheral module clock AH (PCLKAH)	The peripheral module clock AH (PCLKAH) is used as the operating clock and/or bus clock for high-speed peripheral modules such as PCIE, LCDC (clk_a), GMAC unit 1, 2 (ACLK), DDRSS (A0, A1, A4), and Main bus A. The PCLKAH frequency is fixed to 400 MHz, and cannot be user-specified.
Peripheral module clock AM (PCLKAM)	The peripheral module clock AM (PCLKAM) is used as the operating clock and/or bus clock for middle-speed peripheral modules such as USB, GMAC unit 1, 2 (HCLK), and SDHI. The PCLKAM frequency is fixed to 200 MHz, and cannot be user-specified.
Peripheral module clock AL (PCLKAL)	The peripheral module clock AL (PCLKAL) is used as the operating clock and/or bus clock for low-speed peripheral modules such as LCDC (clk_p). The PCLKAL frequency is fixed to 100 MHz, and cannot be user-specified.
SDHI clock (SDHI_clkhs)	The SDHI clock is used as the operating clock for SDHI module. The SDHI_clkhs frequency is fixed to 800 MHz, and cannot be user-specified.
DDR controller DFI clock (DFICLK)	The DDR controller clock (DFICLK) is used as the operating clock for DDR controller. The DFICLK frequency is fixed to 800 MHz, and cannot be user-specified.
LCDC clock (LCDC_clkd)	LCDC clock (LCDC_clkd) is used as the operating clock for LCDC. The LCDC_clkd frequency is adjustable by selecting PLL3 output clock frequency setting and external divider setting. PLL3_VCO_CTR0 and PLL3_VCO_CTR1 registers specify PLL3 output clock frequency and SCKCR3.LCDDIVSEL bits specify external divider outside PLL3.
Peripheral module clock H (PCLKH)	The peripheral module clock H (PCLKH) is used as the operating clock and/or bus clock for high-speed peripheral modules such as MTU3, GPT connected to LLPP bus, TFU, DSMIF, POE3, POEG (LLPP), ADC12 unit 0, 1 (bus), xSPI (bus), DMAC, MPU, GMAC unit 0 (ACLK), SHOSTIF (pclk, mhclk), ENCIFSS (bus), ELC, RSIP, System SRAM, DDRSS (R2, R3), MBXSEM, and Main bus R. The PCLKH frequency is fixed to 250 MHz, and cannot be user-specified.
Peripheral module clock M (PCLKM)	The peripheral module clock M (PCLKM) is used as the operating clock and/or bus clock for middle-speed peripheral modules such as CRC, SCI (bus), SPI (bus), GPT (Non Safety, Safety), CANFD (bus), ETHSW (bus), GMAC unit 0 (HCLK), OTP, Ethernet SS Reg, GPIO, CLMA, ADC12 unit 2 (bus), and ICU. The PCLKM frequency is fixed to 125 MHz, and cannot be user-specified.
Peripheral module clock L (PCLKL)	The peripheral module clock L (PCLKL) is used as the operating clock and/or bus clock for low-speed peripheral modules such as POEG (Non Safety, Safety), IIC, DOC, CMT, CMTW, WDT, RTC, TSU, DDRSS (PHY_Reg), and ADC12 (conversion). The PCLKL frequency is fixed to 62.5 MHz, and cannot be user-specified.
Peripheral module clock for GPT in LLPP (PCLKGPTL)	The peripheral module clock for GPT in LLPP (PCLKGPTL) is used as the operational clock for GPT in LLPP. Frequency of PCLKGPTL is fixed to 500 MHz, and cannot be user-specified.
Peripheral module clock for DSMIFn (PCLKDSMIFn) (n = 0 to 9)	The peripheral module clock for DSMIF (PCLKDSMIFn) is used as the operational clock for DSMIFn. The PCLKDSMIFn frequency is specified by the DSMIFn.DSCCSCR.CLKSEL0 bit. It is selectable from 250 MHz or 400 MHz.
Peripheral module clock for SCI / SCIE (PCLKSCIn / PCLKSCIEm) (n = 0 to 5, m = 0 to 11)	The peripheral module clock for SCIn/SCIEm (PCLKSCIn/PCLKSCIEm) is used as the operating clock for SCIn/SCIEm. The PCLKSCIn/PCLKSCIEm frequency is specified by the combination of SCKCR3.SCInASYNCSEL (n = 0 to 4) or SCKCR2.SCInASYNCSEL (n = 5) or SCKCR4.SCIEmASYNCSEL (m = 0 to 11) bits and the SCIn/SCIEm.CCR3.BPEN bit. SCIn/SCIEm.CCR3.BPEN bit specifies which clock is used, synchronous clock (1) or asynchronous clock (0). When the bit is set to 1 (synchronous clock), the frequency of PCLKSCIn/PCLKSCIEm is the same as PCLKM (125 MHz). When the bit is set to 0 (asynchronous clock), the frequency of PCKLSCIn/PCLKSCIEm is the clock frequency selected by SCKCR3.SCInASYNCSEL (n = 0 to 4) or SCKCR2.SCInASYNCSEL (n = 5) or SCKCR4.SCIEmASYNCSEL (m = 0 to 11) bits. It is selectable from 75, 80, 96, or 100 MHz. Setting is independent for each channel.

Table 7.13 Overview of Clock Generation Circuit Specifications (Internal Clock) (3 of 4)

Clock name	Description
Peripheral module clock for SPI (PCLKSPIn) (n = 0 to 3)	The peripheral module clock for SPIn (PCLKSPIn) is used as the operating clock for SPIn. The PCLKSPIn frequency is specified by combination of the SCKCR3.SPInASYNCSEL (n = 0 to 2) or SCKCR2.SPInASYNCSEL (n = 3) bits and the SPIn.SPCR.BPEN bit. SPIn.SPCR.BPEN bit specifies which clock is used, synchronous clock (1) or asynchronous clock (0). When the bit is set to 1 (synchronous clock), the frequency of PCLKSPIn is the same as PCLKM (125 MHz). When the bit is set to 0 (asynchronous clock), the frequency of PCKLSPIn is the clock frequency selected by SCKCR3.SPInASYNCSEL (n = 0 to 2) or SCKCR2.SPInASYNCSEL (n = 3) bits. It is selectable from 75, 80, 96, or 100 MHz. Setting is independent for each channel.
Peripheral module clock for CANFD (PCLKCAN)	The peripheral module clock for CANFD (PCLKCAN) is used as the serial clock for CANFD. The PCLKCAN frequency is specified by the SCKCR.FSELCANFD bit. It is selectable from 40 MHz or 80 MHz.
Peripheral module clock for SHOSTIF (PCLKSHOST)	The peripheral module clock for SHOSTIF (PCLKSHOST) is used as the operating clock for SHOSTIF. The PCLKSHOST frequency is fixed to 400 MHz, and cannot be user-specified.
External bus clock (BSC_CLK, CKIO)	The external bus clock (CKIO) is used as an operating clock. The CKIO operating frequency is selectable from PCLKH/2, PCLKH/3, PCLKH/4, PCLKH/5, PCLKH/6, PCLKH/7, or PCLKH/8 by the SCKCR.CKIO bits.
Peripheral module clock for RTC (PCLKRTC)	The peripheral module clock for RTC (PCLKRTC) is used as the operational clock for RTC. The PCLKRTC frequency is fixed to 195.3 kHz, and cannot be user-specified.
USB clock (USB_CLK)	The USB clock is used as the reference clock for USB PHY layer. The USB clock frequency is fixed to 50 MHz, and cannot be user-specified.
xSPIn serial clock (XSPI_CLKn) (n = 0, 1)	The xSPI serial clock for xSPIn (XSPI_CLKn) is used as the serial clock for xSPIn. Clock frequency is selectable by combination of the SCKCR.FSELXSPIn and DIVSELXSPIn bits.
Ethernet clock A (ETCLKA)	The Ethernet clock A (ETCLKA) is used as the operating clock for ETHSW. The ETCLKA frequency is fixed to 200 MHz, and cannot be user-specified.
Ethernet clock B (ETCLKB)	The Ethernet clock B (ETCLKB) is a clock for GMAC, ETHSW, and MII converter. The ETCLKB frequency is fixed to 125 MHz, and cannot be user-specified.
Ethernet clock C (ETCLKC)	The Ethernet clock C (ETCLKC) is a clock for ESC. The ETCLKC frequency is fixed to 100 MHz, and cannot be user-specified.
Ethernet clock D (ETCLKD)	The Ethernet clock D (ETCLKD) is a clock for MII Converter. The ETCLKC frequency is fixed to 50 MHz, and cannot be user-specified.
Ethernet clock E (ETCLKE)	The Ethernet clock E (ETCLKE) is a clock for ESC and MII converter. The ETCLKE frequency is fixed to 25 MHz, and cannot be user-specified.
CLMAn sampling clock (CLMAMCLKA) (n = 0, 1, 2, 4)	The CLMAn (n = 0, 1, 2, 4) sampling clocks (CLMAMCLKA) are for use with CLMA0, CLMA1, CLMA2, and CLMA4. The CLMAMCLKA frequency is fixed to 12.5 MHz, and cannot be user-specified.
CLMA3 sampling clock (CLMAMCLKB)	The CLMA3 sampling clock (CLMAMCLKB) is for use with CLMA3. The CLMAMCLKB frequency is fixed to 1.56 MHz, and cannot be user-specified.
CLMA5 sampling clock (CLMAMCLKC)	The CLMA5 sampling clock (CLMAMCLKC) is for use with CLMA5. The CLMAMCLKC frequency is fixed to 97.6 kHz, and cannot be user-specified.
CLMA0 monitor clock (CLMAPLCLK0)	The CLMA0 monitor clock (CLMAPLCLK0) is the monitoring clock for use with CLMA0. The CLMAPLCLK0 frequency is fixed to 75 MHz, and cannot be user-specified.
CLMA1 monitor clock (CLMAPLCLK1)	The CLMA1 monitor clock (CLMAPLCLK1) is the monitoring clock for use with CLMA1. The CLMAPLCLK1 frequency is fixed to 62.5 MHz, and cannot be user-specified.
CLMA2 monitor clock (CLMAPLCLK2)	The CLMA2 monitor clock (CLMAPLCLK2) is the monitoring clock for use with CLMA2. The CLMAPLCLK2 frequency is fixed to 50 MHz, and cannot be user-specified.
CLMA3 monitor clock (CLMAPLCLK3)	The CLMA3 monitor clock (CLMAPLCLK3) is the monitoring clock for use with CLMA3. The CLMAPLCLK3 frequency is selectable from 6.25 MHz to 107.5 MHz.
CLMA4 monitor clock (CLMAPLCLK4)	The CLMA4 monitor clock (CLMAPLCLK4) is the monitoring clock for use with CLMA4. The CLMAPLCLK4 frequency is fixed to 75 MHz, and cannot be user-specified.
CLMA5 monitor clock (CLMALCLK)	The CLMA5 monitor clock (CLMAPLCLK) is the monitoring clock for use with CLMA5. The CLMALCLK frequency is fixed to 1 MHz, and cannot be user-specified.
CLMA6 monitor clock (CLMAMCLK)	The CLMA6 monitor clock (CLMAMCLK) is the monitoring clock for use with CLMA6. The CLMAMCLK frequency is fixed to 12.5 MHz, and cannot be user-specified.
JTAG clock (JTAGTCK)	The JTAG clock (JTAGTCK) is the operating clock for JTAG. JTAGTCK is generated by the external clock for JTAG (TCK).

Table 7.13 Overview of Clock Generation Circuit Specifications (Internal Clock) (4 of 4)

Clock name	Description
Serial Host I/F clock (SHIFCLK)	The Serial Host I/F clock (SHIFCLK) is used as the interface clock for SHOSTIF. The SHIFCLK frequency is up to 40 MHz.
EnDat Clock (PCLKENDAT)	The EnDat clock (PCLKENDAT) is used as the operating clock for EnDat. The PCLKENDAT is fixed to 100 MHz, and cannot be user-specified.
BiSS-C Clock (PCLKBISS)	The BiSS-C clock (PCLKBISS) is used as the operating clock for BiSS-C. The PCLKBISS is fixed to 80 MHz, and cannot be user-specified.
HIPERFACE DSL Clock (PCLKHDSL)	The HIPERFACE DSL clock (PCLKHDSL) is used as the operating clock for HIPERFACE. The PCLKHDSL is fixed to 75 MHz, and cannot be user-specified.
A-format Clock (PCLKAFMT)	The A-format clock (PCLKAFMT) is used as the operating clock for A-format. The PCLKAFMT is fixed to 80 MHz, and cannot be user-specified.
ENCOUT clock (PCLKENCO)	The ENCOUT clock (PCLKENCO) is used as the operating clock for ENCOUT. The PCLKENCO frequency is specified by the SCKCR4.ENCOUTCLK bit. It is selectable from 20 MHz or 80 MHz.

7.11 Reference Clock to the External Ethernet PHY

The LSI provides reference clocks to the external Ethernet PHY.

Table 7.14 List of reference clock to the external Ethernet PHY

Clock name	Description
ETHn_REFCLK (n = 0 to 3)	25 MHz reference clock to the external Ethernet PHY. Clock source is selectable from main clock or frequency-dividing clock for PLL1 by the SCKCR.PHYSEL bit.
RMIIn_REFCLK (n = 0 to 3)	50 MHz reference clock to the external Ethernet PHY for RMII.

7.12 Usage Notes

7.12.1 Notes on Clock Generation Circuit

1. The operating frequency of xSPI interface, the external bus clock (CKIO), and CANFD serial clock that are supplied to the modules based on the SCKCR and SCKCR2 register settings varies before and after the frequency is changed. Do not change the clock frequency during access via the external interface. To start access via the external interface after the clock frequency is changed, confirm that the change to the frequency has been completed before you start access via the interface.
2. Do not change the clock frequency of the PCLKSPIn or PCLKSCIn/PCKJSCIEm while the SPIn (channels 0 to 3) or the SCIn/SCIEm (channels 0 to 5 for SCIn and channels 0 to 11 for SCIEm) is operating. To cancel the module-stop state after the clock frequency is changed, confirm that the change to the frequency is completed before you cancel the module-stop state.
3. In order to secure the processing after the clock frequency is changed, modify the pertinent clock control register to change the frequency, and then perform a dummy read of the register at least seven times. Then perform the subsequent processing.
4. In order to secure processing after release from the module-stop state, dummy read the same register at least seven times except RTC and LCDC after writing to initiate release from the module-stop state, and only then proceed with the subsequent processing. For RTC, dummy read the same register at least 300 times and for LCDC, at least 100 times.
5. When controlling the module-stop state of the modules that initiate DMA transfer or that otherwise operate in relation with other blocks, confirm that DMAC transfer initiated by such modules and other points of interlinked operation are completed before releasing modules from or placing modules in the module-stop state. For set products in which such target modules are not used, design of the control of the module-stop state is on the assumption that the clocks are initially stopped. Therefore, the absence of unexpected operations in states other than the specified is not guaranteed. For example, operation after releasing modules from or placing modules in the module-stop state while a module is running is not guaranteed. For details on the operation and stopping of individual modules, see the section on processing by the corresponding block.

7.12.2 Notes on Resonator

Since various resonator characteristics relate closely to the user's board design, adequate evaluation is required on the user side before use, referencing the resonator connection example shown in this section. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, the circuit constants should be determined in full consultation with the resonator manufacturer. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

7.12.3 Notes on Designing the Board

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 7.5. This prevents electromagnetic induction from interfering with correct oscillation.

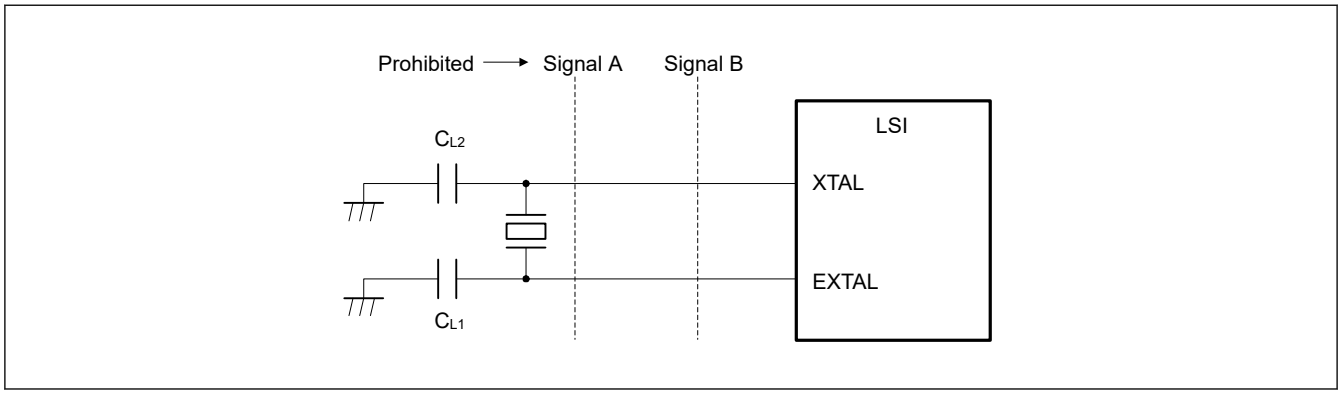


Figure 7.5 Notes on board design for oscillation circuit

8. Clock Monitor Circuit (CLMA)

8.1 Overview

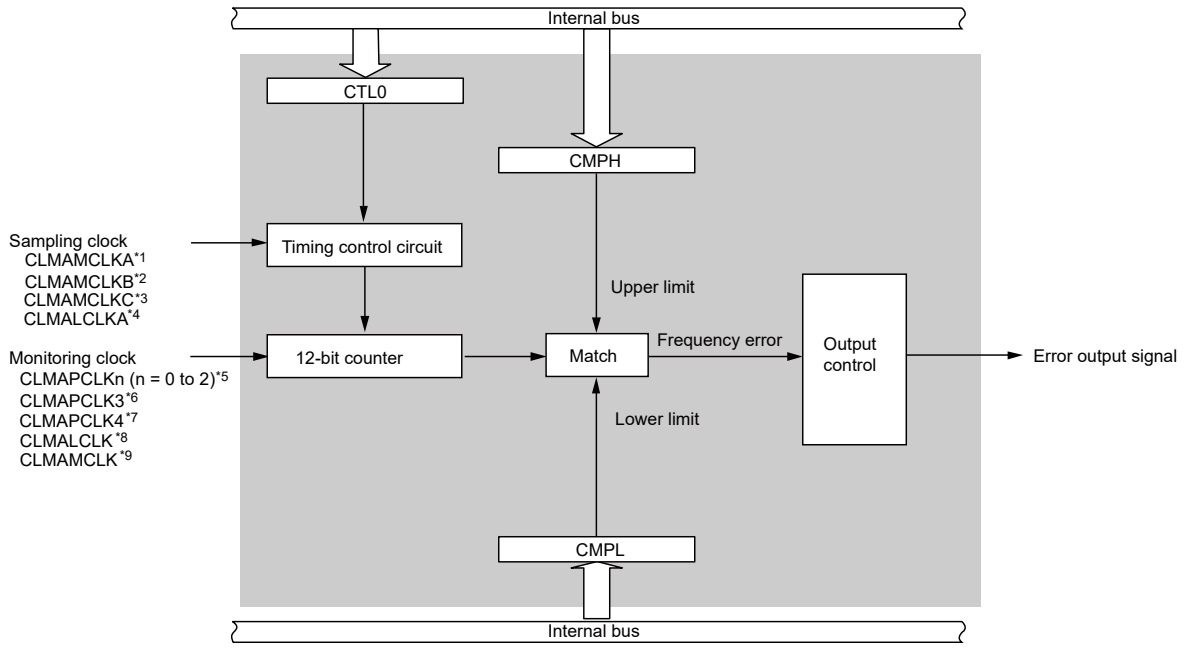
A clock monitor circuit, CLMAm (m = 0 to 6), can detect a frequency error in PLL0 to PLL4 output, low-speed on-chip oscillator (LOCO) output, or main clock. During 16 cycles of the sampling clock, it counts the rising edges of the monitoring clock (frequency-dividing clock from PLL0 to PLL4, LOCO, or main clock), and then compares the counter value with the compare register.

When CLMAm detects an error, error events are generated and sent to the interrupt controller (ICU).

Table 8.1 lists the specifications of CLMAm and Figure 8.1 shows a block diagram.

Table 8.1 CLMAm (m = 0 to 6) specifications

Parameter	Description
Monitoring clock	The following monitoring clock frequency errors can be detected: • PLL0 output clock divided by 16 (CLMAPLCLK0, which is to be supplied to CLMA0): 75 MHz • PLL1 output clock divided by 16 (CLMAPLCLK1, which is to be supplied to CLMA1): 62.5 MHz • PLL2 output clock divided by 16 (CLMAPLCLK2, which is to be supplied to CLMA2): 50 MHz • PLL3 output clock divided by 4 (CLMAPLCLK3, which is to be supplied to CLMA3): 6.25 to 107.5 MHz • PLL4 output clock divided by 32 (CLMAPLCLK4, which is to be supplied to CLMA4): 75 MHz • Low-speed on-chip oscillator (LOCO) output clock (CLMALCLK, which is to be supplied to CLMA5): 1 MHz • Clock which equals to the main clock frequency divided by 2 (CLMAMCLK, which is to be supplied to CLMA6): 12.5 MHz
Sampling clock	The following clock frequency errors are monitored as the sampling clock: • Clock which equals to the main clock frequency divided by 2 (CLMAMCLKA, which is to be supplied to CLMA0): 12.5 MHz • Clock which equals to the main clock frequency divided by 2 (CLMAMCLKA, which is to be supplied to CLMA1): 12.5 MHz • Clock which equals to the main clock frequency divided by 2 (CLMAMCLKA, which is to be supplied to CLMA2): 12.5 MHz • Clock which equals to the main clock frequency divided by 16 (CLMAMCLKA, which is to be supplied to CLMA3): 1.5625 MHz • Clock which equals to the main clock frequency divided by 2 (CLMAMCLKB, which is to be supplied to CLMA4): 12.5 MHz • Clock which equals to the main clock frequency divided by 256 (CLMAMCLKC, which is to be supplied to CLMA5): 97.7 kHz • Clock which equals to LOCO frequency (CLMALCLKA, which is to be supplied to CLMA6): 1 MHz
Error event	When CLMAm detects a frequency error, error events are generated and sent to the interrupt controller (ICU). • CLMA0 error detection interrupt • CLMA1 error detection interrupt • CLMA2 error detection interrupt • CLMA3 error detection interrupt • CLMA4 error detection interrupt • CLMA5 error detection interrupt • CLMA6 error detection interrupt



Note 1. Main clock frequency divided by 2 to be supplied to CLMA0, CLMA1, CLMA2, and CLMA4.
Note 2. Main clock frequency divided by 16 to be supplied to CLMA3.
Note 3. Main clock frequency divided by 256 to be supplied to CLMA5.
Note 4. Low-speed on-chip oscillator (LOCO) output clock to be supplied to CLMA6.
Note 5. PLL0, PLL1, and PLL2 output clock divided by 16 to be supplied to CLMA0, CLMA1, and CLMA2.
Note 6. PLL3 output clock divided by 4 to be supplied to CLMA3.
Note 7. PLL4 output clock divided by 32 to be supplied to CLMA4.
Note 8. Low-speed on-chip oscillator (LOCO) output clock to be supplied to CLMA5.
Note 9. Main clock frequency divided by 2 to be supplied to CLMA6.

Figure 8.1 Block diagram of CLMAm (m = 0 to 6)

8.2 Register Map

Table 8.2 CLMA register map

Address	Register symbol	Register name	Write protection
0x8128_8000 + 0x20 × m (m = 0 to 6)	CTL0	CLMA Control Register 0	—
0x8128_8008 + 0x20 × m (m = 0 to 6)	CMPL	CLMA Compare Register L	—
0x8128_800C + 0x20 × m (m = 0 to 6)	CMPH	CLMA Compare Register H	—
0x8128_8010 + 0x20 × m (m = 0 to 6)	PCMD	CLMA Command Register	—
0x8128_8014 + 0x20 × m (m = 0 to 6)	PROTSR	CLMA Protection Status Register	—

Table 8.3 CLMA related system control register

Unit	Module reset control register	Module stop control register	Slave access control register
0 (for PLL0)	—	MSTPCRG.MSTPCRG09	SLVACCCTL8.SYSCTRL_SL
1 (for PLL1)	—	MSTPCRG.MSTPCRG10	SLVACCCTL8.SYSCTRL_SL
2 (for PLL2)	—	MSTPCRG.MSTPCRG11	SLVACCCTL8.SYSCTRL_SL
3 (for PLL3)	—	MSTPCRG.MSTPCRG12	SLVACCCTL8.SYSCTRL_SL
4 (for PLL4)	—	MSTPCRG.MSTPCRG13	SLVACCCTL8.SYSCTRL_SL
5 (for LOCO)	—	MSTPCRG.MSTPCRG14	SLVACCCTL8.SYSCTRL_SL
6 (for Main clock oscillator)	—	MSTPCRG.MSTPCRG08	SLVACCCTL8.SYSCTRL_SL

8.3 Register Descriptions

8.3.1 CTL0 : CLMA Control Register 0

Base address: CLMAm = 0x8128_8000 + 0x0020 × m (m = 0 to 6)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CLME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLME	Clock Monitor m Enable (m = 0 to 6) Enables or disables operation of clock monitor circuit CLMAm. 0: Disables CLMAm operation 1: Enables CLMAm operation*1	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Once the CLME bit is set to 1, it cannot be cleared by any operation other than RES# pin reset, system software reset, and error reset.

The CTL0 registers control operation of clock monitor circuit CLMAm. Writing to these registers is protected by a specific command sequence. For details, see (1) Enabling operations.

8.3.2 CMPL : CLMA Compare Register L

Base address: CLMAm = 0x8128_8000 + 0x0020 × m (m = 0 to 6)

Offset address: 0x08

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CMPL[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
11:0	CMPL[11:0]	Clock Monitor m Compare L (m = 0 to 6) Specify the lower-limit threshold for the frequency domain.*1 ● For details, see (2) Method of calculating threshold values, the CMPL and CMPH registers. Recommended value: $f_{CLMATMON}$ (min) / $f_{CLMATSMP}$ (max) × 16 - 1 $f_{CLMATMON}$: Monitoring clock frequency $f_{CLMATSMP}$: Sampling clock frequency ● Minimum value: 0x0001	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. To set the CMPL registers, the following conditions must be met:
1. 1 ≤ CMPL
2. CMPL + 3 ≤ CMPH

The CMPL registers set the lower limit for comparing frequency domains. Values can be written to these registers when CTL0.CLME = 0. When CTL0.CLME = 1, writing to these registers has no effect.

8.3.3 CMPH : CLMA Compare Register H

Base address: CLMAm = 0x8128_8000 + 0x0020 × m (m = 0 to 6)

Offset address: 0x0C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CMPH[11:0]											
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
11:0	CMPH[11:0]	Clock Monitor m Compare H (m = 0 to 6) Specify the upper-limit threshold for the frequency domain.*1 ● For details, see (2) Method of calculating threshold values, the CMPL and CMPH registers. Recommended value: $f_{CLMATMON}$ (max) / $f_{CLMATSMP}$ (min) × 16 + 1 $f_{CLMATMON}$: Monitoring clock frequency $f_{CLMATSMP}$: Sampling clock frequency ● Minimum value: CMPL + 0x0003	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. To set the CMPH registers, the following conditions must be met:
- 1 ≤ CMPL
- CMPL + 3 ≤ CMPH

The CMPH registers set the upper limit for comparing frequency domains. Values can be written to these registers when CTL0.CLME = 0. When CTL0.CLME = 1, writing to these registers has no effect.

8.3.4 PCMD : CLMA Command Register

Base address: CLMAm = 0x8128_8000 + 0x0020 × m (m = 0 to 6)

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	n/a	CLMAm Protect Key Code (m = 0 to 6) Write a specific command sequence.	W

The PCMD registers control writing to the protected registers.

For details, see (1) Enabling operations.

8.3.5 PROTSR : CLMA Protection Status Register

Base address: CLMAm = 0x8128_8000 + 0x0020 × m (m = 0 to 6)

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PRERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRERR	CLMAm Error (m = 0 to 6) 0: No protection error occurred 1: A protection error occurred	R
7:1	—	These bits are read as 0.	R

The PROTSR registers indicate whether writing to the protected register is performed correctly. If writing is not performed correctly, a protection error occurs and the PROTSR.PRERR bit is set to 1.

For details, see (1) Enabling operations.

8.4 Operation

8.4.1 CLMAm Operation

(1) Enabling operations

Setting the CTL0.CLME bit to 1 starts monitoring the monitoring clock (PLL0 to PLL2 output divided by 16, PLL3 output divided by 4, PLL4 output divided by 32, LOCO output, and main clock divided by 2) by using clock monitor circuit CLMAm (m = 0 to 6) output. To write 1 in the CTL0.CLME bit, use the following command sequence:

1. Write 0xA5 to the PCMD register.
2. Writing to CTL0 is performed with the following sequence:
   (a) Write the target setting value (0x01).
   (b) Write the reversed value of the target (0xFE).
   (c) Write the target value (0x01) again.
3. Read CTL0.
   When the value of CTL0 shows 0x01, the operation of CLMAm is enabled. If it shows another value, check the value of PROTSR register.

When PROTSR = 0x01, the command sequence is not performed correctly. Execute the sequence again from step 1 to perform writing.

(2) Stopping operations

When a sampling clock stops due to register operation, the corresponding clock monitor circuit (CLMAm) is disabled automatically. After that, when the monitoring clock starts oscillating again and stabilizes, CLMAm resumes operation.

Table 8.4 Monitoring clock and sampling clock each CLMAm

Parameter		CLMA0	CLMA1	CLMA2	CLMA3	CLMA4	CLMA5	CLMA6
Detection target	Clock	PLL0 output clock	PLL1 output clock	PLL2 output clock	PLL3 output clock	PLL4 output clock	LOCO	Main clock
	Frequency	1200 MHz	1000 MHz	800 MHz	25 MHz to 430 MHz	2400 MHz	1 MHz	25 MHz
Monitoring clock	Clock	PLL0 output clock divided by 16	PLL1 output clock divided by 16	PLL2 output clock divided by 16	PLL3 output clock divided by 4	PLL4 output clock divided by 32	LOCO	Main clock divided by 2
	Frequency	75.0 MHz	62.5 MHz	50.0 MHz	6.25 MHz to 107.5 MHz	75.0 MHz	1.0 MHz	12.5 MHz
Sampling clock	Clock	Main clock divided by 2	Main clock divided by 2	Main clock divided by 2	Main clock divided by 16	Main clock divided by 2	Main clock divided by 256	LOCO
	Frequency	12.5 MHz	12.5 MHz	12.5 MHz	1.56 MHz	12.5 MHz	97.7 kHz	1.0 MHz

8.4.2 Detecting Error Clock Frequency

(1) Detection Method

1. During 16 cycles of the sampling clock (main clock frequency dividing clock and LOCO), CLMAm counts the rising edge of the monitoring clock (PLL0 to PLL2 output divided by 16, PLL3 output divided by 4, PLL4 output divided by 32, LOCO output, and main clock divided by 2), and compares the counter value with the set threshold (m = 0 to 6).
   - CMPL.CMPL[11:0] set the lower-limit threshold of the frequency domain.
   - CMPH.CMPH[11:0] set the upper-limit threshold of the frequency domain.
2. If the monitoring clock shows a value lower than the expected frequency, the counter value shows a value lower than the setting of CMPL.CMPL[11:0].
3. If the monitoring clock shows a value higher than the expected frequency, the counter value shows a value higher than the setting of CMPH.CMPH[11:0].

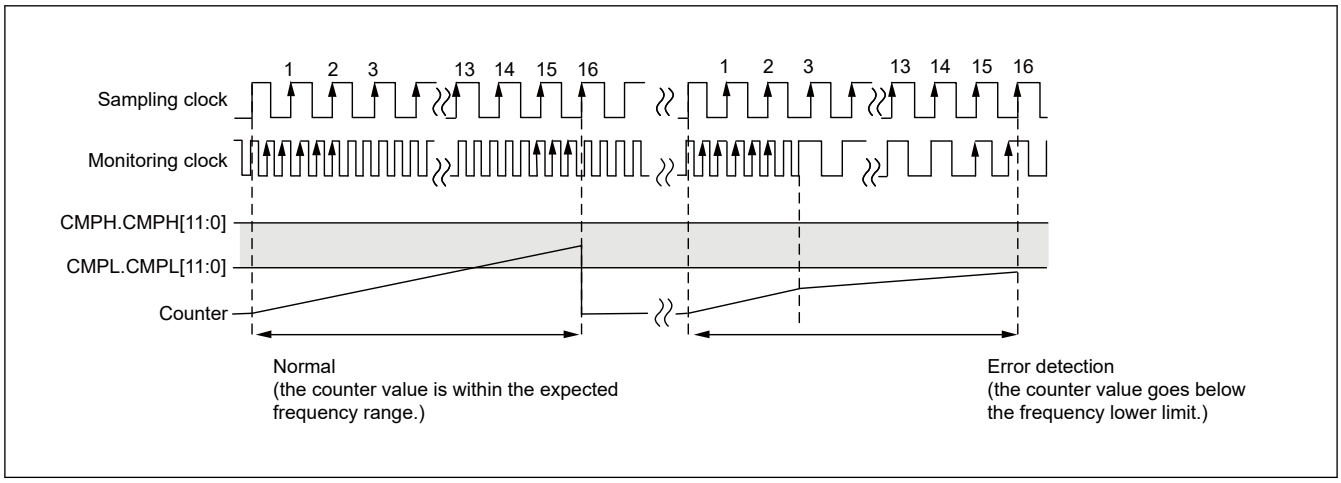


Figure 8.2 Example when the monitoring clock shows a value lower than the expected frequency

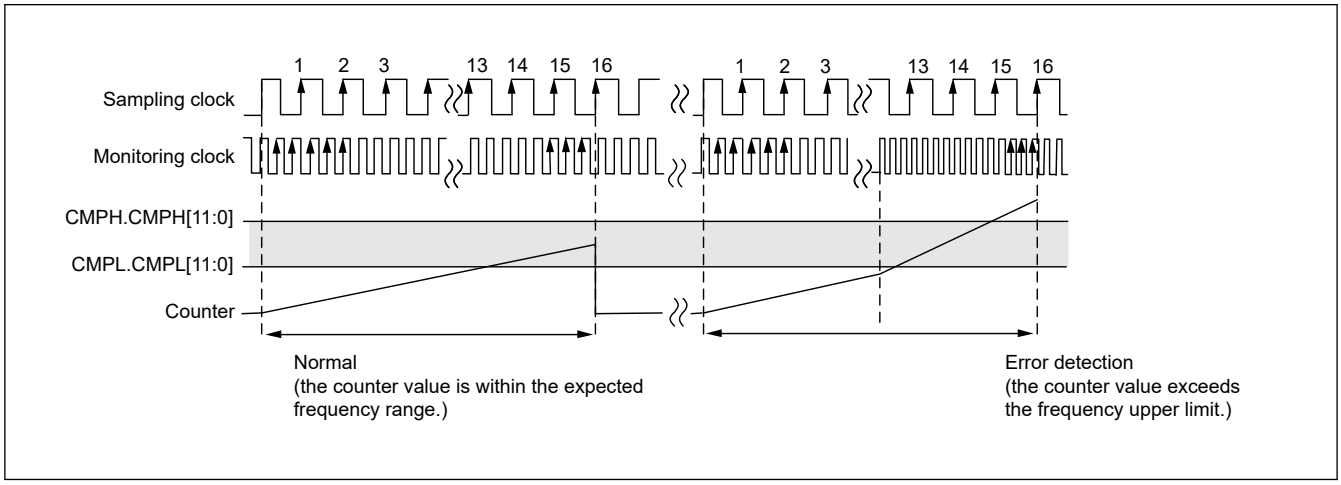


Figure 8.3 Example when the monitoring clock shows a value higher than the expected frequency

Note: No error is detected if the monitoring clock frequency changes within a sampling interval, and it eventually stays within a valid counter value. A monitoring clock error is detected after a sampling interval (16 cycles of the sampling clock).

(2) Method of calculating threshold values, the CMPL and CMPH registers

For compare registers CMPL and CMPH, set the minimum and maximum values of the number of cycles (the number of rising edges) for the monitoring clock which is assumed to be normal within 16 cycles of the sampling clock (main clock frequency dividing clock and LOCO).

$f_{CLMATSMP}$ indicates the sampling clock frequency, $f_{CLMATMON}$ indicates the monitoring clock frequency, and N indicates the number of cycles (rising edges) of the monitoring clock, which is expected within 16 cycles of the sampling clock.

$$N = \frac{f_{CLMATMON}}{f_{CLMATSMP}} \times 16$$

Consider the allowable frequency deviations for the monitoring clock and the sampling clock, use the following formula to calculate the threshold:

$$\text{Lower-limit threshold} = N_{min} = \frac{f_{CLMATMON(min)}}{f_{CLMATSMP(max)}} \times 16 - 1$$

$$\text{Upper-limit threshold} = N_{max} = \frac{f_{CLMATMON(max)}}{f_{CLMATSMP(min)}} \times 16 + 1$$

Note: Set the threshold within the following range:
CMPL ≥ 0x0001
CMPH ≥ CMPL + 0x0003

(3) Example: For CLMA0

For example, when the sampling clock equals to the main clock frequency divided by 2, $f_{CLMATSMP}$ = 12.5 MHz (±5%), or the monitoring clock equals to PLL0 output divided by 16, $f_{CLMATMON}$ = 75 MHz (±5%), the recommended threshold is calculated as follows:

$$N_{min} = \frac{f_{CLMATMON(min)}}{f_{CLMATSMP(max)}} = \frac{71.25}{13.125} \times 16 - 1 = 85.86$$

$$CMPL = 86 = 0x0056$$

$$N_{max} = \frac{f_{CLMATMON(max)}}{f_{CLMATSMP(min)}} = \frac{78.75}{11.875} \times 16 + 1 = 107.11$$

$$CMPH = 107 = 0x006B$$

8.4.3 Detecting Error Clock Frequency

When the monitoring clock frequency (PLL0 to PLL2 output divided by 16, PLL3 output divided by 4, PLL4 output divided by 32, LOCO output, and main clock divided by 2) is higher than the upper-limit threshold or lower than the lower-limit threshold, any of the following signals is sent to the interrupt controller (ICU) as error event:

- CLMA0 error detection interrupt
- CLMA1 error detection interrupt
- CLMA2 error detection interrupt
- CLMA3 error detection interrupt
- CLMA4 error detection interrupt
- CLMA5 error detection interrupt
- CLMA6 error detection interrupt

Error signal can be cleared by RES# pin reset, System software reset, or Error reset only.

8.5 Notes on Using CLMAm

- Do not use a clock for which CLMAm detected an error. If you use the clock, operation of the device is not guaranteed.
- No multiple clocks but only single monitoring clock error is assumed at the same time.
- Abnormality of the main clock oscillator may cause not only CLMA6 error but also other CLMA errors because it is also used as sampling clock. Abnormality of the LOCO may cause not only CLMA5 but also other CLMA errors.

- In case of higher frequency than expectation, the product may not work correctly due to outside of the operation range.
- Glitch may be generated when switching to the alternative clock in case main oscillator clock frequency is lower than one-eight of the LOCO frequency.
- In case monitoring clock stops instantly, abnormality may not be detected.
- The clock is switched to the alternative clock when the clock abnormality is detected as below table.

CLMAm error			Alternative clock			
CLMA6 (MOSC) error	CLMA5 (LOCO) error	CLMAn (PLLn) error (n = 0 to 4)	MOSC clock	PLLn output clock	Other PLL than PLLn output clock	Description
0	0	0	MOSC	PLLn	Other PLL than PLLn	Normal operation
1	0	X	LOCO	LOCO or PLLn	LOCO or Other PLL than PLLn	MOSC error*1 *2 *3
0	1	0	MOSC	PLLn	Other PLL than PLLn	LOCO error
1	1	X	LOCO	LOCO or PLLn	LOCO or Other PLL than PLLn	Either MOSC or LOCO error*1 *2 *3
0	0	1	MOSC	MOSC	Other PLL than PLLn	PLLn error
Other than the above			Indefinite			Indefinite

Note: MOSC is main clock oscillator. It is assumed that all CLMAm are enabled.
Note 1. Set LOCOCR.LCSTP to 1 when the clock is switched to LOCO in case of detecting abnormality since LOCO is stopped at the reset value. Be careful that if the reset is asserted after switching to the LOCO, this LSI may not restart because LOCO is disabled at the reset value in case clock abnormality originates from the hard error.
Note 2. PLL output clocks are selected by SCKCR.CLMASEL.
Note 3. When PLL is selected as alternative clock, PLL output clock is free run and the clock frequency is not guaranteed.

9. Low-Power Consumption Function

9.1 Overview

This LSI has several functions for reducing power consumption, including the Cortex®-A55 and Cortex®-R52 standby function and the module-stop function that stops functions independently for each peripheral module. Power consumption using clock control such as CKIO output control is also possible.

Table 9.1 lists the specifications of low-power consumption functions. Table 9.2 describes how to stop each peripheral module and exit the stop state.

Table 9.1 Specifications of low-power consumption function

Parameter	Specifications
Low-power consumption mode	Standby mode (Cortex-A55 and Cortex-R52)
Module-stop function	Functions can be stopped independently for each peripheral module
CKIO output control function	CKIO clock output or stop (held at the low level) can be selected

Table 9.2 Stopping peripheral modules and exiting module-stop state (1 of 2)

Module	How to stop operation and exit the stop state	Initial state*1
Cortex-A55 Cortex-R52	Standby mode condition: Execution of the Wait For Interrupt (WFI) instruction Condition for release from the Standby mode: interrupt Additionally, set the control register to enter or exit the module-stop state.	Second boot CPU: Operating Other CPU: Stop
Internal bus	Always operating	Operating
Tightly coupled memory (ATCM or BTCM)	Operate only during access	Operate only during access
System SRAM (SYSRAM)	Operate only during access	Operate only during access
Interrupt controller unit (ICU)	Operation is always enabled	Operating
Bus state controller (BSC)	Set the control register to enter or exit the module-stop state	Stop
Direct memory access controller (DMAC)	Operation is always enabled	Operating
General-purpose I/O ports	Operation is always enabled	Operating
Event link controller (ELC)	Operation is always enabled	Operating
Multi-function timer pulse unit 3 (MTU3)	Set the control register to enter or exit the module-stop state	Stop
General PWM timer (GPT)	Set the control register to enter or exit the module-stop state	Stop
Compare match timer (CMT)	Set the control register to enter or exit the module-stop state	Stop
Compare match timer W (CMTW)	Set the control register to enter or exit the module-stop state	Stop
Watchdog timer (WDT)	Operation is always enabled	Operating
Port output enable 3 (POE3)	Operation is always enabled	Operating
Port output enable for GPT (POEG)	Operation is always enabled	Operating
Real time clock (RTC)	Set the control register to enter or exit the module-stop state	Stop
Ethernet MAC (GMAC)	Set the control register to stop the module or exit the module-stop state	Stop
Ethernet switch (ETHSW)	Set the control register to stop the module or exit the module-stop state	Stop
EtherCAT slave controller (ESC)	Set the control register to stop the module or exit the module-stop state	Stop

Table 9.2 Stopping peripheral modules and exiting module-stop state (2 of 2)

Module	How to stop operation and exit the stop state	Initial state*1
USB 2.0 HS host/function module	Set the control register to stop the module or exit the module-stop state	Stop*2
Serial communication interface (SCI)	Set the control register to enter or exit the module-stop state	Stop*2
I2C bus interface (IIC)	Set the control register to enter or exit the module-stop state	Stop
CANFD interface (CANFD)	Set the control register to enter or exit the module-stop state	Stop
Serial peripheral interface (SPI)	Set the control register to enter or exit the module-stop state	Stop
Expanded serial peripheral interface (xSPI)	Set the control register to enter or exit the module-stop state	Stop*2
ΔΣ interface (DSMIF)	Set the control register to enter or exit the module-stop state	Stop
Trigonometric function unit (TFU)	Set the control register to enter or exit the module-stop state	Stop
12-bit A/D converter (ADC12)	Set the control register to enter or exit the module-stop state	Stop
Temperature sensor unit (TSU)	Set the control register to enter or exit the module-stop state	Stop
CRC operation unit (CRC)	Set the control register to enter or exit the module-stop state	Stop
Clock monitor circuit (CLMA)	Set the control register to enter or exit the module-stop state	Stop
Data operation circuit (DOC)	Set the control register to enter or exit the module-stop state	Stop
Serial host interface (SHOSTIF)	Set the control register to enter or exit the module-stop state	Stop
EnDat 2.2 (ENDAT)	Set the control register to enter or exit the module-stop state	Stop
BiSS-C (BISS)	Set the control register to enter or exit the module-stop state	Stop
HIPERFACE DSL (HDSL)	Set the control register to enter or exit the module-stop state	Stop
A-format (AFMT)	Set the control register to enter or exit the module-stop state	Stop
ENCOUT	Set the control register to enter or exit the module-stop state	Stop
LCD Controller (LCDC)	Set the control register to enter or exit the module-stop state	Stop
SD/eMMC Host controller (SD)	Set the control register to enter or exit the module-stop state	Stop*2
PCI Express Gen3 (PCIE)	Set the control register to enter or exit the module-stop state	Stop
LPDDR4 SDRAM Subsystem (DDRSS)	Set the control register to enter or exit the module-stop state	Stop

Note 1. Each module is returned to the initial state by means of the RES# pin reset, error reset, or system software reset.
Note 2. The state of the module is Operating if its boot mode is selected.

9.2 Register Map

Table 9.3 Low-Power Consumption Function register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x8028_0300	MSTPCRA	Module Stop Control Register A	PRCRN.PRC1
0x8028_0304	MSTPCRB	Module Stop Control Register B	PRCRN.PRC1
0x8028_0308	MSTPCRC	Module Stop Control Register C	PRCRN.PRC1
0x8028_030C	MSTPCRD	Module Stop Control Register D	PRCRN.PRC1
0x8028_0310	MSTPCRE	Module Stop Control Register E	PRCRN.PRC1
0x8128_0318	MSTPCRG	Module Stop Control Register G	PRCRS.PRC1
0x8128_0320	MSTPCRI	Module Stop Control Register I	PRCRS.PRC1
0x8028_0324	MSTPCRJ	Module Stop Control Register J	PRCRN.PRC1
0x8028_0328	MSTPCRK	Module Stop Control Register K	PRCRN.PRC1
0x8028_032C	MSTPCRL	Module Stop Control Register L	PRCRN.PRC1

Table 9.3 Low-Power Consumption Function register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x8028_0330	MSTPCRM	Module Stop Control Register M	PRCRN.PRC1
0x8128_0334	MSTPCRN	Module Stop Control Register N	PRCRS.PRC1

Table 9.4 Low Power Consumption Function related system control register

Target	Module Reset- Control Register	Module Stop Control Register	Slave Access Control Register
All the registers	—	—	SLVACCCTL8.SYSCTRL_SL

9.3 Register Descriptions

The registers are applicable to the register write protection function. To write to the registers, specify bit 1 or 3 of the Protect Register (PRCRN and PRCRS) to cancel the write protection. For details, see section 11, Register Write Protection Function.

9.3.1 MSTPCRA : Module Stop Control Register A

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x0300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	MSTPCRA27	MSTPCRA26	MSTPCRA25	MSTPCRA24	MSTPCRA23	MSTPCRA22	MSTPCRA21	MSTPCRA20	MSTPCRA19	MSTPCRA18	MSTPCRA17	MSTPCRA16
Value after reset:	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MSTPCRA12	MSTPCRA11	MSTPCRA10	MSTPCRA09	MSTPCRA08	—	—	MSTPCRA05	MSTPCRA04	—	—	—	MSTPCRA00
Value after reset:	0	0	0	1	1	1	1	1*1	0	0	1*1	1*1	0	0	0	1

Bit	Symbol	Function	R/W
0	MSTPCRA00	BSC Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	MSTPCRA04	xSPI Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
5	MSTPCRA05	xSPI Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	MSTPCRA08	SCI Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
9	MSTPCRA09	SCI Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
10	MSTPCRA10	SCI Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
11	MSTPCRA11	SCI Unit 3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Function	R/W
12	MSTPCRA12	SCI Unit 4 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	MSTPCRA16	SCIE Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
17	MSTPCRA17	SCIE Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
18	MSTPCRA18	SCIE Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
19	MSTPCRA19	SCIE Unit 3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
20	MSTPCRA20	SCIE Unit 4 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
21	MSTPCRA21	SCIE Unit 5 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
22	MSTPCRA22	SCIE Unit 6 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
23	MSTPCRA23	SCIE Unit 7 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
24	MSTPCRA24	SCIE Unit 8 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
25	MSTPCRA25	SCIE Unit 9 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
26	MSTPCRA26	SCIE Unit 10 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
27	MSTPCRA27	SCIE Unit 11 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value changes depending on the operation mode setting.

The MSTPCRA register controls the module-stop state. This register is available in Non Safety region.

9.3.2 MSTPCRB : Module Stop Control Register B

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x0304

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	MSTPCRB06	MSTPCRB05	MSTPCRB04	—	—	MSTPCRB01	MSTPCRB00
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1

Bit	Symbol	Function	R/W
0	MSTPCRB00	IIC Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRB01	IIC Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	MSTPCRB04	SPI Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
5	MSTPCRB05	SPI Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
6	MSTPCRB06	SPI Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRB register controls the module-stop state. This register is available in Non Safety region.

9.3.3 MSTPCRC : Module Stop Control Register C

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x308

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	MSTPCRC25	MSTPCRC24	MSTPCRC23	MSTPCRC22	MSTPCRC21	MSTPCRC20	MSTPCRC19	MSTPCRC18	MSTPCRC17	MSTPCRC16
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	MSTPCRC07	MSTPCRC06	MSTPCRC05	—	—	MSTPCRC02	MSTPCRC01	MSTPCRC00
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1

Bit	Symbol	Function	R/W
0	MSTPCRC00	MTU3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRC01	GPT Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRC02	GPT Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	MSTPCRC05	TFU Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
6	MSTPCRC06	ADC12 Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7	MSTPCRC07	ADC12 Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	MSTPCRC16	GPT Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
17	MSTPCRC17	GPT Unit 3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
18	MSTPCRC18	GPT Unit 4 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
19	MSTPCRC19	GPT Unit 5 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
20	MSTPCRC20	GPT Unit 6 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
21	MSTPCRC21	GPT Unit 7 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
22	MSTPCRC22	GPT Unit 8 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
23	MSTPCRC23	GPT Unit 9 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
24	MSTPCRC24	TFU Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
25	MSTPCRC25	ADC12 Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRC register controls the module-stop state. This register is available in Non Safety region.

9.3.4 MSTPCRD : Module Stop Control Register D

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x30C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	MSTPCRD23	MSTPCRD22	MSTPCRD21	MSTPCRD20	MSTPCRD19	MSTPCRD18	MSTPCRD17	MSTPCRD16
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MSTPCRD11	MSTPCRD10	MSTPCRD09	MSTPCRD08	MSTPCRD07	MSTPCRD06	MSTPCRD05	MSTPCRD04	MSTPCRD03	MSTPCRD02	MSTPCRD01	MSTPCRD00
Value after reset:	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPCRD00	DSMIF Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRD01	DSMIF Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRD02	CMT Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3	MSTPCRD03	CMT Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
4	MSTPCRD04	CMT Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
5	MSTPCRD05	CMTW Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
6	MSTPCRD06	CMTW Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7	MSTPCRD07	TSU Module Stop 0: Release from the module-stop state 1: Transition to the module stop state is made	R/W
8	MSTPCRD08	DOC Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
9	MSTPCRD09	CRC Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
10	MSTPCRD10	CANFD Module Stop 0: Release from the module-stop state 1: Transition to the module stop state is made	R/W
11	MSTPCRD11	CKIO Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	MSTPCRD16	DSMIF Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
17	MSTPCRD17	DSMIF Unit 3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
18	MSTPCRD18	DSMIF Unit 4 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
19	MSTPCRD19	DSMIF Unit 5 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
20	MSTPCRD20	DSMIF Unit 6 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
21	MSTPCRD21	DSMIF Unit 7 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
22	MSTPCRD22	DSMIF Unit 8 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
23	MSTPCRD23	DSMIF Unit 9 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRD register controls the module-stop state. This register is available in Non Safety region.

9.3.5 MSTPCRE : Module Stop Control Register E

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x310

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPCRE17	MSTPCRE16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MSTPCRE08	—	—	—	—	MSTPCRE03	MSTPCRE02	MSTPCRE01	MSTPCRE00
Value after reset:	0	0	0	0	0	0	0	1*1	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPCRE00	GMAC Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRE01	ETHSW Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRE02	ESC Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Function	R/W
3	MSTPCRE03	Ethernet Subsystem Register Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	MSTPCRE08	USB Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	MSTPCRE16	GMAC Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
17	MSTPCRE17	GMAC Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value changes depending on the operation mode setting.

The MSTPCRE register controls the module-stop state. This register is available in Non Safety region.

9.3.6 MSTPCRG : Module Stop Control Register G

Base address: SYSC_S = 0x8128_0000

Offset address: 0x318

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MSTPCRG14	MSTPCRG13	MSTPCRG12	MSTPCRG11	MSTPCRG10	MSTPCRG09	MSTPCRG08	—	—	MSTPCRG05	MSTPCRG04	MSTPCRG03	MSTPCRG02	MSTPCRG01	MSTPCRG00
Value after reset:	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPCRG00	SCI Unit 5 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRG01	IIC Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRG02	SPI Unit 3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3	MSTPCRG03	GPT Unit 10 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
4	MSTPCRG04	CRC Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
5	MSTPCRG05	RTC Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	MSTPCRG08	CLMA6 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
9	MSTPCRG09	CLMA0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
10	MSTPCRG10	CLMA1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
11	MSTPCRG11	CLMA2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
12	MSTPCRG12	CLMA3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
13	MSTPCRG13	CLMA4 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
14	MSTPCRG14	CLMA5 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRG register controls the module-stop state. This register is available in Safety region.

9.3.7 MSTPCRI : Module Stop Control Register I

Base address: SYSC_S = 0x8128_0000

Offset address: 0x320

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPCRI01	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	MSTPCRI01	SHOSTIF Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRI register controls the module-stop state. This register is available in Safety region.

9.3.8 MSTPCRJ : Module Stop Control Register J

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x324

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTPCRJ31	MSTPCRJ30	MSTPCRJ29	MSTPCRJ28	MSTPCRJ27	MSTPCRJ26	MSTPCRJ25	MSTPCRJ24	MSTPCRJ23	MSTPCRJ22	MSTPCRJ21	MSTPCRJ20	MSTPCRJ19	MSTPCRJ18	MSTPCRJ17	MSTPCRJ16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSTPCRJ15	MSTPCRJ14	MSTPCRJ13	MSTPCRJ12	MSTPCRJ11	MSTPCRJ10	MSTPCRJ09	MSTPCRJ08	MSTPCRJ07	MSTPCRJ06	MSTPCRJ05	MSTPCRJ04	MSTPCRJ03	MSTPCRJ02	MSTPCRJ01	MSTPCRJ00
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPCRJ00	AFMT Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRJ01	HDSL Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRJ02	BISS Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3	MSTPCRJ03	ENDAT Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
4	MSTPCRJ04	AFMT Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
5	MSTPCRJ05	HDSL Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
6	MSTPCRJ06	BISS Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7	MSTPCRJ07	ENDAT Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
8	MSTPCRJ08	AFMT Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
9	MSTPCRJ09	HDSL Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
10	MSTPCRJ10	BISS Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
11	MSTPCRJ11	ENDAT Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
12	MSTPCRJ12	AFMT Unit 3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Function	R/W
13	MSTPCRJ13	HDSL Unit 3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
14	MSTPCRJ14	BISS Unit 3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
15	MSTPCRJ15	ENDAT Unit 3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
16	MSTPCRJ16	AFMT Unit 4 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
17	MSTPCRJ17	HDSL Unit 4 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
18	MSTPCRJ18	BISS Unit 4 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
19	MSTPCRJ19	ENDAT Unit 4 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
20	MSTPCRJ20	AFMT Unit 5 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
21	MSTPCRJ21	HDSL Unit 5 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
22	MSTPCRJ22	BISS Unit 5 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
23	MSTPCRJ23	ENDAT Unit 5 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
24	MSTPCRJ24	AFMT Unit 6 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
25	MSTPCRJ25	HDSL Unit 6 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
26	MSTPCRJ26	BISS Unit 6 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
27	MSTPCRJ27	ENDAT Unit 6 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
28	MSTPCRJ28	AFMT Unit 7 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
29	MSTPCRJ29	HDSL Unit 7 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
30	MSTPCRJ30	BISS Unit 7 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Function	R/W
31	MSTPCRJ31	ENDAT Unit 7 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

The MSTPCRJ register controls the module-stop state. This register is available in Non Safety region.

9.3.9 MSTPCRK : Module Stop Control Register K

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x328

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTPCRK31	MSTPCRK30	MSTPCRK29	MSTPCRK28	MSTPCRK27	MSTPCRK26	MSTPCRK25	MSTPCRK24	MSTPCRK23	MSTPCRK22	MSTPCRK21	MSTPCRK20	MSTPCRK19	MSTPCRK18	MSTPCRK17	MSTPCRK16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSTPCRK15	MSTPCRK14	MSTPCRK13	MSTPCRK12	MSTPCRK11	MSTPCRK10	MSTPCRK09	MSTPCRK08	MSTPCRK07	MSTPCRK06	MSTPCRK05	MSTPCRK04	MSTPCRK03	MSTPCRK02	MSTPCRK01	MSTPCRK00
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPCRK00	AFMT Unit 8 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRK01	HDSL Unit 8 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRK02	BISS Unit 8 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3	MSTPCRK03	ENDAT Unit 8 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
4	MSTPCRK04	AFMT Unit 9 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
5	MSTPCRK05	HDSL Unit 9 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
6	MSTPCRK06	BISS Unit 9 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7	MSTPCRK07	ENDAT Unit 9 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
8	MSTPCRK08	AFMT Unit 10 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
9	MSTPCRK09	HDSL Unit 10 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Function	R/W
10	MSTPCRK10	BISS Unit 10 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
11	MSTPCRK11	ENDAT Unit 10 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
12	MSTPCRK12	AFMT Unit 11 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
13	MSTPCRK13	HDSL Unit 11 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
14	MSTPCRK14	BISS Unit 11 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
15	MSTPCRK15	ENDAT Unit 11 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
16	MSTPCRK16	AFMT Unit 12 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
17	MSTPCRK17	HDSL Unit 12 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
18	MSTPCRK18	BISS Unit 12 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
19	MSTPCRK19	ENDAT Unit 12 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
20	MSTPCRK20	AFMT Unit 13 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
21	MSTPCRK21	HDSL Unit 13 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
22	MSTPCRK22	BISS Unit 13 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
23	MSTPCRK23	ENDAT Unit 13 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
24	MSTPCRK24	AFMT Unit 14 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
25	MSTPCRK25	HDSL Unit 14 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
26	MSTPCRK26	BISS Unit 14 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
27	MSTPCRK27	ENDAT Unit 14 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Function	R/W
28	MSTPCRK28	AFMT Unit 15 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
29	MSTPCRK29	HDSL Unit 15 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
30	MSTPCRK30	BISS Unit 15 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31	MSTPCRK31	ENDAT Unit 15 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

The MSTPCRK register controls the module-stop state. This register is available in Non Safety region.

9.3.10 MSTPCRL : Module Stop Control Register L

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x32C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPCRL00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	MSTPCRL00	ENCOUT Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRL register controls the module-stop state. This register is available in Non Safety region.

9.3.11 MSTPCRM : Module Stop Control Register M

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x330

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	MSTPCRM13	MSTPCRM12	—	—	—	MSTPCRM08	—	—	—	MSTPCRM04	—	—	—	MSTPCRM00
Value after reset:	0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	1

Bit	Symbol	Function	R/W
0	MSTPCRM00	DDRSS Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	MSTPCRM04	LCDC Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	MSTPCRM08	PCIE Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	MSTPCRM12	SDHI Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
13	MSTPCRM13	SDHI Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRM register controls the module-stop state. This register is available in Non Safety region.

9.3.12 MSTPCRN : Module Stop Control Register N

Base address: SYSC_S = 0x8128_0000

Offset address: 0x334

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	MSTPCRN05	MSTPCRN04	MSTPCRN03	MSTPCRN02	MSTPCRN01	MSTPCRN00
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	1	1*1	1	1*1

Bit	Symbol	Function	R/W
0	MSTPCRN00	Cortex-R52 CPU0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRN01	Cortex-R52 CPU1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRN02	Cortex-A55 Core0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3	MSTPCRN03	Cortex-A55 Core1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
4	MSTPCRN04	Cortex-A55 Core2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Function	R/W
5	MSTPCRN05	Cortex-A55 Core3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value changes depending on the operation mode setting.

The MSTPCRN register controls the module-stop state. This register is available in Safety region.

9.4 Operation

9.4.1 Module-Stop Function

The module-stop function can stop operation of each module of on-chip peripheral functions. When the MSTPCRmi bit (m = A to N, i = 31 to 00) in the MSTPCRA to MSTPCRN registers is set to 1, the specified module stops operating and enters the module-stop state. Clearing the MSTPCRmi bit (m = A to N, i = 31 to 00) to 0 leads to release from the module-stop state. Follow the procedure given below when releasing any of the peripheral modules listed in Table 9.5 from the module-stop state. This procedure is not required for peripheral modules not listed in Table 9.5. However, release from the module-stop state should be in accord with the procedure for initialization described in the section for the given peripheral module.

Note that in case of AXI-type slave module, the access to the slave I/F needs to be stopped by slave stop function before changing module stop state. For details, see section 13.4.3. Slave Stop Function.

(1) Procedure

1. Clear the corresponding bit in the relevant Module Stop Control Register (MSTPCRm, m = A to N) then immediately dummy-read the MSTPCRm register once.
2. Dummy-read any register of the peripheral module which is being released from the module-stop state at least 7 times except RTC and LCDC. For RTC, at least 300 times and for LCDC, at least 100 times. After that, all registers of that peripheral module can be accessible.

Note: The MPU having set the access control attribute for the peripheral I/O register region to 'Device nGnRnE' or 'Device nGnRE' is a prerequisite for the procedure.

(2) Example code

```
volatile unsigned long dummy;           // Declared volatile to prevent optimization being applied

R_SYSC_NS->MSTPCRD_b.MSTPCRD06 = 0;     // Setting to release CMTW unit 1 from the module-stop state
dummy = R_SYSC_NS->MSTPCRD;             // Step 1: Dummy-read the MSTPCRm register.

dummy = R_CMTW1->CMWCOR;                // Step 2: Dummy-read any register of CMTW unit 1 at least 7 times.
dummy = R_CMTW1->CMWCOR;
dummy = R_CMTW1->CMWCOR;
dummy = R_CMTW1->CMWCOR;
dummy = R_CMTW1->CMWCOR;
dummy = R_CMTW1->CMWCOR;
dummy = R_CMTW1->CMWCOR;
R_CMTW1->CMWIOR = 0x0001U;              // The first setting for CMTW unit 1 (value is an example)
```

For details about the initial state after a release from the reset state, see Table 9.2.

Note: Directly after a module is set to the module-stop state, writing to the control register of that module might still be possible.

Table 9.5 Peripheral functions requiring the above procedure for release from the module-stop state (1 of 3)

Peripheral functions	Corresponding Module-Stop Control Register
BSC	MSTPCRA register, MSTPCRA00 bit
xSPI Unit 0, 1*1	MSTPCRA register, MSTPCRA04, MSTPCRA05 bit

Table 9.5 Peripheral functions requiring the above procedure for release from the module-stop state (2 of 3)

Peripheral functions	Corresponding Module-Stop Control Register
SCI Unit 0 to 4	MSTPCRA register, MSTPCRA08 to MSTPCRA12 bit
SCI Unit 5	MSTPCRG register, MSTPCRG00 bit
SCIE Unit 0 to 11	MSTPCRA register, MSTPCRA16 to MSTPCRA27
IIC Unit 0, 1	MSTPCRB register, MSTPCRB00, MSTPCRB01 bit
IIC Unit 2	MSTPCRG register, MSTPCRG01 bit
SPI Unit 0 to 2	MSTPCRB register, MSTPCRB04 to MSTPCRB06 bit
SPI Unit 3	MSTPCRG register, MSTPCRG02 bit
MTU3	MSTPCRC register, MSTPCRC00 bit
GPT Unit 0 to 9	MSTPCRC register, MSTPCRC01, MSTPCRC02, MSTPCRC16 to MSTPCRC23 bit
GPT Unit 10	MSTPCRG register, MSTPCRG03 bit
TFU Unit 0, 1	MSTPCRC register, MSTPCRC05, MSTPCRC24 bit
ADC Unit 0 to 2	MSTPCRC register, MSTPCRC06, MSTPCRC07, MSTPCRC25 bit
DSMIF Unit 0 to 9	MSTPCRD register, MSTPCRD00, MSTPCRD01, MSTPCRD16 to MSTPCRD23 bit
CMT Unit 0 to 2	MSTPCRD register, MSTPCRD02 to MSTPCRD04 bit
CMTW Unit 0, 1	MSTPCRD register, MSTPCRD05, MSTPCRD06 bit
TSU	MSTPCRD register, MSTPCRD07 bit
DOC	MSTPCRD register, MSTPCRD08 bit
CRC Unit 0	MSTPCRD register, MSTPCRD09 bit
CRC Unit 1	MSTPCRG register, MSTPCRG04 bit
CANFD	MSTPCRD register, MSTPCRD10 bit
CKIO	MSTPCRD register, MSTPCRD11 bit
GMAC Unit 0 to 2	MSTPCRE register, MSTPCRE00, MSTPCRE16, MSTPCRE17 bit
ETHSW	MSTPCRE register, MSTPCRE01 bit
ESC	MSTPCRE register, MSTPCRE02 bit
Ethernet Subsystem register	MSTPCRE register, MSTPCRE03 bit
USB	MSTPCRE register, MSTPCRE08 bit
RTC	MSTPCRG register, MSTPCRG05 bit
CLMA0 to CLMA6	MSTPCRG register, MSTPCRG09 to MSTPCRG14, MSTPCRG08 bit
SHOSTIF	MSTPCRI register, MSTPCRI01 bit
AFMT Unit 0 to 7	MSTPCRJ register, MSTPCRJ00, MSTPCRJ04, MSTPCRJ08, MSTPCRJ12, MSTPCRJ16, MSTPCRJ20, MSTPCRJ24, MSTPCRJ28 bit
AFMT Unit 8 to 15	MSTPCRK register, MSTPCRK00, MSTPCRK04, MSTPCRK08, MSTPCRK12, MSTPCRK16, MSTPCRK20, MSTPCRK24, MSTPCRK28 bit
HDSL Unit 0 to 7	MSTPCRJ register, MSTPCRJ01, MSTPCRJ05, MSTPCRJ09, MSTPCRJ13, MSTPCRJ17, MSTPCRJ21, MSTPCRJ25, MSTPCRJ29 bit
HDSL Unit 8 to 15	MSTPCRK register, MSTPCRK01, MSTPCRK05, MSTPCRK09, MSTPCRK13, MSTPCRK17, MSTPCRK21, MSTPCRK25, MSTPCRK29 bit
BISC Unit 0 to 7	MSTPCRJ register, MSTPCRJ02, MSTPCRJ06, MSTPCRJ10, MSTPCRJ14, MSTPCRJ18, MSTPCRJ22, MSTPCRJ26, MSTPCRJ30 bit
BISS Unit 8 to 15	MSTPCRK register, MSTPCRK02, MSTPCRK06, MSTPCRK10, MSTPCRK14, MSTPCRK18, MSTPCRK22, MSTPCRK26, MSTPCRK30 bit
ENDAT Unit 0 to 7	MSTPCRJ register, MSTPCRJ03, MSTPCRJ07, MSTPCRJ11, MSTPCRJ15, MSTPCRJ19, MSTPCRJ23, MSTPCRJ27, MSTPCRJ31 bit
ENDAT Unit 8 to 15	MSTPCRK register, MSTPCRK03, MSTPCRK07, MSTPCRK11, MSTPCRK15, MSTPCRK19, MSTPCRK23, MSTPCRK27, MSTPCRK31 bit

Table 9.5 Peripheral functions requiring the above procedure for release from the module-stop state (3 of 3)

Peripheral functions	Corresponding Module-Stop Control Register
ENCOUT	MSTPCRL register, MSTPCRL00 bit
DDRSS*1	MSTPCRM register, MSTPCRM00 bit
LCDC	MSTPCRM register, MSTPCRM04 bit
PCIE*1	MSTPCRM register, MSTPCRM08 bit
SDHI Unit 0, 1*1	MSTPCRM register, MSTPCRM12, MSTPCRM13 bit
Cortex-R52 CPU0*1	MSTPCRN register, MSTPCRN00 bit
Cortex-R52 CPU1*1	MSTPCRN register, MSTPCRN01 bit
Cortex-A55 Core0 to Core3	MSTPCRN register, MSTPCRN02 to MSTPCRN05 bit

Note 1. Operation of the slave stop function is needed together. For details, see section 13.4.3. Slave Stop Function.

9.4.2 Cortex-A55/Cortex-R52 Standby Mode

9.4.2.1 Transition to Cortex-A55/Cortex-R52 Standby Mode

The Cortex-A55/Cortex-R52 enters Standby mode by execution of a WFI instruction. In Standby mode, the Cortex-A55/Cortex-R52 stops operating, thus reducing power consumption. For details, see the technical reference manual provided by Arm.

9.4.2.2 Release from Cortex-A55/Cortex-R52 Standby Mode

Release from the Cortex-A55/Cortex-R52 Standby mode is initiated by any interrupt, the RES# pin reset, an error reset, system software reset, or CPU software reset.

(1) Release triggered by an interrupt signal

Generation of an interrupt in a CPU triggers release from Standby mode or Sleep mode of the CPU and the interrupt exception handling starts. Release is triggered by a non-maskable interrupt or a maskable interrupt of which interrupt request is permitted by using the Interrupt Enable register.

(2) Release by a reset

After the RES# pin reset, error reset, or system software reset is cleared, the Cortex-R52 CPU0 starts the reset exception handling but the other CPUs are still in the reset state. After CPU software reset of target CPU is cleared and corresponding bit in MSTPCRN register is cleared, the corresponding CPU starts the reset exception handling. For details about resets, see section 6, Reset.

9.5 Usage Notes

9.5.1 I/O Port State

To reduce I/O power consumption, pin processing based on I/O control is required. For details, see section 17, I/O Ports.

9.5.2 On-Chip Peripheral Module Interrupts in Module-Stop State

Peripheral modules cannot interrupt in a module-stop state. Therefore, if the module-stop state is set during interrupt processing of the module or during DMA transfer by the DMAC, a CPU interrupt source or a DMAC startup source cannot be cleared. For this reason, disable the relevant interrupts before setting the module-stop state.

9.5.3 USB Low Power Consumption

For details about how to clear the USB module-stop state and how to enter the USB in low-power consumption mode, see section 31, USB 2.0 Host Module (USBHC) and section 32, USB 2.0 HS Function Module (USBf).

9.5.4 Low Power Consumption for Ethernet-Related Functions

Ethernet-related functions are stopped and reset in the initial state. To use the functions, specify the MSTPCRE.MSTPCRE00 to MSTPCRE03 bits for release from the module-stop state. After the dummy read, specify the MRCTLE.MRCTLE00 to MRCTLE06 bits for release from reset state.

Note: After release from the module-stop state, the module-stop state cannot be set again. Operation is not guaranteed after release from the module-stop state is initiated the second time. After the modules are returned to the initial state (stop state) by a reset, they can be released from the module-stop state again.

9.5.5 Write Protection Function

The Module Stop Control registers (MSTPCRA to MSTPCRN) can be write-protected. To write to the MSTPCRA to MSTPCRN registers, specify bit 1 of the Protect Register (PRCRN for MSTPCRA to MSTPCRE and MSTPCRJ to MSTPCRM, and PRCRS for MSTPCRG, MSTPCRI, and MSTPCRN) to unlock the write protection. For details, see section 11, Register Write Protection Function.

9.5.6 Transition to Cortex-A55/Cortex-R52 Module-Stop State

When transiting Cortex-A55/Cortex-R52 to module-stop state, follow the below procedure.

In case of Cortex-A55 Cluster including all cores:

1. Set 0x00000000 to SWR55ARC register.
2. Issue WFI instruction from all CA55 cores.
3. Set 0x4321A501 to SWR55C register from other CPU.
4. Set 0xF to MSTPCRN.MSTPCRN0i (i = 2 to 5) from other CPU.

Note that transition to module-stop state each CA55 core only is not supported. Use Standby Mode transited by WFI instruction only instead.

In case of Cortex-R52 CPUn (n = 0 or 1):

1. Issue WFI instruction from Cortex-R52 CPUn.
2. Set 0x4321A501 to SWRCPUn register from other CPU.
3. Set 1 to MSTPCRN.MSTPCRN0n from other CPU.

Note that for CPU0, code in TCM can't be changed until releasing module-stop and starting fetch again. That means CPU0 starts from code stored before transition to module-stop.

For CPU1, code in TCM can be changed before starting fetch again via AXIS. Please see section 3.6.2. How to Start Up Cortex-R52 CPU1.

10. Debugging Interface

The LSI has an internal debugging interface that adopts an architecture in which Cortex-A55 quad cores and two Cortex-R52 are integrated by CoreSight. This LSI supports debugging functions, such as downloading, running, and breaking a program, and trace function, which outputs execution history of programs.

10.1 Overview

The LSI supports JTAG and SWD interfaces for debugging and ETR interface for trace.

The LSI has TAP controllers for boundary scan and for CoreSight debugging, which can be selected by the input level of the BSCANP pin. To use the debugging function, set the input level of the BSCANP pin to low. For details on boundary scan, see section 39, Boundary Scan.

Table 10.1 lists the specifications of CoreSight and Figure 10.1 shows a block diagram of CoreSight. For details on CoreSight, see the *Arm Technical Reference Manual*.

Table 10.1 CoreSight specifications

Parameter	Description
Debugging function	• JTAG interface • SWD (Serial Wire Debug) interface
Trace function	• ETF (Embedded Trace FIFO): 16 KB for CA55 funnel, 16 KB for two CR52 funnel and 8 KB for last funnel • ETR (Embedded Trace Router)

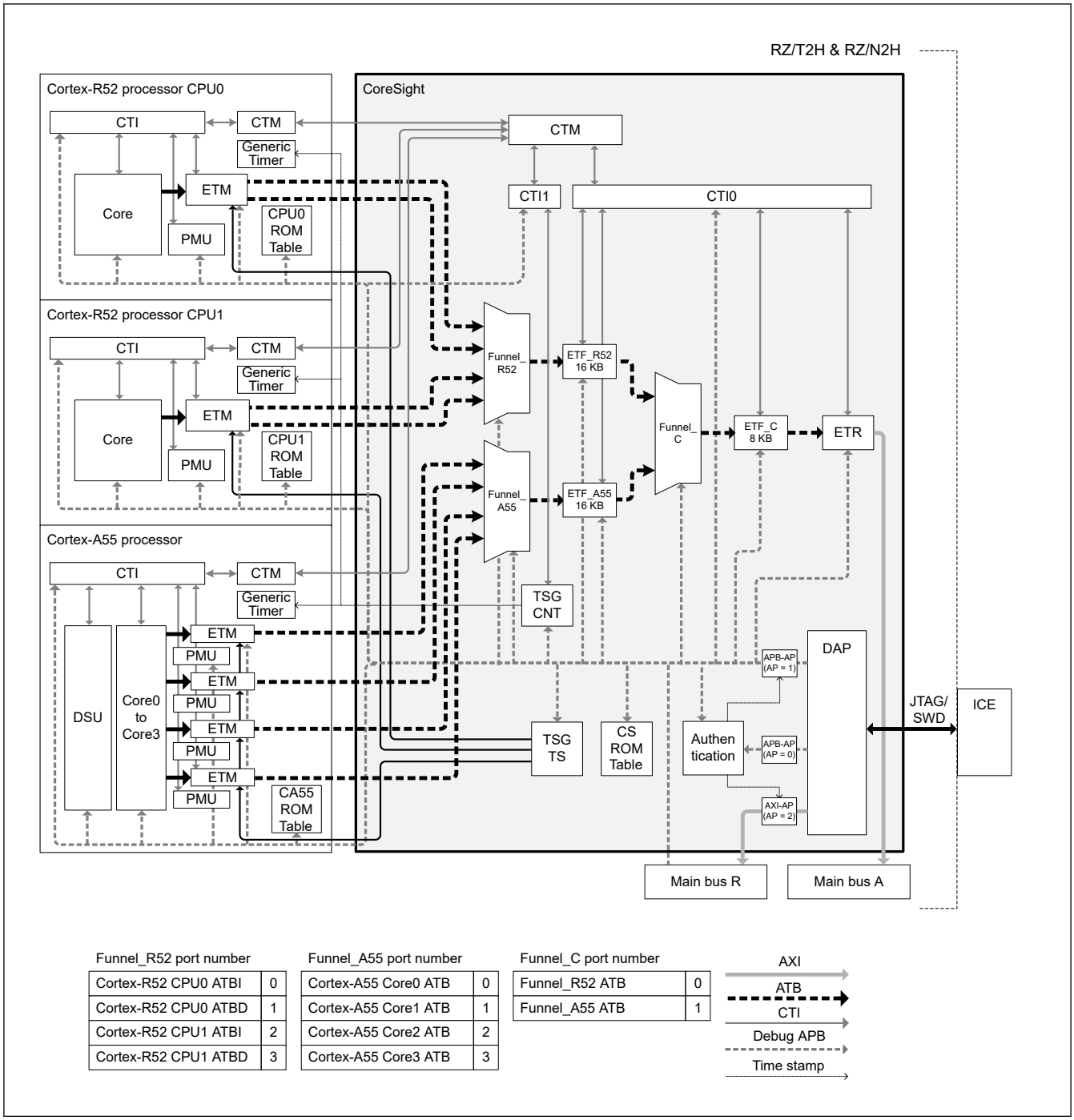


Figure 10.1 CoreSight block diagram

Table 10.2 and Table 10.3 list the input and output for the CTI trigger of CoreSight.

Table 10.2 CTI trigger input and output (CoreSight CTI0) (1 of 2)

CTI input	Description	CTI output	Description
0	ETF_R52 ACQCOMP	0	ETF_R52 FLUSHIN
1	ETF_R52 FULL	1	ETF_R52 TRIGIN
2	ETF_A55 ACQCOMP	2	ETF_A55 FLUSHIN
3	ETF_A55 FULL	3	ETF_A55 TRIGIN
4	ETF_C ACQCOMP	4	ETF_C FLUSHIN
5	ETF_C FULL	5	ETF_C TRIGIN

Table 10.2 CTI trigger input and output (CoreSight CTI0) (2 of 2)

CTI input	Description	CTI output	Description
6	ETR ACQCOMP	6	ETR FLUSHIN
7	ETR FULL	7	ETR TRIGIN

Table 10.3 CTI trigger input and output (CoreSight CTI1)

CTI input	Description	CTI output	Description
0	—	0	TSGCNT HALTREQ
1	—	1	TSGCNT RESTARTREQ
2	—	2	—
3	—	3	—
4	—	4	—
5	—	5	—
6	—	6	—
7	—	7	—

Table 10.4 lists the input and output of the CTI trigger for Cortex-R52 CPU0 and CPU1.

Table 10.4 CTI trigger input and output (Cortex-R52 CPU0 and CPU1)

CTI input	Description	CTI output	Description
0	Cross halt trigger event	0	Requests the processor to enter debug state
1	PMU interrupt	1	Requests the processor to exit debug state
2	—	2	CTI interrupt
3	—	3	—
4	ETM trace unit external output 0	4	ETM trace unit external input 0
5	ETM trace unit external output 1	5	ETM trace unit external input 1
6	ETM trace unit external output 2	6	ETM trace unit external input 2
7	ETM trace unit external output 3	7	ETM trace unit external input 3

Table 10.5 lists the input and output of the CTI trigger for Cortex-A55 Core0 to Core3.

Table 10.5 CTI trigger input and output (Cortex-A55 Core0 to Core3)

CTI input	Description	CTI output	Description
0	Cross-halt trigger event	0	Debug Request trigger event
1	Performance Monitors Overflow trigger event	1	Restart Request trigger event
2	Profiling sample trigger event	2	Generic CTI Interrupt trigger event
3	—	3	—
4	ETM Trace Output trigger event 0	4	Generic Trace External Input trigger event 0
5	ETM Trace Output trigger event 1	5	Generic Trace External Input trigger event 1
6	ETM Trace Output trigger event 2	6	Generic Trace External Input trigger event 2
7	ETM Trace Output trigger event 3	7	Generic Trace External Input trigger event 3

Table 10.6 lists the address map of CoreSight (Debug-APB).

Table 10.6 CoreSight address map (Debug-APB) (1 of 2)

Internal CPU (Cortex-R52 and Cortex-A55) view	Debugger view (AP = 1)	Module
0x8C00_0000 to 0x8C00_FFFF	0x8000_0000 to 0x8000_FFFF	CoreSight/ROM Table
0x8C01_0000 to 0x8C01_FFFF	0x8001_0000 to 0x8001_FFFF	CoreSight/Funnel_R52

Table 10.6 CoreSight address map (Debug-APB) (2 of 2)

Internal CPU (Cortex-R52 and Cortex-A55) view	Debugger view (AP = 1)	Module
0x8C02_0000 to 0x8C02_FFFF	0x8002_0000 to 0x8002_FFFF	CoreSight/Funnel_A55
0x8C03_0000 to 0x8C03_FFFF	0x8003_0000 to 0x8003_FFFF	CoreSight/ETF_R52
0x8C04_0000 to 0x8C04_FFFF	0x8004_0000 to 0x8004_FFFF	CoreSight/ETF_A55
0x8C05_0000 to 0x8C05_FFFF	0x8005_0000 to 0x8005_FFFF	CoreSight/Funnel_C
0x8C06_0000 to 0x8C06_FFFF	0x8006_0000 to 0x8006_FFFF	CoreSight/ETF_C
0x8C07_0000 to 0x8C07_FFFF	0x8007_0000 to 0x8007_FFFF	CoreSight/ETR
0x8C0A_0000 to 0x8C0A_FFFF	0x800A_0000 to 0x800A_FFFF	CoreSight/CTI0
0x8C0B_0000 to 0x8C0B_FFFF	0x800B_0000 to 0x800B_FFFF	CoreSight/CTI1
0x8C0C_0000 to 0x8C0C_FFFF	0x800C_0000 to 0x800C_FFFF	CoreSight/TSG-TS
0x8C20_0000 to 0x8C20_FFFF	0x8020_0000 to 0x8020_FFFF	CoreSight/TSG_CNT
0x8C40_0000 to 0x8C40_FFFF	0x8040_0000 to 0x8040_FFFF	Cortex-R52 CPU0/ROM Table
0x8C41_0000 to 0x8C41_FFFF	0x8041_0000 to 0x8041_FFFF	Cortex-R52 CPU0/Debug
0x8C42_0000 to 0x8C42_FFFF	0x8042_0000 to 0x8042_FFFF	Cortex-R52 CPU0/CTI
0x8C43_0000 to 0x8C43_FFFF	0x8043_0000 to 0x8043_FFFF	Cortex-R52 CPU0/PMU
0x8C44_0000 to 0x8C44_FFFF	0x8044_0000 to 0x8044_FFFF	Cortex-R52 CPU0/ETM
0x8C80_0000 to 0x8C80_FFFF	0x8080_0000 to 0x8080_FFFF	Cortex-R52 CPU1/ROM Table
0x8C81_0000 to 0x8C81_FFFF	0x8081_0000 to 0x8081_FFFF	Cortex-R52 CPU1/Debug
0x8C82_0000 to 0x8C82_FFFF	0x8082_0000 to 0x8082_FFFF	Cortex-R52 CPU1/CTI
0x8C83_0000 to 0x8C83_FFFF	0x8083_0000 to 0x8083_FFFF	Cortex-R52 CPU1/PMU
0x8C84_0000 to 0x8C84_FFFF	0x8084_0000 to 0x8084_FFFF	Cortex-R52 CPU1/ETM
0x8CC0_0000 to 0x8CC0_FFFF	0x80C0_0000 to 0x80C0_FFFF	Cortex-A55/ROM Table
0x8CC1_0000 to 0x8CC1_FFFF	0x80C1_0000 to 0x80C1_FFFF	Cortex-A55 Core0/Debug
0x8CC2_0000 to 0x8CC2_FFFF	0x80C2_0000 to 0x80C2_FFFF	Cortex-A55 Core0/CTI
0x8CC3_0000 to 0x8CC3_FFFF	0x80C3_0000 to 0x80C3_FFFF	Cortex-A55 Core0/PMU
0x8CC4_0000 to 0x8CC4_FFFF	0x80C4_0000 to 0x80C4_FFFF	Cortex-A55 Core0/ETM
0x8CD1_0000 to 0x8CD1_FFFF	0x80D1_0000 to 0x80D1_FFFF	Cortex-A55 Core1/Debug
0x8CD2_0000 to 0x8CD2_FFFF	0x80D2_0000 to 0x80D2_FFFF	Cortex-A55 Core1/CTI
0x8CD3_0000 to 0x8CD3_FFFF	0x80D3_0000 to 0x80D3_FFFF	Cortex-A55 Core1/PMU
0x8CD4_0000 to 0x8CD4_FFFF	0x80D4_0000 to 0x80D4_FFFF	Cortex-A55 Core1/ETM
0x8CE1_0000 to 0x8CE1_FFFF	0x80E1_0000 to 0x80E1_FFFF	Cortex-A55 Core2/Debug
0x8CE2_0000 to 0x8CE2_FFFF	0x80E2_0000 to 0x80E2_FFFF	Cortex-A55 Core2/CTI
0x8CE3_0000 to 0x8CE3_FFFF	0x80E3_0000 to 0x80E3_FFFF	Cortex-A55 Core2/PMU
0x8CE4_0000 to 0x8CE4_FFFF	0x80E4_0000 to 0x80E4_FFFF	Cortex-A55 Core2/ETM
0x8CF1_0000 to 0x8CF1_FFFF	0x80F1_0000 to 0x80F1_FFFF	Cortex-A55 Core3/Debug
0x8CF2_0000 to 0x8CF2_FFFF	0x80F2_0000 to 0x80F2_FFFF	Cortex-A55 Core3/CTI
0x8CF3_0000 to 0x8CF3_FFFF	0x80F3_0000 to 0x80F3_FFFF	Cortex-A55 Core3/PMU
0x8CF4_0000 to 0x8CF4_FFFF	0x80F4_0000 to 0x80F4_FFFF	Cortex-A55 Core3/ETM

Note: No listed areas are reserved. Do not access those areas.

Table 10.7 CoreSight related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
All registers from internal CPU access	—	—	SLVACCCTL7.CSDBGAPB_SL

Table 10.8 lists the input/output pins of the debugging interface.

Table 10.8 Configuration of pins for the debugging interface

Name	Pin name	I/O	Functions
Test clock	TCK	Input	Data is serially supplied from the Test Data Input (TDI) pin to this module, synchronized with this clock, and output from the Test Data Output (TDO) pin. In SWD mode, this pin functions as the SWCLK pin. This pin can also be used as a general-purpose port. The initial function is TCK.
Test mode select	TMS	Input/ Output	Changing the level of this signal, by synchronizing with TCK, determines the status of the TAP (Test Access Port) control circuit. The protocol conforms to the JTAG standard (IEEE Std.1149.1). In SWD mode, this pin functions as the SWDIO pin. This pin can also be used as a general-purpose port. The initial function is TMS.
Test reset	TRST#	Input	This pin accepts input asynchronously with TCK. When the level of this pin is low, TAP (Test Access Port) is reset. When the levels of the TRST# pin and RES# pin are both low, TAP and the debugging circuit are reset.
Test data input	TDI	Input	Changing the level of this pin, by synchronizing with TCK, sends data to this module. This pin can also be used as a general-purpose port. The initial function is TDI.
Test data output	TDO	Output	Reading the level of this pin, by synchronizing with TCK, reads data from this module. This pin can also be used as a general-purpose port. The initial function is TDO.
Boundary scan setting	BSCANP	Input	Input the high level when boundary scan test is performed. Input low when debugging is performed by CoreSight. For details on boundary scan, see section 39, Boundary Scan.

Note: When you design a board on which an emulator can be used, set the TRST# pin to low while the RES# pin is asserted at power up, and configure the board so that control is available by the TRST# pin only. When this pin is not used, fix it to low, or connect it so that the same signal as that on the RES# pin is to be input. For details, see section 10.3.3. Reset Configuration and the Method of Connecting with the Emulator.

10.2 Register Descriptions

10.2.1 OIRn : OCD Authentication ID Code Register n (n = 0 to 3)

Address: 0x0008_0000 + 0x04 × n (the address is the debugger view (AP = 0))

Bit position:	31																												0
Bit field:	IDn[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	IDn[31:0]	128-bit Authentication ID n = 0: Authentication ID[31:0] n = 1: Authentication ID[63:32] n = 2: Authentication ID[95:64] n = 3: Authentication ID[127:96]	W*1

Note 1. Once all four registers are written, further write is blocked.

The OIRn registers are used to write 128-bit authentication ID from the external debugger to access debug system.

10.2.2 OSR : OCD Status Register

Address: 0x0008_0010 (the address is the debugger view (AP = 0))

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	WFLG	RESULT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RESULT[1:0]	Authentication result 0 0: No authentication 0 1: Passed Authentication level 1. Both EL2 and EL1/EL0 accesses are permitted. 1 0: Passed Authentication level 2. Only EL1/EL0 access is permitted. 1 1: Failed or access permanently prohibited	R
2	WFLG	All OIRn registers written flag 0: Not written 1: Written	R
31:3	—	These bits are read as 0.	R

The OSR register indicates authentication result.

10.3 Operation

10.3.1 JTAG Interface

The JTAG interface uses five signals (TCK, TMS, TDO, TDI, and TRST#) to communicate with the host machine (PC) through the emulator. Figure 10.2 shows an example connection, which includes the RES# pin connection.

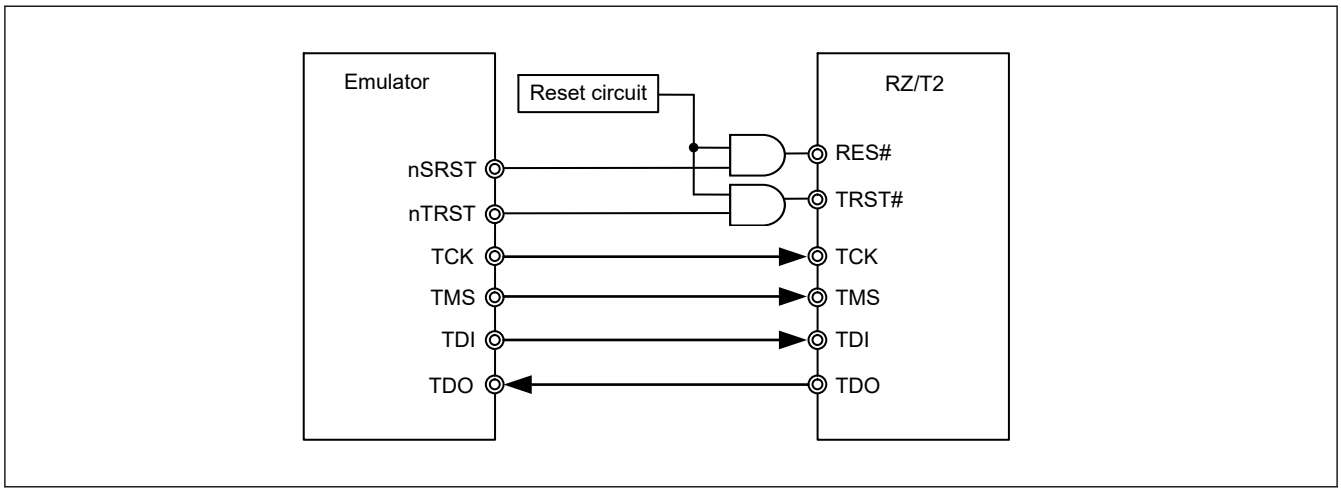


Figure 10.2 Example connection of the JTAG interface

When the JTAG interface is not used, the TCK, TMS, TDI, and TDO pins can be used as general-purpose ports. When you use these pins as general-purpose ports, perform pin settings by referring to section 17, I/O Ports.

10.3.2 SWD Interface

The SWD (Serial Wire Debug) interface uses two signals (SWCLK (TCK) and SWDIO (TMS)) to communicate with the host machine (PC) through the emulator. Figure 10.3 shows an example connection, which includes the RES# pin connection.

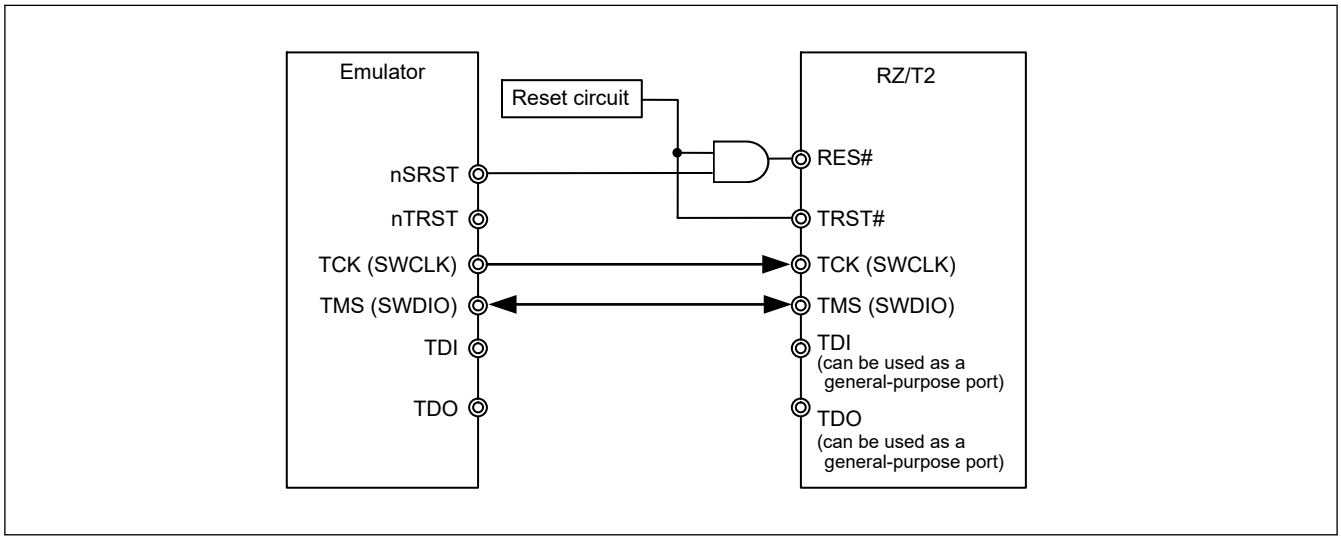


Figure 10.3 Example connection of the SWD interface

When the SWD interface is used for debugging, the TDI and TDO pins can be used as general-purpose ports. When using these pins as general-purpose ports, perform pin settings by referring to section 17, I/O Ports.

Note: In the initial status of the LSI, the debugging interface is in JTAG mode. If you use the TDI and TDO pins as general-purpose ports and the emulator connection for debugging, switch the mode to SWD (Serial Wire Debug) mode using the control from the debugger, and then start debugging.

10.3.3 Reset Configuration and the Method of Connecting with the Emulator

When designing a board on which the emulator can be used, set the TRST# pin to low while the RES# pin is asserted at power up. Also, configure the board so that control is available by the TRST# pin only.

When debugging is performed, if the initial level of the RES# and TRST# pins are both low, the CPU and debugging section become the reset status. Then, setting the TRST# pin to high while the RES# pin is kept to low enables the TAP access to set SWJ-DP registers. After RES# pin is set to high, it is possible to access OCD registers (OIRn and OSR) in AP = 0. Debugging setting is possible after on-chip boot code runs if JTAG authentication passes, or no authentication is needed.

When the emulator is not connected, fix the TRST# pin to low, or let the signal be the same as that on the RES# pin input to the TRST# pin.

Note: When debugging is performed by CoreSight, input low to the BSCANP pin to disable the boundary scan function.

10.3.3.1 Example Connection of the Emulator That Cannot Drive the nTRST Output to High

Figure 10.4 shows an example of connection circuit when an emulator that cannot drive the nTRST output to high is used. The TRST# pin is pulled up and is asserted to low by the emulator.

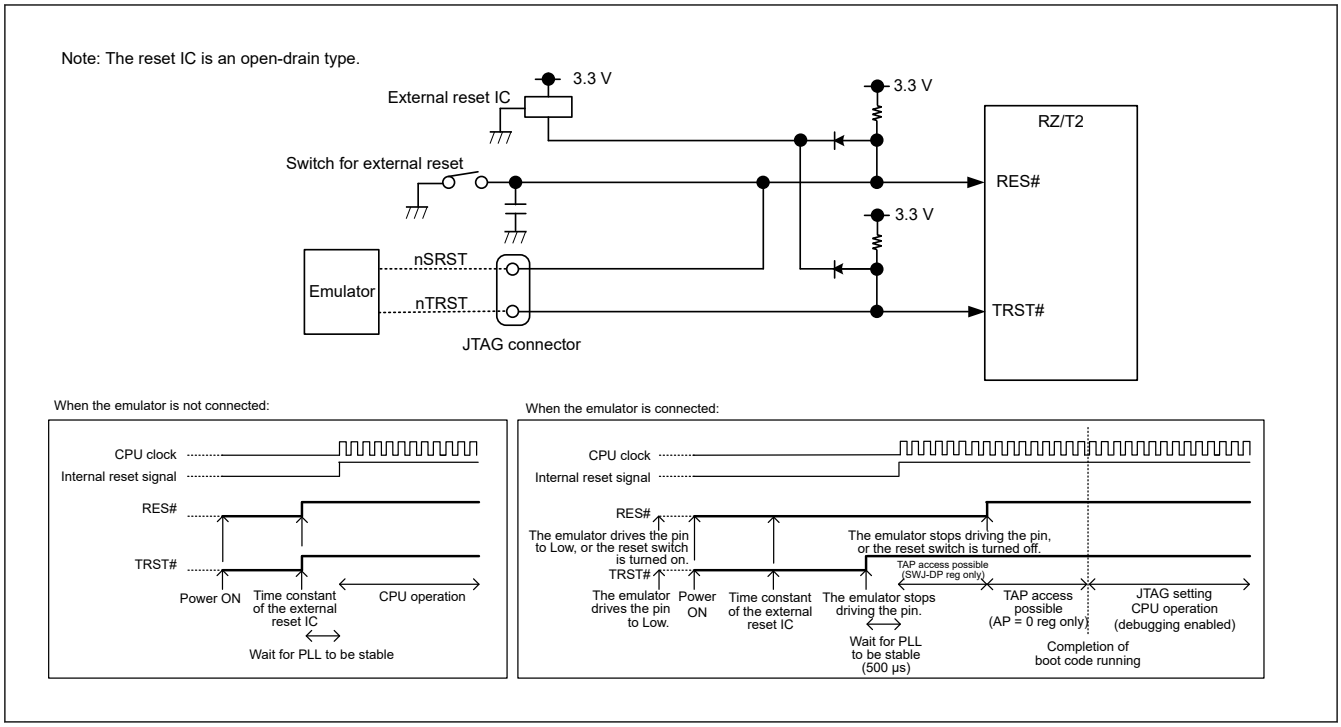


Figure 10.4 Example of connection circuit of an emulator that cannot drive the nTRST output to high

10.3.3.2 Example Connection of the Emulator That Can Drive the nTRST Output to High

Figure 10.5 shows an example of connection circuit when an emulator that can drive the nTRST output to high is used. The TRST# pin (high or low) is controlled by the emulator.

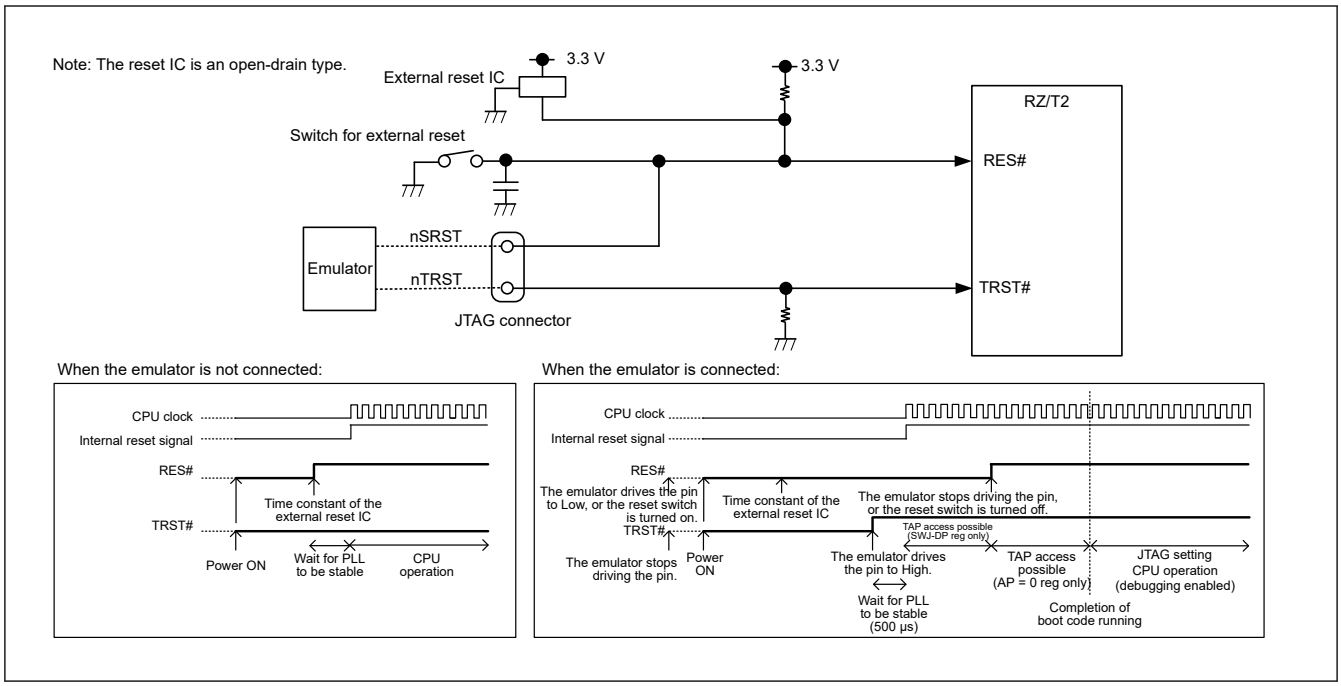


Figure 10.5 Example of connection circuit of an emulator that can drive the nTRST output to high

10.3.4 JTAG Pin Treatment When No Emulator Is Connected

If no emulator is connected, pin treatment is required according to Table 10.9.

Table 10.9 JTAG pin treatment when no emulator is connected

Pin name	Treatment
TCK	Pull down the pin (except when the port is used as a general-purpose port)
TMS	Pull up the pin (except when the port is used as a general-purpose port)
TDI	Pull up the pin (except when the port is used as a general-purpose port)
TDO	Open the pin (except when the port is used as a general-purpose port)
TRST#	Pull down the pin, or let the signal be the same as that on the RES# pin input

10.3.5 Noise Reduction of the TRST# Pin

Analog delay noise reduction is performed for the TRST# pin. This measure against noise can remove noise that is within 100 ns at maximum.

10.3.6 Available Trace Functions

Table 10.10 lists the trace functions that are available through the respective debugging ports (ETR and SWD or JTAG).

Table 10.10 Available trace functions

CPU core	Debugging port	Trace functions
Cortex-R52 CPU0, CPU1, and Cortex-A55	ETR	Full instruction and data tracing through the ETM of the Cortex-R52 CPU0, CPU1, and Cortex-A55. ETF is programmed to be in Circular Buffer mode or Hardware FIFO mode. ETR can output trace data to Main bus A.
	SWD/JTAG	The same information as for the trace port interface can be acquired through the ETF which is programmed to be in Software FIFO mode.

10.3.7 Debug Authentication

Debug authentication mechanism checks access permission for debugging and LSI resources. To obtain full debug functionality, a pass result of the authentication mechanism is required if authentication is required.

Authentication level is selectable from four states by setting AUTHMODEJ area of OTP (One Time Programmable memory):

- No authentication
- Authentication level 1 (both EL2 and EL1/EL0 access permission)
- Authentication level 2 (only EL1/EL0 access permission)
- Permanent prohibition

If authentication level 1 or 2 is selected, 128-bit authentication ID must be written to OCD Authentication ID Code Register n (OIRn) (n = 0 to 3) from debugger to compare with the ID which is already stored in OTP. Once all OIRn registers are written, authentication automatically starts and the authentication result is available in OCD Status Register (OSR). If authentication failed, then no debug connection is permitted. Reset (RES# pin reset, system software reset, or error reset) is required to retry debug authentication. If "No Authentication" is selected, RESULT[1:0] bits in OSR register indicate 01b which means both EL2 and EL1/EL0 accesses are permitted.

10.3.8 Debug Authentication Procedure

Figure 10.6 shows the debug authentication procedure.

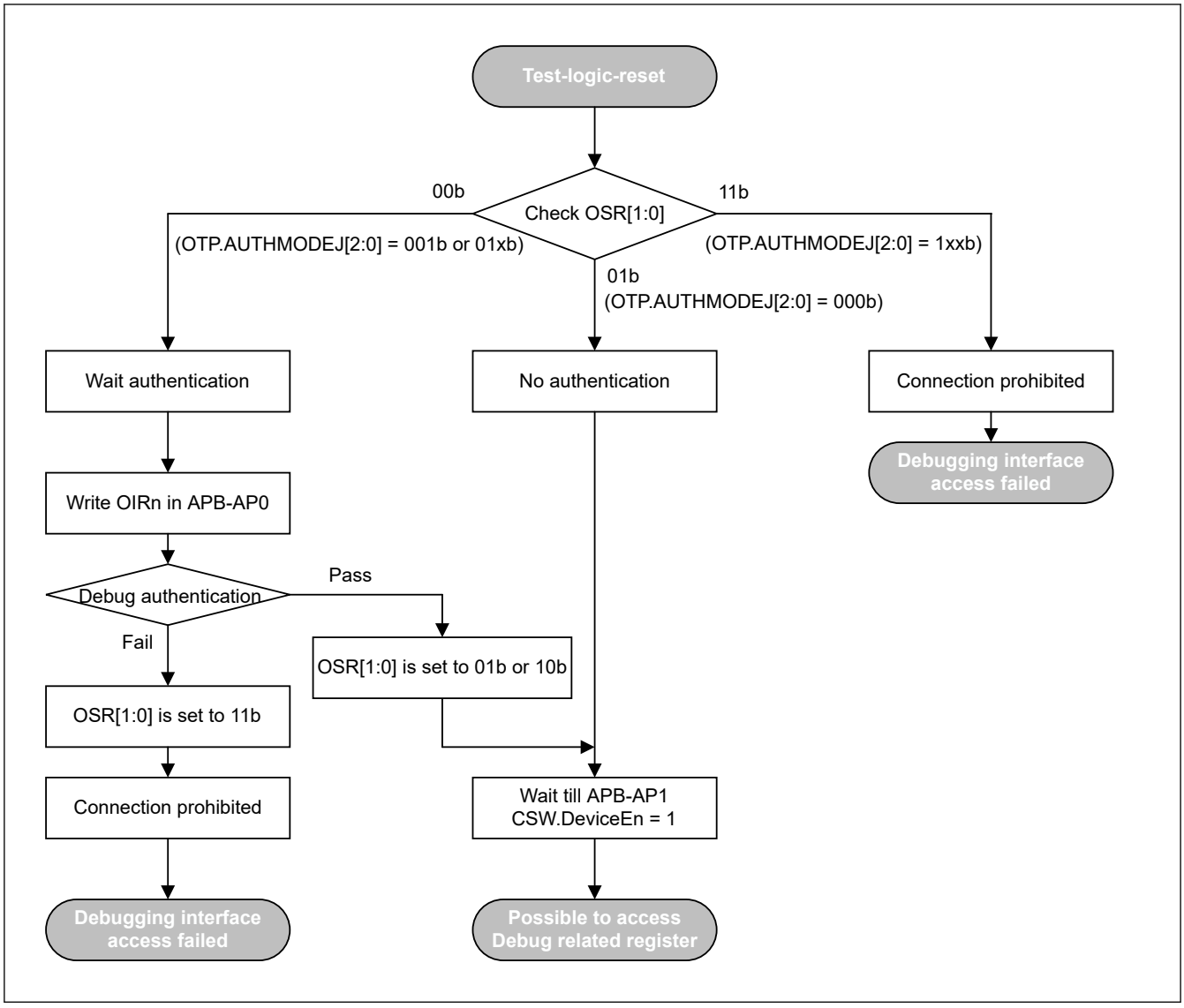


Figure 10.6 Debug authentication procedure (plaintext)

10.4 Usage Notes

10.4.1 Access to the Main Bus

Access to the main bus by the DAP is through either the Cortex-R52 CPU0, CPU1, Cortex-A55, or AXI-AP. When access is to be through the CPU except second boot CPU (Cortex-R52 CPU0 or Cortex-A55), the setting of corresponding CPU Software Reset register and Module Stop register must be changed to release the CPU from the reset state and module stop state beforehand. When access is to be through the AXI-AP, the setting of M-MPU for AXI-AP must be changed to access internal resources.

10.4.2 Authentication Level

Authentication level 2 is available only when debugger can access by EL1/EL0. In case using debugger which can access by EL2 only, do not use authentication level 2.

10.4.3 Debug Reset

The reset request from debugger is prohibited during boot operation.

11. Register Write Protection Function

11.1 Overview

The register write protection function protects important registers from being overwritten in case a program runs out of control. The registers to be protected are set with the protect register (PRCRN and PRCRS).

Table 11.1 and Table 11.2 show the relationship between the PRCRN/PRCRS register bits and the registers to be protected.

Table 11.1 Correspondence between PRCRN Register Bits and Registers to be Protected

PRCRN register	Registers to be protected
PRC0 bit	• Registers related to the clock generation circuit: SCKCR, SCKCR3, SCKCR4
PRC1 bit	• Registers related to the low power consumption functions: MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, MSTPCRE, MSTPCRJ, MSTPCRK, MSTPCRL, MSTPCRM • Reset related registers: RSTSR0, MRCTLA, MRCTLE, MRCTLM
PRC2 bit	• GPIO-related registers: PMC13 to PMC35, PFC13 to PFC35, DRCTL13 to DRCTL35, SLPORT13 to SLPORT35 ELC_PGR1, ELC_PGR2, ELC_PGC1, ELC_PGC2, ELC_PDBF1, ELC_PDBF2, ELC_PEL0 to ELC_PEL3, ELC_DPTC, ELC_ELSR2 SLELC_PGRC, SLELC_PDBF1, SLELC_PDBF2, SLELC_PEL, SLELC_DE
PRC3 bit	• System Control registers: – Encoder interface related registers: ENCODER_CFG0 to ENCODER_CFG7, EnDAT_CFG0 to EnDAT_CFG7 – xSPI interface related registers: IOVOLCTL, CS0ENDAD, CS1STRAD, CS1ENDAD

Table 11.2 Correspondence between PRCRS Register Bits and Registers to be Protected

PRCRS register	Registers to be protected
PRC0 bit	• Registers related to the clock generation circuit: SCKCR2, PMSEL, PLL0EN, PLL0_SSC_CTR, LOCOCR, PLL2EN, PLL2_SSC_CTR, PLL3EN, PLL3_VCO_CTR0, PLL3_VCO_CTR1
PRC1 bit	• Registers related to the low power consumption functions: MSTPCRG, MSTPCRI, MSTPCRN • Reset related registers: SWRSYS, SWRCPU0, SWRCPU1, SWR55C, SWR550, SWR551, SWR552, SWR553, SWR55ARC, MRCTLI
PRC2 bit	• GPIO-related registers: PMC00 to PMC12, PFC00 to PFC12, DRCTL00 to DRCTL12, RSELP00 to RSELP12, SLPORT00 to SLPORT12, SLRSELP, SLPSR, RSELPSR
PRC3 bit	• System Control registers: – CPU related registers: RVBAL0 to RVBAL3, RVBAH0 to RVBAH3, CPU1HALT – Internal Bus related registers: STADD0 to STADD15, ENDADD0 to ENDADD15, ERRINF, ERRINF_R, ERRINF_W, SSTPCR0, SSTPCR1, SSTPCR4 to SSTPCR7, MSTACCCTL0 to MSTACCCTL3, SLVACCCTL0 to SLVACCCTL2, SLVACCCTL4 to SLVACCCTL9, ADXCTL0 to ADXCTL12 – WDT related registers: WDTDCR – System SRAM related registers: SYSRAM_CTRL0 to SYSRAM_CTRL3 – SHOSTIF related registers: SHCFG

11.2 Register Map

Table 11.3 Register Write Protection register map

Address	Register symbol	Register name	Write protection
0x8029_4200	PRCRN	Non_Safety Area Protect Register	—
0x8129_6000	PRCRS	Safety Area Protect Register	—

Table 11.4 Register Write Protection related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
PRCRN and PRCRS	—	—	SLVACCCTL8.PRCR_SL

11.3 Register Descriptions

11.3.1 PRCRN : Non_Safety Area Protect Register

Base address: RWP_NS = 0x8029_4200

Offset address: 0x00

Bit position:	31																15								8				3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRKEY[7:0]								—	—	—	—	PRC3	PRC2	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Protect 0 Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
1	PRC1	Protect 1 Enables writing to the registers related to low power consumption and reset. 0: Write disabled 1: Write enabled	R/W
2	PRC2	Protect 2 Enables writing to the registers related to GPIO settings. 0: Write disabled 1: Write enabled	R/W
3	PRC3	Protect 3 Enables writing to the registers related to System control. 0: Write disabled 1: Write enabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	PRC Key Code These bits control permission and prohibition of writing to the PRCRN register. To modify the PRCRN register, write 0xA5 to the PRKEY[7:0] bits. Any other value is ignored if written to the PRCRN register.*1	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write a value to the PRKEY[7:0] bits at the same time as other bits. When setting PRC0 to 1 and PRC1, PRC2, and PRC3 to 0, for example, write 0x0000A501 to the PRCRN register.

PRCi bit (Protect i) (i = 0 to 3)

Setting the PRCi bit to 1 or 0 enables or disables writing to the corresponding registers to be protected, respectively.

The PRCRN register controls writes to the protected registers for non-safety area. The non-safety area is as follows:

- 0x802C_0000 to 0x802C_FFFF: address area for GPIO-related registers
- 0x8028_0000 to 0x8028_FFFF: address area for clock generation, low power consumption, and reset registers
- 0x8029_0000 to 0x8029_FFFF: address area for system control registers

11.3.2 PRCRS : Safety Area Protect Register

Base address: RWP_S = 0x8129_6000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRKEY[7:0]								—	—	—	—	PRC3	PRC2	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Protect 0 Enables writes to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
1	PRC1	Protect 1 Enables writes to the registers related to low power consumption and reset. 0: Write disabled 1: Write enabled	R/W
2	PRC2	Protect 2 Enables writes to the registers related to GPIO settings. 0: Write disabled 1: Write enabled	R/W
3	PRC3	Protect 3 Enables writes to the system control registers. 0: Write disabled 1: Write enabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	PRC Key Code These bits control permission and prohibition of writing to the PRCRS register. To modify the PRCRS register, write 0xA5 to the PRKEY[7:0] bits. Any other value is ignored if written to the PRCRS register.*1	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write a value to the PRKEY[7:0] bits at the same time as other bits. When setting PRC0 to 1 and PRC1, PRC2, and PRC3 to 0, for example, write 0x0000A501 to the PRCRS register.

PRCi bit (Protect i) (i = 0 to 3)

Setting the PRCi bit to 1 or 0 enables or disables writing to the corresponding registers to be protected, respectively.

The PRCRS register controls writes to the protected registers for safety area. The safety area is as follows:

- 0x802B_0000 to 0x802B_FFFF and 0x812C_0000 to 0x812C_FFFF: address area for GPIO-related registers
- 0x8128_0000 to 0x8128_FFFF: address area for clock generation, low power consumption, and reset registers
- 0x8129_0000 to 0x8129_FFFF: address area for system control registers

12. Interrupt Controller (ICU)

12.1 Overview

The interrupt controller accepts interrupt request from peripheral module, external pins include the external DMA request pins. An interrupt accepted by the interrupt controller is set either as an interrupt to the CPU (Cortex-A55, Cortex-R52 CPU0 and CPU1) or as an activating trigger signal for the DMAC and ELC.

Table 12.1 lists the specifications of the ICU, and Figure 12.1 shows a block diagram of the interrupt controller.

Table 12.1 ICU specifications

Item		Description
Interrupts	Interrupt contact destinations	• Cortex-A55 (via GIC-600) • Cortex-R52 CPU0 • Cortex-R52 CPU1 • Three DMAC units • ELC
	Peripheral function interrupts	• Interrupts from peripheral modules • Interrupt detection: Edge detection/Level detection (dependent on peripheral modules) • GPT/MTU3 events are selectable from max 14 to 4 interrupts each channel. • Encoder interface events are selectable from 5 modules to 1 module each unit. • A part of the assignment of the event source to Cortex-R52 GIC SPI is selectable. • Assignment of the event source to Cortex-A55 GIC-600 SPI is fixed.
	External pin interrupts	Interrupts from pins IRQ0 to IRQ15 Number of sources: 16 IRQ0 to IRQ13: for NONSAFETY IRQ14 to IRQ15: for SAFETY Interrupt detection: Low level/falling edge/rising edge/rising and falling edges. One of these detection methods can be set for each source. Digital noise filter function: Supported
	Interrupts between the CPUs	Mutual interrupts among Cortex-A55, Cortex-R52 CPU0 and CPU1 by software interrupts Number of sources: 16 INTCPU0 to INTCPU13: for NONSAFETY INTCPU14, INTCPU15: for SAFETY
	Error events	• Error events from CPU and peripherals are captured and merged to multiple interrupts with mask. Initial state is all masked. (no interrupt generated) • Hardware reset can be generated by error events.
	DMAC control	Interrupts can be assigned as DMA activation sources.
	External DMA request	External DMA request pins (DREQ) can output DACK, TEND when DMA transfer to external bus. Number of sources: 1 Digital noise filter function: Supported
System error interrupt	SEI pin interrupt	Interrupts from the SEI pin Interrupt detection: Low level/falling edge/rising edge/rising and falling edges. One of these detection methods can be set Digital noise filter function: Supported The SEI pin is connected to SEI and VSEI for Cortex-R52 CPU0 and CPU1, and SPI for Cortex-A55.
Restoration from the sleep status		Restoration due to non-maskable interrupt and all unmasked interrupt sources

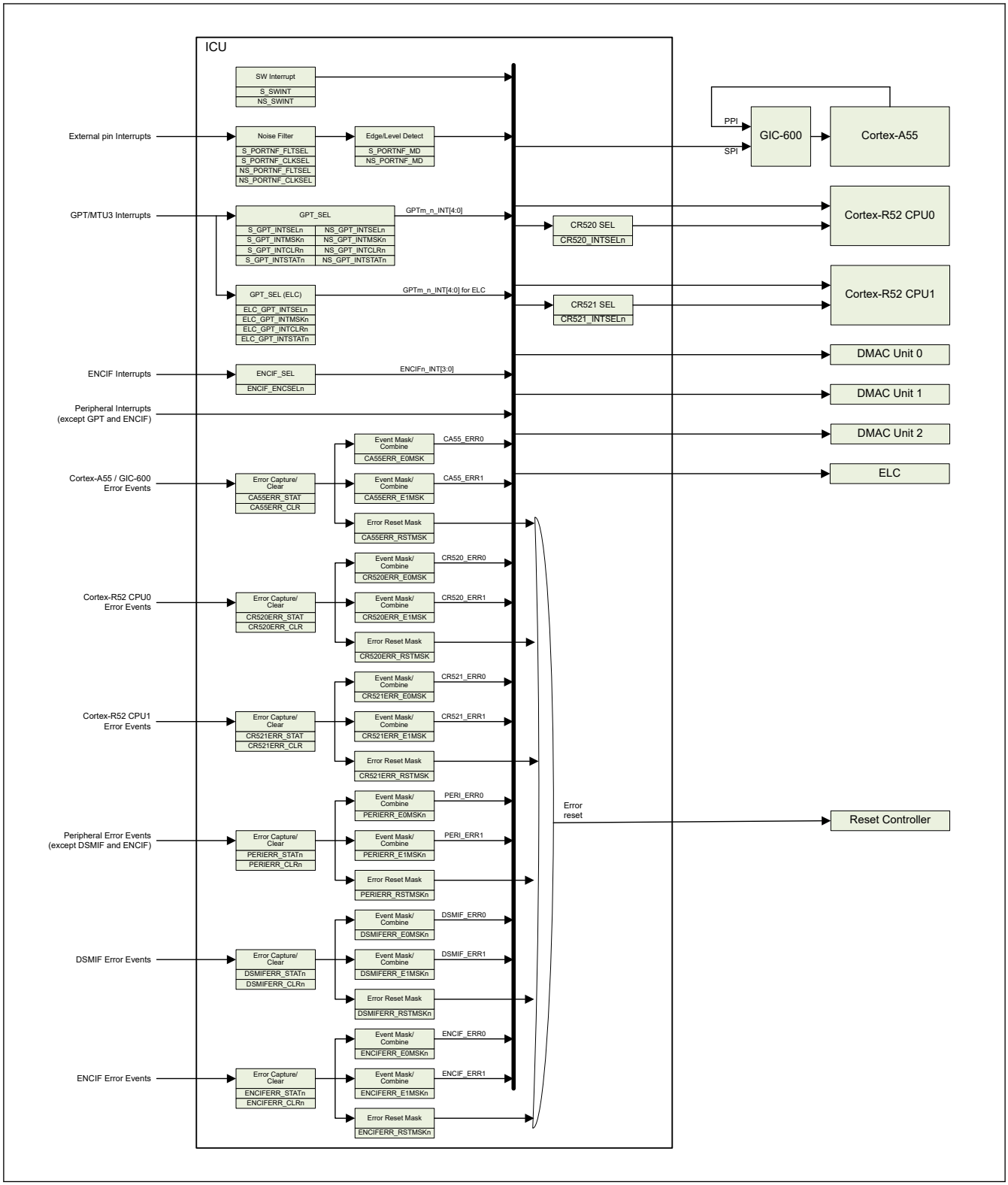


Figure 12.1 Block diagram of interrupt controller unit

12.2 Register Map

Table 12.2 ICU register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x812A_0000	S_SWINT	Software Interrupt Register for Safety Register	—
0x812A_0004	S_PORTNF_FLTSEL	Interrupt Noise Filter Enable Register for Safety Register	—
0x812A_0008	S_PORTNF_CLKSEL	Interrupt Noise Filter Setting Register for Safety Register	—
0x812A_000c	S_PORTNF_MD	Interrupt Edge Detection Setting Register for Safety Register	—
0x812A_0020 + 0x04 × n	S_GPT_INTSELn	Safety GPT Interrupt Select Register n (n = 0, 1)	—
0x812A_0030 + 0x04 × n	S_GPT_INTMSKn	Safety GPT Combined Interrupt Mask Register n (n = 0, 1)	—
0x812A_0038 + 0x04 × n	S_GPT_INTCLRn	Safety GPT Combined Interrupt Status Clear Register n (n = 0, 1)	—
0x812A_0040 + 0x04 × n	S_GPT_INTSTATn	Safety GPT Combined Interrupt Status Register n (n = 0, 1)	—
0x802A_0000	NS_SWINT	Software Interrupt Register	—
0x802A_0004	NS_PORTNF_FLTSEL	Interrupt Noise Filter Enable Register	—
0x802A_0008	NS_PORTNF_CLKSEL	Interrupt Noise Filter Setting Register	—
0x802A_000c	NS_PORTNF_MD	Interrupt Edge Detection Setting Register	—
0x802A_0050	CA55ERR_E0MSK	CA55 E0 Error Event Mask Register	—
0x802A_0054	CA55ERR_E1MSK	CA55 E1 Error Event Mask Register	—
0x802A_0058	CA55ERR_RSTMSK	CA55 Error Event Reset Mask Register	—
0x802A_0060	CA55ERR_CLR	CA55 Error Event Status Clear Register	—
0x802A_0064	CA55ERR_STAT	CA55 Error Event Status Register	—
0x802A_0068	CR520ERR_E0MSK	CR52 CPU0 E0 Error Event Mask Register	—
0x802A_006c	CR520ERR_E1MSK	CR52 CPU0 E1 Error Event Mask Register	—
0x802A_0070	CR520ERR_RSTMSK	CR52 CPU0 Error Event Reset Mask Register	—
0x802A_0078	CR520ERR_CLR	CR52 CPU0 Error Event Status Clear Register	—
0x802A_007c	CR520ERR_STAT	CR52 CPU0 Error Event Status Register	—
0x802A_0080	CR521ERR_E0MSK	CR52 CPU1 E0 Error Event Mask Register	—
0x802A_0084	CR521ERR_E1MSK	CR52 CPU1 E1 Error Event Mask Register	—
0x802A_0088	CR521ERR_RSTMSK	CR52 CPU1 Error Event Reset Mask Register	—
0x802A_0090	CR521ERR_CLR	CR52 CPU1 Error Event Status Clear Register	—
0x802A_0094	CR521ERR_STAT	CR52 CPU1 Error Event Status Register	—
0x802A_0098 + 0x04 × n	PERIERR_E0MSKn	Peripheral E0 Error Event Mask Register n (n = 0 to 2)	—
0x802A_00a4 + 0x04 × n	PERIERR_E1MSKn	Peripheral E1 Error Event Mask Register n (n = 0 to 2)	—
0x802A_00b0 + 0x04 × n	PERIERR_RSTMSKn	Peripheral Error Event Reset Mask Register n (n = 0 to 2)	—
0x802A_00c8 + 0x04 × n	PERIERR_CLRn	Peripheral Error Event Status Clear Register n (n = 0 to 2)	—
0x802A_00d4 + 0x04 × n	PERIERR_STATn	Peripheral Error Event Status Register n (n = 0 to 2)	—
0x802A_00e0 + 0x04 × n	DSMIFERR_E0MSKn	DSMIF E0 Error Event Mask Register n (n = 0 to 11)	—
0x802A_0110 + 0x04 × n	DSMIFERR_E1MSKn	DSMIF E1 Error Event Mask Register n (n = 0 to 11)	—
0x802A_0140 + 0x04 × n	DSMIFERR_RSTMSKn	DSMIF Error Event Reset Mask Register n (n = 0 to 11)	—
0x802A_01A0 + 0x04 × n	DSMIFERR_CLRn	DSMIF Error Event Status Clear Register n (n = 0 to 11)	—
0x802A_01D0 + 0x04 × n	DSMIFERR_STATn	DSMIF Error Event Status Register n (n = 0 to 11)	—
0x802A_0200 + 0x04 × n	ENCIFERR_E0MSKn	ENCIF E0 Error Event Mask Register n (n = 0 to 4)	—
0x802A_0214 + 0x04 × n	ENCIFERR_E1MSKn	ENCIF E1 Error Event Mask Register n (n = 0 to 4)	—

Table 12.2 ICU register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x802A_0228 + 0x04 × n	ENCIFERR_RSTMSKn	ENCIF Error Event Reset Mask Register n (n = 0 to 4)	—
0x802A_0250 + 0x04 × n	ENCIFERR_CLRn	ENCIF Error Event Status Clear Register n (n = 0 to 4)	—
0x802A_0264 + 0x04 × n	ENCIFERR_STATn	ENCIF Error Event Status Register n (n = 0 to 4)	—
0x802A_0290	ENCIF_ENCSEL0	ENCIF Event Select Register 0	—
0x802A_0294	ENCIF_ENCSEL1	ENCIF Event Select Register 1	—
0x802A_02c8 + 0x04 × n	NS_GPT_INTSELn	GPT/MTU3 Interrupt Select Register n (n = 0 to 25)	—
0x802A_0900 + 0x04 × n	NS_GPT_INTMSKn	GPT/MTU3 Combined Interrupt Mask Register n (n = 0 to 25)	—
0x802A_09E0 + 0x04 × n	NS_GPT_INTCLRn	GPT/MTU3 Combined Interrupt Status Clear Register n (n = 0 to 25)	—
0x802A_0AC0 + 0x04 × n	NS_GPT_INTSTATn	GPT/MTU3 Combined Interrupt Status Register n (n = 0 to 25)	—
0x802A_0418 + 0x04 × n	CR520_INTSELn	CR52 CPU0 Interrupt Select Register n (n = 0 to 79)	—
0x802A_0558 + 0x04 × n	CR521_INTSELn	CR52 CPU1 Interrupt Select Register n (n = 0 to 79)	—
0x802A_0330 + 0x04 × n	ELC_GPT_INTSELn	GPT/MTU3 ELC event source Select Register n (n = 0 to 25)	—
0x802A_0970 + 0x04 × n	ELC_GPT_INTMSKn	GPT/MTU3 Combined ELC Event Mask Register n (n = 0 to 25)	—
0x802A_0A50 + 0x04 × n	ELC_GPT_INTCLRn	GPT/MTU3 Combined ELC Event Status Clear Register n (n = 0 to 25)	—
0x802A_0B30 + 0x04 × n	ELC_GPT_INTSTATn	GPT/MTU3 Combined ELC Event Status Register n (n = 0 to 25)	—

Table 12.3 ICU related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
All the registers	—	—	SLVACCCTL8.ICU_SL

12.3 Register Descriptions

This section describes details of the register functions.

12.3.1 Safety Region Registers

12.3.1.1 S_SWINT : Software Interrupt Register for Safety Register

Base address: ICU_S = 0x812A_0000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IC15	IC14
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IC14	Software Interrupt register Occur software interrupt INTCPU14 when this value is set to 1. This bit is read as 0.	W
1	IC15	Software Interrupt register Occur software interrupt INTCPU15 when this value is set to 1. This bit is read as 0.	W

Bit	Symbol	Function	R/W
31:2	—	The write value should be 0.	W

12.3.1.2 S_PORTNF_FLTSEL : Interrupt Noise Filter Enable Register for Safety Register

Base address: ICU_S = 0x812A_0000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	FLTSEI	FLT15	FLT14
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLT14	Noise filter enable for IRQ14 0: Disable 1: Enable	R/W
1	FLT15	Noise filter enable for IRQ15 0: Disable 1: Enable	R/W
2	FLTSEI	Noise filter enable for SEI 0: Disable 1: Enable	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

12.3.1.3 S_PORTNF_CLKSEL : Interrupt Noise Filter Setting Register for Safety Register

Base address: ICU_S = 0x812A_0000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	CLKSELSEI[1:0]		CKSEL15[1:0]		CKSEL14[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKSEL14[1:0]	Select noise filter sampling frequency dividend rate for IRQ14. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W

Bit	Symbol	Function	R/W
3:2	CKSEL15[1:0]	Select noise filter sampling frequency dividend rate for IRQ15. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
5:4	CLKSELSEI[1:0]	Select noise filter sampling frequency dividend rate for SEI. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

12.3.1.4 S_PORTNF_MD : Interrupt Edge Detection Setting Register for Safety Register

Base address: ICU_S = 0x812A_0000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	MDSEI[1:0]		MD15[1:0]		MD14[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1

Bit	Symbol	Function	R/W
1:0	MD14[1:0]	Select detection mode for IRQ14 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
3:2	MD15[1:0]	Select detection mode for IRQ15 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
5:4	MDSEI[1:0]	Select detection mode for SEI 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

12.3.1.5 S_GPT_INTSELn : Safety GPT Interrupt Select Register n (n = 0, 1)

Base address: ICU_S = 0x812A_0000

Offset address: 0x020 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	INTSELY_3[3:0]				INTSELY_2[3:0]				INTSELY_1[3:0]				INTSELY_0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	INTSELX_3[3:0]				INTSELX_2[3:0]				INTSELX_1[3:0]				INTSELX_0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	INTSELX_0[3:0]	Select GPT event source to be output as GPTx_INT0 event	R/W
7:4	INTSELX_1[3:0]	Select GPT event source to be output as GPTx_INT1 event	R/W
11:8	INTSELX_2[3:0]	Select GPT event source to be output as GPTx_INT2 event	R/W
15:12	INTSELX_3[3:0]	Select GPT event source to be output as GPTx_INT3 event	R/W
19:16	INTSELY_0[3:0]	Select GPT event source to be output as GPTy_INT0 event	R/W
23:20	INTSELY_1[3:0]	Select GPT event source to be output as GPTy_INT1 event	R/W
27:24	INTSELY_2[3:0]	Select GPT event source to be output as GPTy_INT2 event	R/W
31:28	INTSELY_3[3:0]	Select GPT event source to be output as GPTy_INT3 event	R/W

Note: Relation among n, x and y is shown in the below table.

n	x	y
0	10_0	10_1
1	10_2	10_3

The S_GPT_INTSELn register specifies a Safety GPT (GPT10) event source to be output as GPTx/y_INT[3:0] event source for Cortex-A55 SPI and Cortex-R52 SPI. Table 12.4 shows which Safety GPT event source is selected by INTSEL bits.

Table 12.4 Safety GPT event source selected by INTSEL bits

	n = 0		n = 1	
	x	y	x	y
INTSEL bits	10_0	10_1	10_2	10_3
0x0	GPTx/y_CCMPA			
0x1	GPTx/y_CCMPB			
0x2	GPTx/y_CMPC			
0x3	GPTx/y_CMPD			
0x4	GPTx/y_CMPE			
0x5	GPTx/y_CMPF			
0x6	GPTx/y_OVF			
0x7	GPTx/y_UDF			
0x8 to 0xF	0			

12.3.1.6 S_GPT_INTMSKn : Safety GPT Combined Interrupt Mask Register n (n = 0, 1)

Base address: ICU_S = 0x812A_0000

Offset address: 0x030 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	IY_MK7	IY_MK6	IY_MK5	IY_MK4	IY_MK3	IY_MK2	IY_MK1	IY_MK0
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IX_MK7	IX_MK6	IX_MK5	IX_MK4	IX_MK3	IX_MK2	IX_MK1	IX_MK0
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	IX_MK0	Mask GPTx_CCMPA event source from GPT combined event, GPTx_INT4 0: No masked 1: Masked	R/W
1	IX_MK1	Mask GPTx_CCMPB event source from GPT combined event, GPTx_INT4 0: No masked 1: Masked	R/W
2	IX_MK2	Mask GPTx_CMPC event source from GPT combined event, GPTx_INT4 0: No masked 1: Masked	R/W
3	IX_MK3	Mask GPTx_CMPD event source from GPT combined event, GPTx_INT4 0: No masked 1: Masked	R/W
4	IX_MK4	Mask GPTx_CMPE event source from GPT combined event, GPTx_INT4 0: No masked 1: Masked	R/W
5	IX_MK5	Mask GPTx_CMPF event source from GPT combined event, GPTx_INT4 0: No masked 1: Masked	R/W
6	IX_MK6	Mask GPTx_OVF event source from GPT combined event, GPTx_INT4 0: No masked 1: Masked	R/W
7	IX_MK7	Mask GPTx_UDF event source from GPT combined event, GPTx_INT4 0: No masked 1: Masked	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	IY_MK0	Mask GPTy_CCMPA event source from GPT combined event, GPTy_INT4 0: No masked 1: Masked	R/W
17	IY_MK1	Mask GPTy_CCMPB event source from GPT combined event, GPTy_INT4 0: No masked 1: Masked	R/W
18	IY_MK2	Mask GPTy_CMPC event source from GPT combined event, GPTy_INT4 0: No masked 1: Masked	R/W
19	IY_MK3	Mask GPTy_CMPD event source from GPT combined event, GPTy_INT4 0: No masked 1: Masked	R/W
20	IY_MK4	Mask GPTy_CMPE event source from GPT combined event, GPTy_INT4 0: No masked 1: Masked	R/W

Bit	Symbol	Function	R/W
21	IY_MK5	Mask GPTy_CMPF event source from GPT combined event, GPTy_INT4 0: No masked 1: Masked	R/W
22	IY_MK6	Mask GPTy_OVF event source from GPT combined event, GPTy_INT4 0: No masked 1: Masked	R/W
23	IY_MK7	Mask GPTy_UDF event source from GPT combined event, GPTy_INT4 0: No masked 1: Masked	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: Relation among n, x and y is shown in the below table.

n	x	y
0	10_0	10_1
1	10_2	10_3

The S_GPT_INTMSKn register is used to mask Safety GPT (GPT10) event sources to be combined as GPTx/y_INT4 event source for Cortex-A55 SPI and Cortex-R52 SPI. When the bit is masked, corresponding event does not occur GPTx/y_INT4 event.

12.3.1.7 S_GPT_INTCLRn : Safety GPT Combined Interrupt Status Clear Register n (n = 0, 1)

Base address: ICU_S = 0x812A_0000

Offset address: 0x038 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	IY_CL7	IY_CL6	IY_CL5	IY_CL4	IY_CL3	IY_CL2	IY_CL1	IY_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IX_CL7	IX_CL6	IX_CL5	IX_CL4	IX_CL3	IX_CL2	IX_CL1	IX_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IX_CL0	Clear GPTx_CCMPA event status captured in S_GPT_INTSTATn register by writing 1	W
1	IX_CL1	Clear GPTx_CCMPB event status captured in S_GPT_INTSTATn register by writing 1	W
2	IX_CL2	Clear GPTx_CMPC event status captured in S_GPT_INTSTATn register by writing 1	W
3	IX_CL3	Clear GPTx_CMPD event status captured in S_GPT_INTSTATn register by writing 1	W
4	IX_CL4	Clear GPTx_CMPE event status captured in S_GPT_INTSTATn register by writing 1	W
5	IX_CL5	Clear GPTx_CMPF event status captured in S_GPT_INTSTATn register by writing 1	W
6	IX_CL6	Clear GPTx_OVF event status captured in S_GPT_INTSTATn register by writing 1	W
7	IX_CL7	Clear GPTx_UDF event status captured in S_GPT_INTSTATn register by writing 1	W
15:8	—	The write value should be 0.	W
16	IY_CL0	Clear GPTy_CCMPA event status captured in S_GPT_INTSTATn register by writing 1	W
17	IY_CL1	Clear GPTy_CCMPB event status captured in S_GPT_INTSTATn register by writing 1	W
18	IY_CL2	Clear GPTy_CMPC event status captured in S_GPT_INTSTATn register by writing 1	W
19	IY_CL3	Clear GPTy_CMPD event status captured in S_GPT_INTSTATn register by writing 1	W

Bit	Symbol	Function	R/W
20	IY_CL4	Clear GPTy_CMPE event status captured in S_GPT_INTSTATn register by writing 1	W
21	IY_CL5	Clear GPTy_CMPF event status captured in S_GPT_INTSTATn register by writing 1	W
22	IY_CL6	Clear GPTy_OVF event status captured in S_GPT_INTSTATn register by writing 1	W
23	IY_CL7	Clear GPTy_UDF event status captured in S_GPT_INTSTATn register by writing 1	W
31:24	—	The write value should be 0.	W

Note: Relation among n, x and y is shown in the below table.

n	x	y
0	10_0	10_1
1	10_2	10_3

The S_GPT_INTCLRn register is used to clear Safety GPT (GPT10) event status captured in S_GPT_INTSTATn register.

12.3.1.8 S_GPT_INTSTATn : Safety GPT Combined Interrupt Status Register n (n = 0, 1)

Base address: ICU_S = 0x812A_0000

Offset address: 0x040 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	IY_ST7	IY_ST6	IY_ST5	IY_ST4	IY_ST3	IY_ST2	IY_ST1	IY_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IX_ST7	IX_ST6	IX_ST5	IX_ST4	IX_ST3	IX_ST2	IX_ST1	IX_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IX_ST0	Indicate captured GPTx_CCMPA event status 0: No event occurs 1: Event occurs	R
1	IX_ST1	Indicate captured GPTx_CCMPB event status 0: No event occurs 1: Event occurs	R
2	IX_ST2	Indicate captured GPTx_CMPC event status 0: No event occurs 1: Event occurs	R
3	IX_ST3	Indicate captured GPTx_CMPD event status 0: No event occurs 1: Event occurs	R
4	IX_ST4	Indicate captured GPTx_CMPE event status 0: No event occurs 1: Event occurs	R
5	IX_ST5	Indicate captured GPTx_CMPF event status 0: No event occurs 1: Event occurs	R
6	IX_ST6	Indicate captured GPTx_OVF event status 0: No event occurs 1: Event occurs	R
7	IX_ST7	Indicate captured GPTx_UDF event status 0: No event occurs 1: Event occurs	R

Bit	Symbol	Function	R/W
15:8	—	These bits are read as 0.	R
16	IY_ST0	Indicate captured GPTy_CCMPA event status 0: No event occurs 1: Event occurs	R
17	IY_ST1	Indicate captured GPTy_CCMPB event status 0: No event occurs 1: Event occurs	R
18	IY_ST2	Indicate captured GPTy_CMPC event status 0: No event occurs 1: Event occurs	R
19	IY_ST3	Indicate captured GPTy_CMPD event status 0: No event occurs 1: Event occurs	R
20	IY_ST4	Indicate captured GPTy_CMPE event status 0: No event occurs 1: Event occurs	R
21	IY_ST5	Indicate captured GPTy_CMPF event status 0: No event occurs 1: Event occurs	R
22	IY_ST6	Indicate captured GPTy_OVF event status 0: No event occurs 1: Event occurs	R
23	IY_ST7	Indicate captured GPTy_UDF event status 0: No event occurs 1: Event occurs	R
31:24	—	These bits are read as 0.	R

Note: Relation among n, x and y is shown in the below table.

n	x	y
0	10_0	10_1
1	10_2	10_3

The S_GPT_INTSTATn register indicates captured status for Safety GPT (GPT10) events for Cortex-A55 SPI and Cortex-R52 SPI.

12.3.2 Non-Safety Region Registers

12.3.2.1 NS_SWINT : Software Interrupt Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IC13	IC12	IC11	IC10	IC9	IC8	IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IC0	Software Interrupt register Occur software interrupt INTCPU0 when this value is set to 1. This bit is read as 0.	W
1	IC1	Software Interrupt register Occur software interrupt INTCPU1 when this value is set to 1. This bit is read as 0.	W
2	IC2	Software Interrupt register Occur software interrupt INTCPU2 when this value is set to 1. This bit is read as 0.	W
3	IC3	Software Interrupt register Occur software interrupt INTCPU3 when this value is set to 1. This bit is read as 0.	W
4	IC4	Software Interrupt register Occur software interrupt INTCPU4 when this value is set to 1. This bit is read as 0.	W
5	IC5	Software Interrupt register Occur software interrupt INTCPU5 when this value is set to 1. This bit is read as 0.	W
6	IC6	Software Interrupt register Occur software interrupt INTCPU6 when this value is set to 1. This bit is read as 0.	W
7	IC7	Software Interrupt register Occur software interrupt INTCPU7 when this value is set to 1. This bit is read as 0.	W
8	IC8	Software Interrupt register Occur software interrupt INTCPU8 when this value is set to 1. This bit is read as 0.	W
9	IC9	Software Interrupt register Occur software interrupt INTCPU9 when this value is set to 1. This bit is read as 0.	W
10	IC10	Software Interrupt register Occur software interrupt INTCPU10 when this value is set to 1. This bit is read as 0.	W
11	IC11	Software Interrupt register Occur software interrupt INTCPU11 when this value is set to 1. This bit is read as 0.	W
12	IC12	Software Interrupt register Occur software interrupt INTCPU12 when this value is set to 1. This bit is read as 0.	W
13	IC13	Software Interrupt register Occur software interrupt INTCPU13 when this value is set to 1. This bit is read as 0.	W
31:14	—	The write value should be 0.	W

12.3.2.2 NS_PORTNF_FLTSEL : Interrupt Noise Filter Enable Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	FLTDRQ	FLT13	FLT12	FLT11	FLT10	FLT9	FLT8	FLT7	FLT6	FLT5	FLT4	FLT3	FLT2	FLT1	FLT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLT0	Noise filter enable for IRQ0 0: Disable 1: Enable	R/W
1	FLT1	Noise filter enable for IRQ1 0: Disable 1: Enable	R/W
2	FLT2	Noise filter enable for IRQ2 0: Disable 1: Enable	R/W
3	FLT3	Noise filter enable for IRQ3 0: Disable 1: Enable	R/W
4	FLT4	Noise filter enable for IRQ4 0: Disable 1: Enable	R/W
5	FLT5	Noise filter enable for IRQ5 0: Disable 1: Enable	R/W
6	FLT6	Noise filter enable for IRQ6 0: Disable 1: Enable	R/W
7	FLT7	Noise filter enable for IRQ7 0: Disable 1: Enable	R/W
8	FLT8	Noise filter enable for IRQ8 0: Disable 1: Enable	R/W
9	FLT9	Noise filter enable for IRQ9 0: Disable 1: Enable	R/W
10	FLT10	Noise filter enable for IRQ10 0: Disable 1: Enable	R/W
11	FLT11	Noise filter enable for IRQ11 0: Disable 1: Enable	R/W
12	FLT12	Noise filter enable for IRQ12 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
13	FLT13	Noise filter enable for IRQ13 0: Disable 1: Enable	R/W
14	FLTDRQ	Noise filter enable for External DMA request (DREQ) 0: Disable 1: Enable	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

12.3.2.3 NS_PORTNF_CLKSEL : Interrupt Noise Filter Setting Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	CKSELDREQ[1:0]		CKSEL13[1:0]		CKSEL12[1:0]		CKSEL11[1:0]		CKSEL10[1:0]		CKSEL9[1:0]		CKSEL8[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CKSEL7[1:0]		CKSEL6[1:0]		CKSEL5[1:0]		CKSEL4[1:0]		CKSEL3[1:0]		CKSEL2[1:0]		CKSEL1[1:0]		CKSEL0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKSEL0[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ0. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
3:2	CKSEL1[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ1. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
5:4	CKSEL2[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ2. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
7:6	CKSEL3[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ3. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
9:8	CKSEL4[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ4. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W

Bit	Symbol	Function	R/W
11:10	CKSEL5[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ5. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
13:12	CKSEL6[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ6. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
15:14	CKSEL7[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ7. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
17:16	CKSEL8[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ8. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
19:18	CKSEL9[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ9. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
21:20	CKSEL10[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ10. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
23:22	CKSEL11[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ11. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
25:24	CKSEL12[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ12. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W

Bit	Symbol	Function	R/W
27:26	CKSEL13[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ13. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
29:28	CKSELDREQ[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for DREQ of DMAC. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

12.3.2.4 NS_PORTNF_MD : Interrupt Edge Detection Setting Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	MDDRQ[1:0]		MD13[1:0]		MD12[1:0]		MD11[1:0]		MD10[1:0]		MD9[1:0]		MD8[1:0]	
Value after reset:	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MD7[1:0]		MD6[1:0]		MD5[1:0]		MD4[1:0]		MD3[1:0]		MD2[1:0]		MD1[1:0]		MD0[1:0]	
Value after reset:	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit	Symbol	Function	R/W
1:0	MD0[1:0]	Select detection mode for IRQ0 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
3:2	MD1[1:0]	Select detection mode for IRQ1 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
5:4	MD2[1:0]	Select detection mode for IRQ2 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
7:6	MD3[1:0]	Select detection mode for IRQ3 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
9:8	MD4[1:0]	Select detection mode for IRQ4 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W

Bit	Symbol	Function	R/W
11:10	MD5[1:0]	Select detection mode for IRQ5 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
13:12	MD6[1:0]	Select detection mode for IRQ6 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
15:14	MD7[1:0]	Select detection mode for IRQ7 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
17:16	MD8[1:0]	Select detection mode for IRQ8 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
19:18	MD9[1:0]	Select detection mode for IRQ9 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
21:20	MD10[1:0]	Select detection mode for IRQ10 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
23:22	MD11[1:0]	Select detection mode for IRQ11 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
25:24	MD12[1:0]	Select detection mode for IRQ12 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
27:26	MD13[1:0]	Select detection mode for IRQ13 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
29:28	MDDRQ[1:0]	Select detection mode for DREQ of DMAC 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

12.3.2.5 CA55ERR_E0MSK : CA55 E0 Error Event Mask Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	E0_MK12	E0_MK11	E0_MK10	E0_MK9	E0_MK8	E0_MK7	E0_MK6	E0_MK5	E0_MK4	E0_MK3	E0_MK2	E0_MK1	E0_MK0
Value after reset:	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E0_MK0	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
1	E0_MK1	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
2	E0_MK2	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
3	E0_MK3	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
4	E0_MK4	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
5	E0_MK5	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
6	E0_MK6	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
7	E0_MK7	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
8	E0_MK8	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
9	E0_MK9	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
10	E0_MK10	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
11	E0_MK11	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
12	E0_MK12	Mask captured error status as an CA55_ERR0 event for CA55ERR_STAT 0: Release 1: Mask	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

The CA55ERR_E0MSK register is used to mask error status for CA55_ERR0 event. When the bit is masked, corresponding error does not occur CA55_ERR0 event.

12.3.2.6 CA55ERR_E1MSK : CA55 E1 Error Event Mask Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x054

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	E1_MK12	E1_MK11	E1_MK10	E1_MK9	E1_MK8	E1_MK7	E1_MK6	E1_MK5	E1_MK4	E1_MK3	E1_MK2	E1_MK1	E1_MK0
Value after reset:	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E1_MK0	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
1	E1_MK1	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
2	E1_MK2	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
3	E1_MK3	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
4	E1_MK4	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
5	E1_MK5	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
6	E1_MK6	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
7	E1_MK7	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
8	E1_MK8	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
9	E1_MK9	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
10	E1_MK10	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
11	E1_MK11	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
12	E1_MK12	Mask captured error status as an CA55_ERR1 event for CA55ERR_STAT 0: Release 1: Mask	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

The CA55ERR_E1MSK register is used to mask error status for CA55_ERR1 event. When the bit is masked, corresponding error does not occur CA55_ERR1 event.

12.3.2.7 CA55ERR_RSTMSK : CA55 Error Event Reset Mask Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x058

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	RS_MK12	RS_MK11	RS_MK10	RS_MK9	RS_MK8	RS_MK7	RS_MK6	RS_MK5	RS_MK4	RS_MK3	RS_MK2	RS_MK1	RS_MK0
Value after reset:	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	RS_MK0	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
1	RS_MK1	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
2	RS_MK2	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
3	RS_MK3	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
4	RS_MK4	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
5	RS_MK5	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
6	RS_MK6	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
7	RS_MK7	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
8	RS_MK8	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
9	RS_MK9	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
10	RS_MK10	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
11	RS_MK11	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
12	RS_MK12	Mask captured error status as a reset event for CA55ERR_STAT 0: Release 1: Mask	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

The CA55ERR_RSTMSK register is used to mask error status for reset event. When the bit is masked, corresponding error does not occur reset event.

12.3.2.8 CA55ERR_CLR : CA55 Error Event Status Clear Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	ER_CL12	ER_CL11	ER_CL10	ER_CL9	ER_CL8	ER_CL7	ER_CL6	ER_CL5	ER_CL4	ER_CL3	ER_CL2	ER_CL1	ER_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_CL0	Clear captured error status for CA55ERR_STAT register by writing 1	W
1	ER_CL1	Clear captured error status for CA55ERR_STAT register by writing 1	W
2	ER_CL2	Clear captured error status for CA55ERR_STAT register by writing 1	W
3	ER_CL3	Clear captured error status for CA55ERR_STAT register by writing 1	W
4	ER_CL4	Clear captured error status for CA55ERR_STAT register by writing 1	W
5	ER_CL5	Clear captured error status for CA55ERR_STAT register by writing 1	W
6	ER_CL6	Clear captured error status for CA55ERR_STAT register by writing 1	W
7	ER_CL7	Clear captured error status for CA55ERR_STAT register by writing 1	W
8	ER_CL8	Clear captured error status for CA55ERR_STAT register by writing 1	W
9	ER_CL9	Clear captured error status for CA55ERR_STAT register by writing 1	W
10	ER_CL10	Clear captured error status for CA55ERR_STAT register by writing 1	W
11	ER_CL11	Clear captured error status for CA55ERR_STAT register by writing 1	W
12	ER_CL12	Clear captured error status for CA55ERR_STAT register by writing 1	W
31:13	—	The write value should be 0.	W

The CA55ERR_CLR register is used to clear error status captured in CA55ERR_STAT register.

12.3.2.9 CA55ERR_STAT : CA55 Error Event Status Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	ER_ST12	ER_ST11	ER_ST10	ER_ST9	ER_ST8	ER_ST7	ER_ST6	ER_ST5	ER_ST4	ER_ST3	ER_ST2	ER_ST1	ER_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_ST0	Indicate captured error status for nFAULTIRQ1 from Cortex-A55 0: No error 1: Error occurred	R
1	ER_ST1	Indicate captured error status for nFAULTIRQ2 from Cortex-A55 0: No error 1: Error occurred	R
2	ER_ST2	Indicate captured error status for nFAULTIRQ3 from Cortex-A55 0: No error 1: Error occurred	R
3	ER_ST3	Indicate captured error status for nFAULTIRQ4 from Cortex-A55 0: No error 1: Error occurred	R
4	ER_ST4	Indicate captured error status for nFAULTIRQ0 from Cortex-A55 0: No error 1: Error occurred	R
5	ER_ST5	Indicate captured error status for nERRIRQ1 from Cortex-A55 0: No error 1: Error occurred	R
6	ER_ST6	Indicate captured error status for nERRIRQ2 from Cortex-A55 0: No error 1: Error occurred	R
7	ER_ST7	Indicate captured error status for nERRIRQ3 from Cortex-A55 0: No error 1: Error occurred	R
8	ER_ST8	Indicate captured error status for nERRIRQ4 from Cortex-A55 0: No error 1: Error occurred	R
9	ER_ST9	Indicate captured error status for nERRIRQ0 from Cortex-A55 0: No error 1: Error occurred	R
10	ER_ST10	Indicate captured error status for fault_int from GIC-600 0: No error 1: Error occurred	R
11	ER_ST11	Indicate captured error status for err_int from GIC-600 0: No error 1: Error occurred	R
12	ER_ST12	Indicate captured error status for pmu_int from GIC-600 0: No error 1: Error occurred	R
31:13	—	These bits are read as 0.	R

The CA55ERR_STAT register indicates captured status for error events from Cortex-A55 and GIC-600.

12.3.2.10 CR520ERR_E0MSK : CR52 CPU0 E0 Error Event Mask Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x068

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	E0_MK25	E0_MK24	E0_MK23	E0_MK22	E0_MK21	E0_MK20	E0_MK19	E0_MK18	E0_MK17	E0_MK16
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E0_MK15	E0_MK14	E0_MK13	E0_MK12	E0_MK11	E0_MK10	E0_MK9	E0_MK8	E0_MK7	E0_MK6	E0_MK5	E0_MK4	E0_MK3	E0_MK2	E0_MK1	E0_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E0_MK0	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
1	E0_MK1	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
2	E0_MK2	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
3	E0_MK3	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
4	E0_MK4	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
5	E0_MK5	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
6	E0_MK6	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
7	E0_MK7	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
8	E0_MK8	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
9	E0_MK9	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
10	E0_MK10	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
11	E0_MK11	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
12	E0_MK12	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
13	E0_MK13	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
14	E0_MK14	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
15	E0_MK15	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
16	E0_MK16	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
17	E0_MK17	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
18	E0_MK18	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
19	E0_MK19	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
20	E0_MK20	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
21	E0_MK21	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
22	E0_MK22	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
23	E0_MK23	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
24	E0_MK24	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
25	E0_MK25	Mask captured error status as an CR520_ERR0 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

12.3.2.11 CR520ERR_E1MSK : CR52 CPU0 E1 Error Event Mask Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x06C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	E1_MK25	E1_MK24	E1_MK23	E1_MK22	E1_MK21	E1_MK20	E1_MK19	E1_MK18	E1_MK17	E1_MK16
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E1_MK15	E1_MK14	E1_MK13	E1_MK12	E1_MK11	E1_MK10	E1_MK9	E1_MK8	E1_MK7	E1_MK6	E1_MK5	E1_MK4	E1_MK3	E1_MK2	E1_MK1	E1_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E1_MK0	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
1	E1_MK1	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
2	E1_MK2	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
3	E1_MK3	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
4	E1_MK4	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
5	E1_MK5	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
6	E1_MK6	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
7	E1_MK7	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
8	E1_MK8	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
9	E1_MK9	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
10	E1_MK10	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
11	E1_MK11	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
12	E1_MK12	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
13	E1_MK13	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
14	E1_MK14	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
15	E1_MK15	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
16	E1_MK16	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
17	E1_MK17	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
18	E1_MK18	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
19	E1_MK19	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
20	E1_MK20	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
21	E1_MK21	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
22	E1_MK22	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
23	E1_MK23	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
24	E1_MK24	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
25	E1_MK25	Mask captured error status as an CR520_ERR1 event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

12.3.2.12 CR520ERR_RSTMSK : CR52 CPU0 Error Event Reset Mask Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	RS_MK25	RS_MK24	RS_MK23	RS_MK22	RS_MK21	RS_MK20	RS_MK19	RS_MK18	RS_MK17	RS_MK16
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RS_MK15	RS_MK14	RS_MK13	RS_MK12	RS_MK11	RS_MK10	RS_MK9	RS_MK8	RS_MK7	RS_MK6	RS_MK5	RS_MK4	RS_MK3	RS_MK2	RS_MK1	RS_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	RS_MK0	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
1	RS_MK1	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
2	RS_MK2	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
3	RS_MK3	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
4	RS_MK4	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
5	RS_MK5	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
6	RS_MK6	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
7	RS_MK7	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
8	RS_MK8	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
9	RS_MK9	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
10	RS_MK10	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
11	RS_MK11	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
12	RS_MK12	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
13	RS_MK13	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
14	RS_MK14	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
15	RS_MK15	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
16	RS_MK16	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
17	RS_MK17	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
18	RS_MK18	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
19	RS_MK19	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
20	RS_MK20	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
21	RS_MK21	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
22	RS_MK22	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
23	RS_MK23	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
24	RS_MK24	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
25	RS_MK25	Mask captured error status as a reset event for NS_CR520ERR_STAT 0: Release 1: Mask	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

12.3.2.13 CR520ERR_CLR : CR52 CPU0 Error Event Status Clear Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x078

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ER_CL25	ER_CL24	ER_CL23	ER_CL22	ER_CL21	ER_CL20	ER_CL19	ER_CL18	ER_CL17	ER_CL16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_CL15	ER_CL14	ER_CL13	ER_CL12	ER_CL11	ER_CL10	ER_CL9	ER_CL8	ER_CL7	ER_CL6	ER_CL5	ER_CL4	ER_CL3	ER_CL2	ER_CL1	ER_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_CL0	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
1	ER_CL1	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
2	ER_CL2	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
3	ER_CL3	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
4	ER_CL4	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
5	ER_CL5	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
6	ER_CL6	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
7	ER_CL7	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
8	ER_CL8	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
9	ER_CL9	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
10	ER_CL10	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
11	ER_CL11	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
12	ER_CL12	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
13	ER_CL13	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
14	ER_CL14	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
15	ER_CL15	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
16	ER_CL16	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
17	ER_CL17	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
18	ER_CL18	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W

Bit	Symbol	Function	R/W
19	ER_CL19	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
20	ER_CL20	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
21	ER_CL21	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
22	ER_CL22	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
23	ER_CL23	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
24	ER_CL24	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
25	ER_CL25	Clear captured error status for NS_CR520ERR_STAT register by writing 1	W
31:26	—	The write value should be 0.	W

12.3.2.14 CR520ERR_STAT : CR52 CPU0 Error Event Status Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x07C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ER_ST25	ER_ST24	ER_ST23	ER_ST22	ER_ST21	ER_ST20	ER_ST19	ER_ST18	ER_ST17	ER_ST16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_ST15	ER_ST14	ER_ST13	ER_ST12	ER_ST11	ER_ST10	ER_ST9	ER_ST8	ER_ST7	ER_ST6	ER_ST5	ER_ST4	ER_ST3	ER_ST2	ER_ST1	ER_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_ST0	Indicate captured error status for ERREVENT0 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
1	ER_ST1	Indicate captured error status for ERREVENT1 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
2	ER_ST2	Indicate captured error status for ERREVENT2 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
3	ER_ST3	Indicate captured error status for ERREVENT3 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
4	ER_ST4	Indicate captured error status for ERREVENT4 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
5	ER_ST5	Indicate captured error status for ERREVENT5 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
6	ER_ST6	Indicate captured error status for ERREVENT6 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
7	ER_ST7	Indicate captured error status for ERREVENT7 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
8	ER_ST8	Indicate captured error status for ERREVENT8 from Cortex-R52 CPU0 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
9	ER_ST9	Indicate captured error status for ERREVENT9 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
10	ER_ST10	Indicate captured error status for ERREVENT10 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
11	ER_ST11	Indicate captured error status for ERREVENT11 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
12	ER_ST12	Indicate captured error status for ERREVENT12 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
13	ER_ST13	Indicate captured error status for ERREVENT13 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
14	ER_ST14	Indicate captured error status for ERREVENT14 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
15	ER_ST15	Indicate captured error status for ERREVENT15 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
16	ER_ST16	Indicate captured error status for ERREVENT16 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
17	ER_ST17	Indicate captured error status for ERREVENT17 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
18	ER_ST18	Indicate captured error status for ERREVENT18 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
19	ER_ST19	Indicate captured error status for ERREVENT19 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
20	ER_ST20	Indicate captured error status for ERREVENT20 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
21	ER_ST21	Indicate captured error status for ERREVENT21 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
22	ER_ST22	Indicate captured error status for ERREVENT22 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
23	ER_ST23	Indicate captured error status for ERREVENT23 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
24	ER_ST24	Indicate captured error status for ERREVENT24 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
25	ER_ST25	Indicate captured error status for ERREVENT25 from Cortex-R52 CPU0 0: No error 1: Error occurred	R
31:26	—	These bits are read as 0.	R

12.3.2.15 CR521ERR_E0MSK : CR52 CPU1 E0 Error Event Mask Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	E0_MK25	E0_MK24	E0_MK23	E0_MK22	E0_MK21	E0_MK20	E0_MK19	E0_MK18	E0_MK17	E0_MK16
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E0_MK15	E0_MK14	E0_MK13	E0_MK12	E0_MK11	E0_MK10	E0_MK9	E0_MK8	E0_MK7	E0_MK6	E0_MK5	E0_MK4	E0_MK3	E0_MK2	E0_MK1	E0_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E0_MK0	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
1	E0_MK1	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
2	E0_MK2	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
3	E0_MK3	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
4	E0_MK4	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
5	E0_MK5	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
6	E0_MK6	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
7	E0_MK7	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
8	E0_MK8	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
9	E0_MK9	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
10	E0_MK10	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
11	E0_MK11	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
12	E0_MK12	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
13	E0_MK13	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
14	E0_MK14	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
15	E0_MK15	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
16	E0_MK16	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
17	E0_MK17	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
18	E0_MK18	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
19	E0_MK19	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
20	E0_MK20	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
21	E0_MK21	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
22	E0_MK22	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
23	E0_MK23	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
24	E0_MK24	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
25	E0_MK25	Mask captured error status as an CR521_ERR0 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

12.3.2.16 CR521ERR_E1MSK : CR52 CPU1 E1 Error Event Mask Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	E1_MK25	E1_MK24	E1_MK23	E1_MK22	E1_MK21	E1_MK20	E1_MK19	E1_MK18	E1_MK17	E1_MK16
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E1_MK15	E1_MK14	E1_MK13	E1_MK12	E1_MK11	E1_MK10	E1_MK9	E1_MK8	E1_MK7	E1_MK6	E1_MK5	E1_MK4	E1_MK3	E1_MK2	E1_MK1	E1_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E1_MK0	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
1	E1_MK1	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
2	E1_MK2	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
3	E1_MK3	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
4	E1_MK4	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
5	E1_MK5	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
6	E1_MK6	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
7	E1_MK7	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
8	E1_MK8	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
9	E1_MK9	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
10	E1_MK10	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
11	E1_MK11	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
12	E1_MK12	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
13	E1_MK13	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
14	E1_MK14	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
15	E1_MK15	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
16	E1_MK16	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
17	E1_MK17	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
18	E1_MK18	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
19	E1_MK19	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
20	E1_MK20	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
21	E1_MK21	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
22	E1_MK22	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
23	E1_MK23	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
24	E1_MK24	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
25	E1_MK25	Mask captured error status as an CR521_ERR1 event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

12.3.2.17 CR521ERR_RSTMSK : CR52 CPU1 Error Event Reset Mask Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x088

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	RS_MK25	RS_MK24	RS_MK23	RS_MK22	RS_MK21	RS_MK20	RS_MK19	RS_MK18	RS_MK17	RS_MK16
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RS_MK15	RS_MK14	RS_MK13	RS_MK12	RS_MK11	RS_MK10	RS_MK9	RS_MK8	RS_MK7	RS_MK6	RS_MK5	RS_MK4	RS_MK3	RS_MK2	RS_MK1	RS_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	RS_MK0	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
1	RS_MK1	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
2	RS_MK2	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
3	RS_MK3	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
4	RS_MK4	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
5	RS_MK5	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
6	RS_MK6	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
7	RS_MK7	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
8	RS_MK8	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
9	RS_MK9	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
10	RS_MK10	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
11	RS_MK11	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
12	RS_MK12	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
13	RS_MK13	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
14	RS_MK14	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
15	RS_MK15	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
16	RS_MK16	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
17	RS_MK17	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
18	RS_MK18	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
19	RS_MK19	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
20	RS_MK20	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
21	RS_MK21	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
22	RS_MK22	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
23	RS_MK23	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
24	RS_MK24	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
25	RS_MK25	Mask captured error status as a reset event for NS_CR521ERR_STAT 0: Release 1: Mask	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

12.3.2.18 CR521ERR_CLR : CR52 CPU1 Error Event Status Clear Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ER_CL25	ER_CL24	ER_CL23	ER_CL22	ER_CL21	ER_CL20	ER_CL19	ER_CL18	ER_CL17	ER_CL16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_CL15	ER_CL14	ER_CL13	ER_CL12	ER_CL11	ER_CL10	ER_CL9	ER_CL8	ER_CL7	ER_CL6	ER_CL5	ER_CL4	ER_CL3	ER_CL2	ER_CL1	ER_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_CL0	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
1	ER_CL1	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
2	ER_CL2	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
3	ER_CL3	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
4	ER_CL4	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
5	ER_CL5	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
6	ER_CL6	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
7	ER_CL7	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
8	ER_CL8	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
9	ER_CL9	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
10	ER_CL10	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
11	ER_CL11	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
12	ER_CL12	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
13	ER_CL13	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
14	ER_CL14	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
15	ER_CL15	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
16	ER_CL16	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
17	ER_CL17	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
18	ER_CL18	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W

Bit	Symbol	Function	R/W
19	ER_CL19	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
20	ER_CL20	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
21	ER_CL21	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
22	ER_CL22	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
23	ER_CL23	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
24	ER_CL24	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
25	ER_CL25	Clear captured error status for NS_CR521ERR_STAT register by writing 1	W
31:26	—	The write value should be 0.	W

12.3.2.19 CR521ERR_STAT : CR52 CPU1 Error Event Status Register

Base address: ICU_NS = 0x802A_0000

Offset address: 0x094

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ER_ST25	ER_ST24	ER_ST23	ER_ST22	ER_ST21	ER_ST20	ER_ST19	ER_ST18	ER_ST17	ER_ST16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_ST15	ER_ST14	ER_ST13	ER_ST12	ER_ST11	ER_ST10	ER_ST9	ER_ST8	ER_ST7	ER_ST6	ER_ST5	ER_ST4	ER_ST3	ER_ST2	ER_ST1	ER_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_ST0	Indicate captured error status for ERREVENT0 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
1	ER_ST1	Indicate captured error status for ERREVENT1 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
2	ER_ST2	Indicate captured error status for ERREVENT2 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
3	ER_ST3	Indicate captured error status for ERREVENT3 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
4	ER_ST4	Indicate captured error status for ERREVENT4 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
5	ER_ST5	Indicate captured error status for ERREVENT5 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
6	ER_ST6	Indicate captured error status for ERREVENT6 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
7	ER_ST7	Indicate captured error status for ERREVENT7 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
8	ER_ST8	Indicate captured error status for ERREVENT8 from Cortex-R52 CPU1 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
9	ER_ST9	Indicate captured error status for ERREVENT9 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
10	ER_ST10	Indicate captured error status for ERREVENT10 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
11	ER_ST11	Indicate captured error status for ERREVENT11 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
12	ER_ST12	Indicate captured error status for ERREVENT12 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
13	ER_ST13	Indicate captured error status for ERREVENT13 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
14	ER_ST14	Indicate captured error status for ERREVENT14 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
15	ER_ST15	Indicate captured error status for ERREVENT15 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
16	ER_ST16	Indicate captured error status for ERREVENT16 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
17	ER_ST17	Indicate captured error status for ERREVENT17 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
18	ER_ST18	Indicate captured error status for ERREVENT18 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
19	ER_ST19	Indicate captured error status for ERREVENT19 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
20	ER_ST20	Indicate captured error status for ERREVENT20 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
21	ER_ST21	Indicate captured error status for ERREVENT21 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
22	ER_ST22	Indicate captured error status for ERREVENT22 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
23	ER_ST23	Indicate captured error status for ERREVENT23 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
24	ER_ST24	Indicate captured error status for ERREVENT24 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
25	ER_ST25	Indicate captured error status for ERREVENT25 from Cortex-R52 CPU1 0: No error 1: Error occurred	R
31:26	—	These bits are read as 0.	R

12.3.2.20 PERIERR_E0MSKn : Peripheral E0 Error Event Mask Register n (n = 0 to 2)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x098 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E0_MK31	E0_MK30	E0_MK29	E0_MK28	E0_MK27	E0_MK26	E0_MK25	E0_MK24	E0_MK23	E0_MK22	E0_MK21	E0_MK20	E0_MK19	E0_MK18	E0_MK17	E0_MK16
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E0_MK15	E0_MK14	E0_MK13	E0_MK12	E0_MK11	E0_MK10	E0_MK9	E0_MK8	E0_MK7	E0_MK6	E0_MK5	E0_MK4	E0_MK3	E0_MK2	E0_MK1	E0_MK0
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E0_MK0	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
1	E0_MK1	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
2	E0_MK2	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
3	E0_MK3	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
4	E0_MK4	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
5	E0_MK5	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
6	E0_MK6	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
7	E0_MK7	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
8	E0_MK8	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
9	E0_MK9	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
10	E0_MK10	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
11	E0_MK11	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
12	E0_MK12	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
13	E0_MK13	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
14	E0_MK14	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
15	E0_MK15	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
16	E0_MK16	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
17	E0_MK17	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
18	E0_MK18	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
19	E0_MK19	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
20	E0_MK20	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
21	E0_MK21	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
22	E0_MK22	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
23	E0_MK23	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
24	E0_MK24	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
25	E0_MK25	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
26	E0_MK26	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
27	E0_MK27	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
28	E0_MK28	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
29	E0_MK29	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W
30	E0_MK30	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
31	E0_MK31	Mask captured error status as PERI_ERR0 event for PERIERR_STATn 0: Release 1: Mask	R/W

Note 1. If bit of status register is reserved, value after reset of the corresponding bit of mask register is 0.

The PERIERR_E0MSKn register is used to mask error status for PERI_ERR0 event. When the bit is masked, corresponding error does not occur PERI_ERR0 event.

12.3.2.21 PERIERR_E1MSKn : Peripheral E1 Error Event Mask Register n (n = 0 to 2)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x0A4 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E1_MK31	E1_MK30	E1_MK29	E1_MK28	E1_MK27	E1_MK26	E1_MK25	E1_MK24	E1_MK23	E1_MK22	E1_MK21	E1_MK20	E1_MK19	E1_MK18	E1_MK17	E1_MK16
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E1_MK15	E1_MK14	E1_MK13	E1_MK12	E1_MK11	E1_MK10	E1_MK9	E1_MK8	E1_MK7	E1_MK6	E1_MK5	E1_MK4	E1_MK3	E1_MK2	E1_MK1	E1_MK0
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E1_MK0	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
1	E1_MK1	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
2	E1_MK2	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
3	E1_MK3	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
4	E1_MK4	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
5	E1_MK5	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
6	E1_MK6	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
7	E1_MK7	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
8	E1_MK8	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
9	E1_MK9	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
10	E1_MK10	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
11	E1_MK11	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
12	E1_MK12	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
13	E1_MK13	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
14	E1_MK14	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
15	E1_MK15	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
16	E1_MK16	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
17	E1_MK17	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
18	E1_MK18	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
19	E1_MK19	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
20	E1_MK20	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
21	E1_MK21	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
22	E1_MK22	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
23	E1_MK23	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
24	E1_MK24	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
25	E1_MK25	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
26	E1_MK26	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
27	E1_MK27	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
28	E1_MK28	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
29	E1_MK29	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
30	E1_MK30	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W
31	E1_MK31	Mask captured error status as PERI_ERR1 event for PERIERR_STATn 0: Release 1: Mask	R/W

Note 1. If bit of status register is reserved, value after reset of the corresponding bit of mask register is 0.

The PERIERR_E1MSKn register is used to mask error status for PERI_ERR1 event. When the bit is masked, corresponding error does not occur PERI_ERR1 event.

12.3.2.22 PERIERR_RSTMSKn : Peripheral Error Event Reset Mask Register n (n = 0 to 2)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x0B0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RS_MK31	RS_MK30	RS_MK29	RS_MK28	RS_MK27	RS_MK26	RS_MK25	RS_MK24	RS_MK23	RS_MK22	RS_MK21	RS_MK20	RS_MK19	RS_MK18	RS_MK17	RS_MK16
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RS_MK15	RS_MK14	RS_MK13	RS_MK12	RS_MK11	RS_MK10	RS_MK9	RS_MK8	RS_MK7	RS_MK6	RS_MK5	RS_MK4	RS_MK3	RS_MK2	RS_MK1	RS_MK0
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	RS_MK0	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
1	RS_MK1	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
2	RS_MK2	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
3	RS_MK3	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
4	RS_MK4	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
5	RS_MK5	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
6	RS_MK6	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
7	RS_MK7	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
8	RS_MK8	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
9	RS_MK9	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
10	RS_MK10	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
11	RS_MK11	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
12	RS_MK12	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
13	RS_MK13	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
14	RS_MK14	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
15	RS_MK15	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
16	RS_MK16	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
17	RS_MK17	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
18	RS_MK18	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
19	RS_MK19	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
20	RS_MK20	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
21	RS_MK21	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
22	RS_MK22	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
23	RS_MK23	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
24	RS_MK24	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
25	RS_MK25	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
26	RS_MK26	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
27	RS_MK27	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
28	RS_MK28	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
29	RS_MK29	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
30	RS_MK30	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W
31	RS_MK31	Mask captured error status as a reset event for PERIERR_STATn 0: Release 1: Mask	R/W

Note 1. If bit of status register is reserved, value after reset of the corresponding bit of mask register is 0.

The PERIERR_RSTMSKn register is used to mask error status for reset event. When the bit is masked, corresponding error does not occur reset event.

12.3.2.23 PERIERR_CLRn : Peripheral Error Event Status Clear Register n (n = 0 to 2)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x0C8 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_CL31	ER_CL30	ER_CL29	ER_CL28	ER_CL27	ER_CL26	ER_CL25	ER_CL24	ER_CL23	ER_CL22	ER_CL21	ER_CL20	ER_CL19	ER_CL18	ER_CL17	ER_CL16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_CL15	ER_CL14	ER_CL13	ER_CL12	ER_CL11	ER_CL10	ER_CL9	ER_CL8	ER_CL7	ER_CL6	ER_CL5	ER_CL4	ER_CL3	ER_CL2	ER_CL1	ER_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_CL0	Clear captured error status for PERIERR_STATn register by writing 1	W
1	ER_CL1	Clear captured error status for PERIERR_STATn register by writing 1	W
2	ER_CL2	Clear captured error status for PERIERR_STATn register by writing 1	W
3	ER_CL3	Clear captured error status for PERIERR_STATn register by writing 1	W
4	ER_CL4	Clear captured error status for PERIERR_STATn register by writing 1	W
5	ER_CL5	Clear captured error status for PERIERR_STATn register by writing 1	W
6	ER_CL6	Clear captured error status for PERIERR_STATn register by writing 1	W
7	ER_CL7	Clear captured error status for PERIERR_STATn register by writing 1	W
8	ER_CL8	Clear captured error status for PERIERR_STATn register by writing 1	W
9	ER_CL9	Clear captured error status for PERIERR_STATn register by writing 1	W

Bit	Symbol	Function	R/W
10	ER_CL10	Clear captured error status for PERIERR_STATn register by writing 1	W
11	ER_CL11	Clear captured error status for PERIERR_STATn register by writing 1	W
12	ER_CL12	Clear captured error status for PERIERR_STATn register by writing 1	W
13	ER_CL13	Clear captured error status for PERIERR_STATn register by writing 1	W
14	ER_CL14	Clear captured error status for PERIERR_STATn register by writing 1	W
15	ER_CL15	Clear captured error status for PERIERR_STATn register by writing 1	W
16	ER_CL16	Clear captured error status for PERIERR_STATn register by writing 1	W
17	ER_CL17	Clear captured error status for PERIERR_STATn register by writing 1	W
18	ER_CL18	Clear captured error status for PERIERR_STATn register by writing 1	W
19	ER_CL19	Clear captured error status for PERIERR_STATn register by writing 1	W
20	ER_CL20	Clear captured error status for PERIERR_STATn register by writing 1	W
21	ER_CL21	Clear captured error status for PERIERR_STATn register by writing 1	W
22	ER_CL22	Clear captured error status for PERIERR_STATn register by writing 1	W
23	ER_CL23	Clear captured error status for PERIERR_STATn register by writing 1	W
24	ER_CL24	Clear captured error status for PERIERR_STATn register by writing 1	W
25	ER_CL25	Clear captured error status for PERIERR_STATn register by writing 1	W
26	ER_CL26	Clear captured error status for PERIERR_STATn register by writing 1	W
27	ER_CL27	Clear captured error status for PERIERR_STATn register by writing 1	W
28	ER_CL28	Clear captured error status for PERIERR_STATn register by writing 1	W
29	ER_CL29	Clear captured error status for PERIERR_STATn register by writing 1	W
30	ER_CL30	Clear captured error status for PERIERR_STATn register by writing 1	W
31	ER_CL31	Clear captured error status for PERIERR_STATn register by writing 1	W

The PERIERR_CLRn register is used to clear error status captured in PERIERR_STATn register.

12.3.2.24 PERIERR_STATn : Peripheral Error Event Status Register n (n = 0 to 2)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x0D4 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_ST31	ER_ST30	ER_ST29	ER_ST28	ER_ST27	ER_ST26	ER_ST25	ER_ST24	ER_ST23	ER_ST22	ER_ST21	ER_ST20	ER_ST19	ER_ST18	ER_ST17	ER_ST16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_ST15	ER_ST14	ER_ST13	ER_ST12	ER_ST11	ER_ST10	ER_ST9	ER_ST8	ER_ST7	ER_ST6	ER_ST5	ER_ST4	ER_ST3	ER_ST2	ER_ST1	ER_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_ST0	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
1	ER_ST1	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
2	ER_ST2	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
3	ER_ST3	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
4	ER_ST4	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
5	ER_ST5	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
6	ER_ST6	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
7	ER_ST7	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
8	ER_ST8	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
9	ER_ST9	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
10	ER_ST10	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
11	ER_ST11	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
12	ER_ST12	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
13	ER_ST13	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
14	ER_ST14	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
15	ER_ST15	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
16	ER_ST16	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
17	ER_ST17	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
18	ER_ST18	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
19	ER_ST19	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
20	ER_ST20	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
21	ER_ST21	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
22	ER_ST22	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
23	ER_ST23	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
24	ER_ST24	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
25	ER_ST25	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
26	ER_ST26	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
27	ER_ST27	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
28	ER_ST28	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
29	ER_ST29	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
30	ER_ST30	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R
31	ER_ST31	Indicate captured error status for error event from peripheral modules 0: No error 1: Error occurred	R

The PERIERR_STATn register indicates captured status for error events from peripheral modules. Table 12.5 lists bit assignment of error events.

Table 12.5 Bit assignment of peripheral error event (1 of 2)

Bit	PERIERR_STATn		
	n = 0	n = 1	n = 2
0	CLMA0_INT	MPU_DMACR0	CR521MPUR
1	CLMA1_INT	MPU_DMACW0	CR521MPUW
2	CLMA2_INT	MPU_DMACR1	MAINBUSA_ERRINT
3	CLMA3_INT	MPU_DMACW1	MAINBUSR_ERRINT
4	CLMA4_INT	MPU_DMACR2	LLPPBUS_ERRINT
5	CLMA5_INT	MPU_DMACW2	TZC_DDR_A0A1
6	CLMA6_INT	MPU_GMACR0	TZC_DDR_A4
7	BSC_WTO	MPU_GMACW0	TZC_PCIE
8	DMAC0_ERR	MPU_GMACR1	TZC_DDR_R2

Table 12.5 Bit assignment of peripheral error event (2 of 2)

Bit	PERIERR_STATn		
	n = 0	n = 1	n = 2
9	DMAC1_ERR	MPU_GMACW1	TZC_DDR_R3
10	DMAC2_ERR	MPU_GMACR2	TZC_SYSRAM
11	WDT_NMIUNDFR0	MPU_GMACW2	TZC_BSC
12	WDT_NMIUNDFR1	MPU_USBH	TZC_XSPI
13	WDT_NMIUNDFA0	MPU_USBF	TZC_TCM
14	WDT_NMIUNDFA1	MPU_DBGR	MAINBUSA_SEC_ERRINT
15	WDT_NMIUNDFA2	MPU_DBGW	MAINBUSR_SEC_ERRINT
16	WDT_NMIUNDFA3	MPU_ETRR	PERI_SEC_ERRINT
17	USB_FDMAERR	MPU_ETRW	PCIE0_INT_SERR_RC
18	DOC_DOPCI	MPU_SHOSTIF	PCIE0_INT_SERR_COR_RC
19	SRAM0_IE1	MPU_LCDCR	PCIE0_INT_NON_FATAL_RC
20	SRAM0_IE2	MPU_LCDCW	PCIE0_INT_SERR_FATL_RC
21	SRAM0_OVF	MPU_SDHIR0	PCIE0_INT_AXI_ERR
22	SRAM1_IE1	MPU_SDHIW0	PCIE1_INT_SERR_RC
23	SRAM1_IE2	MPU_SDHIR1	PCIE1_INT_SERR_COR_RC
24	SRAM1_OVF	MPU_SDHIW1	PCIE1_INT_NON_FATAL_RC
25	SRAM2_IE1	MPU_PCIER0	PCIE1_INT_SERR_FATL_RC
26	SRAM2_IE2	MPU_PCIEW0	PCIE1_INT_AXI_ERR
27	SRAM2_OVF	MPU_PCIER1	perf_mon_data_status0
28	SRAM3_IE1	MPU_PCIEW1	perf_mon_data_status1
29	SRAM3_IE2	CA55MPUR	Reserved (Read as 0)
30	SRAM3_OVF	CA55MPUW	Reserved (Read as 0)
31	Reserved (Read as 0)	Reserved (Read as 0)	Reserved (Read as 0)

12.3.2.25 DSMIFERR_E0MSKn : DSMIF E0 Error Event Mask Register n (n = 0 to 11)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x0E0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E0_MK31	E0_MK30	E0_MK29	E0_MK28	E0_MK27	E0_MK26	E0_MK25	E0_MK24	E0_MK23	E0_MK22	E0_MK21	E0_MK20	E0_MK19	E0_MK18	E0_MK17	E0_MK16
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E0_MK15	E0_MK14	E0_MK13	E0_MK12	E0_MK11	E0_MK10	E0_MK9	E0_MK8	E0_MK7	E0_MK6	E0_MK5	E0_MK4	E0_MK3	E0_MK2	E0_MK1	E0_MK0
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E0_MK0	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
1	E0_MK1	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
2	E0_MK2	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
3	E0_MK3	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
4	E0_MK4	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
5	E0_MK5	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
6	E0_MK6	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
7	E0_MK7	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
8	E0_MK8	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
9	E0_MK9	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
10	E0_MK10	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
11	E0_MK11	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
12	E0_MK12	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
13	E0_MK13	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
14	E0_MK14	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
15	E0_MK15	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
16	E0_MK16	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
17	E0_MK17	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
18	E0_MK18	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
19	E0_MK19	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
20	E0_MK20	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
21	E0_MK21	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
22	E0_MK22	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
23	E0_MK23	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
24	E0_MK24	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
25	E0_MK25	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
26	E0_MK26	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
27	E0_MK27	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
28	E0_MK28	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
29	E0_MK29	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
30	E0_MK30	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
31	E0_MK31	Mask captured error status as DSMIF_ERR0 event for DSMIFERR_STATn 0: Release 1: Mask	R/W

Note 1. If bit of status register is reserved, value after reset of the corresponding bit of mask register is 0.

The DSMIFERR_E0MSKn register is used to mask error status for DSMIF_ERR0 event. When the bit is masked, corresponding error does not occur DSMIF_ERR0 event.

12.3.2.26 DSMIFERR_E1MSKn : DSMIF E1 Error Event Mask Register n (n = 0 to 11)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x110 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E1_MK31	E1_MK30	E1_MK29	E1_MK28	E1_MK27	E1_MK26	E1_MK25	E1_MK24	E1_MK23	E1_MK22	E1_MK21	E1_MK20	E1_MK19	E1_MK18	E1_MK17	E1_MK16
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E1_MK15	E1_MK14	E1_MK13	E1_MK12	E1_MK11	E1_MK10	E1_MK9	E1_MK8	E1_MK7	E1_MK6	E1_MK5	E1_MK4	E1_MK3	E1_MK2	E1_MK1	E1_MK0
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E1_MK0	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
1	E1_MK1	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
2	E1_MK2	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
3	E1_MK3	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
4	E1_MK4	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
5	E1_MK5	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
6	E1_MK6	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
7	E1_MK7	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
8	E1_MK8	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
9	E1_MK9	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
10	E1_MK10	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
11	E1_MK11	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
12	E1_MK12	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
13	E1_MK13	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
14	E1_MK14	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
15	E1_MK15	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
16	E1_MK16	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
17	E1_MK17	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
18	E1_MK18	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
19	E1_MK19	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
20	E1_MK20	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
21	E1_MK21	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
22	E1_MK22	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
23	E1_MK23	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
24	E1_MK24	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
25	E1_MK25	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
26	E1_MK26	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
27	E1_MK27	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
28	E1_MK28	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
29	E1_MK29	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W
30	E1_MK30	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
31	E1_MK31	Mask captured error status as DSMIF_ERR1 event for DSMIFERR_STATn 0: Release 1: Mask	R/W

Note 1. If bit of status register is reserved, value after reset of the corresponding bit of mask register is 0.

The DSMIFERR_E1MSKn register is used to mask error status for DSMIF_ERR1 event. When the bit is masked, corresponding error does not occur DSMIF_ERR1 event.

12.3.2.27 DSMIFERR_RSTMSKn : DSMIF Error Event Reset Mask Register n (n = 0 to 11)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x140 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RS_MK31	RS_MK30	RS_MK29	RS_MK28	RS_MK27	RS_MK26	RS_MK25	RS_MK24	RS_MK23	RS_MK22	RS_MK21	RS_MK20	RS_MK19	RS_MK18	RS_MK17	RS_MK16
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RS_MK15	RS_MK14	RS_MK13	RS_MK12	RS_MK11	RS_MK10	RS_MK9	RS_MK8	RS_MK7	RS_MK6	RS_MK5	RS_MK4	RS_MK3	RS_MK2	RS_MK1	RS_MK0
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	RS_MK0	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
1	RS_MK1	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
2	RS_MK2	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
3	RS_MK3	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
4	RS_MK4	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
5	RS_MK5	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
6	RS_MK6	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
7	RS_MK7	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
8	RS_MK8	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
9	RS_MK9	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
10	RS_MK10	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
11	RS_MK11	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
12	RS_MK12	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
13	RS_MK13	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
14	RS_MK14	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
15	RS_MK15	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
16	RS_MK16	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
17	RS_MK17	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
18	RS_MK18	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
19	RS_MK19	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
20	RS_MK20	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
21	RS_MK21	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
22	RS_MK22	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
23	RS_MK23	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
24	RS_MK24	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
25	RS_MK25	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
26	RS_MK26	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
27	RS_MK27	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
28	RS_MK28	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
29	RS_MK29	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
30	RS_MK30	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W
31	RS_MK31	Mask captured error status as a reset event for DSMIFERR_STATn 0: Release 1: Mask	R/W

Note 1. If bit of status register is reserved, value after reset of the corresponding bit of mask register is 0.

The DSMIFERR_RSTMSKn register is used to mask error status for reset event. When the bit is masked, corresponding error does not occur reset event.

12.3.2.28 DSMIFERR_CLRn : DSMIF Error Event Status Clear Register n (n = 0 to 11)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x1A0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_CL31	ER_CL30	ER_CL29	ER_CL28	ER_CL27	ER_CL26	ER_CL25	ER_CL24	ER_CL23	ER_CL22	ER_CL21	ER_CL20	ER_CL19	ER_CL18	ER_CL17	ER_CL16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_CL15	ER_CL14	ER_CL13	ER_CL12	ER_CL11	ER_CL10	ER_CL9	ER_CL8	ER_CL7	ER_CL6	ER_CL5	ER_CL4	ER_CL3	ER_CL2	ER_CL1	ER_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_CL0	Clear captured error status for DSMIFERR_STATn register by writing 1	W
1	ER_CL1	Clear captured error status for DSMIFERR_STATn register by writing 1	W
2	ER_CL2	Clear captured error status for DSMIFERR_STATn register by writing 1	W
3	ER_CL3	Clear captured error status for DSMIFERR_STATn register by writing 1	W
4	ER_CL4	Clear captured error status for DSMIFERR_STATn register by writing 1	W
5	ER_CL5	Clear captured error status for DSMIFERR_STATn register by writing 1	W
6	ER_CL6	Clear captured error status for DSMIFERR_STATn register by writing 1	W
7	ER_CL7	Clear captured error status for DSMIFERR_STATn register by writing 1	W
8	ER_CL8	Clear captured error status for DSMIFERR_STATn register by writing 1	W
9	ER_CL9	Clear captured error status for DSMIFERR_STATn register by writing 1	W
10	ER_CL10	Clear captured error status for DSMIFERR_STATn register by writing 1	W
11	ER_CL11	Clear captured error status for DSMIFERR_STATn register by writing 1	W
12	ER_CL12	Clear captured error status for DSMIFERR_STATn register by writing 1	W
13	ER_CL13	Clear captured error status for DSMIFERR_STATn register by writing 1	W
14	ER_CL14	Clear captured error status for DSMIFERR_STATn register by writing 1	W
15	ER_CL15	Clear captured error status for DSMIFERR_STATn register by writing 1	W
16	ER_CL16	Clear captured error status for DSMIFERR_STATn register by writing 1	W

Bit	Symbol	Function	R/W
17	ER_CL17	Clear captured error status for DSMIFERR_STATn register by writing 1	W
18	ER_CL18	Clear captured error status for DSMIFERR_STATn register by writing 1	W
19	ER_CL19	Clear captured error status for DSMIFERR_STATn register by writing 1	W
20	ER_CL20	Clear captured error status for DSMIFERR_STATn register by writing 1	W
21	ER_CL21	Clear captured error status for DSMIFERR_STATn register by writing 1	W
22	ER_CL22	Clear captured error status for DSMIFERR_STATn register by writing 1	W
23	ER_CL23	Clear captured error status for DSMIFERR_STATn register by writing 1	W
24	ER_CL24	Clear captured error status for DSMIFERR_STATn register by writing 1	W
25	ER_CL25	Clear captured error status for DSMIFERR_STATn register by writing 1	W
26	ER_CL26	Clear captured error status for DSMIFERR_STATn register by writing 1	W
27	ER_CL27	Clear captured error status for DSMIFERR_STATn register by writing 1	W
28	ER_CL28	Clear captured error status for DSMIFERR_STATn register by writing 1	W
29	ER_CL29	Clear captured error status for DSMIFERR_STATn register by writing 1	W
30	ER_CL30	Clear captured error status for DSMIFERR_STATn register by writing 1	W
31	ER_CL31	Clear captured error status for DSMIFERR_STATn register by writing 1	W

The DSMIFERR_CLRn register is used to clear error status captured in DSMIFERR_STATn register.

12.3.2.29 DSMIFERR_STATn : DSMIF Error Event Status Register n (n = 0 to 11)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x1D0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_ST31	ER_ST30	ER_ST29	ER_ST28	ER_ST27	ER_ST26	ER_ST25	ER_ST24	ER_ST23	ER_ST22	ER_ST21	ER_ST20	ER_ST19	ER_ST18	ER_ST17	ER_ST16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_ST15	ER_ST14	ER_ST13	ER_ST12	ER_ST11	ER_ST10	ER_ST9	ER_ST8	ER_ST7	ER_ST6	ER_ST5	ER_ST4	ER_ST3	ER_ST2	ER_ST1	ER_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_ST0	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
1	ER_ST1	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
2	ER_ST2	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
3	ER_ST3	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
4	ER_ST4	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
5	ER_ST5	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
6	ER_ST6	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
7	ER_ST7	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
8	ER_ST8	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
9	ER_ST9	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
10	ER_ST10	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
11	ER_ST11	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
12	ER_ST12	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
13	ER_ST13	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
14	ER_ST14	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
15	ER_ST15	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
16	ER_ST16	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
17	ER_ST17	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
18	ER_ST18	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
19	ER_ST19	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
20	ER_ST20	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
21	ER_ST21	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
22	ER_ST22	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
23	ER_ST23	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
24	ER_ST24	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
25	ER_ST25	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
26	ER_ST26	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
27	ER_ST27	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
28	ER_ST28	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
29	ER_ST29	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
30	ER_ST30	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R
31	ER_ST31	Indicate captured error status for error event from DSMIF modules 0: No error 1: Error occurred	R

The DSMIFERR_STATn register indicates captured status for error events from DSMIF modules. Table 12.6 lists bit assignment of error events.

Table 12.6 Bit assignment of DSMIF error event (1 of 2)

Bit	DSMIFERR_STATn		
	n = 0 to 9	n = 10	n = 11
0	DSMIFn_OVC0EH0	DSMIF0_SUMERRL	DSMIF5_SUMERRL
1	DSMIFn_OVC1EL0	DSMIF0_SUMERRH	DSMIF5_SUMERRH
2	DSMIFn_OVC1EH0	DSMIF0_SCE0	DSMIF5_SCE0
3	DSMIFn_OVC2EL0	DSMIF0_SCE1	DSMIF5_SCE1
4	DSMIFn_OVC2EH0	DSMIF0_SCE2	DSMIF5_SCE2
5	DSMIFn_OVC0EL1	DSMIF0_OVC0EL0	DSMIF5_OVC0EL0
6	DSMIFn_OVC0EH1	DSMIF1_SUMERRL	DSMIF6_SUMERRL
7	DSMIFn_OVC1EL1	DSMIF1_SUMERRH	DSMIF6_SUMERRH
8	DSMIFn_OVC1EH1	DSMIF1_SCE0	DSMIF6_SCE0
9	DSMIFn_OVC2EL1	DSMIF1_SCE1	DSMIF6_SCE1
10	DSMIFn_OVC2EH1	DSMIF1_SCE2	DSMIF6_SCE2
11	DSMIFn_OVC0EL2	DSMIF1_OVC0EL0	DSMIF6_OVC0EL0
12	DSMIFn_OVC0EH2	DSMIF2_SUMERRL	DSMIF7_SUMERRL
13	DSMIFn_OVC1EL2	DSMIF2_SUMERRH	DSMIF7_SUMERRH
14	DSMIFn_OVC1EH2	DSMIF2_SCE0	DSMIF7_SCE0
15	DSMIFn_OVC2EL2	DSMIF2_SCE1	DSMIF7_SCE1

Table 12.6 Bit assignment of DSMIF error event (2 of 2)

Bit	DSMIFERR_STATn		
	n = 0 to 9	n = 10	n = 11
16	DSMIFn_OVC2EH2	DSMIF2_SCE2	DSMIF7_SCE2
17	DSMIFn_OVC0WN0	DSMIF2_OVC0EL0	DSMIF7_OVC0EL0
18	DSMIFn_OVC1WN0	DSMIF3_SUMERRL	DSMIF8_SUMERRL
19	DSMIFn_OVC2WN0	DSMIF3_SUMERRH	DSMIF8_SUMERRH
20	DSMIFn_OVC3WN0	DSMIF3_SCE0	DSMIF8_SCE0
21	DSMIFn_OVC0WN1	DSMIF3_SCE1	DSMIF8_SCE1
22	DSMIFn_OVC1WN1	DSMIF3_SCE2	DSMIF8_SCE2
23	DSMIFn_OVC2WN1	DSMIF3_OVC0EL0	DSMIF8_OVC0EL0
24	DSMIFn_OVC3WN1	DSMIF4_SUMERRL	DSMIF9_SUMERRL
25	DSMIFn_OVC0WN2	DSMIF4_SUMERRH	DSMIF9_SUMERRH
26	DSMIFn_OVC1WN2	DSMIF4_SCE0	DSMIF9_SCE0
27	DSMIFn_OVC2WN2	DSMIF4_SCE1	DSMIF9_SCE1
28	DSMIFn_OVC3WN2	DSMIF4_SCE2	DSMIF9_SCE2
29	Reserved (Read as 0)	DSMIF4_OVC0EL0	DSMIF9_OVC0EL0
30	Reserved (Read as 0)	Reserved (Read as 0)	Reserved (Read as 0)
31	Reserved (Read as 0)	Reserved (Read as 0)	Reserved (Read as 0)

12.3.2.30 ENCIFERR_E0MSKn : ENCIF E0 Error Event Mask Register n (n = 0 to 4)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x200 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E0_MK31	E0_MK30	E0_MK29	E0_MK28	E0_MK27	E0_MK26	E0_MK25	E0_MK24	E0_MK23	E0_MK22	E0_MK21	E0_MK20	E0_MK19	E0_MK18	E0_MK17	E0_MK16
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E0_MK15	E0_MK14	E0_MK13	E0_MK12	E0_MK11	E0_MK10	E0_MK9	E0_MK8	E0_MK7	E0_MK6	E0_MK5	E0_MK4	E0_MK3	E0_MK2	E0_MK1	E0_MK0
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E0_MK0	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
1	E0_MK1	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
2	E0_MK2	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
3	E0_MK3	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
4	E0_MK4	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
5	E0_MK5	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
6	E0_MK6	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
7	E0_MK7	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
8	E0_MK8	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
9	E0_MK9	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
10	E0_MK10	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
11	E0_MK11	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
12	E0_MK12	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
13	E0_MK13	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
14	E0_MK14	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
15	E0_MK15	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
16	E0_MK16	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
17	E0_MK17	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
18	E0_MK18	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
19	E0_MK19	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
20	E0_MK20	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
21	E0_MK21	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
22	E0_MK22	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
23	E0_MK23	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
24	E0_MK24	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
25	E0_MK25	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
26	E0_MK26	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
27	E0_MK27	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
28	E0_MK28	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
29	E0_MK29	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
30	E0_MK30	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
31	E0_MK31	Mask captured error status as ENCIF_ERR0 event for ENCIFERR_STATn 0: Release 1: Mask	R/W

Note 1. If bit of status register is reserved, value after reset of the corresponding bit of mask register is 0.

The ENCIFERR_E0MSKn register is used to mask error status for ENCIF_ERR0 event. When the bit is masked, corresponding error does not occur ENCIF_ERR0 event.

12.3.2.31 ENCIFERR_E1MSKn : ENCIF E1 Error Event Mask Register n (n = 0 to 4)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x214 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E1_MK31	E1_MK30	E1_MK29	E1_MK28	E1_MK27	E1_MK26	E1_MK25	E1_MK24	E1_MK23	E1_MK22	E1_MK21	E1_MK20	E1_MK19	E1_MK18	E1_MK17	E1_MK16
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E1_MK15	E1_MK14	E1_MK13	E1_MK12	E1_MK11	E1_MK10	E1_MK9	E1_MK8	E1_MK7	E1_MK6	E1_MK5	E1_MK4	E1_MK3	E1_MK2	E1_MK1	E1_MK0
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E1_MK0	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
1	E1_MK1	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
2	E1_MK2	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
3	E1_MK3	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
4	E1_MK4	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
5	E1_MK5	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
6	E1_MK6	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
7	E1_MK7	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
8	E1_MK8	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
9	E1_MK9	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
10	E1_MK10	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
11	E1_MK11	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
12	E1_MK12	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
13	E1_MK13	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
14	E1_MK14	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
15	E1_MK15	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
16	E1_MK16	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
17	E1_MK17	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
18	E1_MK18	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
19	E1_MK19	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
20	E1_MK20	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
21	E1_MK21	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
22	E1_MK22	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
23	E1_MK23	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
24	E1_MK24	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
25	E1_MK25	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
26	E1_MK26	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
27	E1_MK27	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
28	E1_MK28	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
29	E1_MK29	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
30	E1_MK30	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W
31	E1_MK31	Mask captured error status as ENCIF_ERR1 event for ENCIFERR_STATn 0: Release 1: Mask	R/W

Note 1. If bit of status register is reserved, value after reset of the corresponding bit of mask register is 0.

The ENCIFERR_E1MSKn register is used to mask error status for ENCIF_ERR1 event. When the bit is masked, corresponding error does not occur ENCIF_ERR1 event.

12.3.2.32 ENCIFERR_RSTMSKn : ENCIF Error Event Reset Mask Register n (n = 0 to 4)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x228 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RS_MK31	RS_MK30	RS_MK29	RS_MK28	RS_MK27	RS_MK26	RS_MK25	RS_MK24	RS_MK23	RS_MK22	RS_MK21	RS_MK20	RS_MK19	RS_MK18	RS_MK17	RS_MK16
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RS_MK15	RS_MK14	RS_MK13	RS_MK12	RS_MK11	RS_MK10	RS_MK9	RS_MK8	RS_MK7	RS_MK6	RS_MK5	RS_MK4	RS_MK3	RS_MK2	RS_MK1	RS_MK0
Value after reset*1:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	RS_MK0	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
1	RS_MK1	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
2	RS_MK2	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
3	RS_MK3	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
4	RS_MK4	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
5	RS_MK5	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
6	RS_MK6	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
7	RS_MK7	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
8	RS_MK8	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
9	RS_MK9	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
10	RS_MK10	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
11	RS_MK11	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
12	RS_MK12	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
13	RS_MK13	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
14	RS_MK14	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
15	RS_MK15	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
16	RS_MK16	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
17	RS_MK17	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
18	RS_MK18	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
19	RS_MK19	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
20	RS_MK20	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
21	RS_MK21	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
22	RS_MK22	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
23	RS_MK23	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
24	RS_MK24	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
25	RS_MK25	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
26	RS_MK26	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
27	RS_MK27	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
28	RS_MK28	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
29	RS_MK29	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W
30	RS_MK30	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
31	RS_MK31	Mask captured error status as a reset event for ENCIFERR_STATn 0: Release 1: Mask	R/W

Note 1. If bit of status register is reserved, value after reset of the corresponding bit of mask register is 0.

The ENCIFERR_RSTMSKn register is used to mask error status for reset event. When the bit is masked, corresponding error does not occur reset event.

12.3.2.33 ENCIFERR_CLRn : ENCIF Error Event Status Clear Register n (n = 0 to 4)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x250 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_CL31	ER_CL30	ER_CL29	ER_CL28	ER_CL27	ER_CL26	ER_CL25	ER_CL24	ER_CL23	ER_CL22	ER_CL21	ER_CL20	ER_CL19	ER_CL18	ER_CL17	ER_CL16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_CL15	ER_CL14	ER_CL13	ER_CL12	ER_CL11	ER_CL10	ER_CL9	ER_CL8	ER_CL7	ER_CL6	ER_CL5	ER_CL4	ER_CL3	ER_CL2	ER_CL1	ER_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_CL0	Clear captured error status for ENCIFERR_STATn register by writing 1	W
1	ER_CL1	Clear captured error status for ENCIFERR_STATn register by writing 1	W
2	ER_CL2	Clear captured error status for ENCIFERR_STATn register by writing 1	W
3	ER_CL3	Clear captured error status for ENCIFERR_STATn register by writing 1	W
4	ER_CL4	Clear captured error status for ENCIFERR_STATn register by writing 1	W
5	ER_CL5	Clear captured error status for ENCIFERR_STATn register by writing 1	W
6	ER_CL6	Clear captured error status for ENCIFERR_STATn register by writing 1	W
7	ER_CL7	Clear captured error status for ENCIFERR_STATn register by writing 1	W
8	ER_CL8	Clear captured error status for ENCIFERR_STATn register by writing 1	W
9	ER_CL9	Clear captured error status for ENCIFERR_STATn register by writing 1	W
10	ER_CL10	Clear captured error status for ENCIFERR_STATn register by writing 1	W
11	ER_CL11	Clear captured error status for ENCIFERR_STATn register by writing 1	W
12	ER_CL12	Clear captured error status for ENCIFERR_STATn register by writing 1	W
13	ER_CL13	Clear captured error status for ENCIFERR_STATn register by writing 1	W
14	ER_CL14	Clear captured error status for ENCIFERR_STATn register by writing 1	W
15	ER_CL15	Clear captured error status for ENCIFERR_STATn register by writing 1	W
16	ER_CL16	Clear captured error status for ENCIFERR_STATn register by writing 1	W
17	ER_CL17	Clear captured error status for ENCIFERR_STATn register by writing 1	W
18	ER_CL18	Clear captured error status for ENCIFERR_STATn register by writing 1	W
19	ER_CL19	Clear captured error status for ENCIFERR_STATn register by writing 1	W
20	ER_CL20	Clear captured error status for ENCIFERR_STATn register by writing 1	W
21	ER_CL21	Clear captured error status for ENCIFERR_STATn register by writing 1	W
22	ER_CL22	Clear captured error status for ENCIFERR_STATn register by writing 1	W
23	ER_CL23	Clear captured error status for ENCIFERR_STATn register by writing 1	W

Bit	Symbol	Function	R/W
24	ER_CL24	Clear captured error status for ENCIFERR_STATn register by writing 1	W
25	ER_CL25	Clear captured error status for ENCIFERR_STATn register by writing 1	W
26	ER_CL26	Clear captured error status for ENCIFERR_STATn register by writing 1	W
27	ER_CL27	Clear captured error status for ENCIFERR_STATn register by writing 1	W
28	ER_CL28	Clear captured error status for ENCIFERR_STATn register by writing 1	W
29	ER_CL29	Clear captured error status for ENCIFERR_STATn register by writing 1	W
30	ER_CL30	Clear captured error status for ENCIFERR_STATn register by writing 1	W
31	ER_CL31	Clear captured error status for ENCIFERR_STATn register by writing 1	W

The ENCIFERR_CLRn register is used to clear error status captured in ENCIFERR_STATn register.

12.3.2.34 ENCIFERR_STATn : ENCIF Error Event Status Register n (n = 0 to 4)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x264 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_ST31	ER_ST30	ER_ST29	ER_ST28	ER_ST27	ER_ST26	ER_ST25	ER_ST24	ER_ST23	ER_ST22	ER_ST21	ER_ST20	ER_ST19	ER_ST18	ER_ST17	ER_ST16
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_ST15	ER_ST14	ER_ST13	ER_ST12	ER_ST11	ER_ST10	ER_ST9	ER_ST8	ER_ST7	ER_ST6	ER_ST5	ER_ST4	ER_ST3	ER_ST2	ER_ST1	ER_ST0
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_ST0	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
1	ER_ST1	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
2	ER_ST2	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
3	ER_ST3	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
4	ER_ST4	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
5	ER_ST5	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
6	ER_ST6	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
7	ER_ST7	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
8	ER_ST8	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
9	ER_ST9	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
10	ER_ST10	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
11	ER_ST11	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
12	ER_ST12	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
13	ER_ST13	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
14	ER_ST14	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
15	ER_ST15	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
16	ER_ST16	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
17	ER_ST17	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
18	ER_ST18	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
19	ER_ST19	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
20	ER_ST20	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
21	ER_ST21	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
22	ER_ST22	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
23	ER_ST23	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
24	ER_ST24	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
25	ER_ST25	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
26	ER_ST26	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
27	ER_ST27	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
28	ER_ST28	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
29	ER_ST29	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
30	ER_ST30	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R
31	ER_ST31	Indicate captured error status for error event from Encoder Interface modules 0: No error 1: Error occurred	R

Note 1. Value after reset is 1 dependent on bit assignment. See Table 12.7.

The ENCIFERR_STATn register indicates captured status for error events from Encoder Interface modules. Table 12.7 lists bit assignment of error events.

Table 12.7 Bit assignment of Encoder Interface error event (1 of 2)

Bit	ENCIFERR_STATn	
	n = 0	n = 1 to 4
0	BISS00_NER*5	HDSLa_estimator_on*1 *5
1	BISS01_NER*5	HDSLa_safe_channel_err*1 *5
2	BISS02_NER*5	HDSLa_safe_pos_err*1 *5
3	BISS03_NER*5	HDSLa_acceleration_err*1 *5
4	BISS04_NER*5	HDSLa_acc_thr_err*1
5	BISS05_NER*5	HDSLa_encoding_err*1
6	BISS06_NER*5	HDSLa_dev_thr_err*1 *5
7	BISS07_NER*5	Reserved (Read as 0)
8	BISS08_NER*5	HDSLb_estimator_on*2 *5
9	BISS09_NER*5	HDSLb_safe_channel_err*2 *5
10	BISS10_NER*5	HDSLb_safe_pos_err*2 *5
11	BISS11_NER	HDSLb_acceleration_err*2 *5
12	BISS12_NER	HDSLb_acc_thr_err*2
13	BISS13_NER	HDSLb_encoding_err*2
14	BISS14_NER	HDSLb_dev_thr_err*2 *5
15	BISS15_NER	Reserved (Read as 0)
16	AFMT00_TMOUT	HDSLc_estimator_on*3 *5
17	AFMT01_TMOUT	HDSLc_safe_channel_err*3 *5
18	AFMT02_TMOUT	HDSLc_safe_pos_err*3 *5
19	AFMT03_TMOUT	HDSLc_acceleration_err*3 *5
20	AFMT04_TMOUT	HDSLc_acc_thr_err*3

Table 12.7 Bit assignment of Encoder Interface error event (2 of 2)

Bit	ENCIFERR_STATn	
	n = 0	n = 1 to 4
21	AFMT05_TMOUT	HDSLc_encoding_err*3
22	AFMT06_TMOUT	HDSLc_dev_thr_err*3 *5
23	AFMT07_TMOUT	Reserved (Read as 0)
24	AFMT08_TMOUT	HDSLd_estimator_on*4 *5
25	AFMT09_TMOUT	HDSLd_safe_channel_err*4 *5
26	AFMT10_TMOUT	HDSLd_safe_pos_err*4 *5
27	AFMT11_TMOUT	HDSLd_acceleration_err*4 *5
28	AFMT12_TMOUT	HDSLd_acc_thr_err*4
29	AFMT13_TMOUT	HDSLd_encoding_err*4
30	AFMT14_TMOUT	HDSLd_dev_thr_err*4 *5
31	AFMT15_TMOUT	Reserved (Read as 0)

Note 1. a = (n - 1) × 4
Note 2. b = (n - 1) × 4 + 1
Note 3. c = (n - 1) × 4 + 2
Note 4. d = (n - 1) × 4 + 3
Note 5. Value after reset is 1. Clear the bit before operation.

12.3.2.35 ENCIF_ENCSEL0 : ENCIF Event Select Register 0

Base address: ICU_NS = 0x802A_0000

Offset address: 0x290

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ENCSEL7			—	ENCSEL6			—	ENCSEL5			—	ENCSEL4		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ENCSEL3			—	ENCSEL2			—	ENCSEL1			—	ENCSEL0		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ENCSEL0	Select Encoder Interface event source to be output as ENCIF00_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	ENCSEL1	Select Encoder Interface event source to be output as ENCIF01_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10:8	ENCSEL2	Select Encoder Interface event source to be output as ENCIF02_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	ENCSEL3	Select Encoder Interface event source to be output as ENCIF03_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	ENCSEL4	Select Encoder Interface event source to be output as ENCIF04_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	ENCSEL5	Select Encoder Interface event source to be output as ENCIF05_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	ENCSEL6	Select Encoder Interface event source to be output as ENCIF06_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	ENCSEL7	Select Encoder Interface event source to be output as ENCIF07_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The ENCIF_ENCSEL0 register specifies a Encoder Interface event source to be output as ENCIFn_INT[3:0] events. Table 12.8 shows which Encoder Interface event source is selected by ENCSELn bits.

Table 12.8 Encoder Interface event source selected by ENCSELn bits (1 of 2)

ENCSELn bits	ENCIFn_INT0	ENCIFn_INT1	ENCIFn_INT2	ENCIFn_INT3
000b (SCIE)	SCIEn_ERI	SCIEn_RXI	SCIEn_TXI	SCIEn_TEI
001b (ENDAT)	ENDATn_INT1	—	—	—
010b (BISS)	BISSn_EOT	—	—	—

Table 12.8 Encoder Interface event source selected by ENCSELn bits (2 of 2)

ENCSELn bits	ENCIFn_INT0	ENCIFn_INT1	ENCIFn_INT2	ENCIFn_INT3
011b (HDSL)	HDSLn_INT	HDSLn_INTS	HDSLn_FP	HDSLn_SP
100b (AFMT)	AFMTn_EOF	—	—	—
Others	—	—	—	—

Note: n = 0 to 7

12.3.2.36 ENCIF_ENCSEL1 : ENCIF Event Select Register 1

Base address: ICU_NS = 0x802A_0000

Offset address: 0x294

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ENCSEL15			—	ENCSEL14			—	ENCSEL13			—	ENCSEL12		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ENCSEL11			—	ENCSEL10			—	ENCSEL9			—	ENCSEL8		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ENCSEL8	Select Encoder Interface event source to be output as ENCIF08_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	ENCSEL9	Select Encoder Interface event source to be output as ENCIF09_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	ENCSEL10	Select Encoder Interface event source to be output as ENCIF10_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	ENCSEL11	Select Encoder Interface event source to be output as ENCIF11_INT[3:0] event 0 0 0: SCIE 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
18:16	ENCSEL12	Select Encoder Interface event source to be output as ENCIF12_INT[3:0] event 0 0 0: Fixed to 0 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	ENCSEL13	Select Encoder Interface event source to be output as ENCIF13_INT[3:0] event 0 0 0: Fixed to 0 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	ENCSEL14	Select Encoder Interface event source to be output as ENCIF14_INT[3:0] event 0 0 0: Fixed to 0 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	ENCSEL15	Select Encoder Interface event source to be output as ENCIF15_INT[3:0] event 0 0 0: Fixed to 0 0 0 1: ENDAT 0 1 0: BISS 0 1 1: HDSL 1 0 0: AFMT Others: Setting prohibited	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The ENCIF_ENCSEL1 register specifies a Encoder Interface event source to be output as ENCIFn_INT[3:0] events. Table 12.9 shows which Encoder Interface event source is selected by ENCSELn bits.

Table 12.9 Encoder Interface event source selected by ENCSELn bits

ENCSELn bits	ENCIFn_INT0	ENCIFn_INT1	ENCIFn_INT2	ENCIFn_INT3
000b (SCIE)	SCIEn_ERI	SCIEn_RXI	SCIEn_TXI	SCIEn_TEI
001b (ENDAT)	ENDATn_INT1	—	—	—
010b (BISS)	BISSn_EOT	—	—	—
011b (HDSL)	HDSLn_INT	HDSLn_INTS	HDSLn_FP	HDSLn_SP
100b (AFMT)	AFMTn_EOF	—	—	—
Others	—	—	—	—

Note: n = 8 to 15 For n= 12 to 15, SCIE is not available.

12.3.2.37 NS_GPT_INTSELn : GPT/MTU3 Interrupt Select Register n (n = 0 to 25)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x2C8 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	INTSELY_3[3:0]				INTSELY_2[3:0]				INTSELY_1[3:0]				INTSELY_0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	INTSELX_3[3:0]				INTSELX_2[3:0]				INTSELX_1[3:0]				INTSELX_0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	INTSELX_0[3:0]	Select GPT/MTU3 event source to be output as GPTx_INT0 event	R/W
7:4	INTSELX_1[3:0]	Select GPT/MTU3 event source to be output as GPTx_INT1 event	R/W
11:8	INTSELX_2[3:0]	Select GPT/MTU3 event source to be output as GPTx_INT2 event	R/W
15:12	INTSELX_3[3:0]	Select GPT/MTU3 event source to be output as GPTx_INT3 event	R/W
19:16	INTSELY_0[3:0]	Select GPT/MTU3 event source to be output as GPTy_INT0 event	R/W
23:20	INTSELY_1[3:0]	Select GPT/MTU3 event source to be output as GPTy_INT1 event	R/W
27:24	INTSELY_2[3:0]	Select GPT/MTU3 event source to be output as GPTy_INT2 event	R/W
31:28	INTSELY_3[3:0]	Select GPT/MTU3 event source to be output as GPTy_INT3 event	R/W

Note: Relation among n, x and y is shown in the below table.

n	x	y	n	x	y	n	x	y
0	00_0	00_1	10	04_0	04_1	20	08_0	08_1
1	00_2	00_3	11	04_2	04_3	21	08_2	08_3
2	00_4	01_0	12	04_4	05_0	22	08_4	09_0
3	01_1	01_2	13	05_1	05_2	23	09_1	09_2
4	01_3	01_4	14	05_3	05_4	24	09_3	09_4
5	02_0	02_1	15	06_0	06_1	25	09_5	09_6
6	02_2	02_3	16	06_2	06_3	—		
7	02_4	03_0	17	06_4	07_0			
8	03_1	03_2	18	07_1	07_2			
9	03_3	03_4	19	07_3	07_4			

The NS_GPT_INTSELn register specifies GPT (GPT00 to GPT09) and MTU3 event source to be output as GPTx/y_INT[3:0] event source for Cortex-A55 SPI, Cortex-R52 SPI and DMAC.

Table 12.10 shows which GPT/MTU3 event source is selected by INTSEL bits. For n = 0 to 4, MTU3 event is selectable as well as GPT event.

Table 12.10 GPT/MTU3 event source selected by INTSEL bits (1 of 2)

INTSEL bits	n = 0		n = 1		n = 2		n = 3		n = 4		n = 5 to 25
	x	y	x	y	x	y	x	y	x	y	x/y
	00_0	00_1	00_2	00_3	00_4	01_0	01_1	01_2	01_3	01_4	02_0 to 09_6
0x0	GPTx/y_CCMPA										
0x1	GPTx/y_CCMPB										

Table 12.10 GPT/MTU3 event source selected by INTSEL bits (2 of 2)

INTSEL bits	n = 0		n = 1		n = 2		n = 3		n = 4		n = 5 to 25
	x	y	x	y	x	y	x	y	x	y	x/y
	00_0	00_1	00_2	00_3	00_4	01_0	01_1	01_2	01_3	01_4	02_0 to 09_6
0x2	GPTx/y_CMPC										
0x3	GPTx/y_CMPD										
0x4	GPTx/y_CMPE										
0x5	GPTx/y_CMPF										
0x6	GPTx/y_OVF										
0x7	GPTx/y_UDF										
0x8	GPTx/y_DTE										
0x9	TGIA0	TGIE0	TCIU1	TGIA3	TGIA4	TGIU5	TGIA6	TGIA7	TGIA8	0	0
0xA	TGIB0	TGIF0	TGIA2	TGIB3	TGIB4	TGIV5	TGIB6	TGIB7	TGIB8	0	0
0xB	TGIC0	TGIA1	TGIB2	TGIC3	TGIC4	TGIW5	TGIC6	TGIC7	TGIC8	0	0
0xC	TGID0	TGIB1	TCIV2	TGID3	TGID4	0	TGID6	TGID7	TGID8	0	0
0xD	TCIV0	TCIV1	TCIU2	TCIV3	TCIV4	0	TCIV6	TCIV7	TCIV8	0	0
0xE	0	0	0	0	0	0	0	0	0	0	0
0xF	0	0	0	0	0	0	0	0	0	0	0

12.3.2.38 NS_GPT_INTMSKn : GPT/MTU3 Combined Interrupt Mask Register n (n = 0 to 25)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x900 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	IY_MK 13	IY_MK 12	IY_MK 11	IY_MK 10	IY_MK 9	IY_MK 8	IY_MK 7	IY_MK 6	IY_MK 5	IY_MK 4	IY_MK 3	IY_MK 2	IY_MK 1	IY_MK 0
Value after reset:	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IX_MK 13	IX_MK 12	IX_MK 11	IX_MK 10	IX_MK 9	IX_MK 8	IX_MK 7	IX_MK 6	IX_MK 5	IX_MK 4	IX_MK 3	IX_MK 2	IX_MK 1	IX_MK 0
Value after reset:	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	IX_MK0	Mask GPTx_CCMPA event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
1	IX_MK1	Mask GPTx_CCMPB event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
2	IX_MK2	Mask GPTx_CMPC event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
3	IX_MK3	Mask GPTx_CMPD event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
4	IX_MK4	Mask GPTx_CMPE event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W

Bit	Symbol	Function	R/W
5	IX_MK5	Mask GPTx_CMPF event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
6	IX_MK6	Mask GPTx_OVF event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
7	IX_MK7	Mask GPTx_UDF event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
8	IX_MK8	Mask GPTx_DTE event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
9	IX_MK9	Mask MTU3 event source 0 from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
10	IX_MK10	Mask MTU3 event source 1 from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
11	IX_MK11	Mask MTU3 event source 2 from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
12	IX_MK12	Mask MTU3 event source 3 from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
13	IX_MK13	Mask MTU3 event source 4 from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	IY_MK0	Mask GPTy_CCMPA event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
17	IY_MK1	Mask GPTy_CCMPB event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
18	IY_MK2	Mask GPTy_CMPC event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
19	IY_MK3	Mask GPTy_CMPD event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
20	IY_MK4	Mask GPTy_CMPE event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
21	IY_MK5	Mask GPTy_CMPF event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
22	IY_MK6	Mask GPTy_OVF event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
23	IY_MK7	Mask GPTy_UDF event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
24	IY_MK8	Mask GPTy_DTE event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W

Bit	Symbol	Function	R/W
25	IY_MK9	Mask MTU3 event source 0 from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
26	IY_MK10	Mask MTU3 event source 1 from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
27	IY_MK11	Mask MTU3 event source 2 from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
28	IY_MK12	Mask MTU3 event source 3 from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
29	IY_MK13	Mask MTU3 event source 4 from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: Relation among n, x and y is shown in the below table.

n	x	y	n	x	y	n	x	y
0	00_0	00_1	10	04_0	04_1	20	08_0	08_1
1	00_2	00_3	11	04_2	04_3	21	08_2	08_3
2	00_4	01_0	12	04_4	05_0	22	08_4	09_0
3	01_1	01_2	13	05_1	05_2	23	09_1	09_2
4	01_3	01_4	14	05_3	05_4	24	09_3	09_4
5	02_0	02_1	15	06_0	06_1	25	09_5	09_6
6	02_2	02_3	16	06_2	06_3	—		
7	02_4	03_0	17	06_4	07_0			
8	03_1	03_2	18	07_1	07_2			
9	03_3	03_4	19	07_3	07_4			

The NS_GPT_INTMSKn register is used to mask GPT (GPT00 to GPT09) and MTU3 event sources to be combined as GPTx/y_INT4 event source for Cortex-A55 SPI, Cortex-R52 SPI and DMAC. When the bit is masked, corresponding event does not occur GPTx/y_INT4 event.

The bits for MTU3 event sources are valid only for n = 0 to 4. Table 12.11 shows MTU3 event source mapping.

Table 12.11 MTU3 event source mapping

	n = 0		n = 1		n = 2		n = 3		n = 4		n = 5 to 25
	x	y	x	y	x	y	x	y	x	y	x/y
Bit	00_0	00_1	00_2	00_3	00_4	01_0	01_1	01_2	01_3	01_4	02_0 to 09_6
9 / 25	TGIA0	TGIE0	TCIU1	TGIA3	TGIA4	TGIU5	TGIA6	TGIA7	TGIA8	—	—
10 / 26	TGIB0	TGIF0	TGIA2	TGIB3	TGIB4	TGIV5	TGIB6	TGIB7	TGIB8	—	—
11 / 27	TGIC0	TGIA1	TGIB2	TGIC3	TGIC4	TGIW5	TGIC6	TGIC7	TGIC8	—	—
12 / 28	TGID0	TGIB1	TCIV2	TGID3	TGID4	—	TGID6	TGID7	TGID8	—	—
13 / 29	TCIV0	TCIV1	TCIU2	TCIV3	TCIV4	—	TCIV6	TCIV7	TCIV8	—	—

12.3.2.39 NS_GPT_INTCLRn : GPT/MTU3 Combined Interrupt Status Clear Register n (n = 0 to 25)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x9E0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	IY_CL 13	IY_CL 12	IY_CL 11	IY_CL 10	IY_CL 9	IY_CL 8	IY_CL 7	IY_CL 6	IY_CL 5	IY_CL 4	IY_CL 3	IY_CL 2	IY_CL 1	IY_CL 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IX_CL 13	IX_CL 12	IX_CL 11	IX_CL 10	IX_CL 9	IX_CL 8	IX_CL 7	IX_CL 6	IX_CL 5	IX_CL 4	IX_CL 3	IX_CL 2	IX_CL 1	IX_CL 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IX_CL0	Clear GPTx_CCMPA event status captured in NS_GPT_INTSTATn register by writing 1	W
1	IX_CL1	Clear GPTx_CCMPB event status captured in NS_GPT_INTSTATn register by writing 1	W
2	IX_CL2	Clear GPTx_CMPC event status captured in NS_GPT_INTSTATn register by writing 1	W
3	IX_CL3	Clear GPTx_CMPD event status captured in NS_GPT_INTSTATn register by writing 1	W
4	IX_CL4	Clear GPTx_CMPE event status captured in NS_GPT_INTSTATn register by writing 1	W
5	IX_CL5	Clear GPTx_CMPF event status captured in NS_GPT_INTSTATn register by writing 1	W
6	IX_CL6	Clear GPTx_OVF event status captured in NS_GPT_INTSTATn register by writing 1	W
7	IX_CL7	Clear GPTx_UDF event status captured in NS_GPT_INTSTATn register by writing 1	W
8	IX_CL8	Clear GPTx_DTE event status captured in NS_GPT_INTSTATn register by writing 1	W
9	IX_CL9	Clear MTU3 event status 0 captured in NS_GPT_INTSTATn register by writing 1	W
10	IX_CL10	Clear MTU3 event status 1 captured in NS_GPT_INTSTATn register by writing 1	W
11	IX_CL11	Clear MTU3 event status 2 captured in NS_GPT_INTSTATn register by writing 1	W
12	IX_CL12	Clear MTU3 event status 3 captured in NS_GPT_INTSTATn register by writing 1	W
13	IX_CL13	Clear MTU3 event status 4 captured in NS_GPT_INTSTATn register by writing 1	W
15:14	—	The write value should be 0.	W
16	IY_CL0	Clear GPTy_CCMPA event status captured in NS_GPT_INTSTATn register by writing 1	W
17	IY_CL1	Clear GPTy_CCMPB event status captured in NS_GPT_INTSTATn register by writing 1	W
18	IY_CL2	Clear GPTy_CMPC event status captured in NS_GPT_INTSTATn register by writing 1	W
19	IY_CL3	Clear GPTy_CMPD event status captured in NS_GPT_INTSTATn register by writing 1	W
20	IY_CL4	Clear GPTy_CMPE event status captured in NS_GPT_INTSTATn register by writing 1	W
21	IY_CL5	Clear GPTy_CMPF event status captured in NS_GPT_INTSTATn register by writing 1	W
22	IY_CL6	Clear GPTy_OVF event status captured in NS_GPT_INTSTATn register by writing 1	W
23	IY_CL7	Clear GPTy_UDF event status captured in NS_GPT_INTSTATn register by writing 1	W
24	IY_CL8	Clear GPTy_DTE event status captured in NS_GPT_INTSTATn register by writing 1	W
25	IY_CL9	Clear MTU3 event status 0 captured in NS_GPT_INTSTATn register by writing 1	W
26	IY_CL10	Clear MTU3 event status 1 captured in NS_GPT_INTSTATn register by writing 1	W
27	IY_CL11	Clear MTU3 event status 2 captured in NS_GPT_INTSTATn register by writing 1	W
28	IY_CL12	Clear MTU3 event status 3 captured in NS_GPT_INTSTATn register by writing 1	W
29	IY_CL13	Clear MTU3 event status 4 captured in NS_GPT_INTSTATn register by writing 1	W
31:30	—	The write value should be 0.	W

Note: Relation among n, x and y is shown in the below table.

n	x	y	n	x	y	n	x	y
0	00_0	00_1	10	04_0	04_1	20	08_0	08_1
1	00_2	00_3	11	04_2	04_3	21	08_2	08_3
2	00_4	01_0	12	04_4	05_0	22	08_4	09_0
3	01_1	01_2	13	05_1	05_2	23	09_1	09_2
4	01_3	01_4	14	05_3	05_4	24	09_3	09_4
5	02_0	02_1	15	06_0	06_1	25	09_5	09_6
6	02_2	02_3	16	06_2	06_3	—		
7	02_4	03_0	17	06_4	07_0			
8	03_1	03_2	18	07_1	07_2			
9	03_3	03_4	19	07_3	07_4			

The NS_GPT_INTCLRn register is used to clear GPT (GPT00 to GPT09) and MTU3 event status captured in NS_GPT_INTSTATn register.

12.3.2.40 NS_GPT_INTSTATn : GPT/MTU3 Combined Interrupt Status Register n (n = 0 to 25)

Base address: ICU_NS = 0x802A_0000

Offset address: 0xAC0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	IY_ST13	IY_ST12	IY_ST11	IY_ST10	IY_ST9	IY_ST8	IY_ST7	IY_ST6	IY_ST5	IY_ST4	IY_ST3	IY_ST2	IY_ST1	IY_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IX_ST13	IX_ST12	IX_ST11	IX_ST10	IX_ST9	IX_ST8	IX_ST7	IX_ST6	IX_ST5	IX_ST4	IX_ST3	IX_ST2	IX_ST1	IX_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IX_ST0	Indicate captured GPTx_CCMPA event status 0: No event occurs 1: Event occurs	R
1	IX_ST1	Indicate captured GPTx_CCMPB event status 0: No event occurs 1: Event occurs	R
2	IX_ST2	Indicate captured GPTx_CMPC event status 0: No event occurs 1: Event occurs	R
3	IX_ST3	Indicate captured GPTx_CMPD event status 0: No event occurs 1: Event occurs	R
4	IX_ST4	Indicate captured GPTx_CMPE event status 0: No event occurs 1: Event occurs	R
5	IX_ST5	Indicate captured GPTx_CMPF event status 0: No event occurs 1: Event occurs	R
6	IX_ST6	Indicate captured GPTx_OVF event status 0: No event occurs 1: Event occurs	R

Bit	Symbol	Function	R/W
7	IX_ST7	Indicate captured GPTx_UDF event status 0: No event occurs 1: Event occurs	R
8	IX_ST8	Indicate captured GPTx_DTE event status 0: No event occurs 1: Event occurs	R
9	IX_ST9	Indicate captured MTU3 event status 0 0: No event occurs 1: Event occurs	R
10	IX_ST10	Indicate captured MTU3 event status 1 0: No event occurs 1: Event occurs	R
11	IX_ST11	Indicate captured MTU3 event status 2 0: No event occurs 1: Event occurs	R
12	IX_ST12	Indicate captured MTU3 event status 3 0: No event occurs 1: Event occurs	R
13	IX_ST13	Indicate captured MTU3 event status 4 0: No event occurs 1: Event occurs	R
15:14	—	These bits are read as 0.	R
16	IY_ST0	Indicate captured GPTy_CCMPA event status 0: No event occurs 1: Event occurs	R
17	IY_ST1	Indicate captured GPTy_CCMPB event status 0: No event occurs 1: Event occurs	R
18	IY_ST2	Indicate captured GPTy_CMPC event status 0: No event occurs 1: Event occurs	R
19	IY_ST3	Indicate captured GPTy_CMPD event status 0: No event occurs 1: Event occurs	R
20	IY_ST4	Indicate captured GPTy_CMPE event status 0: No event occurs 1: Event occurs	R
21	IY_ST5	Indicate captured GPTy_CMPF event status 0: No event occurs 1: Event occurs	R
22	IY_ST6	Indicate captured GPTy_OVF event status 0: No event occurs 1: Event occurs	R
23	IY_ST7	Indicate captured GPTy_UDF event status 0: No event occurs 1: Event occurs	R
24	IY_ST8	Indicate captured GPTy_DTE event status 0: No event occurs 1: Event occurs	R
25	IY_ST9	Indicate captured MTU3 event status 0 0: No event occurs 1: Event occurs	R
26	IY_ST10	Indicate captured MTU3 event status 1 0: No event occurs 1: Event occurs	R

Bit	Symbol	Function	R/W
27	IY_ST11	Indicate captured MTU3 event status 2 0: No event occurs 1: Event occurs	R
28	IY_ST12	Indicate captured MTU3 event status 3 0: No event occurs 1: Event occurs	R
29	IY_ST13	Indicate captured MTU3 event status 4 0: No event occurs 1: Event occurs	R
31:30	—	These bits are read as 0.	R

Note: Relation among n, x and y is shown in the below table.

n	x	y	n	x	y	n	x	y
0	00_0	00_1	10	04_0	04_1	20	08_0	08_1
1	00_2	00_3	11	04_2	04_3	21	08_2	08_3
2	00_4	01_0	12	04_4	05_0	22	08_4	09_0
3	01_1	01_2	13	05_1	05_2	23	09_1	09_2
4	01_3	01_4	14	05_3	05_4	24	09_3	09_4
5	02_0	02_1	15	06_0	06_1	25	09_5	09_6
6	02_2	02_3	16	06_2	06_3	—		
7	02_4	03_0	17	06_4	07_0			
8	03_1	03_2	18	07_1	07_2			
9	03_3	03_4	19	07_3	07_4			

The NS_GPT_INTSTATn register indicates captured status for GPT (GPT00 to GPT09) and MTU3 events for Cortex-A55 SPI, Cortex-R52 SPI and DMAC.

The bits for MTU3 event status are valid only for n = 0 to 4. Table 12.12 shows MTU3 event status mapping.

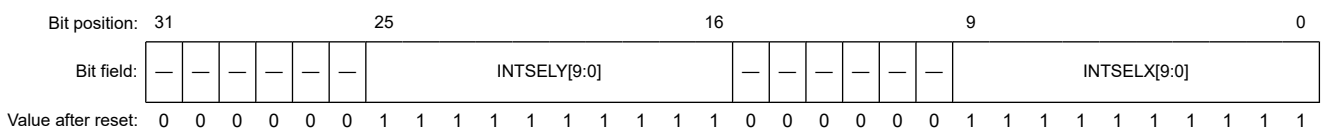
Table 12.12 MTU3 event status mapping

	n = 0		n = 1		n = 2		n = 3		n = 4		n = 5 to 25
	x	y	x	y	x	y	x	y	x	y	x/y
Bit	00_0	00_1	00_2	00_3	00_4	01_0	01_1	01_2	01_3	01_4	02_0 to 09_6
9 / 25	TGIA0	TGIE0	TCIU1	TGIA3	TGIA4	TGIU5	TGIA6	TGIA7	TGIA8	—	—
10 / 26	TGIB0	TGIF0	TGIA2	TGIB3	TGIB4	TGIV5	TGIB6	TGIB7	TGIB8	—	—
11 / 27	TGIC0	TGIA1	TGIB2	TGIC3	TGIC4	TGIW5	TGIC6	TGIC7	TGIC8	—	—
12 / 28	TGID0	TGIB1	TCIV2	TGID3	TGID4	—	TGID6	TGID7	TGID8	—	—
13 / 29	TCIV0	TCIV1	TCIU2	TCIV3	TCIV4	—	TCIV6	TCIV7	TCIV8	—	—

12.3.2.41 CR520_INTSELn : CR52 CPU0 Interrupt Select Register n (n = 0 to 79)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x418 + 0x04 × n



Bit	Symbol	Function	R/W
9:0	INTSELX[9:0]	Set the number of the event source to be select as Cortex-R52 CPU0 selected peripheral interrupt x (SELECT0_IRQx). Event source is selectable from Table 12.17. If SELECT0_IRQx is not used, set 0x3FF (Fixed to 0 output).*1 Setting of the number not listed in Table 12.17 is prohibited.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	INTSELY[9:0]	Set the number of the event source to be select as Cortex-R52 CPU0 selected peripheral interrupt y (SELECT0_IRQy). Event source is selectable from Table 12.17. If SELECT0_IRQy is not used, set 0x3FF (Fixed to 0 output).*2 Setting of the number not listed in Table 12.17 is prohibited.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. x = 2 × n
Note 2. y = 2 × n + 1

12.3.2.42 CR521_INTSELn : CR52 CPU1 Interrupt Select Register n (n = 0 to 79)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x558 + 0x04 × n

Bit position:	31						25									16						9								0
Bit field:	—	—	—	—	—	—	INTSELY[9:0]									—	—	—	—	—	—	INTSELX[9:0]									
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
9:0	INTSELX[9:0]	Set the number of the event source to be select as Cortex-R52 CPU1 selected peripheral interrupt x (SELECT1_IRQx). Event source is selectable from Table 12.17. If SELECT1_IRQx is not used, set 0x3FF (Fixed to 0 output).*1 Setting of the number not listed in Table 12.17 is prohibited.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	INTSELY[9:0]	Set the number of the event source to be select as Cortex-R52 CPU1 selected peripheral interrupt y (SELECT1_IRQy). Event source is selectable from Table 12.17. If SELECT1_IRQy is not used, set 0x3FF (Fixed to 0 output).*2 Setting of the number not listed in Table 12.17 is prohibited.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. x = 2 × n
Note 2. y = 2 × n + 1

12.3.3 ELC Related Registers

12.3.3.1 ELC_GPT_INTSELn : GPT/MTU3 ELC event source Select Register n (n = 0 to 25)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x330 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	INTSELY_3[3:0]				INTSELY_2[3:0]				INTSELY_1[3:0]				INTSELY_0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	INTSELX_3[3:0]				INTSELX_2[3:0]				INTSELX_1[3:0]				INTSELX_0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	INTSELX_0[3:0]	Select GPT/MTU3 event source to be output as GPTx_INT0 event	R/W
7:4	INTSELX_1[3:0]	Select GPT/MTU3 event source to be output as GPTx_INT1 event	R/W
11:8	INTSELX_2[3:0]	Select GPT/MTU3 event source to be output as GPTx_INT2 event	R/W
15:12	INTSELX_3[3:0]	Select GPT/MTU3 event source to be output as GPTx_INT3 event	R/W
19:16	INTSELY_0[3:0]	Select GPT/MTU3 event source to be output as GPTy_INT0 event	R/W
23:20	INTSELY_1[3:0]	Select GPT/MTU3 event source to be output as GPTy_INT1 event	R/W
27:24	INTSELY_2[3:0]	Select GPT/MTU3 event source to be output as GPTy_INT2 event	R/W
31:28	INTSELY_3[3:0]	Select GPT/MTU3 event source to be output as GPTy_INT3 event	R/W

Note: Relation among n, x and y is shown in the below table.

n	x	y	n	x	y	n	x	y
0	00_0	00_1	10	04_0	04_1	20	08_0	08_1
1	00_2	00_3	11	04_2	04_3	21	08_2	08_3
2	00_4	01_0	12	04_4	05_0	22	08_4	09_0
3	01_1	01_2	13	05_1	05_2	23	09_1	09_2
4	01_3	01_4	14	05_3	05_4	24	09_3	09_4
5	02_0	02_1	15	06_0	06_1	25	09_5	09_6
6	02_2	02_3	16	06_2	06_3	—		
7	02_4	03_0	17	06_4	07_0			
8	03_1	03_2	18	07_1	07_2			
9	03_3	03_4	19	07_3	07_4			

The ELC_GPT_INTSELn register specifies GPT (GPT00 to GPT09) and MTU3 event source to be output as GPTx/y_INT[3:0] event source for ELC.

Table 12.13 shows which GPT/MTU3 event source is selected by INTSEL bits. For n = 0 to 4, MTU3 event is selectable as well as GPT event.

Table 12.13 GPT/MTU3 event source selected by INTSEL bits (1 of 2)

INTSEL bits	n = 0		n = 1		n = 2		n = 3		n = 4		n = 5 to 25
	x	y	x	y	x	y	x	y	x	y	x/y
	00_0	00_1	00_2	00_3	00_4	01_0	01_1	01_2	01_3	01_4	02_0 to 09_6
0x0	GPTx/y_CCMPA										
0x1	GPTx/y_CCMPB										
0x2	GPTx/y_CMPC										
0x3	GPTx/y_CMPD										
0x4	GPTx/y_CMPE										
0x5	GPTx/y_CMPF										
0x6	GPTx/y_OVF										
0x7	GPTx/y_UDF										
0x8	GPTx/y_DTE										
0x9	TGIA0	TGIE0	TCIU1	TGIA3	TGIA4	TGIU5	TGIA6	TGIA7	TGIA8	0	0
0xA	TGIB0	TGIF0	TGIA2	TGIB3	TGIB4	TGIV5	TGIB6	TGIB7	TGIB8	0	0
0xB	TGIC0	TGIA1	TGIB2	TGIC3	TGIC4	TGIW5	TGIC6	TGIC7	TGIC8	0	0
0xC	TGID0	TGIB1	TCIV2	TGID3	TGID4	0	TGID6	TGID7	TGID8	0	0

Table 12.13 GPT/MTU3 event source selected by INTSEL bits (2 of 2)

INTSEL bits	n = 0		n = 1		n = 2		n = 3		n = 4		n = 5 to 25
	x	y	x	y	x	y	x	y	x	y	x/y
	00_0	00_1	00_2	00_3	00_4	01_0	01_1	01_2	01_3	01_4	02_0 to 09_6
0xD	TCIV0	TCIV1	TCIU2	TCIV3	TCIV4	0	TCIV6	TCIV7	TCIV8	0	0
0xE	0	0	0	0	0	0	0	0	0	0	0
0xF	0	0	0	0	0	0	0	0	0	0	0

12.3.3.2 ELC_GPT_INTMSKn : GPT/MTU3 Combined ELC Event Mask Register n (n = 0 to 25)

Base address: ICU_NS = 0x802A_0000

Offset address: 0x970 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	IY_MK 13	IY_MK 12	IY_MK 11	IY_MK 10	IY_MK 9	IY_MK 8	IY_MK 7	IY_MK 6	IY_MK 5	IY_MK 4	IY_MK 3	IY_MK 2	IY_MK 1	IY_MK 0
Value after reset:	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IX_MK 13	IX_MK 12	IX_MK 11	IX_MK 10	IX_MK 9	IX_MK 8	IX_MK 7	IX_MK 6	IX_MK 5	IX_MK 4	IX_MK 3	IX_MK 2	IX_MK 1	IX_MK 0
Value after reset:	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	IX_MK0	Mask GPTx_CCMPA event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
1	IX_MK1	Mask GPTx_CCMPB event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
2	IX_MK2	Mask GPTx_CMPC event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
3	IX_MK3	Mask GPTx_CMPD event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
4	IX_MK4	Mask GPTx_CMPE event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
5	IX_MK5	Mask GPTx_CMPF event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
6	IX_MK6	Mask GPTx_OVF event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
7	IX_MK7	Mask GPTx_UDF event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
8	IX_MK8	Mask GPTx_DTE event source from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
9	IX_MK9	Mask MTU3 event source 0 from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W

Bit	Symbol	Function	R/W
10	IX_MK10	Mask MTU3 event source 1 from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
11	IX_MK11	Mask MTU3 event source 2 from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
12	IX_MK12	Mask MTU3 event source 3 from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
13	IX_MK13	Mask MTU3 event source 4 from GPT/MTU3 combined event, GPTx_INT4 0: No masked 1: Masked	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	IY_MK0	Mask GPTy_CCMPA event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
17	IY_MK1	Mask GPTy_CCMPB event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
18	IY_MK2	Mask GPTy_CMPC event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
19	IY_MK3	Mask GPTy_CMPD event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
20	IY_MK4	Mask GPTy_CMPE event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
21	IY_MK5	Mask GPTy_CMPF event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
22	IY_MK6	Mask GPTy_OVF event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
23	IY_MK7	Mask GPTy_UDF event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
24	IY_MK8	Mask GPTy_DTE event source from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
25	IY_MK9	Mask MTU3 event source 0 from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
26	IY_MK10	Mask MTU3 event source 1 from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
27	IY_MK11	Mask MTU3 event source 2 from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
28	IY_MK12	Mask MTU3 event source 3 from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W
29	IY_MK13	Mask MTU3 event source 4 from GPT/MTU3 combined event, GPTy_INT4 0: No masked 1: Masked	R/W

Bit	Symbol	Function	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: Relation among n, x and y is shown in the below table.

n	x	y	n	x	y	n	x	y
0	00_0	00_1	10	04_0	04_1	20	08_0	08_1
1	00_2	00_3	11	04_2	04_3	21	08_2	08_3
2	00_4	01_0	12	04_4	05_0	22	08_4	09_0
3	01_1	01_2	13	05_1	05_2	23	09_1	09_2
4	01_3	01_4	14	05_3	05_4	24	09_3	09_4
5	02_0	02_1	15	06_0	06_1	25	09_5	09_6
6	02_2	02_3	16	06_2	06_3	—		
7	02_4	03_0	17	06_4	07_0			
8	03_1	03_2	18	07_1	07_2			
9	03_3	03_4	19	07_3	07_4			

The ELC_GPT_INTMSKn register is used to mask GPT (GPT00 to GPT09) and MTU3 event sources to be combined as GPTx/y_INT4 event source for ELC. When the bit is masked, corresponding event does not occur GPTx/y_INT4 event.

The bits for MTU3 event sources are valid only for n = 0 to 4. Table 12.14 shows MTU3 event source mapping.

Table 12.14 MTU3 event source mapping

	n = 0		n = 1		n = 2		n = 3		n = 4		n = 5 to 25
	x	y	x	y	x	y	x	y	x	y	x/y
Bit	00_0	00_1	00_2	00_3	00_4	01_0	01_1	01_2	01_3	01_4	02_0 to 09_6
9 / 25	TGIA0	TGIE0	TCIU1	TGIA3	TGIA4	TGIU5	TGIA6	TGIA7	TGIA8	—	—
10 / 26	TGIB0	TGIF0	TGIA2	TGIB3	TGIB4	TGIV5	TGIB6	TGIB7	TGIB8	—	—
11 / 27	TGIC0	TGIA1	TGIB2	TGIC3	TGIC4	TGIW5	TGIC6	TGIC7	TGIC8	—	—
12 / 28	TGID0	TGIB1	TCIV2	TGID3	TGID4	—	TGID6	TGID7	TGID8	—	—
13 / 29	TCIV0	TCIV1	TCIU2	TCIV3	TCIV4	—	TCIV6	TCIV7	TCIV8	—	—

12.3.3.3 ELC_GPT_INTCLRn : GPT/MTU3 Combined ELC Event Status Clear Register n (n = 0 to 25)

Base address: ICU_NS = 0x802A_0000

Offset address: 0xA50 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	IY_CL 13	IY_CL 12	IY_CL 11	IY_CL 10	IY_CL 9	IY_CL 8	IY_CL 7	IY_CL 6	IY_CL 5	IY_CL 4	IY_CL 3	IY_CL 2	IY_CL 1	IY_CL 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IX_CL 13	IX_CL 12	IX_CL 11	IX_CL 10	IX_CL 9	IX_CL 8	IX_CL 7	IX_CL 6	IX_CL 5	IX_CL 4	IX_CL 3	IX_CL 2	IX_CL 1	IX_CL 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IX_CL0	Clear GPTx_CCMPA event status captured in ELC_GPT_INTSTATn register by writing 1	W
1	IX_CL1	Clear GPTx_CCMPB event status captured in ELC_GPT_INTSTATn register by writing 1	W
2	IX_CL2	Clear GPTx_CMPC event status captured in ELC_GPT_INTSTATn register by writing 1	W

Bit	Symbol	Function	R/W
3	IX_CL3	Clear GPTx_CMPD event status captured in ELC_GPT_INTSTATn register by writing 1	W
4	IX_CL4	Clear GPTx_CMPE event status captured in ELC_GPT_INTSTATn register by writing 1	W
5	IX_CL5	Clear GPTx_CMPF event status captured in ELC_GPT_INTSTATn register by writing 1	W
6	IX_CL6	Clear GPTx_OVF event status captured in ELC_GPT_INTSTATn register by writing 1	W
7	IX_CL7	Clear GPTx_UDF event status captured in ELC_GPT_INTSTATn register by writing 1	W
8	IX_CL8	Clear GPTx_DTE event status captured in ELC_GPT_INTSTATn register by writing 1	W
9	IX_CL9	Clear MTU3 event status 0 captured in ELC_GPT_INTSTATn register by writing 1	W
10	IX_CL10	Clear MTU3 event status 1 captured in ELC_GPT_INTSTATn register by writing 1	W
11	IX_CL11	Clear MTU3 event status 2 captured in ELC_GPT_INTSTATn register by writing 1	W
12	IX_CL12	Clear MTU3 event status 3 captured in ELC_GPT_INTSTATn register by writing 1	W
13	IX_CL13	Clear MTU3 event status 4 captured in ELC_GPT_INTSTATn register by writing 1	W
15:14	—	The write value should be 0.	W
16	IY_CL0	Clear GPTy_CCMPA event status captured in ELC_GPT_INTSTATn register by writing 1	W
17	IY_CL1	Clear GPTy_CCMPB event status captured in ELC_GPT_INTSTATn register by writing 1	W
18	IY_CL2	Clear GPTy_CMPC event status captured in ELC_GPT_INTSTATn register by writing 1	W
19	IY_CL3	Clear GPTy_CMPD event status captured in ELC_GPT_INTSTATn register by writing 1	W
20	IY_CL4	Clear GPTy_CMPE event status captured in ELC_GPT_INTSTATn register by writing 1	W
21	IY_CL5	Clear GPTy_CMPF event status captured in ELC_GPT_INTSTATn register by writing 1	W
22	IY_CL6	Clear GPTy_OVF event status captured in ELC_GPT_INTSTATn register by writing 1	W
23	IY_CL7	Clear GPTy_UDF event status captured in ELC_GPT_INTSTATn register by writing 1	W
24	IY_CL8	Clear GPTy_DTE event status captured in ELC_GPT_INTSTATn register by writing 1	W
25	IY_CL9	Clear MTU3 event status 0 captured in ELC_GPT_INTSTATn register by writing 1	W
26	IY_CL10	Clear MTU3 event status 1 captured in ELC_GPT_INTSTATn register by writing 1	W
27	IY_CL11	Clear MTU3 event status 2 captured in ELC_GPT_INTSTATn register by writing 1	W
28	IY_CL12	Clear MTU3 event status 3 captured in ELC_GPT_INTSTATn register by writing 1	W
29	IY_CL13	Clear MTU3 event status 4 captured in ELC_GPT_INTSTATn register by writing 1	W
31:30	—	The write value should be 0.	W

Note: Relation among n, x and y is shown in the below table.

n	x	y	n	x	y	n	x	y
0	00_0	00_1	10	04_0	04_1	20	08_0	08_1
1	00_2	00_3	11	04_2	04_3	21	08_2	08_3
2	00_4	01_0	12	04_4	05_0	22	08_4	09_0
3	01_1	01_2	13	05_1	05_2	23	09_1	09_2
4	01_3	01_4	14	05_3	05_4	24	09_3	09_4
5	02_0	02_1	15	06_0	06_1	25	09_5	09_6
6	02_2	02_3	16	06_2	06_3	—		
7	02_4	03_0	17	06_4	07_0			
8	03_1	03_2	18	07_1	07_2			
9	03_3	03_4	19	07_3	07_4			

The ELC_GPT_INTCLRn register is used to clear GPT (GPT00 to GPT09) and MTU3 event status captured in ELC_GPT_INTSTATn register.

12.3.3.4 ELC_GPT_INTSTATn : GPT/MTU3 Combined ELC Event Status Register n (n = 0 to 25)

Base address: ICU_NS = 0x802A_0000

Offset address: 0xB30 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	IY_ST 13	IY_ST 12	IY_ST 11	IY_ST 10	IY_ST 9	IY_ST 8	IY_ST 7	IY_ST 6	IY_ST 5	IY_ST 4	IY_ST 3	IY_ST 2	IY_ST 1	IY_ST 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IX_ST 13	IX_ST 12	IX_ST 11	IX_ST 10	IX_ST 9	IX_ST 8	IX_ST 7	IX_ST 6	IX_ST 5	IX_ST 4	IX_ST 3	IX_ST 2	IX_ST 1	IX_ST 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IX_ST0	Indicate captured GPTx_CCMPA event status 0: No event occurs 1: Event occurs	R
1	IX_ST1	Indicate captured GPTx_CCMPB event status 0: No event occurs 1: Event occurs	R
2	IX_ST2	Indicate captured GPTx_CMPC event status 0: No event occurs 1: Event occurs	R
3	IX_ST3	Indicate captured GPTx_CMPD event status 0: No event occurs 1: Event occurs	R
4	IX_ST4	Indicate captured GPTx_CMPE event status 0: No event occurs 1: Event occurs	R
5	IX_ST5	Indicate captured GPTx_CMPF event status 0: No event occurs 1: Event occurs	R
6	IX_ST6	Indicate captured GPTx_OVF event status 0: No event occurs 1: Event occurs	R
7	IX_ST7	Indicate captured GPTx_UDF event status 0: No event occurs 1: Event occurs	R
8	IX_ST8	Indicate captured GPTx_DTE event status 0: No event occurs 1: Event occurs	R
9	IX_ST9	Indicate captured MTU3 event status 0 0: No event occurs 1: Event occurs	R
10	IX_ST10	Indicate captured MTU3 event status 1 0: No event occurs 1: Event occurs	R
11	IX_ST11	Indicate captured MTU3 event status 2 0: No event occurs 1: Event occurs	R
12	IX_ST12	Indicate captured MTU3 event status 3 0: No event occurs 1: Event occurs	R

Bit	Symbol	Function	R/W
13	IX_ST13	Indicate captured MTU3 event status 4 0: No event occurs 1: Event occurs	R
15:14	—	These bits are read as 0.	R
16	IY_ST0	Indicate captured GPTy_CCMPA event status 0: No event occurs 1: Event occurs	R
17	IY_ST1	Indicate captured GPTy_CCMPB event status 0: No event occurs 1: Event occurs	R
18	IY_ST2	Indicate captured GPTy_CMPC event status 0: No event occurs 1: Event occurs	R
19	IY_ST3	Indicate captured GPTy_CMPD event status 0: No event occurs 1: Event occurs	R
20	IY_ST4	Indicate captured GPTy_CMPE event status 0: No event occurs 1: Event occurs	R
21	IY_ST5	Indicate captured GPTy_CMPF event status 0: No event occurs 1: Event occurs	R
22	IY_ST6	Indicate captured GPTy_OVF event status 0: No event occurs 1: Event occurs	R
23	IY_ST7	Indicate captured GPTy_UDF event status 0: No event occurs 1: Event occurs	R
24	IY_ST8	Indicate captured GPTy_DTE event status 0: No event occurs 1: Event occurs	R
25	IY_ST9	Indicate captured MTU3 event status 0 0: No event occurs 1: Event occurs	R
26	IY_ST10	Indicate captured MTU3 event status 1 0: No event occurs 1: Event occurs	R
27	IY_ST11	Indicate captured MTU3 event status 2 0: No event occurs 1: Event occurs	R
28	IY_ST12	Indicate captured MTU3 event status 3 0: No event occurs 1: Event occurs	R
29	IY_ST13	Indicate captured MTU3 event status 4 0: No event occurs 1: Event occurs	R
31:30	—	These bits are read as 0.	R

Note: Relation among n, x and y is shown in the below table.

n	x	y	n	x	y	n	x	y
0	00_0	00_1	10	04_0	04_1	20	08_0	08_1
1	00_2	00_3	11	04_2	04_3	21	08_2	08_3
2	00_4	01_0	12	04_4	05_0	22	08_4	09_0

n	x	y	n	x	y	n	x	y
3	01_1	01_2	13	05_1	05_2	23	09_1	09_2
4	01_3	01_4	14	05_3	05_4	24	09_3	09_4
5	02_0	02_1	15	06_0	06_1	25	09_5	09_6
6	02_2	02_3	16	06_2	06_3	—		
7	02_4	03_0	17	06_4	07_0			
8	03_1	03_2	18	07_1	07_2			
9	03_3	03_4	19	07_3	07_4			

The ELC_GPT_INTSTATn register indicates captured status for GPT (GPT00 to GPT09) and MTU3 events for ELC.

The bits for MTU3 event status are valid only for n = 0 to 4. Table 12.15 shows MTU3 event status mapping.

Table 12.15 MTU3 event status mapping

	n = 0		n = 1		n = 2		n = 3		n = 4		n = 5 to 25
	x	y	x	y	x	y	x	y	x	y	x/y
Bit	00_0	00_1	00_2	00_3	00_4	01_0	01_1	01_2	01_3	01_4	02_0 to 09_6
9 / 25	TGIA0	TGIE0	TCIU1	TGIA3	TGIA4	TGIU5	TGIA6	TGIA7	TGIA8	—	—
10 / 26	TGIB0	TGIF0	TGIA2	TGIB3	TGIB4	TGIV5	TGIB6	TGIB7	TGIB8	—	—
11 / 27	TGIC0	TGIA1	TGIB2	TGIC3	TGIC4	TGIW5	TGIC6	TGIC7	TGIC8	—	—
12 / 28	TGID0	TGIB1	TCIV2	TGID3	TGID4	—	TGID6	TGID7	TGID8	—	—
13 / 29	TCIV0	TCIV1	TCIU2	TCIV3	TCIV4	—	TCIV6	TCIV7	TCIV8	—	—

12.4 Operations

12.4.1 Event Handling

Events and interrupts from internal modules and external pins include the external DMA request pins are connected to the GIC (Generic Interrupt Controller) SPI (Shared Peripheral Interrupts) of the CPUs (Cortex-A55, Cortex-R52 CPU0, and Cortex-R52 CPU1) and/or the DMAC and ELC as an activating trigger signal.

GPT/MTU3 events are selectable from max 14 to 4 interrupts each channel by setting NS_GPT_INTSELn (n = 0 to 25) register and S_GPT_INTSELn (n = 0, 1) register. Additionally, a combined interrupt is available each channel. Original event sources of combined interrupt are maskable by NS_GPT_INTMSKn (n = 0 to 25) register and S_GPT_INTMSKn (n = 0, 1) register, and status of the event sources is capturable to identify the event source which causes assert of the combined interrupt. Figure 12.2 shows the diagram of GPT/MTU3 event selector (GPT_SEL). GPT/MTU3 selected interrupt can be selected for CPU/DMAC interrupt and ELC source separately.

Encoder interface (ENCIF) events are selectable from 5 modules to 1 module each unit by setting ENCIF_ENCSELn (n = 0, 1) register. Figure 12.3 shows the diagram of ENCIF event selector (ENCIF_SEL).

Assignment of the event source to Cortex-R52 GIC SPI is partially fixed and partially selectable. SELECT0_IRQ[159:0] for CPU0 and SELECT1_IRQ[159:0] for CPU1 are assigned as selectable events via Cortex-R52 SPI event selectors (CR520_SEL and CR521_SEL). Its assignment is set by CR520_INTSELn (n =0 to 79) register and CR521_INTSELn (n = 0 to 79) register. Assignment to Cortex-A55 GIC-600 SPI is fixed for all events. Figure 12.4 shows the interrupt connection diagram.

Table 12.16 shows the relation between selected interrupts and control register each selector. Table 12.17 shows event table of this LSI. The checked event in the Cortex-A55 GIC-600 SPI, Cortex-R52 CPU0, CPU1 GIC SPI , DMAC Activation Request, and ELC Source columns indicates that there is a connection to the modules. Event number is common to GIC SPI number, DMACn_RSSELi register (n = 0 to 2 and i = 0 to 5), and ELC_SSELn register (n = 0 to 64). Table 12.18 shows Cortex-A55 PPI (Private Peripheral Interrupts) events. Table 12.19 and Table 12.20 show Cortex-R52 CPU0 and CPU1 selectable events, respectively. Table 12.21 shows GPT events to be input to GPT_SEL. Table 12.22 shows ENCIF events to be input to ENCIF_SEL. Event number is not assigned to these events since registers in Table 12.16 are used to select event.

GIC handles these interrupts. When specifying as INTID, add 32 to the SPI Interface number (event number) considering INTID of SGI (0 to 15) and PPI (16 to 31). Regarding to GIC-600 and GIC in Cortex-R52, please refer to the below documents.

- Arm® CoreLink™ GIC-600 Generic Interrupt Controller Technical Reference Manual
- Arm® Cortex®-R52 Processor Technical Reference Manual

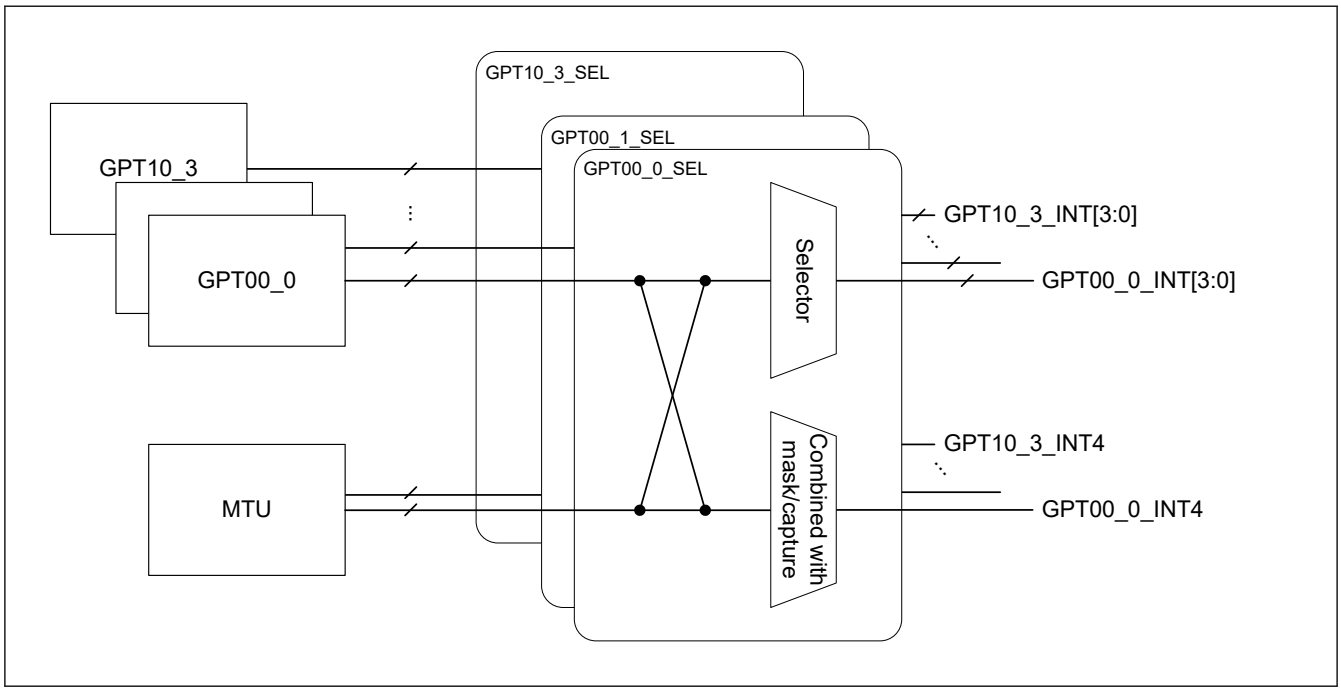


Figure 12.2 Block diagram of GPT/MUT3 event selector (GPT_SEL)

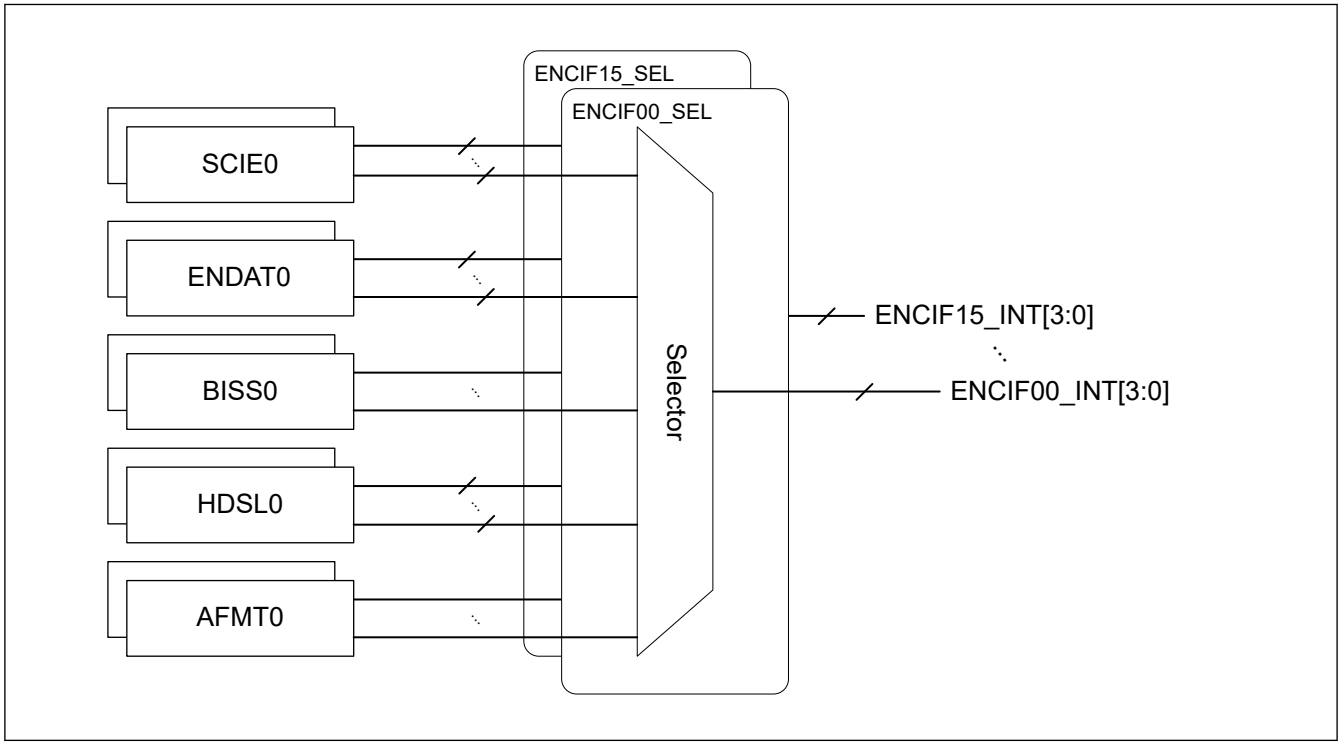


Figure 12.3 Block diagram of encoder interface event selector (ENCIF_SEL)

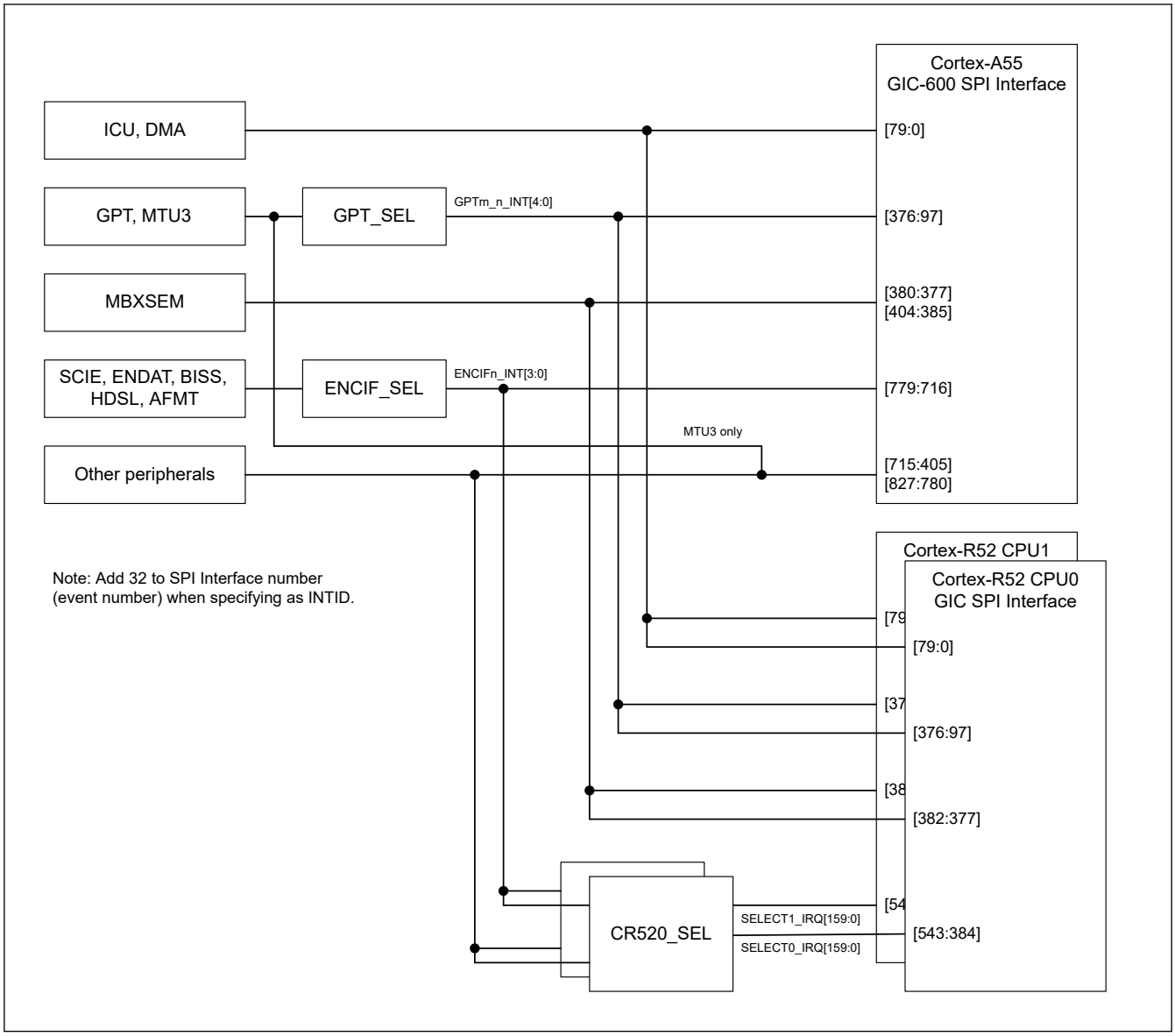


Figure 12.4 Interrupt connection diagram

Table 12.16 Selected interrupts and control register each selector (1 of 2)

Selector	Target	Selected interrupt	Control register	
GPT_SEL	Cortex-A55, Cortex-R52 CPU0, Cortex-R52 CPU1, DMAC	GPTm_n_INT[3:0] (m: unit number (00 to 10), n: channel number (depend on unit))	NS_GPT_INTSELn (n = 0 to 25) S_GPT_INTSELn (n = 0, 1)	
		GPTm_n_INT4 (m: unit number (00 to 10), n: channel number (depend on unit))	Mask	NS_GPT_INTMSKn (n = 0 to 25) S_GPT_INTMSKn (n = 0, 1)
			Status Clear	NS_GPT_INTCLRn (n = 0 to 25) S_GPT_INTCLRn (n = 0, 1)
			Status	NS_GPT_INTSTATn (n = 0 to 25) S_GPT_INTSTATn (n = 0, 1)
GPT_SEL (ELC)	ELC	GPTm_n_INT[3:0] (m: unit number (0 to 9), n: channel number (depend on unit))	ELC_GPT_INTSELn (n = 0 to 25)	
		GPTm_n_INT4 (m: unit number (0 to 9), n: channel number (depend on unit))	Mask	ELC_GPT_INTMSKn (n = 0 to 25)
			Status Clear	ELC_GPT_INTCLRn (n = 0 to 25)
			Status	ELC_GPT_INTSTATn (n = 0 to 25)

Table 12.16 Selected interrupts and control register each selector (2 of 2)

Selector	Target	Selected interrupt	Control register
ENCIF_SEL	Cortex-A55, Cortex-R52 CPU0, Cortex-R52 CPU1, DMAC, ELC	ENCIFn_INT[3:0] (n = 0 to 15)	ENCIF_ENCSELn (n = 0, 1)
CR520_SEL	Cortex-R52 CPU0	SELECT0_IRQ[159:0]	CR520_INTSELn (n = 0 to 79)
CR521_SEL	Cortex-R52 CPU1	SELECT1_IRQ[159:0]	CR521_INTSELn (n = 0 to 79)

12.4.2 Event Table

Table 12.17 shows event table of this device.

Table 12.17 Event table (1 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
CA55	0	—	nCLUSTERPMUIRQ	Cluster PMU interrupt request	Level	405 (0x195)	✓	#CR520_SEL	#CR521_SEL	—	—
ICU	—	—	INTCPU0	Software interrupt 0	Edge	0 (0x000)	✓	✓	✓	✓	✓
			INTCPU1	Software interrupt 1	Edge	1 (0x001)	✓	✓	✓	✓	✓
			INTCPU2	Software interrupt 2	Edge	2 (0x002)	✓	✓	✓	✓	✓
			INTCPU3	Software interrupt 3	Edge	3 (0x003)	✓	✓	✓	✓	✓
			INTCPU4	Software interrupt 4	Edge	4 (0x004)	✓	✓	✓	✓	✓
			INTCPU5	Software interrupt 5	Edge	5 (0x005)	✓	✓	✓	✓	✓
			INTCPU6	Software interrupt 6	Edge	6 (0x006)	✓	✓	✓	✓	✓
			INTCPU7	Software interrupt 7	Edge	7 (0x007)	✓	✓	✓	✓	✓
			INTCPU8	Software interrupt 8	Edge	8 (0x008)	✓	✓	✓	✓	✓
			INTCPU9	Software interrupt 9	Edge	9 (0x009)	✓	✓	✓	✓	✓
			INTCPU10	Software interrupt 10	Edge	10 (0x00A)	✓	✓	✓	✓	✓
			INTCPU11	Software interrupt 11	Edge	11 (0x00B)	✓	✓	✓	✓	✓
			INTCPU12	Software interrupt 12	Edge	12 (0x00C)	✓	✓	✓	✓	✓
			INTCPU13	Software interrupt 13	Edge	13 (0x00D)	✓	✓	✓	✓	✓
			INTCPU14	Software interrupt 14	Edge	14 (0x00E)	✓	✓	✓	—	—
			INTCPU15	Software interrupt 15	Edge	15 (0x00F)	✓	✓	✓	—	—

Table 12.17 Event table (2 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
ICU	—	—	IRQ0	External pin interrupt 0	Level/ Edge	16 (0x010)	✓	✓	✓	✓	—
			IRQ1	External pin interrupt 1	Level/ Edge	17 (0x011)	✓	✓	✓	✓	—
			IRQ2	External pin interrupt 2	Level/ Edge	18 (0x012)	✓	✓	✓	✓	—
			IRQ3	External pin interrupt 3	Level/ Edge	19 (0x013)	✓	✓	✓	✓	—
			IRQ4	External pin interrupt 4	Level/ Edge	20 (0x014)	✓	✓	✓	✓	—
			IRQ5	External pin interrupt 5	Level/ Edge	21 (0x015)	✓	✓	✓	✓	—
			IRQ6	External pin interrupt 6	Level/ Edge	22 (0x016)	✓	✓	✓	✓	—
			IRQ7	External pin interrupt 7	Level/ Edge	23 (0x017)	✓	✓	✓	✓	—
			IRQ8	External pin interrupt 8	Level/ Edge	24 (0x018)	✓	✓	✓	✓	—
			IRQ9	External pin interrupt 9	Level/ Edge	25 (0x019)	✓	✓	✓	✓	—
			IRQ10	External pin interrupt 10	Level/ Edge	26 (0x01A)	✓	✓	✓	✓	—
			IRQ11	External pin interrupt 11	Level/ Edge	27 (0x01B)	✓	✓	✓	✓	—
			IRQ12	External pin interrupt 12	Level/ Edge	28 (0x01C)	✓	✓	✓	✓	—
			IRQ13	External pin interrupt 13	Level/ Edge	29 (0x01D)	✓	✓	✓	✓	—
			IRQ14	External pin interrupt 14	Level/ Edge	30 (0x01E)	✓	✓	✓	—	—
			IRQ15	External pin interrupt 15	Level/ Edge	31 (0x01F)	✓	✓	✓	—	—
			DREQ	External DMA request	Level/ Edge	840 (0x348)	—	—	—	✓	—
			IOPORT_GROUP1	Input edge detection of input port group 1	Edge	857 (0x359)	—	—	—	—	✓
			IOPORT_GROUP2	Input edge detection of input port group 2	Edge	858 (0x35A)	—	—	—	—	✓
			IOPORT_SINGLE0	Input edge detection of single input port 0	Edge	859 (0x35B)	—	—	—	—	✓
			IOPORT_SINGLE1	Input edge detection of single input port 1	Edge	860 (0x35C)	—	—	—	—	✓
			IOPORT_SINGLE2	Input edge detection of single input port 2	Edge	861 (0x35D)	—	—	—	—	✓
			IOPORT_SINGLE3	Input edge detection of single input port 3	Edge	862 (0x35E)	—	—	—	—	✓
			SEI	System error interrupt	Level/ Edge	406 (0x196)	✓	SEI/VSEI	SEI/VSEI	—	—
			CA55_ERR0	Cortex-A55 error event 0	Edge	407 (0x197)	✓	#CR520_SEL	#CR521_SEL	—	—
			CA55_ERR1	Cortex-A55 error event 1	Edge	408 (0x198)	✓	#CR520_SEL	#CR521_SEL	—	—

Table 12.17 Event table (3 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
ICU	—	—	CR520_ERR0	Cortex-R52 CPU0 error event 0	Edge	409 (0x199)	✓	#CR520_SEL	#CR521_SEL	—	—
			CR520_ERR1	Cortex-R52 CPU0 error event 1	Edge	410 (0x19A)	✓	#CR520_SEL	#CR521_SEL	—	—
			CR521_ERR0	Cortex-R52 CPU1 error event 0	Edge	411 (0x19B)	✓	#CR520_SEL	#CR521_SEL	—	—
			CR521_ERR1	Cortex-R52 CPU1 error event 1	Edge	412 (0x19C)	✓	#CR520_SEL	#CR521_SEL	—	—
			PERI_ERR0	Peripherals error event 0	Edge	413 (0x19D)	✓	#CR520_SEL	#CR521_SEL	—	—
			PERI_ERR1	Peripherals error event 1	Edge	414 (0x19E)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF_ERR0	DSMIF error event 0	Edge	415 (0x19F)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF_ERR1	DSMIF error event 1	Edge	416 (0x1A0)	✓	#CR520_SEL	#CR521_SEL	—	—
			ENCIF_ERR0	ENCIF error event 0	Edge	417 (0x1A1)	✓	#CR520_SEL	#CR521_SEL	—	—
			ENCIF_ERR1	ENCIF error event 1	Edge	418 (0x1A2)	✓	#CR520_SEL	#CR521_SEL	—	—
DMAC	0	—	DMAC0_INT0	DMAC0 transfer completion 0	Level/ Edge	32 (0x020)	✓	✓	✓	—	✓
			DMAC0_INT1	DMAC0 transfer completion 1	Level/ Edge	33 (0x021)	✓	✓	✓	—	✓
			DMAC0_INT2	DMAC0 transfer completion 2	Level/ Edge	34 (0x022)	✓	✓	✓	—	✓
			DMAC0_INT3	DMAC0 transfer completion 3	Level/ Edge	35 (0x023)	✓	✓	✓	—	✓
			DMAC0_INT4	DMAC0 transfer completion 4	Level/ Edge	36 (0x024)	✓	✓	✓	—	✓
			DMAC0_INT5	DMAC0 transfer completion 5	Level/ Edge	37 (0x025)	✓	✓	✓	—	✓
			DMAC0_INT6	DMAC0 transfer completion 6	Level/ Edge	38 (0x026)	✓	✓	✓	—	✓
			DMAC0_INT7	DMAC0 transfer completion 7	Level/ Edge	39 (0x027)	✓	✓	✓	—	✓
			DMAC0_INT8	DMAC0 transfer completion 8	Level/ Edge	40 (0x028)	✓	✓	✓	—	—
			DMAC0_INT9	DMAC0 transfer completion 9	Level/ Edge	41 (0x029)	✓	✓	✓	—	—
			DMAC0_INT10	DMAC0 transfer completion 10	Level/ Edge	42 (0x02A)	✓	✓	✓	—	—
			DMAC0_INT11	DMAC0 transfer completion 11	Level/ Edge	43 (0x02B)	✓	✓	✓	—	—
			DMAC0_INT12	DMAC0 transfer completion 12	Level/ Edge	44 (0x02C)	✓	✓	✓	—	—
			DMAC0_INT13	DMAC0 transfer completion 13	Level/ Edge	45 (0x02D)	✓	✓	✓	—	—
			DMAC0_INT14	DMAC0 transfer completion 14	Level/ Edge	46 (0x02E)	✓	✓	✓	—	—
			DMAC0_INT15	DMAC0 transfer completion 15	Level/ Edge	47 (0x02F)	✓	✓	✓	—	—

Table 12.17 Event table (4 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
DMAC	1	—	DMAC1_INT0	DMAC1 transfer completion 0	Level/ Edge	48 (0x030)	✓	✓	✓	—	✓
			DMAC1_INT1	DMAC1 transfer completion 1	Level/ Edge	49 (0x031)	✓	✓	✓	—	✓
			DMAC1_INT2	DMAC1 transfer completion 2	Level/ Edge	50 (0x032)	✓	✓	✓	—	✓
			DMAC1_INT3	DMAC1 transfer completion 3	Level/ Edge	51 (0x033)	✓	✓	✓	—	✓
			DMAC1_INT4	DMAC1 transfer completion 4	Level/ Edge	52 (0x034)	✓	✓	✓	—	✓
			DMAC1_INT5	DMAC1 transfer completion 5	Level/ Edge	53 (0x035)	✓	✓	✓	—	✓
			DMAC1_INT6	DMAC1 transfer completion 6	Level/ Edge	54 (0x036)	✓	✓	✓	—	✓
			DMAC1_INT7	DMAC1 transfer completion 7	Level/ Edge	55 (0x037)	✓	✓	✓	—	✓
			DMAC1_INT8	DMAC1 transfer completion 8	Level/ Edge	56 (0x038)	✓	✓	✓	—	—
			DMAC1_INT9	DMAC1 transfer completion 9	Level/ Edge	57 (0x039)	✓	✓	✓	—	—
			DMAC1_INT10	DMAC1 transfer completion 10	Level/ Edge	58 (0x03A)	✓	✓	✓	—	—
			DMAC1_INT11	DMAC1 transfer completion 11	Level/ Edge	59 (0x03B)	✓	✓	✓	—	—
			DMAC1_INT12	DMAC1 transfer completion 12	Level/ Edge	60 (0x03C)	✓	✓	✓	—	—
			DMAC1_INT13	DMAC1 transfer completion 13	Level/ Edge	61 (0x03D)	✓	✓	✓	—	—
			DMAC1_INT14	DMAC1 transfer completion 14	Level/ Edge	62 (0x03E)	✓	✓	✓	—	—
			DMAC1_INT15	DMAC1 transfer completion 15	Level/ Edge	63 (0x03F)	✓	✓	✓	—	—
	2	—	DMAC2_INT0	DMAC2 transfer completion 0	Level/ Edge	64 (0x040)	✓	✓	✓	—	✓
			DMAC2_INT1	DMAC2 transfer completion 1	Level/ Edge	65 (0x041)	✓	✓	✓	—	✓
			DMAC2_INT2	DMAC2 transfer completion 2	Level/ Edge	66 (0x042)	✓	✓	✓	—	✓
			DMAC2_INT3	DMAC2 transfer completion 3	Level/ Edge	67 (0x043)	✓	✓	✓	—	✓
			DMAC2_INT4	DMAC2 transfer completion 4	Level/ Edge	68 (0x044)	✓	✓	✓	—	✓
			DMAC2_INT5	DMAC2 transfer completion 5	Level/ Edge	69 (0x045)	✓	✓	✓	—	✓
			DMAC2_INT6	DMAC2 transfer completion 6	Level/ Edge	70 (0x046)	✓	✓	✓	—	✓
			DMAC2_INT7	DMAC2 transfer completion 7	Level/ Edge	71 (0x047)	✓	✓	✓	—	✓
			DMAC2_INT8	DMAC2 transfer completion 8	Level/ Edge	72 (0x048)	✓	✓	✓	—	—
			DMAC2_INT9	DMAC2 transfer completion 9	Level/ Edge	73 (0x049)	✓	✓	✓	—	—
			DMAC2_INT10	DMAC2 transfer completion 10	Level/ Edge	74 (0x04A)	✓	✓	✓	—	—

Table 12.17 Event table (5 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
DMAC	2	—	DMAC2_INT11	DMAC2 transfer completion 11	Level/ Edge	75 (0x04B)	✓	✓	✓	—	—
			DMAC2_INT12	DMAC2 transfer completion 12	Level/ Edge	76 (0x04C)	✓	✓	✓	—	—
			DMAC2_INT13	DMAC2 transfer completion 13	Level/ Edge	77 (0x04D)	✓	✓	✓	—	—
			DMAC2_INT14	DMAC2 transfer completion 14	Level/ Edge	78 (0x04E)	✓	✓	✓	—	—
			DMAC2_INT15	DMAC2 transfer completion 15	Level/ Edge	79 (0x04F)	✓	✓	✓	—	—
MTU3	0	0	TGIA0	MTU0.TGRA input capture/compare match	Edge	420 (0x1A4)	✓	—	—	—	—
			TGIB0	MTU0.TGRB input capture/compare match	Edge	421 (0x1A5)	✓	—	—	—	—
			TGIC0	MTU0.TGRC input capture/compare match	Edge	422 (0x1A6)	✓	—	—	—	—
			TGID0	MTU0.TGRD input capture/compare match	Edge	423 (0x1A7)	✓	—	—	—	—
			TCIV0	MTU0.TCNT overflow	Edge	424 (0x1A8)	✓	—	—	—	—
			TGIE0	MTU0.TGRE compare match	Edge	425 (0x1A9)	✓	—	—	—	—
			TGIF0	MTU0.TGRF compare match	Edge	426 (0x1AA)	✓	—	—	—	—
		1	TGIA1	MTU1.TGRA input capture/compare match	Edge	427 (0x1AB)	✓	—	—	—	—
			TGIB1	MTU1.TGRB input capture/compare match	Edge	428 (0x1AC)	✓	—	—	—	—
			TCIV1	MTU1.TCNT overflow	Edge	429 (0x1AD)	✓	—	—	—	—
			TCIU1	MTU1.TCNT underflow	Edge	430 (0x1AE)	✓	—	—	—	—
		2	TGIA2	MTU2.TGRA input capture/compare match	Edge	431 (0x1AF)	✓	—	—	—	—
			TGIB2	MTU2.TGRB input capture/compare match	Edge	432 (0x1B0)	✓	—	—	—	—
			TCIV2	MTU2.TCNT overflow	Edge	433 (0x1B1)	✓	—	—	—	—
			TCIU2	MTU2.TCNT underflow	Edge	434 (0x1B2)	✓	—	—	—	—

Table 12.17 Event table (6 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
MTU3	0	3	TGIA3	MTU3.TGRA input capture/compare match	Edge	435 (0x1B3)	✓	—	—	—	—
			TGIB3	MTU3.TGRB input capture/compare match	Edge	436 (0x1B4)	✓	—	—	—	—
			TGIC3	MTU3.TGRC input capture/compare match	Edge	437 (0x1B5)	✓	—	—	—	—
			TGID3	MTU3.TGRD input capture/compare match	Edge	438 (0x1B6)	✓	—	—	—	—
			TCIV3	MTU3.TCNT overflow	Edge	439 (0x1B7)	✓	—	—	—	—
		4	TGIA4	MTU4.TGRA input capture/compare match	Edge	440 (0x1B8)	✓	—	—	—	—
			TGIB4	MTU4.TGRB input capture/compare match	Edge	441 (0x1B9)	✓	—	—	—	—
			TGIC4	MTU4.TGRC input capture/compare match	Edge	442 (0x1BA)	✓	—	—	—	—
			TGID4	MTU4.TGRD input capture/compare match	Edge	443 (0x1BB)	✓	—	—	—	—
			TCIV4	MTU4.TCNT overflow/underflow	Edge	444 (0x1BC)	✓	—	—	—	—
			TCIV4_OF	MTU4.TCNT overflow	Edge	863 (0x35F)	—	—	—	—	✓
			TCIV4_UF	MTU4.TCNT underflow	Edge	864 (0x360)	—	—	—	—	✓
		5	TGIU5	MTU5.TGRU input capture/compare match	Edge	445 (0x1BD)	✓	—	—	—	—
			TGIV5	MTU5.TGRV input capture/compare match	Edge	446 (0x1BE)	✓	—	—	—	—
			TGIW5	MTU5.TGRW input capture/compare match	Edge	447 (0x1BF)	✓	—	—	—	—
		6	TGIA6	MTU6.TGRA input capture/compare match	Edge	448 (0x1C0)	✓	—	—	—	—
			TGIB6	MTU6.TGRB input capture/compare match	Edge	449 (0x1C1)	✓	—	—	—	—
			TGIC6	MTU6.TGRC input capture/compare match	Edge	450 (0x1C2)	✓	—	—	—	—
			TGID6	MTU6.TGRD input capture/compare match	Edge	451 (0x1C3)	✓	—	—	—	—
			TCIV6	MTU6.TCNT overflow	Edge	452 (0x1C4)	✓	—	—	—	—

Table 12.17 Event table (7 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
MTU3	0	7	TGIA7	MTU7.TGRA input capture/compare match	Edge	453 (0x1C5)	✓	—	—	—	—
			TGIB7	MTU7.TGRB input capture/compare match	Edge	454 (0x1C6)	✓	—	—	—	—
			TGIC7	MTU7.TGRC input capture/compare match	Edge	455 (0x1C7)	✓	—	—	—	—
			TGID7	MTU7.TGRD input capture/compare match	Edge	456 (0x1C8)	✓	—	—	—	—
			TCIV7	MTU7.TCNT overflow/underflow*1	Edge	457 (0x1C9)	✓	—	—	—	—
			TCIV7_OF	MTU7.TCNT overflow	Edge	865 (0x361)	—	—	—	—	✓
			TCIV7_UF	MTU7.TCNT underflow	Edge	866 (0x362)	—	—	—	—	✓
		8	TGIA8	MTU8.TGRA input capture/compare match	Edge	458 (0x1CA)	✓	—	—	—	—
			TGIB8	MTU8.TGRB input capture/compare match	Edge	459 (0x1CB)	✓	—	—	—	—
			TGIC8	MTU8.TGRC input capture/compare match	Edge	460 (0x1CC)	✓	—	—	—	—
			TGID8	MTU8.TGRD input capture/compare match	Edge	461 (0x1CD)	✓	—	—	—	—
			TCIV8	MTU8.TCNT overflow	Edge	462 (0x1CE)	✓	—	—	—	—
POE3	0	—	OEI1	Output enable interrupt 1	Level	463 (0x1CF)	✓	#CR520_SEL	#CR521_SEL	—	—
			OEI2	Output enable interrupt 2	Level	464 (0x1D0)	✓	#CR520_SEL	#CR521_SEL	—	—
			OEI3	Output enable interrupt 3	Level	465 (0x1D1)	✓	#CR520_SEL	#CR521_SEL	—	—
			OEI4	Output enable interrupt 4	Level	466 (0x1D2)	✓	#CR520_SEL	#CR521_SEL	—	—
GPT	0	0	GPT00_0_ADTRGA	GPT00_0 GTADTRA compare match (A/D Converter Start Request A)	Edge	867 (0x363)	—	—	—	—	✓
			GPT00_0_ADTRGB	GPT00_0 GTADTRB compare match (A/D Converter Start Request B)	Edge	868 (0x364)	—	—	—	—	✓
			GPT00_0_INT0	GPT00_0 selected interrupt 0	Edge	97 (0x061)	✓	✓	✓	✓	✓
			GPT00_0_INT1	GPT00_0 selected interrupt 1	Edge	98 (0x062)	✓	✓	✓	✓	✓
			GPT00_0_INT2	GPT00_0 selected interrupt 2	Edge	99 (0x063)	✓	✓	✓	✓	✓
			GPT00_0_INT3	GPT00_0 selected interrupt 3	Edge	100 (0x064)	✓	✓	✓	✓	✓
			GPT00_0_INT4	GPT00_0 combined interrupt	Edge	101 (0x065)	✓	✓	✓	✓	✓

Table 12.17 Event table (8 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	0	1	GPT00_1_ADTRGA	GPT00_1 GTADTRA compare match (A/D Converter Start Request A)	Edge	869 (0x365)	—	—	—	—	✓
			GPT00_1_ADTRGB	GPT00_1 GTADTRB compare match (A/D Converter Start Request B)	Edge	870 (0x366)	—	—	—	—	✓
			GPT00_1_INT0	GPT00_1 selected interrupt 0	Edge	102 (0x066)	✓	✓	✓	✓	✓
			GPT00_1_INT1	GPT00_1 selected interrupt 1	Edge	103 (0x067)	✓	✓	✓	✓	✓
			GPT00_1_INT2	GPT00_1 selected interrupt 2	Edge	104 (0x068)	✓	✓	✓	✓	✓
			GPT00_1_INT3	GPT00_1 selected interrupt 3	Edge	105 (0x069)	✓	✓	✓	✓	✓
			GPT00_1_INT4	GPT00_1 combined interrupt	Edge	106 (0x06A)	✓	✓	✓	✓	✓
		2	GPT00_2_ADTRGA	GPT00_2 GTADTRA compare match (A/D Converter Start Request A)	Edge	871 (0x367)	—	—	—	—	✓
			GPT00_2_ADTRGB	GPT00_2 GTADTRB compare match (A/D Converter Start Request B)	Edge	872 (0x368)	—	—	—	—	✓
			GPT00_2_INT0	GPT00_2 selected interrupt 0	Edge	107 (0x06B)	✓	✓	✓	✓	✓
			GPT00_2_INT1	GPT00_2 selected interrupt 1	Edge	108 (0x06C)	✓	✓	✓	✓	✓
			GPT00_2_INT2	GPT00_2 selected interrupt 2	Edge	109 (0x06D)	✓	✓	✓	✓	✓
			GPT00_2_INT3	GPT00_2 selected interrupt 3	Edge	110 (0x06E)	✓	✓	✓	✓	✓
			GPT00_2_INT4	GPT00_2 combined interrupt	Edge	111 (0x06F)	✓	✓	✓	✓	✓
		3	GPT00_3_ADTRGA	GPT00_3 GTADTRA compare match (A/D Converter Start Request A)	Edge	873 (0x369)	—	—	—	—	✓
			GPT00_3_ADTRGB	GPT00_3 GTADTRB compare match (A/D Converter Start Request B)	Edge	874 (0x36A)	—	—	—	—	✓
			GPT00_3_INT0	GPT00_3 selected interrupt 0	Edge	112 (0x070)	✓	✓	✓	✓	✓
			GPT00_3_INT1	GPT00_3 selected interrupt 1	Edge	113 (0x071)	✓	✓	✓	✓	✓
			GPT00_3_INT2	GPT00_3 selected interrupt 2	Edge	114 (0x072)	✓	✓	✓	✓	✓
			GPT00_3_INT3	GPT00_3 selected interrupt 3	Edge	115 (0x073)	✓	✓	✓	✓	✓
			GPT00_3_INT4	GPT00_3 combined interrupt	Edge	116 (0x074)	✓	✓	✓	✓	✓

Table 12.17 Event table (9 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	0	4	GPT00_4_ADTRGA	GPT00_4 GTADTRA compare match (A/D Converter Start Request A)	Edge	875 (0x36B)	—	—	—	—	✓
			GPT00_4_ADTRGB	GPT00_4 GTADTRB compare match (A/D Converter Start Request B)	Edge	876 (0x36C)	—	—	—	—	✓
			GPT00_4_INT0	GPT00_4 selected interrupt 0	Edge	117 (0x075)	✓	✓	✓	✓	✓
			GPT00_4_INT1	GPT00_4 selected interrupt 1	Edge	118 (0x076)	✓	✓	✓	✓	✓
			GPT00_4_INT2	GPT00_4 selected interrupt 2	Edge	119 (0x077)	✓	✓	✓	✓	✓
			GPT00_4_INT3	GPT00_4 selected interrupt 3	Edge	120 (0x078)	✓	✓	✓	✓	✓
			GPT00_4_INT4	GPT00_4 combined interrupt	Edge	121 (0x079)	✓	✓	✓	✓	✓
	1	0	GPT01_0_ADTRGA	GPT01_0 GTADTRA compare match (A/D Converter Start Request A)	Edge	877 (0x36D)	—	—	—	—	✓
			GPT01_0_ADTRGB	GPT01_0 GTADTRB compare match (A/D Converter Start Request B)	Edge	878 (0x36E)	—	—	—	—	✓
			GPT01_0_INT0	GPT01_0 selected interrupt 0	Edge	122 (0x07A)	✓	✓	✓	✓	✓
			GPT01_0_INT1	GPT01_0 selected interrupt 1	Edge	123 (0x07B)	✓	✓	✓	✓	✓
			GPT01_0_INT2	GPT01_0 selected interrupt 2	Edge	124 (0x07C)	✓	✓	✓	✓	✓
			GPT01_0_INT3	GPT01_0 selected interrupt 3	Edge	125 (0x07D)	✓	✓	✓	✓	✓
			GPT01_0_INT4	GPT01_0 combined interrupt	Edge	126 (0x07E)	✓	✓	✓	✓	✓
		1	GPT01_1_ADTRGA	GPT01_1 GTADTRA compare match (A/D Converter Start Request A)	Edge	879 (0x36F)	—	—	—	—	✓
			GPT01_1_ADTRGB	GPT01_1 GTADTRB compare match (A/D Converter Start Request B)	Edge	880 (0x370)	—	—	—	—	✓
			GPT01_1_INT0	GPT01_1 selected interrupt 0	Edge	127 (0x07F)	✓	✓	✓	✓	✓
			GPT01_1_INT1	GPT01_1 selected interrupt 1	Edge	128 (0x080)	✓	✓	✓	✓	✓
			GPT01_1_INT2	GPT01_1 selected interrupt 2	Edge	129 (0x081)	✓	✓	✓	✓	✓
			GPT01_1_INT3	GPT01_1 selected interrupt 3	Edge	130 (0x082)	✓	✓	✓	✓	✓
			GPT01_1_INT4	GPT01_1 combined interrupt	Edge	131 (0x083)	✓	✓	✓	✓	✓

Table 12.17 Event table (10 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	1	2	GPT01_2_ADTRGA	GPT01_2 GTADTRA compare match (A/D Converter Start Request A)	Edge	881 (0x371)	—	—	—	—	✓
			GPT01_2_ADTRGB	GPT01_2 GTADTRB compare match (A/D Converter Start Request B)	Edge	882 (0x372)	—	—	—	—	✓
			GPT01_2_INT0	GPT01_2 selected interrupt 0	Edge	132 (0x084)	✓	✓	✓	✓	✓
			GPT01_2_INT1	GPT01_2 selected interrupt 1	Edge	133 (0x085)	✓	✓	✓	✓	✓
			GPT01_2_INT2	GPT01_2 selected interrupt 2	Edge	134 (0x086)	✓	✓	✓	✓	✓
			GPT01_2_INT3	GPT01_2 selected interrupt 3	Edge	135 (0x087)	✓	✓	✓	✓	✓
			GPT01_2_INT4	GPT01_2 combined interrupt	Edge	136 (0x088)	✓	✓	✓	✓	✓
		3	GPT01_3_ADTRGA	GPT01_3 GTADTRA compare match (A/D Converter Start Request A)	Edge	883 (0x373)	—	—	—	—	✓
			GPT01_3_ADTRGB	GPT01_3 GTADTRB compare match (A/D Converter Start Request B)	Edge	884 (0x374)	—	—	—	—	✓
			GPT01_3_INT0	GPT01_3 selected interrupt 0	Edge	137 (0x089)	✓	✓	✓	✓	✓
			GPT01_3_INT1	GPT01_3 selected interrupt 1	Edge	138 (0x08A)	✓	✓	✓	✓	✓
			GPT01_3_INT2	GPT01_3 selected interrupt 2	Edge	139 (0x08B)	✓	✓	✓	✓	✓
			GPT01_3_INT3	GPT01_3 selected interrupt 3	Edge	140 (0x08C)	✓	✓	✓	✓	✓
			GPT01_3_INT4	GPT01_3 combined interrupt	Edge	141 (0x08D)	✓	✓	✓	✓	✓
		4	GPT01_4_ADTRGA	GPT01_4 GTADTRA compare match (A/D Converter Start Request A)	Edge	885 (0x375)	—	—	—	—	✓
			GPT01_4_ADTRGB	GPT01_4 GTADTRB compare match (A/D Converter Start Request B)	Edge	886 (0x376)	—	—	—	—	✓
			GPT01_4_INT0	GPT01_4 selected interrupt 0	Edge	142 (0x08E)	✓	✓	✓	✓	✓
			GPT01_4_INT1	GPT01_4 selected interrupt 1	Edge	143 (0x08F)	✓	✓	✓	✓	✓
			GPT01_4_INT2	GPT01_4 selected interrupt 2	Edge	144 (0x090)	✓	✓	✓	✓	✓
			GPT01_4_INT3	GPT01_4 selected interrupt 3	Edge	145 (0x091)	✓	✓	✓	✓	✓
			GPT01_4_INT4	GPT01_4 combined interrupt	Edge	146 (0x092)	✓	✓	✓	✓	✓

Table 12.17 Event table (11 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	2	0	GPT02_0_ADTRGA	GPT02_0 GTADTRA compare match (A/D Converter Start Request A)	Edge	887 (0x377)	—	—	—	—	✓
			GPT02_0_ADTRGB	GPT02_0 GTADTRB compare match (A/D Converter Start Request B)	Edge	888 (0x378)	—	—	—	—	✓
			GPT02_0_INT0	GPT02_0 selected interrupt 0	Edge	147 (0x093)	✓	✓	✓	✓	✓
			GPT02_0_INT1	GPT02_0 selected interrupt 1	Edge	148 (0x094)	✓	✓	✓	✓	✓
			GPT02_0_INT2	GPT02_0 selected interrupt 2	Edge	149 (0x095)	✓	✓	✓	✓	✓
			GPT02_0_INT3	GPT02_0 selected interrupt 3	Edge	150 (0x096)	✓	✓	✓	✓	✓
			GPT02_0_INT4	GPT02_0 combined interrupt	Edge	151 (0x097)	✓	✓	✓	✓	✓
		1	GPT02_1_ADTRGA	GPT02_1 GTADTRA compare match (A/D Converter Start Request A)	Edge	889 (0x379)	—	—	—	—	✓
			GPT02_1_ADTRGB	GPT02_1 GTADTRB compare match (A/D Converter Start Request B)	Edge	890 (0x37A)	—	—	—	—	✓
			GPT02_1_INT0	GPT02_1 selected interrupt 0	Edge	152 (0x098)	✓	✓	✓	✓	✓
			GPT02_1_INT1	GPT02_1 selected interrupt 1	Edge	153 (0x099)	✓	✓	✓	✓	✓
			GPT02_1_INT2	GPT02_1 selected interrupt 2	Edge	154 (0x09A)	✓	✓	✓	✓	✓
			GPT02_1_INT3	GPT02_1 selected interrupt 3	Edge	155 (0x09B)	✓	✓	✓	✓	✓
			GPT02_1_INT4	GPT02_1 combined interrupt	Edge	156 (0x09C)	✓	✓	✓	✓	✓
		2	GPT02_2_ADTRGA	GPT02_2 GTADTRA compare match (A/D Converter Start Request A)	Edge	891 (0x37B)	—	—	—	—	✓
			GPT02_2_ADTRGB	GPT02_2 GTADTRB compare match (A/D Converter Start Request B)	Edge	892 (0x37C)	—	—	—	—	✓
			GPT02_2_INT0	GPT02_2 selected interrupt 0	Edge	157 (0x09D)	✓	✓	✓	✓	✓
			GPT02_2_INT1	GPT02_2 selected interrupt 1	Edge	158 (0x09E)	✓	✓	✓	✓	✓
			GPT02_2_INT2	GPT02_2 selected interrupt 2	Edge	159 (0x09F)	✓	✓	✓	✓	✓
			GPT02_2_INT3	GPT02_2 selected interrupt 3	Edge	160 (0x0A0)	✓	✓	✓	✓	✓
			GPT02_2_INT4	GPT02_2 combined interrupt	Edge	161 (0x0A1)	✓	✓	✓	✓	✓

Table 12.17 Event table (12 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	2	3	GPT02_3_ADTRGA	GPT02_3 GTADTRA compare match (A/D Converter Start Request A)	Edge	893 (0x37D)	—	—	—	—	✓
			GPT02_3_ADTRGB	GPT02_3 GTADTRB compare match (A/D Converter Start Request B)	Edge	894 (0x37E)	—	—	—	—	✓
			GPT02_3_INT0	GPT02_3 selected interrupt 0	Edge	162 (0x0A2)	✓	✓	✓	✓	✓
			GPT02_3_INT1	GPT02_3 selected interrupt 1	Edge	163 (0x0A3)	✓	✓	✓	✓	✓
			GPT02_3_INT2	GPT02_3 selected interrupt 2	Edge	164 (0x0A4)	✓	✓	✓	✓	✓
			GPT02_3_INT3	GPT02_3 selected interrupt 3	Edge	165 (0x0A5)	✓	✓	✓	✓	✓
			GPT02_3_INT4	GPT02_3 combined interrupt	Edge	166 (0x0A6)	✓	✓	✓	✓	✓
		4	GPT02_4_ADTRGA	GPT02_4 GTADTRA compare match (A/D Converter Start Request A)	Edge	895 (0x37F)	—	—	—	—	✓
			GPT02_4_ADTRGB	GPT02_4 GTADTRB compare match (A/D Converter Start Request B)	Edge	896 (0x380)	—	—	—	—	✓
			GPT02_4_INT0	GPT02_4 selected interrupt 0	Edge	167 (0x0A7)	✓	✓	✓	✓	✓
			GPT02_4_INT1	GPT02_4 selected interrupt 1	Edge	168 (0x0A8)	✓	✓	✓	✓	✓
			GPT02_4_INT2	GPT02_4 selected interrupt 2	Edge	169 (0x0A9)	✓	✓	✓	✓	✓
			GPT02_4_INT3	GPT02_4 selected interrupt 3	Edge	170 (0x0AA)	✓	✓	✓	✓	✓
			GPT02_4_INT4	GPT02_4 combined interrupt	Edge	171 (0x0AB)	✓	✓	✓	✓	✓
	3	0	GPT03_0_ADTRGA	GPT03_0 GTADTRA compare match (A/D Converter Start Request A)	Edge	897 (0x381)	—	—	—	—	✓
			GPT03_0_ADTRGB	GPT03_0 GTADTRB compare match (A/D Converter Start Request B)	Edge	898 (0x382)	—	—	—	—	✓
			GPT03_0_INT0	GPT03_0 selected interrupt 0	Edge	172 (0x0AC)	✓	✓	✓	✓	✓
			GPT03_0_INT1	GPT03_0 selected interrupt 1	Edge	173 (0x0AD)	✓	✓	✓	✓	✓
			GPT03_0_INT2	GPT03_0 selected interrupt 2	Edge	174 (0x0AE)	✓	✓	✓	✓	✓
			GPT03_0_INT3	GPT03_0 selected interrupt 3	Edge	175 (0x0AF)	✓	✓	✓	✓	✓
			GPT03_0_INT4	GPT03_0 combined interrupt	Edge	176 (0x0B0)	✓	✓	✓	✓	✓

Table 12.17 Event table (13 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	3	1	GPT03_1_ADTRGA	GPT03_1 GTADTRA compare match (A/D Converter Start Request A)	Edge	899 (0x383)	—	—	—	—	✓
			GPT03_1_ADTRGB	GPT03_1 GTADTRB compare match (A/D Converter Start Request B)	Edge	900 (0x384)	—	—	—	—	✓
			GPT03_1_INT0	GPT03_1 selected interrupt 0	Edge	177 (0x0B1)	✓	✓	✓	✓	✓
			GPT03_1_INT1	GPT03_1 selected interrupt 1	Edge	178 (0x0B2)	✓	✓	✓	✓	✓
			GPT03_1_INT2	GPT03_1 selected interrupt 2	Edge	179 (0x0B3)	✓	✓	✓	✓	✓
			GPT03_1_INT3	GPT03_1 selected interrupt 3	Edge	180 (0x0B4)	✓	✓	✓	✓	✓
			GPT03_1_INT4	GPT03_1 combined interrupt	Edge	181 (0x0B5)	✓	✓	✓	✓	✓
		2	GPT03_2_ADTRGA	GPT03_2 GTADTRA compare match (A/D Converter Start Request A)	Edge	901 (0x385)	—	—	—	—	✓
			GPT03_2_ADTRGB	GPT03_2 GTADTRB compare match (A/D Converter Start Request B)	Edge	902 (0x386)	—	—	—	—	✓
			GPT03_2_INT0	GPT03_2 selected interrupt 0	Edge	182 (0x0B6)	✓	✓	✓	✓	✓
			GPT03_2_INT1	GPT03_2 selected interrupt 1	Edge	183 (0x0B7)	✓	✓	✓	✓	✓
			GPT03_2_INT2	GPT03_2 selected interrupt 2	Edge	184 (0x0B8)	✓	✓	✓	✓	✓
			GPT03_2_INT3	GPT03_2 selected interrupt 3	Edge	185 (0x0B9)	✓	✓	✓	✓	✓
			GPT03_2_INT4	GPT03_2 combined interrupt	Edge	186 (0x0BA)	✓	✓	✓	✓	✓
		3	GPT03_3_ADTRGA	GPT03_3 GTADTRA compare match (A/D Converter Start Request A)	Edge	903 (0x387)	—	—	—	—	✓
			GPT03_3_ADTRGB	GPT03_3 GTADTRB compare match (A/D Converter Start Request B)	Edge	904 (0x388)	—	—	—	—	✓
			GPT03_3_INT0	GPT03_3 selected interrupt 0	Edge	187 (0x0BB)	✓	✓	✓	✓	✓
			GPT03_3_INT1	GPT03_3 selected interrupt 1	Edge	188 (0x0BC)	✓	✓	✓	✓	✓
			GPT03_3_INT2	GPT03_3 selected interrupt 2	Edge	189 (0x0BD)	✓	✓	✓	✓	✓
			GPT03_3_INT3	GPT03_3 selected interrupt 3	Edge	190 (0x0BE)	✓	✓	✓	✓	✓
			GPT03_3_INT4	GPT03_3 combined interrupt	Edge	191 (0x0BF)	✓	✓	✓	✓	✓

Table 12.17 Event table (14 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	3	4	GPT03_4_ADTRGA	GPT03_4 GTADTRA compare match (A/D Converter Start Request A)	Edge	905 (0x389)	—	—	—	—	✓
			GPT03_4_ADTRGB	GPT03_4 GTADTRB compare match (A/D Converter Start Request B)	Edge	906 (0x38A)	—	—	—	—	✓
			GPT03_4_INT0	GPT03_4 selected interrupt 0	Edge	192 (0x0C0)	✓	✓	✓	✓	✓
			GPT03_4_INT1	GPT03_4 selected interrupt 1	Edge	193 (0x0C1)	✓	✓	✓	✓	✓
			GPT03_4_INT2	GPT03_4 selected interrupt 2	Edge	194 (0x0C2)	✓	✓	✓	✓	✓
			GPT03_4_INT3	GPT03_4 selected interrupt 3	Edge	195 (0x0C3)	✓	✓	✓	✓	✓
			GPT03_4_INT4	GPT03_4 combined interrupt	Edge	196 (0x0C4)	✓	✓	✓	✓	✓
	4	0	GPT04_0_ADTRGA	GPT04_0 GTADTRA compare match (A/D Converter Start Request A)	Edge	907 (0x38B)	—	—	—	—	✓
			GPT04_0_ADTRGB	GPT04_0 GTADTRB compare match (A/D Converter Start Request B)	Edge	908 (0x38C)	—	—	—	—	✓
			GPT04_0_INT0	GPT04_0 selected interrupt 0	Edge	197 (0x0C5)	✓	✓	✓	✓	✓
			GPT04_0_INT1	GPT04_0 selected interrupt 1	Edge	198 (0x0C6)	✓	✓	✓	✓	✓
			GPT04_0_INT2	GPT04_0 selected interrupt 2	Edge	199 (0x0C7)	✓	✓	✓	✓	✓
			GPT04_0_INT3	GPT04_0 selected interrupt 3	Edge	200 (0x0C8)	✓	✓	✓	✓	✓
			GPT04_0_INT4	GPT04_0 combined interrupt	Edge	201 (0x0C9)	✓	✓	✓	✓	✓
		1	GPT04_1_ADTRGA	GPT04_1 GTADTRA compare match (A/D Converter Start Request A)	Edge	909 (0x38D)	—	—	—	—	✓
			GPT04_1_ADTRGB	GPT04_1 GTADTRB compare match (A/D Converter Start Request B)	Edge	910 (0x38E)	—	—	—	—	✓
			GPT04_1_INT0	GPT04_1 selected interrupt 0	Edge	202 (0x0CA)	✓	✓	✓	✓	✓
			GPT04_1_INT1	GPT04_1 selected interrupt 1	Edge	203 (0x0CB)	✓	✓	✓	✓	✓
			GPT04_1_INT2	GPT04_1 selected interrupt 2	Edge	204 (0x0CC)	✓	✓	✓	✓	✓
			GPT04_1_INT3	GPT04_1 selected interrupt 3	Edge	205 (0x0CD)	✓	✓	✓	✓	✓
			GPT04_1_INT4	GPT04_1 combined interrupt	Edge	206 (0x0CE)	✓	✓	✓	✓	✓

Table 12.17 Event table (15 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	4	2	GPT04_2_ADTRGA	GPT04_2 GTADTRA compare match (A/D Converter Start Request A)	Edge	911 (0x38F)	—	—	—	—	✓
			GPT04_2_ADTRGB	GPT04_2 GTADTRB compare match (A/D Converter Start Request B)	Edge	912 (0x390)	—	—	—	—	✓
			GPT04_2_INT0	GPT04_2 selected interrupt 0	Edge	207 (0x0CF)	✓	✓	✓	✓	✓
			GPT04_2_INT1	GPT04_2 selected interrupt 1	Edge	208 (0x0D0)	✓	✓	✓	✓	✓
			GPT04_2_INT2	GPT04_2 selected interrupt 2	Edge	209 (0x0D1)	✓	✓	✓	✓	✓
			GPT04_2_INT3	GPT04_2 selected interrupt 3	Edge	210 (0x0D2)	✓	✓	✓	✓	✓
			GPT04_2_INT4	GPT04_2 combined interrupt	Edge	211 (0x0D3)	✓	✓	✓	✓	✓
		3	GPT04_3_ADTRGA	GPT04_3 GTADTRA compare match (A/D Converter Start Request A)	Edge	913 (0x391)	—	—	—	—	✓
			GPT04_3_ADTRGB	GPT04_3 GTADTRB compare match (A/D Converter Start Request B)	Edge	914 (0x392)	—	—	—	—	✓
			GPT04_3_INT0	GPT04_3 selected interrupt 0	Edge	212 (0x0D4)	✓	✓	✓	✓	✓
			GPT04_3_INT1	GPT04_3 selected interrupt 1	Edge	213 (0x0D5)	✓	✓	✓	✓	✓
			GPT04_3_INT2	GPT04_3 selected interrupt 2	Edge	214 (0x0D6)	✓	✓	✓	✓	✓
			GPT04_3_INT3	GPT04_3 selected interrupt 3	Edge	215 (0x0D7)	✓	✓	✓	✓	✓
			GPT04_3_INT4	GPT04_3 combined interrupt	Edge	216 (0x0D8)	✓	✓	✓	✓	✓
		4	GPT04_4_ADTRGA	GPT04_4 GTADTRA compare match (A/D Converter Start Request A)	Edge	915 (0x393)	—	—	—	—	✓
			GPT04_4_ADTRGB	GPT04_4 GTADTRB compare match (A/D Converter Start Request B)	Edge	916 (0x394)	—	—	—	—	✓
			GPT04_4_INT0	GPT04_4 selected interrupt 0	Edge	217 (0x0D9)	✓	✓	✓	✓	✓
			GPT04_4_INT1	GPT04_4 selected interrupt 1	Edge	218 (0x0DA)	✓	✓	✓	✓	✓
			GPT04_4_INT2	GPT04_4 selected interrupt 2	Edge	219 (0x0DB)	✓	✓	✓	✓	✓
			GPT04_4_INT3	GPT04_4 selected interrupt 3	Edge	220 (0x0DC)	✓	✓	✓	✓	✓
			GPT04_4_INT4	GPT04_4 combined interrupt	Edge	221 (0x0DD)	✓	✓	✓	✓	✓

Table 12.17 Event table (16 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	5	0	GPT05_0_ADTRGA	GPT05_0 GTADTRA compare match (A/D Converter Start Request A)	Edge	917 (0x395)	—	—	—	—	✓
			GPT05_0_ADTRGB	GPT05_0 GTADTRB compare match (A/D Converter Start Request B)	Edge	918 (0x396)	—	—	—	—	✓
			GPT05_0_INT0	GPT05_0 selected interrupt 0	Edge	222 (0x0DE)	✓	✓	✓	✓	✓
			GPT05_0_INT1	GPT05_0 selected interrupt 1	Edge	223 (0x0DF)	✓	✓	✓	✓	✓
			GPT05_0_INT2	GPT05_0 selected interrupt 2	Edge	224 (0x0E0)	✓	✓	✓	✓	✓
			GPT05_0_INT3	GPT05_0 selected interrupt 3	Edge	225 (0x0E1)	✓	✓	✓	✓	✓
			GPT05_0_INT4	GPT05_0 combined interrupt	Edge	226 (0x0E2)	✓	✓	✓	✓	✓
		1	GPT05_1_ADTRGA	GPT05_1 GTADTRA compare match (A/D Converter Start Request A)	Edge	919 (0x397)	—	—	—	—	✓
			GPT05_1_ADTRGB	GPT05_1 GTADTRB compare match (A/D Converter Start Request B)	Edge	920 (0x398)	—	—	—	—	✓
			GPT05_1_INT0	GPT05_1 selected interrupt 0	Edge	227 (0x0E3)	✓	✓	✓	✓	✓
			GPT05_1_INT1	GPT05_1 selected interrupt 1	Edge	228 (0x0E4)	✓	✓	✓	✓	✓
			GPT05_1_INT2	GPT05_1 selected interrupt 2	Edge	229 (0x0E5)	✓	✓	✓	✓	✓
			GPT05_1_INT3	GPT05_1 selected interrupt 3	Edge	230 (0x0E6)	✓	✓	✓	✓	✓
			GPT05_1_INT4	GPT05_1 combined interrupt	Edge	231 (0x0E7)	✓	✓	✓	✓	✓
		2	GPT05_2_ADTRGA	GPT05_2 GTADTRA compare match (A/D Converter Start Request A)	Edge	921 (0x399)	—	—	—	—	✓
			GPT05_2_ADTRGB	GPT05_2 GTADTRB compare match (A/D Converter Start Request B)	Edge	922 (0x39A)	—	—	—	—	✓
			GPT05_2_INT0	GPT05_2 selected interrupt 0	Edge	232 (0x0E8)	✓	✓	✓	✓	✓
			GPT05_2_INT1	GPT05_2 selected interrupt 1	Edge	233 (0x0E9)	✓	✓	✓	✓	✓
			GPT05_2_INT2	GPT05_2 selected interrupt 2	Edge	234 (0x0EA)	✓	✓	✓	✓	✓
			GPT05_2_INT3	GPT05_2 selected interrupt 3	Edge	235 (0x0EB)	✓	✓	✓	✓	✓
			GPT05_2_INT4	GPT05_2 combined interrupt	Edge	236 (0x0EC)	✓	✓	✓	✓	✓

Table 12.17 Event table (17 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	5	3	GPT05_3_ADTRGA	GPT05_3 GTADTRA compare match (A/D Converter Start Request A)	Edge	923 (0x39B)	—	—	—	—	✓
			GPT05_3_ADTRGB	GPT05_3 GTADTRB compare match (A/D Converter Start Request B)	Edge	924 (0x39C)	—	—	—	—	✓
			GPT05_3_INT0	GPT05_3 selected interrupt 0	Edge	237 (0x0ED)	✓	✓	✓	✓	✓
			GPT05_3_INT1	GPT05_3 selected interrupt 1	Edge	238 (0x0EE)	✓	✓	✓	✓	✓
			GPT05_3_INT2	GPT05_3 selected interrupt 2	Edge	239 (0x0EF)	✓	✓	✓	✓	✓
			GPT05_3_INT3	GPT05_3 selected interrupt 3	Edge	240 (0x0F0)	✓	✓	✓	✓	✓
			GPT05_3_INT4	GPT05_3 combined interrupt	Edge	241 (0x0F1)	✓	✓	✓	✓	✓
		4	GPT05_4_ADTRGA	GPT05_4 GTADTRA compare match (A/D Converter Start Request A)	Edge	925 (0x39D)	—	—	—	—	✓
			GPT05_4_ADTRGB	GPT05_4 GTADTRB compare match (A/D Converter Start Request B)	Edge	926 (0x39E)	—	—	—	—	✓
			GPT05_4_INT0	GPT05_4 selected interrupt 0	Edge	242 (0x0F2)	✓	✓	✓	✓	✓
			GPT05_4_INT1	GPT05_4 selected interrupt 1	Edge	243 (0x0F3)	✓	✓	✓	✓	✓
			GPT05_4_INT2	GPT05_4 selected interrupt 2	Edge	244 (0x0F4)	✓	✓	✓	✓	✓
			GPT05_4_INT3	GPT05_4 selected interrupt 3	Edge	245 (0x0F5)	✓	✓	✓	✓	✓
			GPT05_4_INT4	GPT05_4 combined interrupt	Edge	246 (0x0F6)	✓	✓	✓	✓	✓
	6	0	GPT06_0_ADTRGA	GPT06_0 GTADTRA compare match (A/D Converter Start Request A)	Edge	927 (0x39F)	—	—	—	—	✓
			GPT06_0_ADTRGB	GPT06_0 GTADTRB compare match (A/D Converter Start Request B)	Edge	928 (0x3A0)	—	—	—	—	✓
			GPT06_0_INT0	GPT06_0 selected interrupt 0	Edge	247 (0x0F7)	✓	✓	✓	✓	✓
			GPT06_0_INT1	GPT06_0 selected interrupt 1	Edge	248 (0x0F8)	✓	✓	✓	✓	✓
			GPT06_0_INT2	GPT06_0 selected interrupt 2	Edge	249 (0x0F9)	✓	✓	✓	✓	✓
			GPT06_0_INT3	GPT06_0 selected interrupt 3	Edge	250 (0x0FA)	✓	✓	✓	✓	✓
			GPT06_0_INT4	GPT06_0 combined interrupt	Edge	251 (0x0FB)	✓	✓	✓	✓	✓

Table 12.17 Event table (18 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	6	1	GPT06_1_ADTRGA	GPT06_1 GTADTRA compare match (A/D Converter Start Request A)	Edge	929 (0x3A1)	—	—	—	—	✓
			GPT06_1_ADTRGB	GPT06_1 GTADTRB compare match (A/D Converter Start Request B)	Edge	930 (0x3A2)	—	—	—	—	✓
			GPT06_1_INT0	GPT06_1 selected interrupt 0	Edge	252 (0x0FC)	✓	✓	✓	✓	✓
			GPT06_1_INT1	GPT06_1 selected interrupt 1	Edge	253 (0x0FD)	✓	✓	✓	✓	✓
			GPT06_1_INT2	GPT06_1 selected interrupt 2	Edge	254 (0x0FE)	✓	✓	✓	✓	✓
			GPT06_1_INT3	GPT06_1 selected interrupt 3	Edge	255 (0x0FF)	✓	✓	✓	✓	✓
			GPT06_1_INT4	GPT06_1 combined interrupt	Edge	256 (0x100)	✓	✓	✓	✓	✓
		2	GPT06_2_ADTRGA	GPT06_2 GTADTRA compare match (A/D Converter Start Request A)	Edge	931 (0x3A3)	—	—	—	—	✓
			GPT06_2_ADTRGB	GPT06_2 GTADTRB compare match (A/D Converter Start Request B)	Edge	932 (0x3A4)	—	—	—	—	✓
			GPT06_2_INT0	GPT06_2 selected interrupt 0	Edge	257 (0x101)	✓	✓	✓	✓	✓
			GPT06_2_INT1	GPT06_2 selected interrupt 1	Edge	258 (0x102)	✓	✓	✓	✓	✓
			GPT06_2_INT2	GPT06_2 selected interrupt 2	Edge	259 (0x103)	✓	✓	✓	✓	✓
			GPT06_2_INT3	GPT06_2 selected interrupt 3	Edge	260 (0x104)	✓	✓	✓	✓	✓
			GPT06_2_INT4	GPT06_2 combined interrupt	Edge	261 (0x105)	✓	✓	✓	✓	✓
		3	GPT06_3_ADTRGA	GPT06_3 GTADTRA compare match (A/D Converter Start Request A)	Edge	933 (0x3A5)	—	—	—	—	✓
			GPT06_3_ADTRGB	GPT06_3 GTADTRB compare match (A/D Converter Start Request B)	Edge	934 (0x3A6)	—	—	—	—	✓
			GPT06_3_INT0	GPT06_3 selected interrupt 0	Edge	262 (0x106)	✓	✓	✓	✓	✓
			GPT06_3_INT1	GPT06_3 selected interrupt 1	Edge	263 (0x107)	✓	✓	✓	✓	✓
			GPT06_3_INT2	GPT06_3 selected interrupt 2	Edge	264 (0x108)	✓	✓	✓	✓	✓
			GPT06_3_INT3	GPT06_3 selected interrupt 3	Edge	265 (0x109)	✓	✓	✓	✓	✓
			GPT06_3_INT4	GPT06_3 combined interrupt	Edge	266 (0x10A)	✓	✓	✓	✓	✓

Table 12.17 Event table (19 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	6	4	GPT06_4_ADTRGA	GPT06_4 GTADTRA compare match (A/D Converter Start Request A)	Edge	935 (0x3A7)	—	—	—	—	✓
			GPT06_4_ADTRGB	GPT06_4 GTADTRB compare match (A/D Converter Start Request B)	Edge	936 (0x3A8)	—	—	—	—	✓
			GPT06_4_INT0	GPT06_4 selected interrupt 0	Edge	267 (0x10B)	✓	✓	✓	✓	✓
			GPT06_4_INT1	GPT06_4 selected interrupt 1	Edge	268 (0x10C)	✓	✓	✓	✓	✓
			GPT06_4_INT2	GPT06_4 selected interrupt 2	Edge	269 (0x10D)	✓	✓	✓	✓	✓
			GPT06_4_INT3	GPT06_4 selected interrupt 3	Edge	270 (0x10E)	✓	✓	✓	✓	✓
			GPT06_4_INT4	GPT06_4 combined interrupt	Edge	271 (0x10F)	✓	✓	✓	✓	✓
	7	0	GPT07_0_ADTRGA	GPT07_0 GTADTRA compare match (A/D Converter Start Request A)	Edge	937 (0x3A9)	—	—	—	—	✓
			GPT07_0_ADTRGB	GPT07_0 GTADTRB compare match (A/D Converter Start Request B)	Edge	938 (0x3AA)	—	—	—	—	✓
			GPT07_0_INT0	GPT07_0 selected interrupt 0	Edge	272 (0x110)	✓	✓	✓	✓	✓
			GPT07_0_INT1	GPT07_0 selected interrupt 1	Edge	273 (0x111)	✓	✓	✓	✓	✓
			GPT07_0_INT2	GPT07_0 selected interrupt 2	Edge	274 (0x112)	✓	✓	✓	✓	✓
			GPT07_0_INT3	GPT07_0 selected interrupt 3	Edge	275 (0x113)	✓	✓	✓	✓	✓
			GPT07_0_INT4	GPT07_0 combined interrupt	Edge	276 (0x114)	✓	✓	✓	✓	✓
		1	GPT07_1_ADTRGA	GPT07_1 GTADTRA compare match (A/D Converter Start Request A)	Edge	939 (0x3AB)	—	—	—	—	✓
			GPT07_1_ADTRGB	GPT07_1 GTADTRB compare match (A/D Converter Start Request B)	Edge	940 (0x3AC)	—	—	—	—	✓
			GPT07_1_INT0	GPT07_1 selected interrupt 0	Edge	277 (0x115)	✓	✓	✓	✓	✓
			GPT07_1_INT1	GPT07_1 selected interrupt 1	Edge	278 (0x116)	✓	✓	✓	✓	✓
			GPT07_1_INT2	GPT07_1 selected interrupt 2	Edge	279 (0x117)	✓	✓	✓	✓	✓
			GPT07_1_INT3	GPT07_1 selected interrupt 3	Edge	280 (0x118)	✓	✓	✓	✓	✓
			GPT07_1_INT4	GPT07_1 combined interrupt	Edge	281 (0x119)	✓	✓	✓	✓	✓

Table 12.17 Event table (20 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	7	2	GPT07_2_ADTRGA	GPT07_2 GTADTRA compare match (A/D Converter Start Request A)	Edge	941 (0x3AD)	—	—	—	—	✓
			GPT07_2_ADTRGB	GPT07_2 GTADTRB compare match (A/D Converter Start Request B)	Edge	942 (0x3AE)	—	—	—	—	✓
			GPT07_2_INT0	GPT07_2 selected interrupt 0	Edge	282 (0x11A)	✓	✓	✓	✓	✓
			GPT07_2_INT1	GPT07_2 selected interrupt 1	Edge	283 (0x11B)	✓	✓	✓	✓	✓
			GPT07_2_INT2	GPT07_2 selected interrupt 2	Edge	284 (0x11C)	✓	✓	✓	✓	✓
			GPT07_2_INT3	GPT07_2 selected interrupt 3	Edge	285 (0x11D)	✓	✓	✓	✓	✓
			GPT07_2_INT4	GPT07_2 combined interrupt	Edge	286 (0x11E)	✓	✓	✓	✓	✓
		3	GPT07_3_ADTRGA	GPT07_3 GTADTRA compare match (A/D Converter Start Request A)	Edge	943 (0x3AF)	—	—	—	—	✓
			GPT07_3_ADTRGB	GPT07_3 GTADTRB compare match (A/D Converter Start Request B)	Edge	944 (0x3B0)	—	—	—	—	✓
			GPT07_3_INT0	GPT07_3 selected interrupt 0	Edge	287 (0x11F)	✓	✓	✓	✓	✓
			GPT07_3_INT1	GPT07_3 selected interrupt 1	Edge	288 (0x120)	✓	✓	✓	✓	✓
			GPT07_3_INT2	GPT07_3 selected interrupt 2	Edge	289 (0x121)	✓	✓	✓	✓	✓
			GPT07_3_INT3	GPT07_3 selected interrupt 3	Edge	290 (0x122)	✓	✓	✓	✓	✓
			GPT07_3_INT4	GPT07_3 combined interrupt	Edge	291 (0x123)	✓	✓	✓	✓	✓
		4	GPT07_4_ADTRGA	GPT07_4 GTADTRA compare match (A/D Converter Start Request A)	Edge	945 (0x3B1)	—	—	—	—	✓
			GPT07_4_ADTRGB	GPT07_4 GTADTRB compare match (A/D Converter Start Request B)	Edge	946 (0x3B2)	—	—	—	—	✓
			GPT07_4_INT0	GPT07_4 selected interrupt 0	Edge	292 (0x124)	✓	✓	✓	✓	✓
			GPT07_4_INT1	GPT07_4 selected interrupt 1	Edge	293 (0x125)	✓	✓	✓	✓	✓
			GPT07_4_INT2	GPT07_4 selected interrupt 2	Edge	294 (0x126)	✓	✓	✓	✓	✓
			GPT07_4_INT3	GPT07_4 selected interrupt 3	Edge	295 (0x127)	✓	✓	✓	✓	✓
			GPT07_4_INT4	GPT07_4 combined interrupt	Edge	296 (0x128)	✓	✓	✓	✓	✓

Table 12.17 Event table (21 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	8	0	GPT08_0_ADTRGA	GPT08_0 GTADTRA compare match (A/D Converter Start Request A)	Edge	947 (0x3B3)	—	—	—	—	✓
			GPT08_0_ADTRGB	GPT08_0 GTADTRB compare match (A/D Converter Start Request B)	Edge	948 (0x3B4)	—	—	—	—	✓
			GPT08_0_INT0	GPT08_0 selected interrupt 0	Edge	297 (0x129)	✓	✓	✓	✓	✓
			GPT08_0_INT1	GPT08_0 selected interrupt 1	Edge	298 (0x12A)	✓	✓	✓	✓	✓
			GPT08_0_INT2	GPT08_0 selected interrupt 2	Edge	299 (0x12B)	✓	✓	✓	✓	✓
			GPT08_0_INT3	GPT08_0 selected interrupt 3	Edge	300 (0x12C)	✓	✓	✓	✓	✓
			GPT08_0_INT4	GPT08_0 combined interrupt	Edge	301 (0x12D)	✓	✓	✓	✓	✓
		1	GPT08_1_ADTRGA	GPT08_1 GTADTRA compare match (A/D Converter Start Request A)	Edge	949 (0x3B5)	—	—	—	—	✓
			GPT08_1_ADTRGB	GPT08_1 GTADTRB compare match (A/D Converter Start Request B)	Edge	950 (0x3B6)	—	—	—	—	✓
			GPT08_1_INT0	GPT08_1 selected interrupt 0	Edge	302 (0x12E)	✓	✓	✓	✓	✓
			GPT08_1_INT1	GPT08_1 selected interrupt 1	Edge	303 (0x12F)	✓	✓	✓	✓	✓
			GPT08_1_INT2	GPT08_1 selected interrupt 2	Edge	304 (0x130)	✓	✓	✓	✓	✓
			GPT08_1_INT3	GPT08_1 selected interrupt 3	Edge	305 (0x131)	✓	✓	✓	✓	✓
			GPT08_1_INT4	GPT08_1 combined interrupt	Edge	306 (0x132)	✓	✓	✓	✓	✓
		2	GPT08_2_ADTRGA	GPT08_2 GTADTRA compare match (A/D Converter Start Request A)	Edge	951 (0x3B7)	—	—	—	—	✓
			GPT08_2_ADTRGB	GPT08_2 GTADTRB compare match (A/D Converter Start Request B)	Edge	952 (0x3B8)	—	—	—	—	✓
			GPT08_2_INT0	GPT08_2 selected interrupt 0	Edge	307 (0x133)	✓	✓	✓	✓	✓
			GPT08_2_INT1	GPT08_2 selected interrupt 1	Edge	308 (0x134)	✓	✓	✓	✓	✓
			GPT08_2_INT2	GPT08_2 selected interrupt 2	Edge	309 (0x135)	✓	✓	✓	✓	✓
			GPT08_2_INT3	GPT08_2 selected interrupt 3	Edge	310 (0x136)	✓	✓	✓	✓	✓
			GPT08_2_INT4	GPT08_2 combined interrupt	Edge	311 (0x137)	✓	✓	✓	✓	✓

Table 12.17 Event table (22 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	8	3	GPT08_3_ADTRGA	GPT08_3 GTADTRA compare match (A/D Converter Start Request A)	Edge	953 (0x3B9)	—	—	—	—	✓
			GPT08_3_ADTRGB	GPT08_3 GTADTRB compare match (A/D Converter Start Request B)	Edge	954 (0x3BA)	—	—	—	—	✓
			GPT08_3_INT0	GPT08_3 selected interrupt 0	Edge	312 (0x138)	✓	✓	✓	✓	✓
			GPT08_3_INT1	GPT08_3 selected interrupt 1	Edge	313 (0x139)	✓	✓	✓	✓	✓
			GPT08_3_INT2	GPT08_3 selected interrupt 2	Edge	314 (0x13A)	✓	✓	✓	✓	✓
			GPT08_3_INT3	GPT08_3 selected interrupt 3	Edge	315 (0x13B)	✓	✓	✓	✓	✓
			GPT08_3_INT4	GPT08_3 combined interrupt	Edge	316 (0x13C)	✓	✓	✓	✓	✓
		4	GPT08_4_ADTRGA	GPT08_4 GTADTRA compare match (A/D Converter Start Request A)	Edge	955 (0x3BB)	—	—	—	—	✓
			GPT08_4_ADTRGB	GPT08_4 GTADTRB compare match (A/D Converter Start Request B)	Edge	956 (0x3BC)	—	—	—	—	✓
			GPT08_4_INT0	GPT08_4 selected interrupt 0	Edge	317 (0x13D)	✓	✓	✓	✓	✓
			GPT08_4_INT1	GPT08_4 selected interrupt 1	Edge	318 (0x13E)	✓	✓	✓	✓	✓
			GPT08_4_INT2	GPT08_4 selected interrupt 2	Edge	319 (0x13F)	✓	✓	✓	✓	✓
			GPT08_4_INT3	GPT08_4 selected interrupt 3	Edge	320 (0x140)	✓	✓	✓	✓	✓
			GPT08_4_INT4	GPT08_4 combined interrupt	Edge	321 (0x141)	✓	✓	✓	✓	✓
	9	0	GPT09_0_ADTRGA	GPT09_0 GTADTRA compare match (A/D Converter Start Request A)	Edge	957 (0x3BD)	—	—	—	—	✓
			GPT09_0_ADTRGB	GPT09_0 GTADTRB compare match (A/D Converter Start Request B)	Edge	958 (0x3BE)	—	—	—	—	✓
			GPT09_0_INT0	GPT09_0 selected interrupt 0	Edge	322 (0x142)	✓	✓	✓	✓	✓
			GPT09_0_INT1	GPT09_0 selected interrupt 1	Edge	323 (0x143)	✓	✓	✓	✓	✓
			GPT09_0_INT2	GPT09_0 selected interrupt 2	Edge	324 (0x144)	✓	✓	✓	✓	✓
			GPT09_0_INT3	GPT09_0 selected interrupt 3	Edge	325 (0x145)	✓	✓	✓	✓	✓
			GPT09_0_INT4	GPT09_0 combined interrupt	Edge	326 (0x146)	✓	✓	✓	✓	✓

Table 12.17 Event table (23 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	9	1	GPT09_1_ADTRGA	GPT09_1 GTADTRA compare match (A/D Converter Start Request A)	Edge	959 (0x3BF)	—	—	—	—	✓
			GPT09_1_ADTRGB	GPT09_1 GTADTRB compare match (A/D Converter Start Request B)	Edge	960 (0x3C0)	—	—	—	—	✓
			GPT09_1_INT0	GPT09_1 selected interrupt 0	Edge	327 (0x147)	✓	✓	✓	✓	✓
			GPT09_1_INT1	GPT09_1 selected interrupt 1	Edge	328 (0x148)	✓	✓	✓	✓	✓
			GPT09_1_INT2	GPT09_1 selected interrupt 2	Edge	329 (0x149)	✓	✓	✓	✓	✓
			GPT09_1_INT3	GPT09_1 selected interrupt 3	Edge	330 (0x14A)	✓	✓	✓	✓	✓
			GPT09_1_INT4	GPT09_1 combined interrupt	Edge	331 (0x14B)	✓	✓	✓	✓	✓
		2	GPT09_2_ADTRGA	GPT09_2 GTADTRA compare match (A/D Converter Start Request A)	Edge	961 (0x3C1)	—	—	—	—	✓
			GPT09_2_ADTRGB	GPT09_2 GTADTRB compare match (A/D Converter Start Request B)	Edge	962 (0x3C2)	—	—	—	—	✓
			GPT09_2_INT0	GPT09_2 selected interrupt 0	Edge	332 (0x14C)	✓	✓	✓	✓	✓
			GPT09_2_INT1	GPT09_2 selected interrupt 1	Edge	333 (0x14D)	✓	✓	✓	✓	✓
			GPT09_2_INT2	GPT09_2 selected interrupt 2	Edge	334 (0x14E)	✓	✓	✓	✓	✓
			GPT09_2_INT3	GPT09_2 selected interrupt 3	Edge	335 (0x14F)	✓	✓	✓	✓	✓
			GPT09_2_INT4	GPT09_2 combined interrupt	Edge	336 (0x150)	✓	✓	✓	✓	✓
		3	GPT09_3_ADTRGA	GPT09_3 GTADTRA compare match (A/D Converter Start Request A)	Edge	963 (0x3C3)	—	—	—	—	✓
			GPT09_3_ADTRGB	GPT09_3 GTADTRB compare match (A/D Converter Start Request B)	Edge	964 (0x3C4)	—	—	—	—	✓
			GPT09_3_INT0	GPT09_3 selected interrupt 0	Edge	337 (0x151)	✓	✓	✓	✓	✓
			GPT09_3_INT1	GPT09_3 selected interrupt 1	Edge	338 (0x152)	✓	✓	✓	✓	✓
			GPT09_3_INT2	GPT09_3 selected interrupt 2	Edge	339 (0x153)	✓	✓	✓	✓	✓
			GPT09_3_INT3	GPT09_3 selected interrupt 3	Edge	340 (0x154)	✓	✓	✓	✓	✓
			GPT09_3_INT4	GPT09_3 combined interrupt	Edge	341 (0x155)	✓	✓	✓	✓	✓

Table 12.17 Event table (24 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	9	4	GPT09_4_ADTRGA	GPT09_4 GTADTRA compare match (A/D Converter Start Request A)	Edge	965 (0x3C5)	—	—	—	—	✓
			GPT09_4_ADTRGB	GPT09_4 GTADTRB compare match (A/D Converter Start Request B)	Edge	966 (0x3C6)	—	—	—	—	✓
			GPT09_4_INT0	GPT09_4 selected interrupt 0	Edge	342 (0x156)	✓	✓	✓	✓	✓
			GPT09_4_INT1	GPT09_4 selected interrupt 1	Edge	343 (0x157)	✓	✓	✓	✓	✓
			GPT09_4_INT2	GPT09_4 selected interrupt 2	Edge	344 (0x158)	✓	✓	✓	✓	✓
			GPT09_4_INT3	GPT09_4 selected interrupt 3	Edge	345 (0x159)	✓	✓	✓	✓	✓
			GPT09_4_INT4	GPT09_4 combined interrupt	Edge	346 (0x15A)	✓	✓	✓	✓	✓
		5	GPT09_5_ADTRGA	GPT09_5 GTADTRA compare match (A/D Converter Start Request A)	Edge	967 (0x3C7)	—	—	—	—	✓
			GPT09_5_ADTRGB	GPT09_5 GTADTRB compare match (A/D Converter Start Request B)	Edge	968 (0x3C8)	—	—	—	—	✓
			GPT09_5_INT0	GPT09_5 selected interrupt 0	Edge	347 (0x15B)	✓	✓	✓	✓	✓
			GPT09_5_INT1	GPT09_5 selected interrupt 1	Edge	348 (0x15C)	✓	✓	✓	✓	✓
			GPT09_5_INT2	GPT09_5 selected interrupt 2	Edge	349 (0x15D)	✓	✓	✓	✓	✓
			GPT09_5_INT3	GPT09_5 selected interrupt 3	Edge	350 (0x15E)	✓	✓	✓	✓	✓
			GPT09_5_INT4	GPT09_5 combined interrupt	Edge	351 (0x15F)	✓	✓	✓	✓	✓
		6	GPT09_6_ADTRGA	GPT09_6 GTADTRA compare match (A/D Converter Start Request A)	Edge	969 (0x3C9)	—	—	—	—	✓
			GPT09_6_ADTRGB	GPT09_6 GTADTRB compare match (A/D Converter Start Request B)	Edge	970 (0x3CA)	—	—	—	—	✓
			GPT09_6_INT0	GPT09_6 selected interrupt 0	Edge	352 (0x160)	✓	✓	✓	✓	✓
			GPT09_6_INT1	GPT09_6 selected interrupt 1	Edge	353 (0x161)	✓	✓	✓	✓	✓
			GPT09_6_INT2	GPT09_6 selected interrupt 2	Edge	354 (0x162)	✓	✓	✓	✓	✓
			GPT09_6_INT3	GPT09_6 selected interrupt 3	Edge	355 (0x163)	✓	✓	✓	✓	✓
			GPT09_6_INT4	GPT09_6 combined interrupt	Edge	356 (0x164)	✓	✓	✓	✓	✓

Table 12.17 Event table (25 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GPT	10	0	GPT10_0_INT0	GPT10_0 selected interrupt 0	Edge	357 (0x165)	✓	✓	✓	—	—
			GPT10_0_INT1	GPT10_0 selected interrupt 1	Edge	358 (0x166)	✓	✓	✓	—	—
			GPT10_0_INT2	GPT10_0 selected interrupt 2	Edge	359 (0x167)	✓	✓	✓	—	—
			GPT10_0_INT3	GPT10_0 selected interrupt 3	Edge	360 (0x168)	✓	✓	✓	—	—
			GPT10_0_INT4	GPT10_0 combined interrupt	Edge	361 (0x169)	✓	✓	✓	—	—
		1	GPT10_1_INT0	GPT10_1 selected interrupt 0	Edge	362 (0x16A)	✓	✓	✓	—	—
			GPT10_1_INT1	GPT10_1 selected interrupt 1	Edge	363 (0x16B)	✓	✓	✓	—	—
			GPT10_1_INT2	GPT10_1 selected interrupt 2	Edge	364 (0x16C)	✓	✓	✓	—	—
			GPT10_1_INT3	GPT10_1 selected interrupt 3	Edge	365 (0x16D)	✓	✓	✓	—	—
			GPT10_1_INT4	GPT10_1 combined interrupt	Edge	366 (0x16E)	✓	✓	✓	—	—
		2	GPT10_2_INT0	GPT10_2 selected interrupt 0	Edge	367 (0x16F)	✓	✓	✓	—	—
			GPT10_2_INT1	GPT10_2 selected interrupt 1	Edge	368 (0x170)	✓	✓	✓	—	—
			GPT10_2_INT2	GPT10_2 selected interrupt 2	Edge	369 (0x171)	✓	✓	✓	—	—
			GPT10_2_INT3	GPT10_2 selected interrupt 3	Edge	370 (0x172)	✓	✓	✓	—	—
			GPT10_2_INT4	GPT10_2 combined interrupt	Edge	371 (0x173)	✓	✓	✓	—	—
		3	GPT10_3_INT0	GPT10_3 selected interrupt 0	Edge	372 (0x174)	✓	✓	✓	—	—
			GPT10_3_INT1	GPT10_3 selected interrupt 1	Edge	373 (0x175)	✓	✓	✓	—	—
			GPT10_3_INT2	GPT10_3 selected interrupt 2	Edge	374 (0x176)	✓	✓	✓	—	—
			GPT10_3_INT3	GPT10_3 selected interrupt 3	Edge	375 (0x177)	✓	✓	✓	—	—
			GPT10_3_INT4	GPT10_3 combined interrupt	Edge	376 (0x178)	✓	✓	✓	—	—
POEG	0	0	POEG0_GROUP0	POEG group A interrupt for channels in LLPP	Level	467 (0x1D3)	✓	#CR520_SEL	#CR521_SEL	—	—
		1	POEG0_GROUP1	POEG group B interrupt for channels in LLPP	Level	468 (0x1D4)	✓	#CR520_SEL	#CR521_SEL	—	—
		2	POEG0_GROUP2	POEG group C interrupt for channels in LLPP	Level	469 (0x1D5)	✓	#CR520_SEL	#CR521_SEL	—	—
		3	POEG0_GROUP3	POEG group D interrupt for channels in LLPP	Level	470 (0x1D6)	✓	#CR520_SEL	#CR521_SEL	—	—

Table 12.17 Event table (26 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
POEG	1	0	POEG1_GROUP0	POEG group A interrupt for channels in NONSAFETY	Level	471 (0x1D7)	✓	#CR520_SEL	#CR521_SEL	—	—
		1	POEG1_GROUP1	POEG group B interrupt for channels in NONSAFETY	Level	472 (0x1D8)	✓	#CR520_SEL	#CR521_SEL	—	—
		2	POEG1_GROUP2	POEG group C interrupt for channels in NONSAFETY	Level	473 (0x1D9)	✓	#CR520_SEL	#CR521_SEL	—	—
		3	POEG1_GROUP3	POEG group D interrupt for channels in NONSAFETY	Level	474 (0x1DA)	✓	#CR520_SEL	#CR521_SEL	—	—
	2	0	POEG2_GROUP0	POEG group A interrupt for channels in SAFETY	Level	475 (0x1DB)	✓	#CR520_SEL	#CR521_SEL	—	—
		1	POEG2_GROUP1	POEG group B interrupt for channels in SAFETY	Level	476 (0x1DC)	✓	#CR520_SEL	#CR521_SEL	—	—
		2	POEG2_GROUP2	POEG group C interrupt for channels in SAFETY	Level	477 (0x1DD)	✓	#CR520_SEL	#CR521_SEL	—	—
		3	POEG2_GROUP3	POEG group D interrupt for channels in SAFETY	Level	478 (0x1DE)	✓	#CR520_SEL	#CR521_SEL	—	—
CMT	0	0	CMT0_CMI	CMT0 Compare match	Edge	479 (0x1DF)	✓	#CR520_SEL	#CR521_SEL	✓	—
		1	CMT1_CMI	CMT1 Compare match	Edge	480 (0x1E0)	✓	#CR520_SEL	#CR521_SEL	✓	—
	1	0	CMT2_CMI	CMT2 Compare match	Edge	481 (0x1E1)	✓	#CR520_SEL	#CR521_SEL	✓	—
		1	CMT3_CMI	CMT3 Compare match	Edge	482 (0x1E2)	✓	#CR520_SEL	#CR521_SEL	✓	—
	2	0	CMT4_CMI	CMT4 Compare match	Edge	483 (0x1E3)	✓	#CR520_SEL	#CR521_SEL	✓	—
		1	CMT5_CMI	CMT5 Compare match	Edge	484 (0x1E4)	✓	#CR520_SEL	#CR521_SEL	✓	—
CMTW	0	—	CMTW0_CMWI	CMTW0 Compare match	Edge	485 (0x1E5)	✓	#CR520_SEL	#CR521_SEL	✓	—
			CMTW0_IC0I	CMTW0 Input capture of register 0	Edge	486 (0x1E6)	✓	#CR520_SEL	#CR521_SEL	✓	—
			CMTW0_IC1I	CMTW0 Input capture of register 1	Edge	487 (0x1E7)	✓	#CR520_SEL	#CR521_SEL	✓	—
			CMTW0_OC0I	CMTW0 Output compare of register 0	Edge	488 (0x1E8)	✓	#CR520_SEL	#CR521_SEL	✓	—
			CMTW0_OC1I	CMTW0 Output compare of register 1	Edge	489 (0x1E9)	✓	#CR520_SEL	#CR521_SEL	✓	—
	1	—	CMTW1_CMWI	CMTW1 Compare match	Edge	490 (0x1EA)	✓	#CR520_SEL	#CR521_SEL	✓	—
			CMTW1_IC0I	CMTW1 Input capture of register 0	Edge	491 (0x1EB)	✓	#CR520_SEL	#CR521_SEL	✓	—
			CMTW1_IC1I	CMTW1 Input capture of register 1	Edge	492 (0x1EC)	✓	#CR520_SEL	#CR521_SEL	✓	—
			CMTW1_OC0I	CMTW1 Output compare of register 0	Edge	493 (0x1ED)	✓	#CR520_SEL	#CR521_SEL	✓	—
			CMTW1_OC1I	CMTW1 Output compare of register 1	Edge	494 (0x1EE)	✓	#CR520_SEL	#CR521_SEL	✓	—

Table 12.17 Event table (27 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
RTC	0	—	RTC_ALM	Alarm interrupt	Edge	495 (0x1EF)	✓	#CR520_SEL	#CR521_SEL	—	—
			RTC_1S	1 second interrupt	Edge	496 (0x1F0)	✓	#CR520_SEL	#CR521_SEL	—	—
			RTC_PRD	Fixed interval interrupt	Edge	497 (0x1F1)	✓	#CR520_SEL	#CR521_SEL	—	—
GMAC	0	—	GMAC0_sbd_intr_o	GMAC0 Subsystem interrupt	Level	498 (0x1F2)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_lpi_intr_o	GMAC0 LPI interrupt	Level	499 (0x1F3)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_pmt_intr_o	GMAC0 PMT Interrupt	Level	500 (0x1F4)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_tx_intr_o0	GMAC0 Per channel Transmit Interrupt 0	Level/ Edge	501 (0x1F5)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_tx_intr_o1	GMAC0 Per channel Transmit Interrupt 1	Level/ Edge	502 (0x1F6)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_tx_intr_o2	GMAC0 Per channel Transmit Interrupt 2	Level/ Edge	503 (0x1F7)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_tx_intr_o3	GMAC0 Per channel Transmit Interrupt 3	Level/ Edge	504 (0x1F8)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_tx_intr_o4	GMAC0 Per channel Transmit Interrupt 4	Level/ Edge	505 (0x1F9)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_tx_intr_o5	GMAC0 Per channel Transmit Interrupt 5	Level/ Edge	506 (0x1FA)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_tx_intr_o6	GMAC0 Per channel Transmit Interrupt 6	Level/ Edge	507 (0x1FB)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_tx_intr_o7	GMAC0 Per channel Transmit Interrupt 7	Level/ Edge	508 (0x1FC)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_rx_intr_o0	GMAC0 Per channel Receive Interrupt 0	Level/ Edge	509 (0x1FD)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_rx_intr_o1	GMAC0 Per channel Receive Interrupt 1	Level/ Edge	510 (0x1FE)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_rx_intr_o2	GMAC0 Per channel Receive Interrupt 2	Level/ Edge	511 (0x1FF)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_rx_intr_o3	GMAC0 Per channel Receive Interrupt 3	Level/ Edge	512 (0x200)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_rx_intr_o4	GMAC0 Per channel Receive Interrupt 4	Level/ Edge	513 (0x201)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_rx_intr_o5	GMAC0 Per channel Receive Interrupt 5	Level/ Edge	514 (0x202)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_rx_intr_o6	GMAC0 Per channel Receive Interrupt 6	Level/ Edge	515 (0x203)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC0_sbd_perch_rx_intr_o7	GMAC0 Per channel Receive Interrupt 7	Level/ Edge	516 (0x204)	✓	#CR520_SEL	#CR521_SEL	—	—
	1	—	GMAC1_sbd_intr_o	GMAC1 Subsystem interrupt	Level	517 (0x205)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_lpi_intr_o	GMAC1 LPI interrupt	Level	518 (0x206)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_pmt_intr_o	GMAC1 PMT Interrupt	Level	519 (0x207)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_tx_intr_o0	GMAC1 Per channel Transmit Interrupt 0	Level/ Edge	520 (0x208)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_tx_intr_o1	GMAC1 Per channel Transmit Interrupt 1	Level/ Edge	521 (0x209)	✓	#CR520_SEL	#CR521_SEL	—	—

Table 12.17 Event table (28 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GMAC	1	—	GMAC1_sbd_perch_tx_intr_o2	GMAC1 Per channel Transmit Interrupt 2	Level/ Edge	522 (0x20A)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_tx_intr_o3	GMAC1 Per channel Transmit Interrupt 3	Level/ Edge	523 (0x20B)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_tx_intr_o4	GMAC1 Per channel Transmit Interrupt 4	Level/ Edge	524 (0x20C)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_tx_intr_o5	GMAC1 Per channel Transmit Interrupt 5	Level/ Edge	525 (0x20D)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_tx_intr_o6	GMAC1 Per channel Transmit Interrupt 6	Level/ Edge	526 (0x20E)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_tx_intr_o7	GMAC1 Per channel Transmit Interrupt 7	Level/ Edge	527 (0x20F)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_rx_intr_o0	GMAC1 Per channel Receive Interrupt 0	Level/ Edge	528 (0x210)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_rx_intr_o1	GMAC1 Per channel Receive Interrupt 1	Level/ Edge	529 (0x211)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_rx_intr_o2	GMAC1 Per channel Receive Interrupt 2	Level/ Edge	530 (0x212)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_rx_intr_o3	GMAC1 Per channel Receive Interrupt 3	Level/ Edge	531 (0x213)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_rx_intr_o4	GMAC1 Per channel Receive Interrupt 4	Level/ Edge	532 (0x214)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_rx_intr_o5	GMAC1 Per channel Receive Interrupt 5	Level/ Edge	533 (0x215)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_rx_intr_o6	GMAC1 Per channel Receive Interrupt 6	Level/ Edge	534 (0x216)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC1_sbd_perch_rx_intr_o7	GMAC1 Per channel Receive Interrupt 7	Level/ Edge	535 (0x217)	✓	#CR520_SEL	#CR521_SEL	—	—
	2	—	GMAC2_sbd_intr_o	GMAC2 Subsystem interrupt	Level	536 (0x218)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_lpi_intr_o	GMAC2 LPI interrupt	Level	537 (0x219)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_pmt_intr_o	GMAC2 PMT Interrupt	Level	538 (0x21A)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_tx_intr_o0	GMAC2 Per channel Transmit Interrupt 0	Level/ Edge	539 (0x21B)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_tx_intr_o1	GMAC2 Per channel Transmit Interrupt 1	Level/ Edge	540 (0x21C)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_tx_intr_o2	GMAC2 Per channel Transmit Interrupt 2	Level/ Edge	541 (0x21D)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_tx_intr_o3	GMAC2 Per channel Transmit Interrupt 3	Level/ Edge	542 (0x21E)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_tx_intr_o4	GMAC2 Per channel Transmit Interrupt 4	Level/ Edge	543 (0x21F)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_tx_intr_o5	GMAC2 Per channel Transmit Interrupt 5	Level/ Edge	544 (0x220)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_tx_intr_o6	GMAC2 Per channel Transmit Interrupt 6	Level/ Edge	545 (0x221)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_tx_intr_o7	GMAC2 Per channel Transmit Interrupt 7	Level/ Edge	546 (0x222)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_rx_intr_o0	GMAC2 Per channel Receive Interrupt 0	Level/ Edge	547 (0x223)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_rx_intr_o1	GMAC2 Per channel Receive Interrupt 1	Level/ Edge	548 (0x224)	✓	#CR520_SEL	#CR521_SEL	—	—

Table 12.17 Event table (29 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
GMAC	2	—	GMAC2_sbd_perch_rx_intr_o2	GMAC2 Per channel Receive Interrupt 2	Level/Edge	549 (0x225)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_rx_intr_o3	GMAC2 Per channel Receive Interrupt 3	Level/Edge	550 (0x226)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_rx_intr_o4	GMAC2 Per channel Receive Interrupt 4	Level/Edge	551 (0x227)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_rx_intr_o5	GMAC2 Per channel Receive Interrupt 5	Level/Edge	552 (0x228)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_rx_intr_o6	GMAC2 Per channel Receive Interrupt 6	Level/Edge	553 (0x229)	✓	#CR520_SEL	#CR521_SEL	—	—
			GMAC2_sbd_perch_rx_intr_o7	GMAC2 Per channel Receive Interrupt 7	Level/Edge	554 (0x22A)	✓	#CR520_SEL	#CR521_SEL	—	—
ETHSW	0	—	ETHSW_INTR	Ethernet Switch interrupt	Level	555 (0x22B)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_DLR	Ethernet Switch DLR interrupt	Level	556 (0x22C)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PRP	Ethernet Switch PRP interrupt	Level	557 (0x22D)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_IHUB	Ethernet Switch Integrated Hub interrupt	Level	558 (0x22E)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN0	Ethernet Switch RX Pattern Matcher interrupt 0	Level	559 (0x22F)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN1	Ethernet Switch RX Pattern Matcher interrupt 1	Level	560 (0x230)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN2	Ethernet Switch RX Pattern Matcher interrupt 2	Level	561 (0x231)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN3	Ethernet Switch RX Pattern Matcher interrupt 3	Level	562 (0x232)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN4	Ethernet Switch RX Pattern Matcher interrupt 4	Level	563 (0x233)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN5	Ethernet Switch RX Pattern Matcher interrupt 5	Level	564 (0x234)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN6	Ethernet Switch RX Pattern Matcher interrupt 6	Level	565 (0x235)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN7	Ethernet Switch RX Pattern Matcher interrupt 7	Level	566 (0x236)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN8	Ethernet Switch RX Pattern Matcher interrupt 8	Level	567 (0x237)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN9	Ethernet Switch RX Pattern Matcher interrupt 9	Level	568 (0x238)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN10	Ethernet Switch RX Pattern Matcher interrupt 10	Level	569 (0x239)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ETHSW_PTRN11	Ethernet Switch RX Pattern Matcher interrupt 11	Level	570 (0x23A)	✓	#CR520_SEL	#CR521_SEL	✓	—

Table 12.17 Event table (30 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
ETHSW	0	—	ETHSW_PTPOUT0	Ethernet switch timer pulse output 0	Level/ Edge	571 (0x23B)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ETHSW_PTPOUT1	Ethernet switch timer pulse output 1	Level/ Edge	572 (0x23C)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ETHSW_PTPOUT2	Ethernet switch timer pulse output 2	Level/ Edge	573 (0x23D)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ETHSW_PTPOUT3	Ethernet switch timer pulse output 3	Level/ Edge	574 (0x23E)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ETHSW_TDMAOUT0	Ethernet Switch TDMA timer output 0	Level/ Edge	575 (0x23F)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ETHSW_TDMAOUT1	Ethernet Switch TDMA timer output 1	Level/ Edge	576 (0x240)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ETHSW_TDMAOUT2	Ethernet Switch TDMA timer output 2	Level/ Edge	577 (0x241)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ETHSW_TDMAOUT3	Ethernet Switch TDMA timer output 3	Level/ Edge	578 (0x242)	✓	#CR520_SEL	#CR521_SEL	✓	✓
ESC	0	—	ESC_SYNC0	EtherCAT Sync0 interrupt	Level/ Edge	579 (0x243)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ESC_SYNC1	EtherCAT Sync1 interrupt	Level/ Edge	580 (0x244)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ESC_CAT	EtherCAT interrupt	Level	581 (0x245)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ESC_SOF	EtherCAT SOF interrupt	Edge	582 (0x246)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ESC_EOF	EtherCAT EOF interrupt	Edge	583 (0x247)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ESC_WDT	EtherCAT WDT interrupt	Edge	584 (0x248)	✓	#CR520_SEL	#CR521_SEL	—	—
			ESC_RST	EtherCAT RESET interrupt	Level	585 (0x249)	✓	#CR520_SEL	#CR521_SEL	—	—
USBh	0	—	USB_HI	USB (Host) interrupt	Level	586 (0x24A)	✓	#CR520_SEL	#CR521_SEL	✓	—
USBf	0	—	USB_FI	USB (Function) interrupt	Level/ Edge	587 (0x24B)	✓	#CR520_SEL	#CR521_SEL	✓	—
			USB_FDMA0	USB (Function) DMA 0 transmit completion	Level/ Edge	588 (0x24C)	✓	#CR520_SEL	#CR521_SEL	—	—
			USB_FDMA1	USB (Function) DMA 1 transmit completion	Level/ Edge	589 (0x24D)	✓	#CR520_SEL	#CR521_SEL	—	—
SCI	0	—	SCI0_ERI	SCI0 Receive error	Level	590 (0x24E)	✓	#CR520_SEL	#CR521_SEL	—	—
			SCI0_RXI	SCI0 Receive data full	Edge	591 (0x24F)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SCI0_TXI	SCI0 Transmit data empty	Edge	592 (0x250)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SCI0_TEI	SCI0 Transmit end	Level	593 (0x251)	✓	#CR520_SEL	#CR521_SEL	—	—
	1	—	SCI1_ERI	SCI1 Receive error	Level	594 (0x252)	✓	#CR520_SEL	#CR521_SEL	—	—
			SCI1_RXI	SCI1 Receive data full	Edge	595 (0x253)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SCI1_TXI	SCI1 Transmit data empty	Edge	596 (0x254)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SCI1_TEI	SCI1 Transmit end	Level	597 (0x255)	✓	#CR520_SEL	#CR521_SEL	—	—

Table 12.17 Event table (31 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
SCI	2	—	SCI2_ERI	SCI2 Receive error	Level	598 (0x256)	✓	#CR520_SEL	#CR521_SEL	—	—
			SCI2_RXI	SCI2 Receive data full	Edge	599 (0x257)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SCI2_TXI	SCI2 Transmit data empty	Edge	600 (0x258)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SCI2_TEI	SCI2 Transmit end	Level	601 (0x259)	✓	#CR520_SEL	#CR521_SEL	—	—
	3	—	SCI3_ERI	SCI3 Receive error	Level	602 (0x25A)	✓	#CR520_SEL	#CR521_SEL	—	—
			SCI3_RXI	SCI3 Receive data full	Edge	603 (0x25B)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SCI3_TXI	SCI3 Transmit data empty	Edge	604 (0x25C)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SCI3_TEI	SCI3 Transmit end	Level	605 (0x25D)	✓	#CR520_SEL	#CR521_SEL	—	—
	4	—	SCI4_ERI	SCI4 Receive error	Level	606 (0x25E)	✓	#CR520_SEL	#CR521_SEL	—	—
			SCI4_RXI	SCI4 Receive data full	Edge	607 (0x25F)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SCI4_TXI	SCI4 Transmit data empty	Edge	608 (0x260)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SCI4_TEI	SCI4 Transmit end	Level	609 (0x261)	✓	#CR520_SEL	#CR521_SEL	—	—
	5	—	SCI5_ERI	SCI5 Receive error	Level	610 (0x262)	✓	#CR520_SEL	#CR521_SEL	—	—
			SCI5_RXI	SCI5 Receive data full	Edge	611 (0x263)	✓	#CR520_SEL	#CR521_SEL	—	—
			SCI5_TXI	SCI5 Transmit data empty	Edge	612 (0x264)	✓	#CR520_SEL	#CR521_SEL	—	—
			SCI5_TEI	SCI5 Transmit end	Level	613 (0x265)	✓	#CR520_SEL	#CR521_SEL	—	—
IIC	0	—	IIC0_EEI	IIC0 Transfer error or event generation	Level	614 (0x266)	✓	#CR520_SEL	#CR521_SEL	—	—
			IIC0_RXI	IIC0 Receive data full	Edge	615 (0x267)	✓	#CR520_SEL	#CR521_SEL	✓	—
			IIC0_TXI	IIC0 Transmit data empty	Edge	616 (0x268)	✓	#CR520_SEL	#CR521_SEL	✓	—
			IIC0_TEI	IIC0 Transmit end	Level	617 (0x269)	✓	#CR520_SEL	#CR521_SEL	—	—
	1	—	IIC1_EEI	IIC1 Transfer error or event generation	Level	618 (0x26A)	✓	#CR520_SEL	#CR521_SEL	—	—
			IIC1_RXI	IIC1 Receive data full	Edge	619 (0x26B)	✓	#CR520_SEL	#CR521_SEL	✓	—
			IIC1_TXI	IIC1 Transmit data empty	Edge	620 (0x26C)	✓	#CR520_SEL	#CR521_SEL	✓	—
			IIC1_TEI	IIC1 Transmit end	Level	621 (0x26D)	✓	#CR520_SEL	#CR521_SEL	—	—

Table 12.17 Event table (32 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
IIC	2	—	IIC2_EEI	IIC2 Transfer error or event generation	Level	622 (0x26E)	✓	#CR520_SEL	#CR521_SEL	—	—
			IIC2_RXI	IIC2 Receive data full	Edge	623 (0x26F)	✓	#CR520_SEL	#CR521_SEL	—	—
			IIC2_TXI	IIC2 Transmit data empty	Edge	624 (0x270)	✓	#CR520_SEL	#CR521_SEL	—	—
			IIC2_TEI	IIC2 Transmit end	Level	625 (0x271)	✓	#CR520_SEL	#CR521_SEL	—	—
CANFD	0	0	CAN0_TX	CAFND0 Channel TX interrupt	Level	626 (0x272)	✓	#CR520_SEL	#CR521_SEL	—	—
			CAN0_CHERR	CAFND0 Channel CAN error interrupt	Level	627 (0x273)	✓	#CR520_SEL	#CR521_SEL	—	—
			CAN0_COMFRX	CAFND0 Common RX FIFO or TXQ interrupt	Level	628 (0x274)	✓	#CR520_SEL	#CR521_SEL	—	—
			CAN0_CF_DMAREQ	CAFND0 First common FIFO DMA request	Edge	841 (0x349)	—	—	—	✓	—
		1	CAN1_TX	CAFND1 Channel TX interrupt	Level	629 (0x275)	✓	#CR520_SEL	#CR521_SEL	—	—
			CAN1_CHERR	CAFND1 Channel CAN error interrupt	Level	630 (0x276)	✓	#CR520_SEL	#CR521_SEL	—	—
			CAN1_COMFRX	CAFND1 Common RX FIFO or TXQ interrupt	Level	631 (0x277)	✓	#CR520_SEL	#CR521_SEL	—	—
			CAN1_CF_DMAREQ	CAFND1 First common FIFO DMA request	Edge	842 (0x34A)	—	—	—	✓	—
		COM	CAN_RXF	CANFD RX FIFO interrupt	Level	632 (0x278)	✓	#CR520_SEL	#CR521_SEL	—	—
			CAN_GLERR	CANFD Global error Interrupt	Level	633 (0x279)	✓	#CR520_SEL	#CR521_SEL	—	—
			CAN_RF_DMAREQ0	CAFND RX FIFO 0 DMA request	Edge	843 (0x34B)	—	—	—	✓	—
			CAN_RF_DMAREQ1	CAFND RX FIFO 1 DMA request	Edge	844 (0x34C)	—	—	—	✓	—
			CAN_RF_DMAREQ2	CAFND RX FIFO 2 DMA request	Edge	845 (0x34D)	—	—	—	✓	—
			CAN_RF_DMAREQ3	CAFND RX FIFO 3 DMA request	Edge	846 (0x34E)	—	—	—	✓	—
			CAN_RF_DMAREQ4	CAFND RX FIFO 4 DMA request	Edge	847 (0x34F)	—	—	—	✓	—
			CAN_RF_DMAREQ5	CAFND RX FIFO 5 DMA request	Edge	848 (0x350)	—	—	—	✓	—
			CAN_RF_DMAREQ6	CAFND RX FIFO 6 DMA request	Edge	849 (0x351)	—	—	—	✓	—
			CAN_RF_DMAREQ7	CAFND RX FIFO 7 DMA request	Edge	850 (0x352)	—	—	—	✓	—

Table 12.17 Event table (33 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
SPI	0	—	SPI0_SPRI	SPI0 Reception buffer full	Edge	634 (0x27A)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SPI0_SPTI	SPI0 Transmit buffer empty	Edge	635 (0x27B)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SPI0_SPII	SPI0 SPI idle	Level	636 (0x27C)	✓	#CR520_SEL	#CR521_SEL	—	—
			SPI0_SPEI	SPI0 SPI error (mode fault, underrun, overrun, parity error, and received data ready)	Level	637 (0x27D)	✓	#CR520_SEL	#CR521_SEL	—	—
			SPI0_SPCEND	SPI0 Communication complete	Edge	638 (0x27E)	✓	#CR520_SEL	#CR521_SEL	—	—
	1	—	SPI1_SPRI	SPI1 Reception buffer full	Edge	639 (0x27F)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SPI1_SPTI	SPI1 Transmit buffer empty	Edge	640 (0x280)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SPI1_SPII	SPI1 SPI idle	Level	641 (0x281)	✓	#CR520_SEL	#CR521_SEL	—	—
			SPI1_SPEI	SPI1 SPI error (mode fault, underrun, overrun, parity error, and received data ready)	Level	642 (0x282)	✓	#CR520_SEL	#CR521_SEL	—	—
			SPI1_SPCEND	SPI1 Communication complete	Edge	643 (0x283)	✓	#CR520_SEL	#CR521_SEL	—	—
	2	—	SPI2_SPRI	SPI2 Reception buffer full	Edge	644 (0x284)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SPI2_SPTI	SPI2 Transmit buffer empty	Edge	645 (0x285)	✓	#CR520_SEL	#CR521_SEL	✓	—
			SPI2_SPII	SPI2 SPI idle	Level	646 (0x286)	✓	#CR520_SEL	#CR521_SEL	—	—
			SPI2_SPEI	SPI2 SPI error (mode fault, underrun, overrun, parity error, and received data ready)	Level	647 (0x287)	✓	#CR520_SEL	#CR521_SEL	—	—
			SPI2_SPCEND	SPI2 Communication complete	Edge	648 (0x288)	✓	#CR520_SEL	#CR521_SEL	—	—
	3	—	SPI3_SPRI	SPI3 Reception buffer full	Edge	649 (0x289)	✓	#CR520_SEL	#CR521_SEL	—	—
			SPI3_SPTI	SPI3 Transmit buffer empty	Edge	650 (0x28A)	✓	#CR520_SEL	#CR521_SEL	—	—
			SPI3_SPII	SPI3 SPI idle	Level	651 (0x28B)	✓	#CR520_SEL	#CR521_SEL	—	—
			SPI3_SPEI	SPI3 SPI error (mode fault, underrun, overrun, parity error, and received data ready)	Level	652 (0x28C)	✓	#CR520_SEL	#CR521_SEL	—	—
			SPI3_SPCEND	SPI3 Communication complete	Edge	653 (0x28D)	✓	#CR520_SEL	#CR521_SEL	—	—

Table 12.17 Event table (34 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
xSPI	0	—	XSPI0_INT	xSPI0 Interrupt	Level	654 (0x28E)	✓	#CR520_SEL	#CR521_SEL	—	—
			XSPI0_INTERR	xSPI0 Error interrupt	Level	655 (0x28F)	✓	#CR520_SEL	#CR521_SEL	—	—
	1	—	XSPI1_INT	xSPI1 Interrupt	Level	656 (0x290)	✓	#CR520_SEL	#CR521_SEL	—	—
			XSPI1_INTERR	xSPI1 Error interrupt	Level	657 (0x291)	✓	#CR520_SEL	#CR521_SEL	—	—
DSMIF	0	COM	DSMIF0_CDRUI	DSMIF0 Current Data register Update (Combined with ch0 to ch2)	Edge	658 (0x292)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF0_CDUPDC	DSMIF0 Current Data register Update Common	Edge	659 (0x293)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF0_CDACUPDC	DSMIF0 Current Capture Data register A Update Common	Edge	660 (0x294)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF0_CDBCUPDC	DSMIF0 Current Capture Data register B Update Common	Edge	661 (0x295)	✓	#CR520_SEL	#CR521_SEL	✓	—
	1	COM	DSMIF1_CDRUI	DSMIF1 Current Data register Update (Combined with ch0 to ch2)	Edge	662 (0x296)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF1_CDUPDC	DSMIF1 Current Data register Update Common	Edge	663 (0x297)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF1_CDACUPDC	DSMIF1 Current Capture Data register A Update Common	Edge	664 (0x298)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF1_CDBCUPDC	DSMIF1 Current Capture Data register B Update Common	Edge	665 (0x299)	✓	#CR520_SEL	#CR521_SEL	✓	—
	2	COM	DSMIF2_CDRUI	DSMIF2 Current Data register Update (Combined with ch0 to ch2)	Edge	666 (0x29A)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF2_CDUPDC	DSMIF2 Current Data register Update Common	Edge	667 (0x29B)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF2_CDACUPDC	DSMIF2 Current Capture Data register A Update Common	Edge	668 (0x29C)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF2_CDBCUPDC	DSMIF2 Current Capture Data register B Update Common	Edge	669 (0x29D)	✓	#CR520_SEL	#CR521_SEL	✓	—
	3	COM	DSMIF3_CDRUI	DSMIF3 Current Data register Update (Combined with ch0 to ch2)	Edge	670 (0x29E)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF3_CDUPDC	DSMIF3 Current Data register Update Common	Edge	671 (0x29F)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF3_CDACUPDC	DSMIF3 Current Capture Data register A Update Common	Edge	672 (0x2A0)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF3_CDBCUPDC	DSMIF3 Current Capture Data register B Update Common	Edge	673 (0x2A1)	✓	#CR520_SEL	#CR521_SEL	✓	—

Table 12.17 Event table (35 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
DSMIF	4	COM	DSMIF4_CDRUI	DSMIF4 Current Data register Update (Combined with ch0 to ch2)	Edge	674 (0x2A2)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF4_CDUPDC	DSMIF4 Current Data register Update Common	Edge	675 (0x2A3)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF4_CDACUPDC	DSMIF4 Current Capture Data register A Update Common	Edge	676 (0x2A4)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF4_CDBCUPDC	DSMIF4 Current Capture Data register B Update Common	Edge	677 (0x2A5)	✓	#CR520_SEL	#CR521_SEL	✓	—
	5	COM	DSMIF5_CDRUI	DSMIF5 Current Data register Update (Combined with ch0 to ch2)	Edge	678 (0x2A6)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF5_CDUPDC	DSMIF5 Current Data register Update Common	Edge	679 (0x2A7)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF5_CDACUPDC	DSMIF5 Current Capture Data register A Update Common	Edge	680 (0x2A8)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF5_CDBCUPDC	DSMIF5 Current Capture Data register B Update Common	Edge	681 (0x2A9)	✓	#CR520_SEL	#CR521_SEL	✓	—
	6	COM	DSMIF6_CDRUI	DSMIF6 Current Data register Update (Combined with ch0 to ch2)	Edge	682 (0x2AA)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF6_CDUPDC	DSMIF6 Current Data register Update Common	Edge	683 (0x2AB)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF6_CDACUPDC	DSMIF6 Current Capture Data register A Update Common	Edge	684 (0x2AC)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF6_CDBCUPDC	DSMIF6 Current Capture Data register B Update Common	Edge	685 (0x2AD)	✓	#CR520_SEL	#CR521_SEL	✓	—
	7	COM	DSMIF7_CDRUI	DSMIF7 Current Data register Update (Combined with ch0 to ch2)	Edge	686 (0x2AE)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF7_CDUPDC	DSMIF7 Current Data register Update Common	Edge	687 (0x2AF)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF7_CDACUPDC	DSMIF7 Current Capture Data register A Update Common	Edge	688 (0x2B0)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF7_CDBCUPDC	DSMIF7 Current Capture Data register B Update Common	Edge	689 (0x2B1)	✓	#CR520_SEL	#CR521_SEL	✓	—

Table 12.17 Event table (36 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
DSMIF	8	COM	DSMIF8_CDRUI	DSMIF8 Current Data register Update (Combined with ch0 to ch2)	Edge	690 (0x2B2)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF8_CDUPDC	DSMIF8 Current Data register Update Common	Edge	691 (0x2B3)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF8_CDACUPDC	DSMIF8 Current Capture Data register A Update Common	Edge	692 (0x2B4)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF8_CDBCUPDC	DSMIF8 Current Capture Data register B Update Common	Edge	693 (0x2B5)	✓	#CR520_SEL	#CR521_SEL	✓	—
	9	COM	DSMIF9_CDRUI	DSMIF9 Current Data register Update (Combined with ch0 to ch2)	Edge	694 (0x2B6)	✓	#CR520_SEL	#CR521_SEL	—	—
			DSMIF9_CDUPDC	DSMIF9 Current Data register Update Common	Edge	695 (0x2B7)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF9_CDACUPDC	DSMIF9 Current Capture Data register A Update Common	Edge	696 (0x2B8)	✓	#CR520_SEL	#CR521_SEL	✓	—
			DSMIF9_CDBCUPDC	DSMIF9 Current Capture Data register B Update Common	Edge	697 (0x2B9)	✓	#CR520_SEL	#CR521_SEL	✓	—
ADC12	0	—	ADC120_ADI	ADC120 A/D scan end interrupt	Edge	698 (0x2BA)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ADC120_GBADI	ADC120 A/D scan end interrupt for Group B	Edge	699 (0x2BB)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ADC120_GCADI	ADC120 A/D scan end interrupt for Group C	Edge	700 (0x2BC)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ADC120_CMPAI	ADC120 Window A compare match	Level	701 (0x2BD)	✓	#CR520_SEL	#CR521_SEL	—	—
			ADC120_CMPBI	ADC120 Window B compare match	Level	702 (0x2BE)	✓	#CR520_SEL	#CR521_SEL	—	—
			ADC120_WCMPM	ADC120 compare match	Edge	851 (0x353)	—	—	—	✓	✓
			ADC120_WCMPUM	ADC120 compare mismatch.	Edge	852 (0x354)	—	—	—	✓	✓
	1	—	ADC121_ADI	ADC121 A/D scan end interrupt	Edge	703 (0x2BF)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ADC121_GBADI	ADC121 A/D scan end interrupt for Group B	Edge	704 (0x2C0)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ADC121_GCADI	ADC121 A/D scan end interrupt for Group C	Edge	705 (0x2C1)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ADC121_CMPAI	ADC121 Window A compare match	Level	706 (0x2C2)	✓	#CR520_SEL	#CR521_SEL	—	—
			ADC121_CMPBI	ADC121 Window B compare match	Level	707 (0x2C3)	✓	#CR520_SEL	#CR521_SEL	—	—
			ADC121_WCMPM	ADC121 compare match	Edge	853 (0x355)	—	—	—	✓	✓
			ADC121_WCMPUM	ADC121 compare mismatch.	Edge	854 (0x356)	—	—	—	✓	✓

Table 12.17 Event table (37 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
ADC12	2	—	ADC122_ADI	ADC122 A/D scan end interrupt	Edge	708 (0x2C4)	✓	#CR520_SEL	#CR521_SEL	✓	✓
			ADC122_GBADI	ADC122 A/D scan end interrupt for Group B	Edge	709 (0x2C5)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ADC122_GCADI	ADC122 A/D scan end interrupt for Group C	Edge	710 (0x2C6)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ADC122_CMPAI	ADC122 Window A compare match	Level	711 (0x2C7)	✓	#CR520_SEL	#CR521_SEL	—	—
			ADC122_CMPBI	ADC122 Window B compare match	Level	712 (0x2C8)	✓	#CR520_SEL	#CR521_SEL	—	—
			ADC122_WCMPM	ADC122 compare match	Edge	855 (0x357)	—	—	—	✓	✓
			ADC122_WCMPUM	ADC122 compare mismatch.	Edge	856 (0x358)	—	—	—	✓	✓
TSU	0	—	TSUADI	A/D conversion-end	Edge	713 (0x2C9)	✓	#CR520_SEL	#CR521_SEL	—	—
			TSUADCMPI	Comparison result match	Level	714 (0x2CA)	✓	#CR520_SEL	#CR521_SEL	—	—
SHOSTIF	0	—	SHOST_INT	SHOSTIF interrupt	Level	715 (0x2CB)	✓	#CR520_SEL	#CR521_SEL	—	—
MBXSEM	0	—	MBX_INT0	Mailbox (Host to CA55/CR52) interrupt 0	Level	377 (0x179)	✓	✓	✓	✓	✓
			MBX_INT1	Mailbox (Host to CA55/CR52) interrupt 1	Level	378 (0x17A)	✓	✓	✓	✓	✓
			MBX_INT2	Mailbox (Host to CA55/CR52) interrupt 2	Level	379 (0x17B)	✓	✓	✓	✓	✓
			MBX_INT3	Mailbox (Host to CA55/CR52) interrupt 3	Level	380 (0x17C)	✓	✓	✓	✓	✓
			MBX_INTAR0	Mailbox (CA55 to CR52_0) interrupt	Level	381 (0x17D)	—	✓	—	—	—
			MBX_INTR1R0	Mailbox (CR52_1 to CR52_0) interrupt	Level	382 (0x17E)	—	✓	—	—	—
			MBX_INTAR1	Mailbox (CA55 to CR52_1) interrupt	Level	381 (0x17D)	—	—	✓	—	—
			MBX_INTR0R1	Mailbox (CR52_0 to CR52_1) interrupt	Level	382 (0x17E)	—	—	✓	—	—
			MBX_INTR0A0	Mailbox (CR52_0 to CA55 core0) interrupt	Level	385 (0x181)	✓	—	—	—	—
			MBX_INTR0A1	Mailbox (CR52_0 to CA55 core1) interrupt	Level	386 (0x182)	✓	—	—	—	—
			MBX_INTR0A2	Mailbox (CR52_0 to CA55 core2) interrupt	Level	387 (0x183)	✓	—	—	—	—
			MBX_INTR0A3	Mailbox (CR52_0 to CA55 core3) interrupt	Level	388 (0x184)	✓	—	—	—	—
			MBX_INTR1A0	Mailbox (CR52_1 to CA55 core0) interrupt	Level	389 (0x185)	✓	—	—	—	—
			MBX_INTR1A1	Mailbox (CR52_1 to CA55 core1) interrupt	Level	390 (0x186)	✓	—	—	—	—
			MBX_INTR1A2	Mailbox (CR52_1 to CA55 core2) interrupt	Level	391 (0x187)	✓	—	—	—	—

Table 12.17 Event table (38 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
MBXSEM	0	—	MBX_INTR1A3	Mailbox (CR52_1 to CA55 core3) interrupt	Level	392 (0x188)	✓	—	—	—	—
			MBX_INTA0A1	Mailbox (CA55 core0 to CA55 core1) interrupt	Level	393 (0x189)	✓	—	—	—	—
			MBX_INTA0A2	Mailbox (CA55 core0 to CA55 core2) interrupt	Level	394 (0x18A)	✓	—	—	—	—
			MBX_INTA0A3	Mailbox (CA55 core0 to CA55 core3) interrupt	Level	395 (0x18B)	✓	—	—	—	—
			MBX_INTA1A0	Mailbox (CA55 core1 to CA55 core0) interrupt	Level	396 (0x18C)	✓	—	—	—	—
			MBX_INTA1A2	Mailbox (CA55 core1 to CA55 core2) interrupt	Level	397 (0x18D)	✓	—	—	—	—
			MBX_INTA1A3	Mailbox (CA55 core1 to CA55 core3) interrupt	Level	398 (0x18E)	✓	—	—	—	—
			MBX_INTA2A0	Mailbox (CA55 core2 to CA55 core0) interrupt	Level	399 (0x18F)	✓	—	—	—	—
			MBX_INTA2A1	Mailbox (CA55 core2 to CA55 core1) interrupt	Level	400 (0x190)	✓	—	—	—	—
			MBX_INTA2A3	Mailbox (CA55 core2 to CA55 core3) interrupt	Level	401 (0x191)	✓	—	—	—	—
			MBX_INTA3A0	Mailbox (CA55 core3 to CA55 core0) interrupt	Level	402 (0x192)	✓	—	—	—	—
			MBX_INTA3A1	Mailbox (CA55 core3 to CA55 core1) interrupt	Level	403 (0x193)	✓	—	—	—	—
			MBX_INTA3A2	Mailbox (CA55 core3 to CA55 core2) interrupt	Level	404 (0x194)	✓	—	—	—	—
ENCIF	0	—	ENCIF00_INT0	ENCIF00 combined interrupt 0	—	716 (0x2CC)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF00_INT1	ENCIF00 combined interrupt 1	—	717 (0x2CD)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF00_INT2	ENCIF00 combined interrupt 2	—	718 (0x2CE)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF00_INT3	ENCIF00 combined interrupt 3	—	719 (0x2CF)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	1	—	ENCIF01_INT0	ENCIF01 combined interrupt 0	—	720 (0x2D0)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF01_INT1	ENCIF01 combined interrupt 1	—	721 (0x2D1)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF01_INT2	ENCIF01 combined interrupt 2	—	722 (0x2D2)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF01_INT3	ENCIF01 combined interrupt 3	—	723 (0x2D3)	✓	#CR520_SEL	#CR521_SEL	✓	✓

Table 12.17 Event table (39 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
ENCIF	2	—	ENCIF02_INT0	ENCIF02 combined interrupt 0	—	724 (0x2D4)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF02_INT1	ENCIF02 combined interrupt 1	—	725 (0x2D5)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF02_INT2	ENCIF02 combined interrupt 2	—	726 (0x2D6)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF02_INT3	ENCIF02 combined interrupt 3	—	727 (0x2D7)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	3	—	ENCIF03_INT0	ENCIF03 combined interrupt 0	—	728 (0x2D8)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF03_INT1	ENCIF03 combined interrupt 1	—	729 (0x2D9)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF03_INT2	ENCIF03 combined interrupt 2	—	730 (0x2DA)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF03_INT3	ENCIF03 combined interrupt 3	—	731 (0x2DB)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	4	—	ENCIF04_INT0	ENCIF04 combined interrupt 0	—	732 (0x2DC)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF04_INT1	ENCIF04 combined interrupt 1	—	733 (0x2DD)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF04_INT2	ENCIF04 combined interrupt 2	—	734 (0x2DE)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF04_INT3	ENCIF04 combined interrupt 3	—	735 (0x2DF)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	5	—	ENCIF05_INT0	ENCIF05 combined interrupt 0	—	736 (0x2E0)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF05_INT1	ENCIF05 combined interrupt 1	—	737 (0x2E1)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF05_INT2	ENCIF05 combined interrupt 2	—	738 (0x2E2)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF05_INT3	ENCIF05 combined interrupt 3	—	739 (0x2E3)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	6	—	ENCIF06_INT0	ENCIF06 combined interrupt 0	—	740 (0x2E4)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF06_INT1	ENCIF06 combined interrupt 1	—	741 (0x2E5)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF06_INT2	ENCIF06 combined interrupt 2	—	742 (0x2E6)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF06_INT3	ENCIF06 combined interrupt 3	—	743 (0x2E7)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	7	—	ENCIF07_INT0	ENCIF07 combined interrupt 0	—	744 (0x2E8)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF07_INT1	ENCIF07 combined interrupt 1	—	745 (0x2E9)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF07_INT2	ENCIF07 combined interrupt 2	—	746 (0x2EA)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF07_INT3	ENCIF07 combined interrupt 3	—	747 (0x2EB)	✓	#CR520_SEL	#CR521_SEL	✓	✓

Table 12.17 Event table (40 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
ENCIF	8	—	ENCIF08_INT0	ENCIF08 combined interrupt 0	—	748 (0x2EC)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF08_INT1	ENCIF08 combined interrupt 1	—	749 (0x2ED)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF08_INT2	ENCIF08 combined interrupt 2	—	750 (0x2EE)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF08_INT3	ENCIF08 combined interrupt 3	—	751 (0x2EF)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	9	—	ENCIF09_INT0	ENCIF09 combined interrupt 0	—	752 (0x2F0)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF09_INT1	ENCIF09 combined interrupt 1	—	753 (0x2F1)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF09_INT2	ENCIF09 combined interrupt 2	—	754 (0x2F2)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF09_INT3	ENCIF09 combined interrupt 3	—	755 (0x2F3)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	10	—	ENCIF10_INT0	ENCIF10 combined interrupt 0	—	756 (0x2F4)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF10_INT1	ENCIF10 combined interrupt 1	—	757 (0x2F5)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF10_INT2	ENCIF10 combined interrupt 2	—	758 (0x2F6)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF10_INT3	ENCIF10 combined interrupt 3	—	759 (0x2F7)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	11	—	ENCIF11_INT0	ENCIF11 combined interrupt 0	—	760 (0x2F8)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF11_INT1	ENCIF11 combined interrupt 1	—	761 (0x2F9)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF11_INT2	ENCIF11 combined interrupt 2	—	762 (0x2FA)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF11_INT3	ENCIF11 combined interrupt 3	—	763 (0x2FB)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	12	—	ENCIF12_INT0	ENCIF12 combined interrupt 0	—	764 (0x2FC)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF12_INT1	ENCIF12 combined interrupt 1	—	765 (0x2FD)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF12_INT2	ENCIF12 combined interrupt 2	—	766 (0x2FE)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF12_INT3	ENCIF12 combined interrupt 3	—	767 (0x2FF)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	13	—	ENCIF13_INT0	ENCIF13 combined interrupt 0	—	768 (0x300)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF13_INT1	ENCIF13 combined interrupt 1	—	769 (0x301)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF13_INT2	ENCIF13 combined interrupt 2	—	770 (0x302)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF13_INT3	ENCIF13 combined interrupt 3	—	771 (0x303)	✓	#CR520_SEL	#CR521_SEL	✓	✓

Table 12.17 Event table (41 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
ENCIF	14	—	ENCIF14_INT0	ENCIF14 combined interrupt 0	—	772 (0x304)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF14_INT1	ENCIF14 combined interrupt 1	—	773 (0x305)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF14_INT2	ENCIF14 combined interrupt 2	—	774 (0x306)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF14_INT3	ENCIF14 combined interrupt 3	—	775 (0x307)	✓	#CR520_SEL	#CR521_SEL	✓	✓
	15	—	ENCIF15_INT0	ENCIF15 combined interrupt 0	—	776 (0x308)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF15_INT1	ENCIF15 combined interrupt 1	—	777 (0x309)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF15_INT2	ENCIF15 combined interrupt 2	—	778 (0x30A)	✓	#CR520_SEL	#CR521_SEL	✓	—
			ENCIF15_INT3	ENCIF15 combined interrupt 3	—	779 (0x30B)	✓	#CR520_SEL	#CR521_SEL	✓	✓
LCDC	0	—	LCDC_VSPD_INT	LCDC VSPD interrupt	Level	780 (0x30C)	✓	#CR520_SEL	#CR521_SEL	—	—
			LCDC_DU_INT	LCDC DU interrupt	Level	781 (0x30D)	✓	#CR520_SEL	#CR521_SEL	—	—
SDHI	0	—	SDHI0_OXMNIRQ	SDHI0 interrupt	Level	782 (0x30E)	✓	#CR520_SEL	#CR521_SEL	—	—
			SDHI0_OXASIOIRQ	SDHI0 SDIO access interrupt	Level	783 (0x30F)	✓	#CR520_SEL	#CR521_SEL	—	—
	1	—	SDHI1_OXMNIRQ	SDHI1 interrupt	Level	784 (0x310)	✓	#CR520_SEL	#CR521_SEL	—	—
			SDHI1_OXASIOIRQ	SDHI1 SDIO access interrupt	Level	785 (0x311)	✓	#CR520_SEL	#CR521_SEL	—	—
PCIE	0	0	PCIE0_INTA_RC	PCIE0 INTA interrupt for Root complex	Level	786 (0x312)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_INTB_RC	PCIE0 INTB interrupt for Root complex	Level	787 (0x313)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_INTC_RC	PCIE0 INTC interrupt for Root complex	Level	788 (0x314)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_INTD_RC	PCIE0 INTD interrupt for Root complex	Level	789 (0x315)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_INTMSI_RC	PCIE0 MSI interrupt for Root complex	Level	790 (0x316)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_INT_LINK_BANDWIDTH	PCIE0 Link width change interrupt for Root complex	Level	791 (0x317)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_INT_EQUALIZATION_REQUEST	PCIE0 Link Equalization Request interrupt for Root complex	Level	792 (0x318)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_INT_PM_PME	PCIE0 PM_PME Message receive interrupt for Root complex	Level	793 (0x319)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_INT_DMA	PCIE0 Event interrupt of DMA	Level	794 (0x31A)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_INT_PCIE_EVT	PCIE0 Event interrupt of PCIE	Level	795 (0x31B)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_INT_MSG	PCIE0 Message receive interrupt	Level	796 (0x31C)	✓	#CR520_SEL	#CR521_SEL	—	—

Table 12.17 Event table (42 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
PCIE	0	0	PCIE0_INT_ALL	PCIE0 ORed interrupt of all interrupt	Level	797 (0x31D)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_TURN_OFF_EVENT	PCIE0 PME_Turn_Off Msg. receive flag output for Endpoint	Level	798 (0x31E)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_PMU_POWEROFF	PCIE0 POWEROFF indication on L2	Level	799 (0x31F)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_D3_EVENT_F0	PCIE0 Non-D0 State transition request receive output Function #0 for Endpoint	Level	800 (0x320)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_D3_EVENT_F1	PCIE0 Non-D0 State transition request receive output Function #1 for Endpoint	Level	801 (0x321)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_CFG_PMCSR_PME_STATUS_WRITECLEAR_F0	PCIE0 PME_STATUS clear output Function #0 for Endpoint	Level	802 (0x322)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_CFG_PMCSR_PME_STATUS_WRITECLEAR_F1	PCIE0 PME_STATUS clear output Function #1 for Endpoint	Level	803 (0x323)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_FLR_REQ_F0	PCIE0 FLR request Function #0 for Endpoint	Level	804 (0x324)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE0_FLR_REQ_F1	PCIE0 FLR request Function #1 for Endpoint	Level	805 (0x325)	✓	#CR520_SEL	#CR521_SEL	—	—
	0	1	PCIE1_INTA_RC	PCIE1 INTA interrupt for Root complex	Level	806 (0x326)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_INTB_RC	PCIE1 INTB interrupt for Root complex	Level	807 (0x327)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_INTC_RC	PCIE1 INTC interrupt for Root complex	Level	808 (0x328)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_INTD_RC	PCIE1 INTD interrupt for Root complex	Level	809 (0x329)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_INTMSI_RC	PCIE1 MSI interrupt for Root complex	Level	810 (0x32A)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_INT_LINK_BANDWIDTH	PCIE1 Link width change interrupt for Root complex	Level	811 (0x32B)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_INT_EQUALIZATION_REQUEST	PCIE1 Link Equalization Request interrupt for Root complex	Level	812 (0x32C)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_INT_PM_PME	PCIE1 PM_PME Message receive interrupt for Root complex	Level	813 (0x32D)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_INT_DMA	PCIE1 Event interrupt of DMA	Level	814 (0x32E)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_INT_PCIE_EVT	PCIE1 Event interrupt of PCIE	Level	815 (0x32F)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_INT_MSG	PCIE1 Message receive interrupt	Level	816 (0x330)	✓	#CR520_SEL	#CR521_SEL	—	—

Table 12.17 Event table (43 of 43)

Module	UNIT	CH	Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 GIC-600 request (SPI)	Cortex-R52 CPU0 GIC request (SPI)	Cortex-R52 CPU1 GIC request (SPI)	DMAC activation request	ELC source
PCIE	0	1	PCIE1_INT_ALL	PCIE1 ORed interrupt of all interrupt	Level	817 (0x331)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_TURN_OFF_EVENT	PCIE1 PME_Turn_Off Msg. receive flag output for Endpoint	Level	818 (0x332)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_PMU_POWEROFF	PCIE1 POWEROFF indication on L2	Level	819 (0x333)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_D3_EVENT_F0	PCIE1 Non-D0 State transition request receive output Function #0 for Endpoint	Level	820 (0x334)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_D3_EVENT_F1	PCIE1 Non-D0 State transition request receive output Function #1 for Endpoint	Level	821 (0x335)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_CFG_PMCSR_PME_STATUS_WRITECLEAR_F0	PCIE1 PME_STATUS clear output Function #0 for Endpoint	Level	822 (0x336)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_CFG_PMCSR_PME_STATUS_WRITECLEAR_F1	PCIE1 PME_STATUS clear output Function #1 for Endpoint	Level	823 (0x337)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_FLR_REQ_F0	PCIE1 FLR request Function #0 for Endpoint	Level	824 (0x338)	✓	#CR520_SEL	#CR521_SEL	—	—
			PCIE1_FLR_REQ_F1	PCIE1 FLR request Function #1 for Endpoint	Level	825 (0x339)	✓	#CR520_SEL	#CR521_SEL	—	—
DDRSS	0	—	DDRC_INT	Interrupt signal from the DDR controller	Level	826 (0x33A)	✓	#CR520_SEL	#CR521_SEL	—	—
			DDRPHY_INT	Interrupt signal from the DDR PHY	Level	827 (0x33B)	✓	#CR520_SEL	#CR521_SEL	—	—

Table 12.18 Cortex-A55 PPI event (1 of 2)

PPI INTID	Name	Interrupt sources	Interrupt type
16 (0x010)	—	—	—
17 (0x011)	—	—	—
18 (0x012)	—	—	—
19 (0x013)	—	—	—
20 (0x014)	—	—	—
21 (0x015)	—	—	—
22 (0x016)	nCOMMIRQ[3:0]	Comms channel receive or transmit interrupt request	Level
23 (0x017)	nPMUIRQ[3:0]	PMU interrupt request	Level
24 (0x018)	CTIIRQ[3:0]	CTI interrupt	Edge
25 (0x019)	nVCPUMNTIRQ[3:0]	virtual CPU interface maintenance interrupt PPI output	Level
26 (0x01A)	nCNTHPIRQ[3:0]	Hypervisor physical timer event	Level
27 (0x01B)	nCNTVIRQ[3:0]	virtual timer event	Level
28 (0x01C)	nCNTHVIRQ[3:0]	Hypervisor virtual timer event	Level

Table 12.18 Cortex-A55 PPI event (2 of 2)

PPI INTID	Name	Interrupt sources	Interrupt type
29 (0x01D)	nCNTPSIRQ[3:0]	Secure physical timer event	Level
30 (0x01E)	nCNTPNSIRQ[3:0]	Non-secure physical timer event	Level
31 (0x01F)	—	—	—

Table 12.19 Cortex-R52 CPU0 selectable event (1 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT0_IRQ0	Cortex-R52 CPU0 selected peripheral interrupt 0	Level/ Edge	384 (0x180)	—	✓	—	—	—
SELECT0_IRQ1	Cortex-R52 CPU0 selected peripheral interrupt 1	Level/ Edge	385 (0x181)	—	✓	—	—	—
SELECT0_IRQ2	Cortex-R52 CPU0 selected peripheral interrupt 2	Level/ Edge	386 (0x182)	—	✓	—	—	—
SELECT0_IRQ3	Cortex-R52 CPU0 selected peripheral interrupt 3	Level/ Edge	387 (0x183)	—	✓	—	—	—
SELECT0_IRQ4	Cortex-R52 CPU0 selected peripheral interrupt 4	Level/ Edge	388 (0x184)	—	✓	—	—	—
SELECT0_IRQ5	Cortex-R52 CPU0 selected peripheral interrupt 5	Level/ Edge	389 (0x185)	—	✓	—	—	—
SELECT0_IRQ6	Cortex-R52 CPU0 selected peripheral interrupt 6	Level/ Edge	390 (0x186)	—	✓	—	—	—
SELECT0_IRQ7	Cortex-R52 CPU0 selected peripheral interrupt 7	Level/ Edge	391 (0x187)	—	✓	—	—	—
SELECT0_IRQ8	Cortex-R52 CPU0 selected peripheral interrupt 8	Level/ Edge	392 (0x188)	—	✓	—	—	—
SELECT0_IRQ9	Cortex-R52 CPU0 selected peripheral interrupt 9	Level/ Edge	393 (0x189)	—	✓	—	—	—
SELECT0_IRQ10	Cortex-R52 CPU0 selected peripheral interrupt 10	Level/ Edge	394 (0x18A)	—	✓	—	—	—
SELECT0_IRQ11	Cortex-R52 CPU0 selected peripheral interrupt 11	Level/ Edge	395 (0x18B)	—	✓	—	—	—
SELECT0_IRQ12	Cortex-R52 CPU0 selected peripheral interrupt 12	Level/ Edge	396 (0x18C)	—	✓	—	—	—
SELECT0_IRQ13	Cortex-R52 CPU0 selected peripheral interrupt 13	Level/ Edge	397 (0x18D)	—	✓	—	—	—
SELECT0_IRQ14	Cortex-R52 CPU0 selected peripheral interrupt 14	Level/ Edge	398 (0x18E)	—	✓	—	—	—
SELECT0_IRQ15	Cortex-R52 CPU0 selected peripheral interrupt 15	Level/ Edge	399 (0x18F)	—	✓	—	—	—

Table 12.19 Cortex-R52 CPU0 selectable event (2 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT0_IRQ16	Cortex-R52 CPU0 selected peripheral interrupt 16	Level/ Edge	400 (0x190)	—	✓	—	—	—
SELECT0_IRQ17	Cortex-R52 CPU0 selected peripheral interrupt 17	Level/ Edge	401 (0x191)	—	✓	—	—	—
SELECT0_IRQ18	Cortex-R52 CPU0 selected peripheral interrupt 18	Level/ Edge	402 (0x192)	—	✓	—	—	—
SELECT0_IRQ19	Cortex-R52 CPU0 selected peripheral interrupt 19	Level/ Edge	403 (0x193)	—	✓	—	—	—
SELECT0_IRQ20	Cortex-R52 CPU0 selected peripheral interrupt 20	Level/ Edge	404 (0x194)	—	✓	—	—	—
SELECT0_IRQ21	Cortex-R52 CPU0 selected peripheral interrupt 21	Level/ Edge	405 (0x195)	—	✓	—	—	—
SELECT0_IRQ22	Cortex-R52 CPU0 selected peripheral interrupt 22	Level/ Edge	406 (0x196)	—	✓	—	—	—
SELECT0_IRQ23	Cortex-R52 CPU0 selected peripheral interrupt 23	Level/ Edge	407 (0x197)	—	✓	—	—	—
SELECT0_IRQ24	Cortex-R52 CPU0 selected peripheral interrupt 24	Level/ Edge	408 (0x198)	—	✓	—	—	—
SELECT0_IRQ25	Cortex-R52 CPU0 selected peripheral interrupt 25	Level/ Edge	409 (0x199)	—	✓	—	—	—
SELECT0_IRQ26	Cortex-R52 CPU0 selected peripheral interrupt 26	Level/ Edge	410 (0x19A)	—	✓	—	—	—
SELECT0_IRQ27	Cortex-R52 CPU0 selected peripheral interrupt 27	Level/ Edge	411 (0x19B)	—	✓	—	—	—
SELECT0_IRQ28	Cortex-R52 CPU0 selected peripheral interrupt 28	Level/ Edge	412 (0x19C)	—	✓	—	—	—
SELECT0_IRQ29	Cortex-R52 CPU0 selected peripheral interrupt 29	Level/ Edge	413 (0x19D)	—	✓	—	—	—
SELECT0_IRQ30	Cortex-R52 CPU0 selected peripheral interrupt 30	Level/ Edge	414 (0x19E)	—	✓	—	—	—
SELECT0_IRQ31	Cortex-R52 CPU0 selected peripheral interrupt 31	Level/ Edge	415 (0x19F)	—	✓	—	—	—
SELECT0_IRQ32	Cortex-R52 CPU0 selected peripheral interrupt 32	Level/ Edge	416 (0x1A0)	—	✓	—	—	—
SELECT0_IRQ33	Cortex-R52 CPU0 selected peripheral interrupt 33	Level/ Edge	417 (0x1A1)	—	✓	—	—	—

Table 12.19 Cortex-R52 CPU0 selectable event (3 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT0_IRQ34	Cortex-R52 CPU0 selected peripheral interrupt 34	Level/ Edge	418 (0x1A2)	—	✓	—	—	—
SELECT0_IRQ35	Cortex-R52 CPU0 selected peripheral interrupt 35	Level/ Edge	419 (0x1A3)	—	✓	—	—	—
SELECT0_IRQ36	Cortex-R52 CPU0 selected peripheral interrupt 36	Level/ Edge	420 (0x1A4)	—	✓	—	—	—
SELECT0_IRQ37	Cortex-R52 CPU0 selected peripheral interrupt 37	Level/ Edge	421 (0x1A5)	—	✓	—	—	—
SELECT0_IRQ38	Cortex-R52 CPU0 selected peripheral interrupt 38	Level/ Edge	422 (0x1A6)	—	✓	—	—	—
SELECT0_IRQ39	Cortex-R52 CPU0 selected peripheral interrupt 39	Level/ Edge	423 (0x1A7)	—	✓	—	—	—
SELECT0_IRQ40	Cortex-R52 CPU0 selected peripheral interrupt 40	Level/ Edge	424 (0x1A8)	—	✓	—	—	—
SELECT0_IRQ41	Cortex-R52 CPU0 selected peripheral interrupt 41	Level/ Edge	425 (0x1A9)	—	✓	—	—	—
SELECT0_IRQ42	Cortex-R52 CPU0 selected peripheral interrupt 42	Level/ Edge	426 (0x1AA)	—	✓	—	—	—
SELECT0_IRQ43	Cortex-R52 CPU0 selected peripheral interrupt 43	Level/ Edge	427 (0x1AB)	—	✓	—	—	—
SELECT0_IRQ44	Cortex-R52 CPU0 selected peripheral interrupt 44	Level/ Edge	428 (0x1AC)	—	✓	—	—	—
SELECT0_IRQ45	Cortex-R52 CPU0 selected peripheral interrupt 45	Level/ Edge	429 (0x1AD)	—	✓	—	—	—
SELECT0_IRQ46	Cortex-R52 CPU0 selected peripheral interrupt 46	Level/ Edge	430 (0x1AE)	—	✓	—	—	—
SELECT0_IRQ47	Cortex-R52 CPU0 selected peripheral interrupt 47	Level/ Edge	431 (0x1AF)	—	✓	—	—	—
SELECT0_IRQ48	Cortex-R52 CPU0 selected peripheral interrupt 48	Level/ Edge	432 (0x1B0)	—	✓	—	—	—
SELECT0_IRQ49	Cortex-R52 CPU0 selected peripheral interrupt 49	Level/ Edge	433 (0x1B1)	—	✓	—	—	—
SELECT0_IRQ50	Cortex-R52 CPU0 selected peripheral interrupt 50	Level/ Edge	434 (0x1B2)	—	✓	—	—	—
SELECT0_IRQ51	Cortex-R52 CPU0 selected peripheral interrupt 51	Level/ Edge	435 (0x1B3)	—	✓	—	—	—

Table 12.19 Cortex-R52 CPU0 selectable event (4 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT0_IRQ52	Cortex-R52 CPU0 selected peripheral interrupt 52	Level/ Edge	436 (0x1B4)	—	✓	—	—	—
SELECT0_IRQ53	Cortex-R52 CPU0 selected peripheral interrupt 53	Level/ Edge	437 (0x1B5)	—	✓	—	—	—
SELECT0_IRQ54	Cortex-R52 CPU0 selected peripheral interrupt 54	Level/ Edge	438 (0x1B6)	—	✓	—	—	—
SELECT0_IRQ55	Cortex-R52 CPU0 selected peripheral interrupt 55	Level/ Edge	439 (0x1B7)	—	✓	—	—	—
SELECT0_IRQ56	Cortex-R52 CPU0 selected peripheral interrupt 56	Level/ Edge	440 (0x1B8)	—	✓	—	—	—
SELECT0_IRQ57	Cortex-R52 CPU0 selected peripheral interrupt 57	Level/ Edge	441 (0x1B9)	—	✓	—	—	—
SELECT0_IRQ58	Cortex-R52 CPU0 selected peripheral interrupt 58	Level/ Edge	442 (0x1BA)	—	✓	—	—	—
SELECT0_IRQ59	Cortex-R52 CPU0 selected peripheral interrupt 59	Level/ Edge	443 (0x1BB)	—	✓	—	—	—
SELECT0_IRQ60	Cortex-R52 CPU0 selected peripheral interrupt 60	Level/ Edge	444 (0x1BC)	—	✓	—	—	—
SELECT0_IRQ61	Cortex-R52 CPU0 selected peripheral interrupt 61	Level/ Edge	445 (0x1BD)	—	✓	—	—	—
SELECT0_IRQ62	Cortex-R52 CPU0 selected peripheral interrupt 62	Level/ Edge	446 (0x1BE)	—	✓	—	—	—
SELECT0_IRQ63	Cortex-R52 CPU0 selected peripheral interrupt 63	Level/ Edge	447 (0x1BF)	—	✓	—	—	—
SELECT0_IRQ64	Cortex-R52 CPU0 selected peripheral interrupt 64	Level/ Edge	448 (0x1C0)	—	✓	—	—	—
SELECT0_IRQ65	Cortex-R52 CPU0 selected peripheral interrupt 65	Level/ Edge	449 (0x1C1)	—	✓	—	—	—
SELECT0_IRQ66	Cortex-R52 CPU0 selected peripheral interrupt 66	Level/ Edge	450 (0x1C2)	—	✓	—	—	—
SELECT0_IRQ67	Cortex-R52 CPU0 selected peripheral interrupt 67	Level/ Edge	451 (0x1C3)	—	✓	—	—	—
SELECT0_IRQ68	Cortex-R52 CPU0 selected peripheral interrupt 68	Level/ Edge	452 (0x1C4)	—	✓	—	—	—
SELECT0_IRQ69	Cortex-R52 CPU0 selected peripheral interrupt 69	Level/ Edge	453 (0x1C5)	—	✓	—	—	—

Table 12.19 Cortex-R52 CPU0 selectable event (5 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT0_IRQ70	Cortex-R52 CPU0 selected peripheral interrupt 70	Level/ Edge	454 (0x1C6)	—	✓	—	—	—
SELECT0_IRQ71	Cortex-R52 CPU0 selected peripheral interrupt 71	Level/ Edge	455 (0x1C7)	—	✓	—	—	—
SELECT0_IRQ72	Cortex-R52 CPU0 selected peripheral interrupt 72	Level/ Edge	456 (0x1C8)	—	✓	—	—	—
SELECT0_IRQ73	Cortex-R52 CPU0 selected peripheral interrupt 73	Level/ Edge	457 (0x1C9)	—	✓	—	—	—
SELECT0_IRQ74	Cortex-R52 CPU0 selected peripheral interrupt 74	Level/ Edge	458 (0x1CA)	—	✓	—	—	—
SELECT0_IRQ75	Cortex-R52 CPU0 selected peripheral interrupt 75	Level/ Edge	459 (0x1CB)	—	✓	—	—	—
SELECT0_IRQ76	Cortex-R52 CPU0 selected peripheral interrupt 76	Level/ Edge	460 (0x1CC)	—	✓	—	—	—
SELECT0_IRQ77	Cortex-R52 CPU0 selected peripheral interrupt 77	Level/ Edge	461 (0x1CD)	—	✓	—	—	—
SELECT0_IRQ78	Cortex-R52 CPU0 selected peripheral interrupt 78	Level/ Edge	462 (0x1CE)	—	✓	—	—	—
SELECT0_IRQ79	Cortex-R52 CPU0 selected peripheral interrupt 79	Level/ Edge	463 (0x1CF)	—	✓	—	—	—
SELECT0_IRQ80	Cortex-R52 CPU0 selected peripheral interrupt 80	Level/ Edge	464 (0x1D0)	—	✓	—	—	—
SELECT0_IRQ81	Cortex-R52 CPU0 selected peripheral interrupt 81	Level/ Edge	465 (0x1D1)	—	✓	—	—	—
SELECT0_IRQ82	Cortex-R52 CPU0 selected peripheral interrupt 82	Level/ Edge	466 (0x1D2)	—	✓	—	—	—
SELECT0_IRQ83	Cortex-R52 CPU0 selected peripheral interrupt 83	Level/ Edge	467 (0x1D3)	—	✓	—	—	—
SELECT0_IRQ84	Cortex-R52 CPU0 selected peripheral interrupt 84	Level/ Edge	468 (0x1D4)	—	✓	—	—	—
SELECT0_IRQ85	Cortex-R52 CPU0 selected peripheral interrupt 85	Level/ Edge	469 (0x1D5)	—	✓	—	—	—
SELECT0_IRQ86	Cortex-R52 CPU0 selected peripheral interrupt 86	Level/ Edge	470 (0x1D6)	—	✓	—	—	—
SELECT0_IRQ87	Cortex-R52 CPU0 selected peripheral interrupt 87	Level/ Edge	471 (0x1D7)	—	✓	—	—	—

Table 12.19 Cortex-R52 CPU0 selectable event (6 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT0_IRQ88	Cortex-R52 CPU0 selected peripheral interrupt 88	Level/ Edge	472 (0x1D8)	—	✓	—	—	—
SELECT0_IRQ89	Cortex-R52 CPU0 selected peripheral interrupt 89	Level/ Edge	473 (0x1D9)	—	✓	—	—	—
SELECT0_IRQ90	Cortex-R52 CPU0 selected peripheral interrupt 90	Level/ Edge	474 (0x1DA)	—	✓	—	—	—
SELECT0_IRQ91	Cortex-R52 CPU0 selected peripheral interrupt 91	Level/ Edge	475 (0x1DB)	—	✓	—	—	—
SELECT0_IRQ92	Cortex-R52 CPU0 selected peripheral interrupt 92	Level/ Edge	476 (0x1DC)	—	✓	—	—	—
SELECT0_IRQ93	Cortex-R52 CPU0 selected peripheral interrupt 93	Level/ Edge	477 (0x1DD)	—	✓	—	—	—
SELECT0_IRQ94	Cortex-R52 CPU0 selected peripheral interrupt 94	Level/ Edge	478 (0x1DE)	—	✓	—	—	—
SELECT0_IRQ95	Cortex-R52 CPU0 selected peripheral interrupt 95	Level/ Edge	479 (0x1DF)	—	✓	—	—	—
SELECT0_IRQ96	Cortex-R52 CPU0 selected peripheral interrupt 96	Level/ Edge	480 (0x1E0)	—	✓	—	—	—
SELECT0_IRQ97	Cortex-R52 CPU0 selected peripheral interrupt 97	Level/ Edge	481 (0x1E1)	—	✓	—	—	—
SELECT0_IRQ98	Cortex-R52 CPU0 selected peripheral interrupt 98	Level/ Edge	482 (0x1E2)	—	✓	—	—	—
SELECT0_IRQ99	Cortex-R52 CPU0 selected peripheral interrupt 99	Level/ Edge	483 (0x1E3)	—	✓	—	—	—
SELECT0_IRQ100	Cortex-R52 CPU0 selected peripheral interrupt 100	Level/ Edge	484 (0x1E4)	—	✓	—	—	—
SELECT0_IRQ101	Cortex-R52 CPU0 selected peripheral interrupt 101	Level/ Edge	485 (0x1E5)	—	✓	—	—	—
SELECT0_IRQ102	Cortex-R52 CPU0 selected peripheral interrupt 102	Level/ Edge	486 (0x1E6)	—	✓	—	—	—
SELECT0_IRQ103	Cortex-R52 CPU0 selected peripheral interrupt 103	Level/ Edge	487 (0x1E7)	—	✓	—	—	—
SELECT0_IRQ104	Cortex-R52 CPU0 selected peripheral interrupt 104	Level/ Edge	488 (0x1E8)	—	✓	—	—	—
SELECT0_IRQ105	Cortex-R52 CPU0 selected peripheral interrupt 105	Level/ Edge	489 (0x1E9)	—	✓	—	—	—

Table 12.19 Cortex-R52 CPU0 selectable event (7 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT0_IRQ106	Cortex-R52 CPU0 selected peripheral interrupt 106	Level/ Edge	490 (0x1EA)	—	✓	—	—	—
SELECT0_IRQ107	Cortex-R52 CPU0 selected peripheral interrupt 107	Level/ Edge	491 (0x1EB)	—	✓	—	—	—
SELECT0_IRQ108	Cortex-R52 CPU0 selected peripheral interrupt 108	Level/ Edge	492 (0x1EC)	—	✓	—	—	—
SELECT0_IRQ109	Cortex-R52 CPU0 selected peripheral interrupt 109	Level/ Edge	493 (0x1ED)	—	✓	—	—	—
SELECT0_IRQ110	Cortex-R52 CPU0 selected peripheral interrupt 110	Level/ Edge	494 (0x1EE)	—	✓	—	—	—
SELECT0_IRQ111	Cortex-R52 CPU0 selected peripheral interrupt 111	Level/ Edge	495 (0x1EF)	—	✓	—	—	—
SELECT0_IRQ112	Cortex-R52 CPU0 selected peripheral interrupt 112	Level/ Edge	496 (0x1F0)	—	✓	—	—	—
SELECT0_IRQ113	Cortex-R52 CPU0 selected peripheral interrupt 113	Level/ Edge	497 (0x1F1)	—	✓	—	—	—
SELECT0_IRQ114	Cortex-R52 CPU0 selected peripheral interrupt 114	Level/ Edge	498 (0x1F2)	—	✓	—	—	—
SELECT0_IRQ115	Cortex-R52 CPU0 selected peripheral interrupt 115	Level/ Edge	499 (0x1F3)	—	✓	—	—	—
SELECT0_IRQ116	Cortex-R52 CPU0 selected peripheral interrupt 116	Level/ Edge	500 (0x1F4)	—	✓	—	—	—
SELECT0_IRQ117	Cortex-R52 CPU0 selected peripheral interrupt 117	Level/ Edge	501 (0x1F5)	—	✓	—	—	—
SELECT0_IRQ118	Cortex-R52 CPU0 selected peripheral interrupt 118	Level/ Edge	502 (0x1F6)	—	✓	—	—	—
SELECT0_IRQ119	Cortex-R52 CPU0 selected peripheral interrupt 119	Level/ Edge	503 (0x1F7)	—	✓	—	—	—
SELECT0_IRQ120	Cortex-R52 CPU0 selected peripheral interrupt 120	Level/ Edge	504 (0x1F8)	—	✓	—	—	—
SELECT0_IRQ121	Cortex-R52 CPU0 selected peripheral interrupt 121	Level/ Edge	505 (0x1F9)	—	✓	—	—	—
SELECT0_IRQ122	Cortex-R52 CPU0 selected peripheral interrupt 122	Level/ Edge	506 (0x1FA)	—	✓	—	—	—
SELECT0_IRQ123	Cortex-R52 CPU0 selected peripheral interrupt 123	Level/ Edge	507 (0x1FB)	—	✓	—	—	—

Table 12.19 Cortex-R52 CPU0 selectable event (8 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT0_IRQ124	Cortex-R52 CPU0 selected peripheral interrupt 124	Level/ Edge	508 (0x1FC)	—	✓	—	—	—
SELECT0_IRQ125	Cortex-R52 CPU0 selected peripheral interrupt 125	Level/ Edge	509 (0x1FD)	—	✓	—	—	—
SELECT0_IRQ126	Cortex-R52 CPU0 selected peripheral interrupt 126	Level/ Edge	510 (0x1FE)	—	✓	—	—	—
SELECT0_IRQ127	Cortex-R52 CPU0 selected peripheral interrupt 127	Level/ Edge	511 (0x1FF)	—	✓	—	—	—
SELECT0_IRQ128	Cortex-R52 CPU0 selected peripheral interrupt 128	Level/ Edge	512 (0x200)	—	✓	—	—	—
SELECT0_IRQ129	Cortex-R52 CPU0 selected peripheral interrupt 129	Level/ Edge	513 (0x201)	—	✓	—	—	—
SELECT0_IRQ130	Cortex-R52 CPU0 selected peripheral interrupt 130	Level/ Edge	514 (0x202)	—	✓	—	—	—
SELECT0_IRQ131	Cortex-R52 CPU0 selected peripheral interrupt 131	Level/ Edge	515 (0x203)	—	✓	—	—	—
SELECT0_IRQ132	Cortex-R52 CPU0 selected peripheral interrupt 132	Level/ Edge	516 (0x204)	—	✓	—	—	—
SELECT0_IRQ133	Cortex-R52 CPU0 selected peripheral interrupt 133	Level/ Edge	517 (0x205)	—	✓	—	—	—
SELECT0_IRQ134	Cortex-R52 CPU0 selected peripheral interrupt 134	Level/ Edge	518 (0x206)	—	✓	—	—	—
SELECT0_IRQ135	Cortex-R52 CPU0 selected peripheral interrupt 135	Level/ Edge	519 (0x207)	—	✓	—	—	—
SELECT0_IRQ136	Cortex-R52 CPU0 selected peripheral interrupt 136	Level/ Edge	520 (0x208)	—	✓	—	—	—
SELECT0_IRQ137	Cortex-R52 CPU0 selected peripheral interrupt 137	Level/ Edge	521 (0x209)	—	✓	—	—	—
SELECT0_IRQ138	Cortex-R52 CPU0 selected peripheral interrupt 138	Level/ Edge	522 (0x20A)	—	✓	—	—	—
SELECT0_IRQ139	Cortex-R52 CPU0 selected peripheral interrupt 139	Level/ Edge	523 (0x20B)	—	✓	—	—	—
SELECT0_IRQ140	Cortex-R52 CPU0 selected peripheral interrupt 140	Level/ Edge	524 (0x20C)	—	✓	—	—	—
SELECT0_IRQ141	Cortex-R52 CPU0 selected peripheral interrupt 141	Level/ Edge	525 (0x20D)	—	✓	—	—	—

Table 12.19 Cortex-R52 CPU0 selectable event (9 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT0_IRQ142	Cortex-R52 CPU0 selected peripheral interrupt 142	Level/ Edge	526 (0x20E)	—	✓	—	—	—
SELECT0_IRQ143	Cortex-R52 CPU0 selected peripheral interrupt 143	Level/ Edge	527 (0x20F)	—	✓	—	—	—
SELECT0_IRQ144	Cortex-R52 CPU0 selected peripheral interrupt 144	Level/ Edge	528 (0x210)	—	✓	—	—	—
SELECT0_IRQ145	Cortex-R52 CPU0 selected peripheral interrupt 145	Level/ Edge	529 (0x211)	—	✓	—	—	—
SELECT0_IRQ146	Cortex-R52 CPU0 selected peripheral interrupt 146	Level/ Edge	530 (0x212)	—	✓	—	—	—
SELECT0_IRQ147	Cortex-R52 CPU0 selected peripheral interrupt 147	Level/ Edge	531 (0x213)	—	✓	—	—	—
SELECT0_IRQ148	Cortex-R52 CPU0 selected peripheral interrupt 148	Level/ Edge	532 (0x214)	—	✓	—	—	—
SELECT0_IRQ149	Cortex-R52 CPU0 selected peripheral interrupt 149	Level/ Edge	533 (0x215)	—	✓	—	—	—
SELECT0_IRQ150	Cortex-R52 CPU0 selected peripheral interrupt 150	Level/ Edge	534 (0x216)	—	✓	—	—	—
SELECT0_IRQ151	Cortex-R52 CPU0 selected peripheral interrupt 151	Level/ Edge	535 (0x217)	—	✓	—	—	—
SELECT0_IRQ152	Cortex-R52 CPU0 selected peripheral interrupt 152	Level/ Edge	536 (0x218)	—	✓	—	—	—
SELECT0_IRQ153	Cortex-R52 CPU0 selected peripheral interrupt 153	Level/ Edge	537 (0x219)	—	✓	—	—	—
SELECT0_IRQ154	Cortex-R52 CPU0 selected peripheral interrupt 154	Level/ Edge	538 (0x21A)	—	✓	—	—	—
SELECT0_IRQ155	Cortex-R52 CPU0 selected peripheral interrupt 155	Level/ Edge	539 (0x21B)	—	✓	—	—	—
SELECT0_IRQ156	Cortex-R52 CPU0 selected peripheral interrupt 156	Level/ Edge	540 (0x21C)	—	✓	—	—	—
SELECT0_IRQ157	Cortex-R52 CPU0 selected peripheral interrupt 157	Level/ Edge	541 (0x21D)	—	✓	—	—	—
SELECT0_IRQ158	Cortex-R52 CPU0 selected peripheral interrupt 158	Level/ Edge	542 (0x21E)	—	✓	—	—	—
SELECT0_IRQ159	Cortex-R52 CPU0 selected peripheral interrupt 159	Level/ Edge	543 (0x21F)	—	✓	—	—	—

Table 12.20 Cortex-R52 CPU1 selectable event (1 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT1_IRQ0	Cortex-R52 CPU1 selected peripheral interrupt 0	Level/ Edge	384 (0x180)	—	—	✓	—	—
SELECT1_IRQ1	Cortex-R52 CPU1 selected peripheral interrupt 1	Level/ Edge	385 (0x181)	—	—	✓	—	—
SELECT1_IRQ2	Cortex-R52 CPU1 selected peripheral interrupt 2	Level/ Edge	386 (0x182)	—	—	✓	—	—
SELECT1_IRQ3	Cortex-R52 CPU1 selected peripheral interrupt 3	Level/ Edge	387 (0x183)	—	—	✓	—	—
SELECT1_IRQ4	Cortex-R52 CPU1 selected peripheral interrupt 4	Level/ Edge	388 (0x184)	—	—	✓	—	—
SELECT1_IRQ5	Cortex-R52 CPU1 selected peripheral interrupt 5	Level/ Edge	389 (0x185)	—	—	✓	—	—
SELECT1_IRQ6	Cortex-R52 CPU1 selected peripheral interrupt 6	Level/ Edge	390 (0x186)	—	—	✓	—	—
SELECT1_IRQ7	Cortex-R52 CPU1 selected peripheral interrupt 7	Level/ Edge	391 (0x187)	—	—	✓	—	—
SELECT1_IRQ8	Cortex-R52 CPU1 selected peripheral interrupt 8	Level/ Edge	392 (0x188)	—	—	✓	—	—
SELECT1_IRQ9	Cortex-R52 CPU1 selected peripheral interrupt 9	Level/ Edge	393 (0x189)	—	—	✓	—	—
SELECT1_IRQ10	Cortex-R52 CPU1 selected peripheral interrupt 10	Level/ Edge	394 (0x18A)	—	—	✓	—	—
SELECT1_IRQ11	Cortex-R52 CPU1 selected peripheral interrupt 11	Level/ Edge	395 (0x18B)	—	—	✓	—	—
SELECT1_IRQ12	Cortex-R52 CPU1 selected peripheral interrupt 12	Level/ Edge	396 (0x18C)	—	—	✓	—	—
SELECT1_IRQ13	Cortex-R52 CPU1 selected peripheral interrupt 13	Level/ Edge	397 (0x18D)	—	—	✓	—	—
SELECT1_IRQ14	Cortex-R52 CPU1 selected peripheral interrupt 14	Level/ Edge	398 (0x18E)	—	—	✓	—	—
SELECT1_IRQ15	Cortex-R52 CPU1 selected peripheral interrupt 15	Level/ Edge	399 (0x18F)	—	—	✓	—	—
SELECT1_IRQ16	Cortex-R52 CPU1 selected peripheral interrupt 16	Level/ Edge	400 (0x190)	—	—	✓	—	—
SELECT1_IRQ17	Cortex-R52 CPU1 selected peripheral interrupt 17	Level/ Edge	401 (0x191)	—	—	✓	—	—

Table 12.20 Cortex-R52 CPU1 selectable event (2 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT1_IRQ18	Cortex-R52 CPU1 selected peripheral interrupt 18	Level/ Edge	402 (0x192)	—	—	✓	—	—
SELECT1_IRQ19	Cortex-R52 CPU1 selected peripheral interrupt 19	Level/ Edge	403 (0x193)	—	—	✓	—	—
SELECT1_IRQ20	Cortex-R52 CPU1 selected peripheral interrupt 20	Level/ Edge	404 (0x194)	—	—	✓	—	—
SELECT1_IRQ21	Cortex-R52 CPU1 selected peripheral interrupt 21	Level/ Edge	405 (0x195)	—	—	✓	—	—
SELECT1_IRQ22	Cortex-R52 CPU1 selected peripheral interrupt 22	Level/ Edge	406 (0x196)	—	—	✓	—	—
SELECT1_IRQ23	Cortex-R52 CPU1 selected peripheral interrupt 23	Level/ Edge	407 (0x197)	—	—	✓	—	—
SELECT1_IRQ24	Cortex-R52 CPU1 selected peripheral interrupt 24	Level/ Edge	408 (0x198)	—	—	✓	—	—
SELECT1_IRQ25	Cortex-R52 CPU1 selected peripheral interrupt 25	Level/ Edge	409 (0x199)	—	—	✓	—	—
SELECT1_IRQ26	Cortex-R52 CPU1 selected peripheral interrupt 26	Level/ Edge	410 (0x19A)	—	—	✓	—	—
SELECT1_IRQ27	Cortex-R52 CPU1 selected peripheral interrupt 27	Level/ Edge	411 (0x19B)	—	—	✓	—	—
SELECT1_IRQ28	Cortex-R52 CPU1 selected peripheral interrupt 28	Level/ Edge	412 (0x19C)	—	—	✓	—	—
SELECT1_IRQ29	Cortex-R52 CPU1 selected peripheral interrupt 29	Level/ Edge	413 (0x19D)	—	—	✓	—	—
SELECT1_IRQ30	Cortex-R52 CPU1 selected peripheral interrupt 30	Level/ Edge	414 (0x19E)	—	—	✓	—	—
SELECT1_IRQ31	Cortex-R52 CPU1 selected peripheral interrupt 31	Level/ Edge	415 (0x19F)	—	—	✓	—	—
SELECT1_IRQ32	Cortex-R52 CPU1 selected peripheral interrupt 32	Level/ Edge	416 (0x1A0)	—	—	✓	—	—
SELECT1_IRQ33	Cortex-R52 CPU1 selected peripheral interrupt 33	Level/ Edge	417 (0x1A1)	—	—	✓	—	—
SELECT1_IRQ34	Cortex-R52 CPU1 selected peripheral interrupt 34	Level/ Edge	418 (0x1A2)	—	—	✓	—	—
SELECT1_IRQ35	Cortex-R52 CPU1 selected peripheral interrupt 35	Level/ Edge	419 (0x1A3)	—	—	✓	—	—

Table 12.20 Cortex-R52 CPU1 selectable event (3 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT1_IRQ36	Cortex-R52 CPU1 selected peripheral interrupt 36	Level/ Edge	420 (0x1A4)	—	—	✓	—	—
SELECT1_IRQ37	Cortex-R52 CPU1 selected peripheral interrupt 37	Level/ Edge	421 (0x1A5)	—	—	✓	—	—
SELECT1_IRQ38	Cortex-R52 CPU1 selected peripheral interrupt 38	Level/ Edge	422 (0x1A6)	—	—	✓	—	—
SELECT1_IRQ39	Cortex-R52 CPU1 selected peripheral interrupt 39	Level/ Edge	423 (0x1A7)	—	—	✓	—	—
SELECT1_IRQ40	Cortex-R52 CPU1 selected peripheral interrupt 40	Level/ Edge	424 (0x1A8)	—	—	✓	—	—
SELECT1_IRQ41	Cortex-R52 CPU1 selected peripheral interrupt 41	Level/ Edge	425 (0x1A9)	—	—	✓	—	—
SELECT1_IRQ42	Cortex-R52 CPU1 selected peripheral interrupt 42	Level/ Edge	426 (0x1AA)	—	—	✓	—	—
SELECT1_IRQ43	Cortex-R52 CPU1 selected peripheral interrupt 43	Level/ Edge	427 (0x1AB)	—	—	✓	—	—
SELECT1_IRQ44	Cortex-R52 CPU1 selected peripheral interrupt 44	Level/ Edge	428 (0x1AC)	—	—	✓	—	—
SELECT1_IRQ45	Cortex-R52 CPU1 selected peripheral interrupt 45	Level/ Edge	429 (0x1AD)	—	—	✓	—	—
SELECT1_IRQ46	Cortex-R52 CPU1 selected peripheral interrupt 46	Level/ Edge	430 (0x1AE)	—	—	✓	—	—
SELECT1_IRQ47	Cortex-R52 CPU1 selected peripheral interrupt 47	Level/ Edge	431 (0x1AF)	—	—	✓	—	—
SELECT1_IRQ48	Cortex-R52 CPU1 selected peripheral interrupt 48	Level/ Edge	432 (0x1B0)	—	—	✓	—	—
SELECT1_IRQ49	Cortex-R52 CPU1 selected peripheral interrupt 49	Level/ Edge	433 (0x1B1)	—	—	✓	—	—
SELECT1_IRQ50	Cortex-R52 CPU1 selected peripheral interrupt 50	Level/ Edge	434 (0x1B2)	—	—	✓	—	—
SELECT1_IRQ51	Cortex-R52 CPU1 selected peripheral interrupt 51	Level/ Edge	435 (0x1B3)	—	—	✓	—	—
SELECT1_IRQ52	Cortex-R52 CPU1 selected peripheral interrupt 52	Level/ Edge	436 (0x1B4)	—	—	✓	—	—
SELECT1_IRQ53	Cortex-R52 CPU1 selected peripheral interrupt 53	Level/ Edge	437 (0x1B5)	—	—	✓	—	—

Table 12.20 Cortex-R52 CPU1 selectable event (4 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT1_IRQ54	Cortex-R52 CPU1 selected peripheral interrupt 54	Level/ Edge	438 (0x1B6)	—	—	✓	—	—
SELECT1_IRQ55	Cortex-R52 CPU1 selected peripheral interrupt 55	Level/ Edge	439 (0x1B7)	—	—	✓	—	—
SELECT1_IRQ56	Cortex-R52 CPU1 selected peripheral interrupt 56	Level/ Edge	440 (0x1B8)	—	—	✓	—	—
SELECT1_IRQ57	Cortex-R52 CPU1 selected peripheral interrupt 57	Level/ Edge	441 (0x1B9)	—	—	✓	—	—
SELECT1_IRQ58	Cortex-R52 CPU1 selected peripheral interrupt 58	Level/ Edge	442 (0x1BA)	—	—	✓	—	—
SELECT1_IRQ59	Cortex-R52 CPU1 selected peripheral interrupt 59	Level/ Edge	443 (0x1BB)	—	—	✓	—	—
SELECT1_IRQ60	Cortex-R52 CPU1 selected peripheral interrupt 60	Level/ Edge	444 (0x1BC)	—	—	✓	—	—
SELECT1_IRQ61	Cortex-R52 CPU1 selected peripheral interrupt 61	Level/ Edge	445 (0x1BD)	—	—	✓	—	—
SELECT1_IRQ62	Cortex-R52 CPU1 selected peripheral interrupt 62	Level/ Edge	446 (0x1BE)	—	—	✓	—	—
SELECT1_IRQ63	Cortex-R52 CPU1 selected peripheral interrupt 63	Level/ Edge	447 (0x1BF)	—	—	✓	—	—
SELECT1_IRQ64	Cortex-R52 CPU1 selected peripheral interrupt 64	Level/ Edge	448 (0x1C0)	—	—	✓	—	—
SELECT1_IRQ65	Cortex-R52 CPU1 selected peripheral interrupt 65	Level/ Edge	449 (0x1C1)	—	—	✓	—	—
SELECT1_IRQ66	Cortex-R52 CPU1 selected peripheral interrupt 66	Level/ Edge	450 (0x1C2)	—	—	✓	—	—
SELECT1_IRQ67	Cortex-R52 CPU1 selected peripheral interrupt 67	Level/ Edge	451 (0x1C3)	—	—	✓	—	—
SELECT1_IRQ68	Cortex-R52 CPU1 selected peripheral interrupt 68	Level/ Edge	452 (0x1C4)	—	—	✓	—	—
SELECT1_IRQ69	Cortex-R52 CPU1 selected peripheral interrupt 69	Level/ Edge	453 (0x1C5)	—	—	✓	—	—
SELECT1_IRQ70	Cortex-R52 CPU1 selected peripheral interrupt 70	Level/ Edge	454 (0x1C6)	—	—	✓	—	—
SELECT1_IRQ71	Cortex-R52 CPU1 selected peripheral interrupt 71	Level/ Edge	455 (0x1C7)	—	—	✓	—	—

Table 12.20 Cortex-R52 CPU1 selectable event (5 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT1_IRQ72	Cortex-R52 CPU1 selected peripheral interrupt 72	Level/ Edge	456 (0x1C8)	—	—	✓	—	—
SELECT1_IRQ73	Cortex-R52 CPU1 selected peripheral interrupt 73	Level/ Edge	457 (0x1C9)	—	—	✓	—	—
SELECT1_IRQ74	Cortex-R52 CPU1 selected peripheral interrupt 74	Level/ Edge	458 (0x1CA)	—	—	✓	—	—
SELECT1_IRQ75	Cortex-R52 CPU1 selected peripheral interrupt 75	Level/ Edge	459 (0x1CB)	—	—	✓	—	—
SELECT1_IRQ76	Cortex-R52 CPU1 selected peripheral interrupt 76	Level/ Edge	460 (0x1CC)	—	—	✓	—	—
SELECT1_IRQ77	Cortex-R52 CPU1 selected peripheral interrupt 77	Level/ Edge	461 (0x1CD)	—	—	✓	—	—
SELECT1_IRQ78	Cortex-R52 CPU1 selected peripheral interrupt 78	Level/ Edge	462 (0x1CE)	—	—	✓	—	—
SELECT1_IRQ79	Cortex-R52 CPU1 selected peripheral interrupt 79	Level/ Edge	463 (0x1CF)	—	—	✓	—	—
SELECT1_IRQ80	Cortex-R52 CPU1 selected peripheral interrupt 80	Level/ Edge	464 (0x1D0)	—	—	✓	—	—
SELECT1_IRQ81	Cortex-R52 CPU1 selected peripheral interrupt 81	Level/ Edge	465 (0x1D1)	—	—	✓	—	—
SELECT1_IRQ82	Cortex-R52 CPU1 selected peripheral interrupt 82	Level/ Edge	466 (0x1D2)	—	—	✓	—	—
SELECT1_IRQ83	Cortex-R52 CPU1 selected peripheral interrupt 83	Level/ Edge	467 (0x1D3)	—	—	✓	—	—
SELECT1_IRQ84	Cortex-R52 CPU1 selected peripheral interrupt 84	Level/ Edge	468 (0x1D4)	—	—	✓	—	—
SELECT1_IRQ85	Cortex-R52 CPU1 selected peripheral interrupt 85	Level/ Edge	469 (0x1D5)	—	—	✓	—	—
SELECT1_IRQ86	Cortex-R52 CPU1 selected peripheral interrupt 86	Level/ Edge	470 (0x1D6)	—	—	✓	—	—
SELECT1_IRQ87	Cortex-R52 CPU1 selected peripheral interrupt 87	Level/ Edge	471 (0x1D7)	—	—	✓	—	—
SELECT1_IRQ88	Cortex-R52 CPU1 selected peripheral interrupt 88	Level/ Edge	472 (0x1D8)	—	—	✓	—	—
SELECT1_IRQ89	Cortex-R52 CPU1 selected peripheral interrupt 89	Level/ Edge	473 (0x1D9)	—	—	✓	—	—

Table 12.20 Cortex-R52 CPU1 selectable event (6 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT1_IRQ90	Cortex-R52 CPU1 selected peripheral interrupt 90	Level/ Edge	474 (0x1DA)	—	—	✓	—	—
SELECT1_IRQ91	Cortex-R52 CPU1 selected peripheral interrupt 91	Level/ Edge	475 (0x1DB)	—	—	✓	—	—
SELECT1_IRQ92	Cortex-R52 CPU1 selected peripheral interrupt 92	Level/ Edge	476 (0x1DC)	—	—	✓	—	—
SELECT1_IRQ93	Cortex-R52 CPU1 selected peripheral interrupt 93	Level/ Edge	477 (0x1DD)	—	—	✓	—	—
SELECT1_IRQ94	Cortex-R52 CPU1 selected peripheral interrupt 94	Level/ Edge	478 (0x1DE)	—	—	✓	—	—
SELECT1_IRQ95	Cortex-R52 CPU1 selected peripheral interrupt 95	Level/ Edge	479 (0x1DF)	—	—	✓	—	—
SELECT1_IRQ96	Cortex-R52 CPU1 selected peripheral interrupt 96	Level/ Edge	480 (0x1E0)	—	—	✓	—	—
SELECT1_IRQ97	Cortex-R52 CPU1 selected peripheral interrupt 97	Level/ Edge	481 (0x1E1)	—	—	✓	—	—
SELECT1_IRQ98	Cortex-R52 CPU1 selected peripheral interrupt 98	Level/ Edge	482 (0x1E2)	—	—	✓	—	—
SELECT1_IRQ99	Cortex-R52 CPU1 selected peripheral interrupt 99	Level/ Edge	483 (0x1E3)	—	—	✓	—	—
SELECT1_IRQ100	Cortex-R52 CPU1 selected peripheral interrupt 100	Level/ Edge	484 (0x1E4)	—	—	✓	—	—
SELECT1_IRQ101	Cortex-R52 CPU1 selected peripheral interrupt 101	Level/ Edge	485 (0x1E5)	—	—	✓	—	—
SELECT1_IRQ102	Cortex-R52 CPU1 selected peripheral interrupt 102	Level/ Edge	486 (0x1E6)	—	—	✓	—	—
SELECT1_IRQ103	Cortex-R52 CPU1 selected peripheral interrupt 103	Level/ Edge	487 (0x1E7)	—	—	✓	—	—
SELECT1_IRQ104	Cortex-R52 CPU1 selected peripheral interrupt 104	Level/ Edge	488 (0x1E8)	—	—	✓	—	—
SELECT1_IRQ105	Cortex-R52 CPU1 selected peripheral interrupt 105	Level/ Edge	489 (0x1E9)	—	—	✓	—	—
SELECT1_IRQ106	Cortex-R52 CPU1 selected peripheral interrupt 106	Level/ Edge	490 (0x1EA)	—	—	✓	—	—
SELECT1_IRQ107	Cortex-R52 CPU1 selected peripheral interrupt 107	Level/ Edge	491 (0x1EB)	—	—	✓	—	—

Table 12.20 Cortex-R52 CPU1 selectable event (7 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT1_IRQ108	Cortex-R52 CPU1 selected peripheral interrupt 108	Level/ Edge	492 (0x1EC)	—	—	✓	—	—
SELECT1_IRQ109	Cortex-R52 CPU1 selected peripheral interrupt 109	Level/ Edge	493 (0x1ED)	—	—	✓	—	—
SELECT1_IRQ110	Cortex-R52 CPU1 selected peripheral interrupt 110	Level/ Edge	494 (0x1EE)	—	—	✓	—	—
SELECT1_IRQ111	Cortex-R52 CPU1 selected peripheral interrupt 111	Level/ Edge	495 (0x1EF)	—	—	✓	—	—
SELECT1_IRQ112	Cortex-R52 CPU1 selected peripheral interrupt 112	Level/ Edge	496 (0x1F0)	—	—	✓	—	—
SELECT1_IRQ113	Cortex-R52 CPU1 selected peripheral interrupt 113	Level/ Edge	497 (0x1F1)	—	—	✓	—	—
SELECT1_IRQ114	Cortex-R52 CPU1 selected peripheral interrupt 114	Level/ Edge	498 (0x1F2)	—	—	✓	—	—
SELECT1_IRQ115	Cortex-R52 CPU1 selected peripheral interrupt 115	Level/ Edge	499 (0x1F3)	—	—	✓	—	—
SELECT1_IRQ116	Cortex-R52 CPU1 selected peripheral interrupt 116	Level/ Edge	500 (0x1F4)	—	—	✓	—	—
SELECT1_IRQ117	Cortex-R52 CPU1 selected peripheral interrupt 117	Level/ Edge	501 (0x1F5)	—	—	✓	—	—
SELECT1_IRQ118	Cortex-R52 CPU1 selected peripheral interrupt 118	Level/ Edge	502 (0x1F6)	—	—	✓	—	—
SELECT1_IRQ119	Cortex-R52 CPU1 selected peripheral interrupt 119	Level/ Edge	503 (0x1F7)	—	—	✓	—	—
SELECT1_IRQ120	Cortex-R52 CPU1 selected peripheral interrupt 120	Level/ Edge	504 (0x1F8)	—	—	✓	—	—
SELECT1_IRQ121	Cortex-R52 CPU1 selected peripheral interrupt 121	Level/ Edge	505 (0x1F9)	—	—	✓	—	—
SELECT1_IRQ122	Cortex-R52 CPU1 selected peripheral interrupt 122	Level/ Edge	506 (0x1FA)	—	—	✓	—	—
SELECT1_IRQ123	Cortex-R52 CPU1 selected peripheral interrupt 123	Level/ Edge	507 (0x1FB)	—	—	✓	—	—
SELECT1_IRQ124	Cortex-R52 CPU1 selected peripheral interrupt 124	Level/ Edge	508 (0x1FC)	—	—	✓	—	—
SELECT1_IRQ125	Cortex-R52 CPU1 selected peripheral interrupt 125	Level/ Edge	509 (0x1FD)	—	—	✓	—	—

Table 12.20 Cortex-R52 CPU1 selectable event (8 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT1_IRQ126	Cortex-R52 CPU1 selected peripheral interrupt 126	Level/ Edge	510 (0x1FE)	—	—	✓	—	—
SELECT1_IRQ127	Cortex-R52 CPU1 selected peripheral interrupt 127	Level/ Edge	511 (0x1FF)	—	—	✓	—	—
SELECT1_IRQ128	Cortex-R52 CPU1 selected peripheral interrupt 128	Level/ Edge	512 (0x200)	—	—	✓	—	—
SELECT1_IRQ129	Cortex-R52 CPU1 selected peripheral interrupt 129	Level/ Edge	513 (0x201)	—	—	✓	—	—
SELECT1_IRQ130	Cortex-R52 CPU1 selected peripheral interrupt 130	Level/ Edge	514 (0x202)	—	—	✓	—	—
SELECT1_IRQ131	Cortex-R52 CPU1 selected peripheral interrupt 131	Level/ Edge	515 (0x203)	—	—	✓	—	—
SELECT1_IRQ132	Cortex-R52 CPU1 selected peripheral interrupt 132	Level/ Edge	516 (0x204)	—	—	✓	—	—
SELECT1_IRQ133	Cortex-R52 CPU1 selected peripheral interrupt 133	Level/ Edge	517 (0x205)	—	—	✓	—	—
SELECT1_IRQ134	Cortex-R52 CPU1 selected peripheral interrupt 134	Level/ Edge	518 (0x206)	—	—	✓	—	—
SELECT1_IRQ135	Cortex-R52 CPU1 selected peripheral interrupt 135	Level/ Edge	519 (0x207)	—	—	✓	—	—
SELECT1_IRQ136	Cortex-R52 CPU1 selected peripheral interrupt 136	Level/ Edge	520 (0x208)	—	—	✓	—	—
SELECT1_IRQ137	Cortex-R52 CPU1 selected peripheral interrupt 137	Level/ Edge	521 (0x209)	—	—	✓	—	—
SELECT1_IRQ138	Cortex-R52 CPU1 selected peripheral interrupt 138	Level/ Edge	522 (0x20A)	—	—	✓	—	—
SELECT1_IRQ139	Cortex-R52 CPU1 selected peripheral interrupt 139	Level/ Edge	523 (0x20B)	—	—	✓	—	—
SELECT1_IRQ140	Cortex-R52 CPU1 selected peripheral interrupt 140	Level/ Edge	524 (0x20C)	—	—	✓	—	—
SELECT1_IRQ141	Cortex-R52 CPU1 selected peripheral interrupt 141	Level/ Edge	525 (0x20D)	—	—	✓	—	—
SELECT1_IRQ142	Cortex-R52 CPU1 selected peripheral interrupt 142	Level/ Edge	526 (0x20E)	—	—	✓	—	—
SELECT1_IRQ143	Cortex-R52 CPU1 selected peripheral interrupt 143	Level/ Edge	527 (0x20F)	—	—	✓	—	—

Table 12.20 Cortex-R52 CPU1 selectable event (9 of 9)

Name	Interrupt sources	Interrupt type	EVENT number	Cortex-A55 (GIC-600) SPI	Cortex-R52 CPU0 SPI	Cortex-R52 CPU1 SPI	DMAC activation	ELC source
SELECT1_IRQ144	Cortex-R52 CPU1 selected peripheral interrupt 144	Level/ Edge	528 (0x210)	—	—	✓	—	—
SELECT1_IRQ145	Cortex-R52 CPU1 selected peripheral interrupt 145	Level/ Edge	529 (0x211)	—	—	✓	—	—
SELECT1_IRQ146	Cortex-R52 CPU1 selected peripheral interrupt 146	Level/ Edge	530 (0x212)	—	—	✓	—	—
SELECT1_IRQ147	Cortex-R52 CPU1 selected peripheral interrupt 147	Level/ Edge	531 (0x213)	—	—	✓	—	—
SELECT1_IRQ148	Cortex-R52 CPU1 selected peripheral interrupt 148	Level/ Edge	532 (0x214)	—	—	✓	—	—
SELECT1_IRQ149	Cortex-R52 CPU1 selected peripheral interrupt 149	Level/ Edge	533 (0x215)	—	—	✓	—	—
SELECT1_IRQ150	Cortex-R52 CPU1 selected peripheral interrupt 150	Level/ Edge	534 (0x216)	—	—	✓	—	—
SELECT1_IRQ151	Cortex-R52 CPU1 selected peripheral interrupt 151	Level/ Edge	535 (0x217)	—	—	✓	—	—
SELECT1_IRQ152	Cortex-R52 CPU1 selected peripheral interrupt 152	Level/ Edge	536 (0x218)	—	—	✓	—	—
SELECT1_IRQ153	Cortex-R52 CPU1 selected peripheral interrupt 153	Level/ Edge	537 (0x219)	—	—	✓	—	—
SELECT1_IRQ154	Cortex-R52 CPU1 selected peripheral interrupt 154	Level/ Edge	538 (0x21A)	—	—	✓	—	—
SELECT1_IRQ155	Cortex-R52 CPU1 selected peripheral interrupt 155	Level/ Edge	539 (0x21B)	—	—	✓	—	—
SELECT1_IRQ156	Cortex-R52 CPU1 selected peripheral interrupt 156	Level/ Edge	540 (0x21C)	—	—	✓	—	—
SELECT1_IRQ157	Cortex-R52 CPU1 selected peripheral interrupt 157	Level/ Edge	541 (0x21D)	—	—	✓	—	—
SELECT1_IRQ158	Cortex-R52 CPU1 selected peripheral interrupt 158	Level/ Edge	542 (0x21E)	—	—	✓	—	—
SELECT1_IRQ159	Cortex-R52 CPU1 selected peripheral interrupt 159	Level/ Edge	543 (0x21F)	—	—	✓	—	—

Table 12.21 GPT event to be input to GPT_SEL

Module	Name	Interrupt sources	Interrupt type	DMAC activation request	ELC source
GPT	GPTm_n_CCMPA	GPTm_n GTCCRA input capture/compare match	Edge	✓	✓
	GPTm_n_CCMPB	GPTm_n GTCCRB input capture/compare match	Edge	✓	✓
	GPTm_n_CMPC	GPTm_n GTCCRC compare match	Edge	✓	✓
	GPTm_n_CMPD	GPTm_n GTCCRD compare match	Edge	✓	✓
	GPTm_n_CMPE	GPTm_n GTCCRE compare match	Edge	✓	✓
	GPTm_n_CMPF	GPTm_n GTCCRF compare match	Edge	✓	✓
	GPTm_n_OVF	GPTm_n GTCNT overflow (GTPR compare match)	Edge	✓	✓
	GPTm_n_UDF	GPTm_n GTCNT underflow	Edge	✓	✓
	GPTm_n_DTE*1	GPTm_n Dead time error	Edge	—	—

Note: m: unit number (m = 0 to 10)
n: channel number
For m = 0 to 8, n = 0 to 4
For m = 9, n = 0 to 6
For m = 10, n = 0 to 3
Note 1. Not available for m = 10

Table 12.22 ENCIF event to be input to ENCIF_SEL

Module	Name	Interrupt sources	Interrupt type	DMAC activation request	ELC source
SCIE	SCIEn_ERI	SCIEn Receive error	Level	—	—
	SCIEn_RXI	SCIEn Receive data full	Edge	✓	—
	SCIEn_TXI	SCIEn Transmit data empty	Edge	✓	—
	SCIEn_TEI	SCIEn Transmit end	Level	—	—
ENDAT	ENDATn_INT1	ENDATn Interrupt output 1	Level	✓	—
BISS	BISSn_EOT	BISSn End of transmission interrupt	Level	✓	—
HDSL	HDSLn_INT	HDSLn Interrupt output	Level	✓	—
	HDSLn_INTS	HDSLn Interrupt output (safety related)	Level	✓	—
	HDSLn_FP	HDSLn Fast position received interrupt (POSTX1 & ~POSTX0)	Edge	✓	—
	HDSLn_SP	HDSLn Safe position received interrupt (POSTX1 & POSTX0)	Edge	✓	✓
AFMT	AFMTn_EOF	AFMTn Communication completion	Level	✓	—

Note: n = 0 to 15

12.4.3 Error Events

The ICU handles error events from modules. CA55ERR_STAT, CR520ERR_STAT, CR521ERR_STAT, PERIERR_STATn (n = 0 to 2), DSMIFERR_STATn (n = 0 to 11), and ENCIFERR_STATn (n = 0 to 4) registers show each error status. The ICU captures error events and generates combined interrupt signal CA55_ERR0, CR520_ERR0, CR521_ERR0, PERI_ERR0, DSMIF_ERR0, ENCIF_ERR0 and CA55_ERR1, CR520_ERR1, CR521_ERR1, PERI_ERR1, DSMIF_ERR1, ENCIF_ERR1.

Error events can be disabled by CA55ERR_E0MSK, CR520ERR_E0MSK, CR521ERR_E0MSK, PERIERR_E0MSKn (n = 0 to 2), DSMIFERR_E0MSKn (n = 0 to 11), ENCIFERR_E0MSKn (n = 0 to 4) registers and CA55ERR_E1MSK, CR520ERR_E1MSK, CR521ERR_E1MSK, PERIERR_E1MSKn (n = 0 to 2), DSMIFERR_E1MSKn (n = 0 to 11), ENCIFERR_E1MSKn (n = 0 to 4) registers.

Additionally, error events can cause the error reset. Error events that cause error reset are maskable in CA55ERR_RSTMSK, CR520ERR_RSTMSK, CR521ERR_RSTMSK, PERIERR_RSTMSKn (n = 0 to 2), DSMIFERR_RSTMSKn (n = 0 to 11), ENCIFERR_RSTMSKn (n = 0 to 4) registers.

All error events are masked as initial values. *ERR_STAT register is not subject to initialization by Error reset. Since it is not initialized, Error will occur again if the Error reset mask is released. If it returns to normal operation after Error reset, it is necessary to confirm error status and clear error flag after restarting.

Table 12.23 shows the relation among error control registers and combined interrupts. Table 12.24 to Table 12.27 show error events.

Table 12.23 Relation among error control registers and combined interrupts

Module	Error Event Status register	Error Event Status Clear register	Error Event Mask register	Combined interrupt
Cortex-A55, GIC-600	CA55ERR_STAT	CA55ERR_CLR	CA55ERR_E0MSK	CA55_ERR0
			CA55ERR_E1MSK	CA55_ERR1
			CA55ERR_RSTMSK	Error reset
Cortex-R52 CPU0	CR520ERR_STAT	CR520ERR_CLR	CR520ERR_E0MSK	CR520_ERR0
			CR520ERR_E1MSK	CR520_ERR1
			CR520ERR_RSTMSK	Error reset
Cortex-R52 CPU1	CR521ERR_STAT	CR521ERR_CLR	CR521ERR_E0MSK	CR521_ERR0
			CR521ERR_E1MSK	CR521_ERR1
			CR521ERR_RSTMSK	Error reset
Peripheral modules except DSMIF and ENCIF (n = 0 to 2)	PERIERR_STATn	PERIERR_CLRn	PERIERR_E0MSKn	PERI_ERR0
			PERIERR_E1MSKn	PERI_ERR1
			PERIERR_RSTMSKn	Error reset
DSMIF (n = 0 to 11)	DSMIFERR_STATn	DSMIFERR_CLRn	DSMIFERR_E0MSKn	DSMIF_ERR0
			DSMIFERR_E1MSKn	DSMIF_ERR1
			DSMIFERR_RSTMSKn	Error reset
ENCIF (n = 0 to 4)	ENCIFERR_STATn	ENCIFERR_CLRn	ENCIFERR_E0MSKn	ENCIF_ERR0
			ENCIFERR_E1MSKn	ENCIF_ERR1
			ENCIFERR_RSTMSKn	Error reset

Table 12.24 CPU error event table (1 of 4)

Module	UNIT	Core	Name	Interrupt sources	Bit of CA55ERR_XXX	Bit of CR520ERR_XXX	Bit of CR521ERR_XXX
Cortex-A55	0	0	nFAULTIRQ1	Fault indicator for a detected 1-bit or 2-bit ECC or Parity error in the RAMs 0	0	—	—
		1	nFAULTIRQ2	Fault indicator for a detected 1-bit or 2-bit ECC or Parity error in the RAMs 1	1	—	—
		2	nFAULTIRQ3	Fault indicator for a detected 1-bit or 2-bit ECC or Parity error in the RAMs 2	2	—	—
		3	nFAULTIRQ4	Fault indicator for a detected 1-bit or 2-bit ECC or Parity error in the RAMs 3	3	—	—
		COM	nFAULTIRQ0	Fault indicator for a detected 1-bit or 2-bit ECC or Parity error in the RAMs DSU	4	—	—
		0	nERRIRQ1	Error indicator for an ECC error that causes potential data corruption or loss of coherency 0	5	—	—
		1	nERRIRQ2	Error indicator for an ECC error that causes potential data corruption or loss of coherency 1	6	—	—
		2	nERRIRQ3	Error indicator for an ECC error that causes potential data corruption or loss of coherency 2	7	—	—
		3	nERRIRQ4	Error indicator for an ECC error that causes potential data corruption or loss of coherency 3	8	—	—

Table 12.24 CPU error event table (2 of 4)

Module	UNIT	Core	Name	Interrupt sources	Bit of CA55ERR_XXX	Bit of CR520ERR_XXX	Bit of CR521ERR_XXX
Cortex-A55	0	COM	nERRIRQ0	Error indicator for an ECC error that causes potential data corruption or loss of coherency DSU	9	—	—
GIC-600	0	—	fault_int	Fault handling interrupt.	10	—	—
			err_int	Error handling interrupt.	11	—	—
			pmu_int	PMU overflow length.	12	—	—
Cortex-R52	CPU0	—	ERREVENT0[0]	Correctable memory error detected from L1 instruction cache, L1 data cache, ATCM, BTCM, CTCM, or flash. The PMU mnemonic is KITE_CORE_ERR_MEM.	—	0	—
			ERREVENT0[1]	Fatal memory error occurred from ATCM, BTCM, CTCM, or flash. The PMU mnemonic is KITE_FAT_ERR_MEM.	—	1	—
			ERREVENT0[2]	Correctable data payload bus error occurred from AXIM or LLPP. The PMU mnemonic is KITE_BUS_COR_DATA.	—	2	—
			ERREVENT0[3]	Fatal data payload bus error occurred from AXIM or LLPP.	—	3	—
			ERREVENT0[4]	Fatal READY signal bus error occurred from AXIM, Flash, or LLPP.	—	4	—
			ERREVENT0[5]	Fatal non-protocol bus error occurred from AXIM, Flash, or LLPP. These are payload errors. The PMU mnemonic is KITE_BUS_FAT_ANY.	—	5	—
			ERREVENT0[6]	Fatal protocol bus error occurred from AXIM, Flash, or LLPP. These are LEN, ID, LAST, VALID and READY, and interconnect protection errors. The mnemonic is KITE_BUS_PROTOCOL_ANY.	—	6	—
			ERREVENT0[7]	Correctable L1 instruction cache data or L1 instruction cache tag memory error.	—	7	—
			ERREVENT0[8]	Correctable L1 data cache data or L1 data cache tag memory error.	—	8	—
			ERREVENT0[9]	Correctable ATCM, BTCM, or CTCM memory error.	—	9	—
			ERREVENT0[10]	Fatal ATCM, BTCM, or CTCM memory error.	—	10	—
			ERREVENT0[11]	Correctable flash memory error.	—	11	—
			ERREVENT0[12]	Fatal flash memory error.	—	12	—
			ERREVENT0[13]	Timeout for main AXIM bus.	—	13	—
			ERREVENT0[14]	Timeout for Flash bus.	—	14	—
			ERREVENT0[15]	Timeout for LLPP bus.	—	15	—
			ERREVENT0[16]	Correctable memory error occurred L1 instruction cache, L1 data cache, ATCM, BTCM, CTCM, or flash, but was not recorded in the Error Record Registers (ERRs), because they were full or because it was overwritten by a fatal error.	—	16	—
			ERREVENT0[17]	Fatal memory error occurred L1 instruction cache, L1 data cache, ATCM, BTCM, CTCM, or flash, but was not recorded in the ERRs, because they were full.	—	17	—

Table 12.24 CPU error event table (3 of 4)

Module	UNIT	Core	Name	Interrupt sources	Bit of CA55ERR_XXX	Bit of CR520ERR_XXX	Bit of CR521ERR_XXX
Cortex-R52	CPU0	—	ERREVENT0[18]	Interrupts masked in Hyp mode for too long. The IMP_INTMONR watchdog is triggered.	—	18	—
			ERREVENT0[19]	A memory or bus fatal error was synchronously detected in Hyp mode.	—	19	—
			ERREVENT0[20]	An abort exception was taken because an EL2-controlled MPU fault generated an abort.	—	20	—
			ERREVENT0[21]	An Abort exception was taken because an EL1-controlled MPU fault generated an abort.	—	21	—
			ERREVENT0[22]	An Undefined exception was taken due to any cause.	—	22	—
			ERREVENT0[23]	A memory access marked in EL1-controlled MPU as Device was flushed because it was marked as Normal in EL2-controlled MPU.	—	23	—
			ERREVENT0[24]	Processor livelock because of hard errors or exception at exception vector. An exception of the same type and vector offset was consecutively taken 20 times.	—	24	—
			ERREVENT0[25]	Software running in EL2 unlocks TESTR1. TESTR1 is for testing purposes only.	—	25	—
	CPU1	—	ERREVENT0[0]	Correctable memory error detected from L1 instruction cache, L1 data cache, ATCM, BTCM, CTCM, or flash. The PMU mnemonic is KITE_CORE_ERR_MEM.	—	—	0
			ERREVENT0[1]	Fatal memory error occurred from ATCM, BTCM, CTCM, or flash. The PMU mnemonic is KITE_FAT_ERR_MEM.	—	—	1
			ERREVENT0[2]	Correctable data payload bus error occurred from AXIM or LLPP. The PMU mnemonic is KITE_BUS_COR_DATA.	—	—	2
			ERREVENT0[3]	Fatal data payload bus error occurred from AXIM or LLPP.	—	—	3
			ERREVENT0[4]	Fatal READY signal bus error occurred from AXIM, Flash, or LLPP.	—	—	4
			ERREVENT0[5]	Fatal non-protocol bus error occurred from AXIM, Flash, or LLPP. These are payload errors. The PMU mnemonic is KITE_BUS_FAT_ANY.	—	—	5
			ERREVENT0[6]	Fatal protocol bus error occurred from AXIM, Flash, or LLPP. These are LEN, ID, LAST, VALID and READY, and interconnect protection errors. The mnemonic is KITE_BUS_PROTOCOL_ANY.	—	—	6
			ERREVENT0[7]	Correctable L1 instruction cache data or L1 instruction cache tag memory error.	—	—	7
			ERREVENT0[8]	Correctable L1 data cache data or L1 data cache tag memory error.	—	—	8
			ERREVENT0[9]	Correctable ATCM, BTCM, or CTCM memory error.	—	—	9
			ERREVENT0[10]	Fatal ATCM, BTCM, or CTCM memory error.	—	—	10
			ERREVENT0[11]	Correctable flash memory error.	—	—	11
			ERREVENT0[12]	Fatal flash memory error.	—	—	12
			ERREVENT0[13]	Timeout for main AXIM bus.	—	—	13

Table 12.24 CPU error event table (4 of 4)

Module	UNIT	Core	Name	Interrupt sources	Bit of CA55ERR_XXX	Bit of CR520ERR_XXX	Bit of CR521ERR_XXX
Cortex-R52	CPU1	—	ERREVENT0[14]	Timeout for Flash bus.	—	—	14
			ERREVENT0[15]	Timeout for LLPP bus.	—	—	15
			ERREVENT0[16]	Correctable memory error occurred L1 instruction cache, L1 data cache, ATCM, BTCM, CTCM, or flash, but was not recorded in the Error Record Registers (ERRs), because they were full or because it was overwritten by a fatal error.	—	—	16
			ERREVENT0[17]	Fatal memory error occurred L1 instruction cache, L1 data cache, ATCM, BTCM, CTCM, or flash, but was not recorded in the ERRs, because they were full.	—	—	17
			ERREVENT0[18]	Interrupts masked in Hyp mode for too long. The IMP_INTMONR watchdog is triggered.	—	—	18
			ERREVENT0[19]	A memory or bus fatal error was synchronously detected in Hyp mode.	—	—	19
			ERREVENT0[20]	An abort exception was taken because an EL2-controlled MPU fault generated an abort.	—	—	20
			ERREVENT0[21]	An Abort exception was taken because an EL1-controlled MPU fault generated an abort.	—	—	21
			ERREVENT0[22]	An Undefined exception was taken due to any cause.	—	—	22
			ERREVENT0[23]	A memory access marked in EL1-controlled MPU as Device was flushed because it was marked as Normal in EL2-controlled MPU.	—	—	23
			ERREVENT0[24]	Processor livelock because of hard errors or exception at exception vector. An exception of the same type and vector offset was consecutively taken 20 times.	—	—	24
			ERREVENT0[25]	Software running in EL2 unlocks TESTR1. TESTR1 is for testing purposes only.	—	—	25

Table 12.25 Peripheral error event table (except DSMIF and ENCIF error) (1 of 4)

Module	UNIT	CH	Name	Interrupt sources	Bit of PERIERR_XXXn		
					n = 0	n = 1	n = 2
CLMA	0	—	CLMA0_INT	CLMA0 error detection (PLL0)	0	—	—
	1	—	CLMA1_INT	CLMA1 error detection (PLL1)	1	—	—
	2	—	CLMA2_INT	CLMA2 error detection (PLL2)	2	—	—
	3	—	CLMA3_INT	CLMA3 error detection (PLL3)	3	—	—
	4	—	CLMA4_INT	CLMA4 error detection (PLL4)	4	—	—
	5	—	CLMA5_INT	CLMA5 error detection (LOCO)	5	—	—
	6	—	CLMA6_INT	CLMA6 error detection (MOSC)	6	—	—
BSC	0	—	BSC_WTO	External wait timeout detection interrupt	7	—	—
DMAC	0	COM	DMAC0_ERR	DMAC0 transfer error	8	—	—
	1	COM	DMAC1_ERR	DMAC1 transfer error	9	—	—
	2	COM	DMAC2_ERR	DMAC2 transfer error	10	—	—
WDT	0	—	WDT_NMIUNDFR0	Down-counter underflow Refresh error for CR52 CPU0	11	—	—

Table 12.25 Peripheral error event table (except DSMIF and ENCIF error) (2 of 4)

Module	UNIT	CH	Name	Interrupt sources	Bit of PERIERR_XXXn		
					n = 0	n = 1	n = 2
WDT	1	—	WDT_NMIUNDFR1	Down-counter underflow Refresh error for CR52 CPU1	12	—	—
	2	—	WDT_NMIUNDFA0	Down-counter underflow Refresh error for CA55 CORE0	13	—	—
	3	—	WDT_NMIUNDFA1	Down-counter underflow Refresh error for CA55 CORE1	14	—	—
	4	—	WDT_NMIUNDFA2	Down-counter underflow Refresh error for CA55 CORE2	15	—	—
	5	—	WDT_NMIUNDFA3	Down-counter underflow Refresh error for CA55 CORE3	16	—	—
USBf	0	—	USB_FDMAERR	USB (Function) DMA error	17	—	—
DOC	0	—	DOC_DOPCI	DOC interrupt	18	—	—
SYSRAM	0	—	SRAM0_IE1	System SRAM0 ECC 1 bit error	19	—	—
			SRAM0_IE2	System SRAM0 ECC 2 bits error	20	—	—
			SRAM0_OVF	System SRAM0 ECC error address capture overflow	21	—	—
	1	—	SRAM1_IE1	System SRAM1 ECC 1 bit error	22	—	—
			SRAM1_IE2	System SRAM1 ECC 2 bits error	23	—	—
			SRAM1_OVF	System SRAM1 ECC error address capture overflow	24	—	—
	2	—	SRAM2_IE1	System SRAM2 ECC 1 bit error	25	—	—
			SRAM2_IE2	System SRAM2 ECC 2 bits error	26	—	—
			SRAM2_OVF	System SRAM2 ECC error address capture overflow	27	—	—
	3	—	SRAM3_IE1	System SRAM3 ECC 1 bit error	28	—	—
			SRAM3_IE2	System SRAM3 ECC 2 bits error	29	—	—
			SRAM3_OVF	System SRAM3 ECC error address capture overflow	30	—	—
BUS	—	—	MPU_DMACR0	Master MPU Read channel for DMAC0 error	—	0	—
			MPU_DMACW0	Master MPU Write channel for DMAC0 error	—	1	—
			MPU_DMACR1	Master MPU Read channel for DMAC1 error	—	2	—
			MPU_DMACW1	Master MPU Write channel for DMAC1 error	—	3	—
			MPU_DMACR2	Master MPU Read channel for DMAC2 error	—	4	—
			MPU_DMACW2	Master MPU Write channel for DMAC2 error	—	5	—
			MPU_GMACR0	Master MPU Read channel for GMAC0 error	—	6	—
			MPU_GMACW0	Master MPU Write channel for GMAC0 error	—	7	—
			MPU_GMACR1	Master MPU Read channel for GMAC1 error	—	8	—
			MPU_GMACW1	Master MPU Write channel for GMAC1 error	—	9	—
			MPU_GMACR2	Master MPU Read channel for GMAC2 error	—	10	—
			MPU_GMACW2	Master MPU Write channel for GMAC2 error	—	11	—
			MPU_USBH	Master MPU for USB Host error	—	12	—
			MPU_USBF	Master MPU for USB Function error	—	13	—
			MPU_DBGR	Master MPU Read channel for AXI-AP error	—	14	—
			MPU_DBGW	Master MPU Write channel for AXI-AP error	—	15	—
			MPU_ETRR	Master MPU Read channel for ETR error	—	16	—
			MPU_ETRW	Master MPU Write channel for ETR error	—	17	—

Table 12.25 Peripheral error event table (except DSMIF and ENCIF error) (3 of 4)

Module	UNIT	CH	Name	Interrupt sources	Bit of PERIERR_XXXn		
					n = 0	n = 1	n = 2
BUS	—	—	MPU_SHOSTIF	Master MPU for Serial Host I/F error	—	18	—
			MPU_LCDCR	Master MPU Read channel for LCDC error	—	19	—
			MPU_LCDCW	Master MPU Write channel for LCDC error	—	20	—
			MPU_SDHIR0	Master MPU Read channel for SDHI0 error	—	21	—
			MPU_SDHIW0	Master MPU Write channel for SDHI0 error	—	22	—
			MPU_SDHIR1	Master MPU Read channel for SDHI1 error	—	23	—
			MPU_SDHIW1	Master MPU Write channel for SDHI1 error	—	24	—
			MPU_PCIER0	Master MPU Read channel for PCIE0 error	—	25	—
			MPU_PCIEW0	Master MPU Write channel for PCIE0 error	—	26	—
			MPU_PCIER1	Master MPU Read channel for PCIE1 error	—	27	—
			MPU_PCIEW1	Master MPU Write channel for PCIE1 error	—	28	—
			CA55MPUR	Cortex-A55 read error to Master MPU registers	—	29	—
			CA55MPUW	Cortex-A55 write error to Master MPU registers	—	30	—
			CR521MPUR	Cortex-R52 CPU1 read error to Master MPU registers	—	—	0
			CR521MPUW	Cortex-R52 CPU1 write error to Master MPU registers	—	—	1
			MAINBUSA_ERRINT	Internal main bus A error	—	—	2
			MAINBUSR_ERRINT	Internal main bus R error	—	—	3
			LLPPBUS_ERRINT	LLPP bus error	—	—	4
			TZC_DDR_A0A1	TZC-400-0 for DDR A0 and A1 I/F error	—	—	5
			TZC_DDR_A4	TZC-400-1 for DDR A4 I/F error	—	—	6
			TZC_PCIE	TZC-400 for PCIE error	—	—	7
			TZC_DDR_R2	TZC-400-3 for DDR R2 I/F error	—	—	8
			TZC_DDR_R3	TZC-400-4 for DDR R3 I/F error	—	—	9
			TZC_SYSRAM	TZC-400 for SYSRAM error	—	—	10
			TZC_BSC	TZC-400 for BSC error	—	—	11
			TZC_XSPI	TZC-400 for XSPI error	—	—	12
			TZC_TCM	TZC-400 for TCM error	—	—	13
			MAINBUSA_SEC_ERRINT	Internal main bus A error (TZM and TZS)	—	—	14
			MAINBUSR_SEC_ERRINT	Internal main bus R error (TZM and TZS)	—	—	15
			PERI_SEC_ERRINT	Pericore error (TZS)	—	—	16
PCIE	0	0	PCIE0_INT_SERR_RC	PCIE0 Error detect interrupt for Root complex	—	—	17
			PCIE0_INT_SERR_COR_RC	PCIE0 Correctable error detect interrupt for Root complex	—	—	18
			PCIE0_INT_NON_FATAL_RC	PCIE0 Non fatal error detect interrupt for Root complex	—	—	19
			PCIE0_INT_SERR_FATL_RC	PCIE0 Fatal error detect for Root complex	—	—	20
			PCIE0_INT_AXI_ERR	PCIE0 Error detect interrupt of AXI	—	—	21
		1	PCIE1_INT_SERR_RC	PCIE1 Error detect interrupt for Root complex	—	—	22
			PCIE1_INT_SERR_COR_RC	PCIE1 Correctable error detect interrupt for Root complex	—	—	23
			PCIE1_INT_NON_FATAL_RC	PCIE1 Non fatal error detect interrupt for Root complex	—	—	24

Table 12.25 Peripheral error event table (except DSMIF and ENCIF error) (4 of 4)

Module	UNIT	CH	Name	Interrupt sources	Bit of PERIERR_XXXn		
					n = 0	n = 1	n = 2
PCIE	0	1	PCIE1_INT_SERR_FATL_RC	PCIE1 Fatal error detect for Root complex	—	—	25
			PCIE1_INT_AXI_ERR	PCIE1 Error detect interrupt of AXI	—	—	26
DDRSS	0	—	perf_mon_data_status0	Performance monitor signal 0	—	—	27
			perf_mon_data_status1	Performance monitor signal 1	—	—	28

Table 12.26 DSMIF error event table (1 of 3)

Module	UNIT	CH	Name	Interrupt sources	Bit of DSMIFERR_XXXn		
					n = m	n = 10	n = 11
DSMIF	0 to 9	0	DSMIFm_OVC0EH0	DSMIFm Overcurrent Upper limit Detection 0 CH0	0	—	—
		1	DSMIFm_OVC1EL0	DSMIFm Overcurrent Lower limit Detection 1 CH0	1	—	—
			DSMIFm_OVC1EH0	DSMIFm Overcurrent Upper limit Detection 1 CH0	2	—	—
		2	DSMIFm_OVC2EL0	DSMIFm Overcurrent Lower limit Detection 2 CH0	3	—	—
			DSMIFm_OVC2EH0	DSMIFm Overcurrent Upper limit Detection 2 CH0	4	—	—
		0	DSMIFm_OVC0EL1	DSMIFm Overcurrent Lower limit Detection 0 CH1	5	—	—
			DSMIFm_OVC0EH1	DSMIFm Overcurrent Upper limit Detection 0 CH1	6	—	—
		1	DSMIFm_OVC1EL1	DSMIFm Overcurrent Lower limit Detection 1 CH1	7	—	—
			DSMIFm_OVC1EH1	DSMIFm Overcurrent Upper limit Detection 1 CH1	8	—	—
		2	DSMIFm_OVC2EL1	DSMIFm Overcurrent Lower limit Detection 2 CH1	9	—	—
			DSMIFm_OVC2EH1	DSMIFm Overcurrent Upper limit Detection 2 CH1	10	—	—
		0	DSMIFm_OVC0EL2	DSMIFm Overcurrent Lower limit Detection 0 CH2	11	—	—
			DSMIFm_OVC0EH2	DSMIFm Overcurrent Upper limit Detection 0 CH2	12	—	—
		1	DSMIFm_OVC1EL2	DSMIFm Overcurrent Lower limit Detection 1 CH2	13	—	—
			DSMIFm_OVC1EH2	DSMIFm Overcurrent Upper limit Detection 1 CH2	14	—	—
		2	DSMIFm_OVC2EL2	DSMIFm Overcurrent Lower limit Detection 2 CH2	15	—	—
			DSMIFm_OVC2EH2	DSMIFm Overcurrent Upper limit Detection 2 CH2	16	—	—
		0	DSMIFm_OVC0WN0	DSMIFm Overcurrent Detection Window Notification 0 CH0	17	—	—
			DSMIFm_OVC1WN0	DSMIFm Overcurrent Detection Window Notification 1 CH0	18	—	—
			DSMIFm_OVC2WN0	DSMIFm Overcurrent Detection Window Notification 2 CH0	19	—	—
			DSMIFm_OVC3WN0	DSMIFm Overcurrent Detection Window Notification 3 CH0	20	—	—
		1	DSMIFm_OVC0WN1	DSMIFm Overcurrent Detection Window Notification 0 CH1	21	—	—
			DSMIFm_OVC1WN1	DSMIFm Overcurrent Detection Window Notification 1 CH1	22	—	—
			DSMIFm_OVC2WN1	DSMIFm Overcurrent Detection Window Notification 2 CH1	23	—	—
			DSMIFm_OVC3WN1	DSMIFm Overcurrent Detection Window Notification 3 CH1	24	—	—
		2	DSMIFm_OVC0WN2	DSMIFm Overcurrent Detection Window Notification 0 CH2	25	—	—
			DSMIFm_OVC1WN2	DSMIFm Overcurrent Detection Window Notification 1 CH2	26	—	—
			DSMIFm_OVC2WN2	DSMIFm Overcurrent Detection Window Notification 2 CH2	27	—	—
			DSMIFm_OVC3WN2	DSMIFm Overcurrent Detection Window Notification 3 CH2	28	—	—
	0	COM	DSMIF0_SUMERRL	DSMIF0 Overcurrent Sum Error Lower limit detection	—	0	—
		COM	DSMIF0_SUMERRH	DSMIF0 Overcurrent Sum Error Upper limit detection	—	1	—

Table 12.26 DSMIF error event table (2 of 3)

Module	UNIT	CH	Name	Interrupt sources	Bit of DSMIFERR_XXXn		
					n = m	n = 10	n = 11
DSMIF	0	0	DSMIF0_SCE0	DSMIF0 Short Circuit Detection CH0	—	2	—
		1	DSMIF0_SCE1	DSMIF0 Short Circuit Detection CH1	—	3	—
		2	DSMIF0_SCE2	DSMIF0 Short Circuit Detection CH2	—	4	—
		0	DSMIF0_OVC0EL0	DSMIF0 Overcurrent Lower limit Detection 0 CH0	—	5	—
	1	COM	DSMIF1_SUMERRL	DSMIF1 Overcurrent Sum Error Lower limit detection	—	6	—
		COM	DSMIF1_SUMERRH	DSMIF1 Overcurrent Sum Error Upper limit detection	—	7	—
		0	DSMIF1_SCE0	DSMIF1 Short Circuit Detection CH0	—	8	—
		1	DSMIF1_SCE1	DSMIF1 Short Circuit Detection CH1	—	9	—
		2	DSMIF1_SCE2	DSMIF1 Short Circuit Detection CH2	—	10	—
		0	DSMIF1_OVC0EL0	DSMIF1 Overcurrent Lower limit Detection 0 CH0	—	11	—
	2	COM	DSMIF2_SUMERRL	DSMIF2 Overcurrent Sum Error Lower limit detection	—	12	—
		COM	DSMIF2_SUMERRH	DSMIF2 Overcurrent Sum Error Upper limit detection	—	13	—
		0	DSMIF2_SCE0	DSMIF2 Short Circuit Detection CH0	—	14	—
		1	DSMIF2_SCE1	DSMIF2 Short Circuit Detection CH1	—	15	—
		2	DSMIF2_SCE2	DSMIF2 Short Circuit Detection CH2	—	16	—
		0	DSMIF2_OVC0EL0	DSMIF2 Overcurrent Lower limit Detection 0 CH0	—	17	—
	3	COM	DSMIF3_SUMERRL	DSMIF3 Overcurrent Sum Error Lower limit detection	—	18	—
		COM	DSMIF3_SUMERRH	DSMIF3 Overcurrent Sum Error Upper limit detection	—	19	—
		0	DSMIF3_SCE0	DSMIF3 Short Circuit Detection CH0	—	20	—
		1	DSMIF3_SCE1	DSMIF3 Short Circuit Detection CH1	—	21	—
		2	DSMIF3_SCE2	DSMIF3 Short Circuit Detection CH2	—	22	—
		0	DSMIF3_OVC0EL0	DSMIF3 Overcurrent Lower limit Detection 0 CH0	—	23	—
	4	COM	DSMIF4_SUMERRL	DSMIF4 Overcurrent Sum Error Lower limit detection	—	24	—
		COM	DSMIF4_SUMERRH	DSMIF4 Overcurrent Sum Error Upper limit detection	—	25	—
		0	DSMIF4_SCE0	DSMIF4 Short Circuit Detection CH0	—	26	—
		1	DSMIF4_SCE1	DSMIF4 Short Circuit Detection CH1	—	27	—
		2	DSMIF4_SCE2	DSMIF4 Short Circuit Detection CH2	—	28	—
		0	DSMIF4_OVC0EL0	DSMIF4 Overcurrent Lower limit Detection 0 CH0	—	29	—
	5	COM	DSMIF5_SUMERRL	DSMIF5 Overcurrent Sum Error Lower limit detection	—	—	0
		COM	DSMIF5_SUMERRH	DSMIF5 Overcurrent Sum Error Upper limit detection	—	—	1
		0	DSMIF5_SCE0	DSMIF5 Short Circuit Detection CH0	—	—	2
		1	DSMIF5_SCE1	DSMIF5 Short Circuit Detection CH1	—	—	3
		2	DSMIF5_SCE2	DSMIF5 Short Circuit Detection CH2	—	—	4
		0	DSMIF5_OVC0EL0	DSMIF5 Overcurrent Lower limit Detection 0 CH0	—	—	5
	6	COM	DSMIF6_SUMERRL	DSMIF6 Overcurrent Sum Error Lower limit detection	—	—	6
		COM	DSMIF6_SUMERRH	DSMIF6 Overcurrent Sum Error Upper limit detection	—	—	7
		0	DSMIF6_SCE0	DSMIF6 Short Circuit Detection CH0	—	—	8
		1	DSMIF6_SCE1	DSMIF6 Short Circuit Detection CH1	—	—	9
		2	DSMIF6_SCE2	DSMIF6 Short Circuit Detection CH2	—	—	10
		0	DSMIF6_OVC0EL0	DSMIF6 Overcurrent Lower limit Detection 0 CH0	—	—	11

Table 12.26 DSMIF error event table (3 of 3)

Module	UNIT	CH	Name	Interrupt sources	Bit of DSMIFERR_XXXn		
					n = m	n = 10	n = 11
DSMIF	7	COM	DSMIF7_SUMERRL	DSMIF7 Overcurrent Sum Error Lower limit detection	—	—	12
		COM	DSMIF7_SUMERRH	DSMIF7 Overcurrent Sum Error Upper limit detection	—	—	13
		0	DSMIF7_SCE0	DSMIF7 Short Circuit Detection CH0	—	—	14
		1	DSMIF7_SCE1	DSMIF7 Short Circuit Detection CH1	—	—	15
		2	DSMIF7_SCE2	DSMIF7 Short Circuit Detection CH2	—	—	16
		0	DSMIF7_OVC0EL0	DSMIF7 Overcurrent Lower limit Detection 0 CH0	—	—	17
	8	COM	DSMIF8_SUMERRL	DSMIF8 Overcurrent Sum Error Lower limit detection	—	—	18
		COM	DSMIF8_SUMERRH	DSMIF8 Overcurrent Sum Error Upper limit detection	—	—	19
		0	DSMIF8_SCE0	DSMIF8 Short Circuit Detection CH0	—	—	20
		1	DSMIF8_SCE1	DSMIF8 Short Circuit Detection CH1	—	—	21
		2	DSMIF8_SCE2	DSMIF8 Short Circuit Detection CH2	—	—	22
		0	DSMIF8_OVC0EL0	DSMIF8 Overcurrent Lower limit Detection 0 CH0	—	—	23
	9	COM	DSMIF9_SUMERRL	DSMIF9 Overcurrent Sum Error Lower limit detection	—	—	24
		COM	DSMIF9_SUMERRH	DSMIF9 Overcurrent Sum Error Upper limit detection	—	—	25
		0	DSMIF9_SCE0	DSMIF9 Short Circuit Detection CH0	—	—	26
		1	DSMIF9_SCE1	DSMIF9 Short Circuit Detection CH1	—	—	27
		2	DSMIF9_SCE2	DSMIF9 Short Circuit Detection CH2	—	—	28
		0	DSMIF9_OVC0EL0	DSMIF9 Overcurrent Lower limit Detection 0 CH0	—	—	29

Table 12.27 ENCIF error event table (1 of 5)

Module	UNIT	CH	Name	Interrupt sources	Bit of ENCIFERR_XXXn				
					n = 0	n = 1	n = 2	n = 3	n = 4
BISS	0	—	BISS00_NER	BISS00 Error interrupt	0	—	—	—	—
	1	—	BISS01_NER	BISS01 Error interrupt	1	—	—	—	—
	2	—	BISS02_NER	BISS02 Error interrupt	2	—	—	—	—
	3	—	BISS03_NER	BISS03 Error interrupt	3	—	—	—	—
	4	—	BISS04_NER	BISS04 Error interrupt	4	—	—	—	—
	5	—	BISS05_NER	BISS05 Error interrupt	5	—	—	—	—
	6	—	BISS06_NER	BISS06 Error interrupt	6	—	—	—	—
	7	—	BISS07_NER	BISS07 Error interrupt	7	—	—	—	—
	8	—	BISS08_NER	BISS08 Error interrupt	8	—	—	—	—
	9	—	BISS09_NER	BISS09 Error interrupt	9	—	—	—	—
	10	—	BISS10_NER	BISS10 Error interrupt	10	—	—	—	—
	11	—	BISS11_NER	BISS11 Error interrupt	11	—	—	—	—
	12	—	BISS12_NER	BISS12 Error interrupt	12	—	—	—	—
	13	—	BISS13_NER	BISS13 Error interrupt	13	—	—	—	—
	14	—	BISS14_NER	BISS14 Error interrupt	14	—	—	—	—
	15	—	BISS15_NER	BISS15 Error interrupt	15	—	—	—	—

Table 12.27 ENCIF error event table (2 of 5)

Module	UNIT	CH	Name	Interrupt sources	Bit of ENCIFERR_XXXn				
					n = 0	n = 1	n = 2	n = 3	n = 4
AFMT	0	—	AFMT00_TMOUT	AFMT00 Time out error	16	—	—	—	—
	1	—	AFMT01_TMOUT	AFMT01 Time out error	17	—	—	—	—
	2	—	AFMT02_TMOUT	AFMT02 Time out error	18	—	—	—	—
	3	—	AFMT03_TMOUT	AFMT03 Time out error	19	—	—	—	—
	4	—	AFMT04_TMOUT	AFMT04 Time out error	20	—	—	—	—
	5	—	AFMT05_TMOUT	AFMT05 Time out error	21	—	—	—	—
	6	—	AFMT06_TMOUT	AFMT06 Time out error	22	—	—	—	—
	7	—	AFMT07_TMOUT	AFMT07 Time out error	23	—	—	—	—
	8	—	AFMT08_TMOUT	AFMT08 Time out error	24	—	—	—	—
	9	—	AFMT09_TMOUT	AFMT09 Time out error	25	—	—	—	—
	10	—	AFMT10_TMOUT	AFMT10 Time out error	26	—	—	—	—
	11	—	AFMT11_TMOUT	AFMT11 Time out error	27	—	—	—	—
	12	—	AFMT12_TMOUT	AFMT12 Time out error	28	—	—	—	—
	13	—	AFMT13_TMOUT	AFMT13 Time out error	29	—	—	—	—
	14	—	AFMT14_TMOUT	AFMT14 Time out error	30	—	—	—	—
	15	—	AFMT15_TMOUT	AFMT15 Time out error	31	—	—	—	—
HDSL	0	—	HDSL00_estimator_on	HDSL00 Position Estimator activated	—	0	—	—	—
			HDSL00_safe_channel_err	HDSL00 Transmission error in safe channel 1	—	1	—	—	—
			HDSL00_safe_pos_err	HDSL00 Safe position not valid	—	2	—	—	—
			HDSL00_acceleration_err	HDSL00 Fast channel / position error	—	3	—	—	—
			HDSL00_acc_thr_err	HDSL00 Fast channel / position threshold error	—	4	—	—	—
			HDSL00_encoding_err	HDSL00 DSL message encoding error	—	5	—	—	—
			HDSL00_dev_thr_err	HDSL00 Estimator deviation threshold reached	—	6	—	—	—
	1	—	HDSL01_estimator_on	HDSL01 Position Estimator activated	—	8	—	—	—
			HDSL01_safe_channel_err	HDSL01 Transmission error in safe channel 1	—	9	—	—	—
			HDSL01_safe_pos_err	HDSL01 Safe position not valid	—	10	—	—	—
			HDSL01_acceleration_err	HDSL01 Fast channel / position error	—	11	—	—	—
			HDSL01_acc_thr_err	HDSL01 Fast channel / position threshold error	—	12	—	—	—
			HDSL01_encoding_err	HDSL01 DSL message encoding error	—	13	—	—	—
			HDSL01_dev_thr_err	HDSL01 Estimator deviation threshold reached	—	14	—	—	—
	2	—	HDSL02_estimator_on	HDSL02 Position Estimator activated	—	16	—	—	—
			HDSL02_safe_channel_err	HDSL02 Transmission error in safe channel 1	—	17	—	—	—
			HDSL02_safe_pos_err	HDSL02 Safe position not valid	—	18	—	—	—
			HDSL02_acceleration_err	HDSL02 Fast channel / position error	—	19	—	—	—
			HDSL02_acc_thr_err	HDSL02 Fast channel / position threshold error	—	20	—	—	—
			HDSL02_encoding_err	HDSL02 DSL message encoding error	—	21	—	—	—
			HDSL02_dev_thr_err	HDSL02 Estimator deviation threshold reached	—	22	—	—	—

Table 12.27 ENCIF error event table (3 of 5)

Module	UNIT	CH	Name	Interrupt sources	Bit of ENCIFERR_XXXn				
					n = 0	n = 1	n = 2	n = 3	n = 4
HDSL	3	—	HDSL03_estimator_on	HDSL03 Position Estimator activated	—	24	—	—	—
			HDSL03_safe_channel_err	HDSL03 Transmission error in safe channel 1	—	25	—	—	—
			HDSL03_safe_pos_err	HDSL03 Safe position not valid	—	26	—	—	—
			HDSL03_acceleration_err	HDSL03 Fast channel / position error	—	27	—	—	—
			HDSL03_acc_thr_err	HDSL03 Fast channel / position threshold error	—	28	—	—	—
			HDSL03_encoding_err	HDSL03 DSL message encoding error	—	29	—	—	—
			HDSL03_dev_thr_err	HDSL03 Estimator deviation threshold reached	—	30	—	—	—
	4	—	HDSL04_estimator_on	HDSL04 Position Estimator activated	—	—	0	—	—
			HDSL04_safe_channel_err	HDSL04 Transmission error in safe channel 1	—	—	1	—	—
			HDSL04_safe_pos_err	HDSL04 Safe position not valid	—	—	2	—	—
			HDSL04_acceleration_err	HDSL04 Fast channel / position error	—	—	3	—	—
			HDSL04_acc_thr_err	HDSL04 Fast channel / position threshold error	—	—	4	—	—
			HDSL04_encoding_err	HDSL04 DSL message encoding error	—	—	5	—	—
			HDSL04_dev_thr_err	HDSL04 Estimator deviation threshold reached	—	—	6	—	—
	5	—	HDSL05_estimator_on	HDSL05 Position Estimator activated	—	—	8	—	—
			HDSL05_safe_channel_err	HDSL05 Transmission error in safe channel 1	—	—	9	—	—
			HDSL05_safe_pos_err	HDSL05 Safe position not valid	—	—	10	—	—
			HDSL05_acceleration_err	HDSL05 Fast channel / position error	—	—	11	—	—
			HDSL05_acc_thr_err	HDSL05 Fast channel / position threshold error	—	—	12	—	—
			HDSL05_encoding_err	HDSL05 DSL message encoding error	—	—	13	—	—
			HDSL05_dev_thr_err	HDSL05 Estimator deviation threshold reached	—	—	14	—	—
	6	—	HDSL06_estimator_on	HDSL06 Position Estimator activated	—	—	16	—	—
			HDSL06_safe_channel_err	HDSL06 Transmission error in safe channel 1	—	—	17	—	—
			HDSL06_safe_pos_err	HDSL06 Safe position not valid	—	—	18	—	—
			HDSL06_acceleration_err	HDSL06 Fast channel / position error	—	—	19	—	—
			HDSL06_acc_thr_err	HDSL06 Fast channel / position threshold error	—	—	20	—	—
			HDSL06_encoding_err	HDSL06 DSL message encoding error	—	—	21	—	—
			HDSL06_dev_thr_err	HDSL06 Estimator deviation threshold reached	—	—	22	—	—
	7	—	HDSL07_estimator_on	HDSL07 Position Estimator activated	—	—	24	—	—
			HDSL07_safe_channel_err	HDSL07 Transmission error in safe channel 1	—	—	25	—	—
			HDSL07_safe_pos_err	HDSL07 Safe position not valid	—	—	26	—	—
			HDSL07_acceleration_err	HDSL07 Fast channel / position error	—	—	27	—	—
			HDSL07_acc_thr_err	HDSL07 Fast channel / position threshold error	—	—	28	—	—
			HDSL07_encoding_err	HDSL07 DSL message encoding error	—	—	29	—	—
			HDSL07_dev_thr_err	HDSL07 Estimator deviation threshold reached	—	—	30	—	—

Table 12.27 ENCIF error event table (4 of 5)

Module	UNIT	CH	Name	Interrupt sources	Bit of ENCIFERR_XXXn				
					n = 0	n = 1	n = 2	n = 3	n = 4
HDSL	8	—	HDSL08_estimator_on	HDSL08 Position Estimator activated	—	—	—	0	—
			HDSL08_safe_channel_err	HDSL08 Transmission error in safe channel 1	—	—	—	1	—
			HDSL08_safe_pos_err	HDSL08 Safe position not valid	—	—	—	2	—
			HDSL08_acceleration_err	HDSL08 Fast channel / position error	—	—	—	3	—
			HDSL08_acc_thr_err	HDSL08 Fast channel / position threshold error	—	—	—	4	—
			HDSL08_encoding_err	HDSL08 DSL message encoding error	—	—	—	5	—
			HDSL08_dev_thr_err	HDSL08 Estimator deviation threshold reached	—	—	—	6	—
	9	—	HDSL09_estimator_on	HDSL09 Position Estimator activated	—	—	—	8	—
			HDSL09_safe_channel_err	HDSL09 Transmission error in safe channel 1	—	—	—	9	—
			HDSL09_safe_pos_err	HDSL09 Safe position not valid	—	—	—	10	—
			HDSL09_acceleration_err	HDSL09 Fast channel / position error	—	—	—	11	—
			HDSL09_acc_thr_err	HDSL09 Fast channel / position threshold error	—	—	—	12	—
			HDSL09_encoding_err	HDSL09 DSL message encoding error	—	—	—	13	—
			HDSL09_dev_thr_err	HDSL09 Estimator deviation threshold reached	—	—	—	14	—
	10	—	HDSL10_estimator_on	HDSL10 Position Estimator activated	—	—	—	16	—
			HDSL10_safe_channel_err	HDSL10 Transmission error in safe channel 1	—	—	—	17	—
			HDSL10_safe_pos_err	HDSL10 Safe position not valid	—	—	—	18	—
			HDSL10_acceleration_err	HDSL10 Fast channel / position error	—	—	—	19	—
			HDSL10_acc_thr_err	HDSL10 Fast channel / position threshold error	—	—	—	20	—
			HDSL10_encoding_err	HDSL10 DSL message encoding error	—	—	—	21	—
			HDSL10_dev_thr_err	HDSL10 Estimator deviation threshold reached	—	—	—	22	—
	11	—	HDSL11_estimator_on	HDSL11 Position Estimator activated	—	—	—	24	—
			HDSL11_safe_channel_err	HDSL11 Transmission error in safe channel 1	—	—	—	25	—
			HDSL11_safe_pos_err	HDSL11 Safe position not valid	—	—	—	26	—
			HDSL11_acceleration_err	HDSL11 Fast channel / position error	—	—	—	27	—
			HDSL11_acc_thr_err	HDSL11 Fast channel / position threshold error	—	—	—	28	—
			HDSL11_encoding_err	HDSL11 DSL message encoding error	—	—	—	29	—
			HDSL11_dev_thr_err	HDSL11 Estimator deviation threshold reached	—	—	—	30	—
	12	—	HDSL12_estimator_on	HDSL12 Position Estimator activated	—	—	—	—	0
			HDSL12_safe_channel_err	HDSL12 Transmission error in safe channel 1	—	—	—	—	1
			HDSL12_safe_pos_err	HDSL12 Safe position not valid	—	—	—	—	2
			HDSL12_acceleration_err	HDSL12 Fast channel / position error	—	—	—	—	3
			HDSL12_acc_thr_err	HDSL12 Fast channel / position threshold error	—	—	—	—	4
			HDSL12_encoding_err	HDSL12 DSL message encoding error	—	—	—	—	5
			HDSL12_dev_thr_err	HDSL12 Estimator deviation threshold reached	—	—	—	—	6

Table 12.27 ENCIF error event table (5 of 5)

Module	UNIT	CH	Name	Interrupt sources	Bit of ENCIFERR_XXXn				
					n = 0	n = 1	n = 2	n = 3	n = 4
HDSL	13	—	HDSL13_estimator_on	HDSL13 Position Estimator activated	—	—	—	—	8
			HDSL13_safe_channel_err	HDSL13 Transmission error in safe channel 1	—	—	—	—	9
			HDSL13_safe_pos_err	HDSL13 Safe position not valid	—	—	—	—	10
			HDSL13_acceleration_err	HDSL13 Fast channel / position error	—	—	—	—	11
			HDSL13_acc_thr_err	HDSL13 Fast channel / position threshold error	—	—	—	—	12
			HDSL13_encoding_err	HDSL13 DSL message encoding error	—	—	—	—	13
			HDSL13_dev_thr_err	HDSL13 Estimator deviation threshold reached	—	—	—	—	14
	14	—	HDSL14_estimator_on	HDSL14 Position Estimator activated	—	—	—	—	16
			HDSL14_safe_channel_err	HDSL14 Transmission error in safe channel 1	—	—	—	—	17
			HDSL14_safe_pos_err	HDSL14 Safe position not valid	—	—	—	—	18
			HDSL14_acceleration_err	HDSL14 Fast channel / position error	—	—	—	—	19
			HDSL14_acc_thr_err	HDSL14 Fast channel / position threshold error	—	—	—	—	20
			HDSL14_encoding_err	HDSL14 DSL message encoding error	—	—	—	—	21
			HDSL14_dev_thr_err	HDSL14 Estimator deviation threshold reached	—	—	—	—	22
	15	—	HDSL15_estimator_on	HDSL15 Position Estimator activated	—	—	—	—	24
			HDSL15_safe_channel_err	HDSL15 Transmission error in safe channel 1	—	—	—	—	25
			HDSL15_safe_pos_err	HDSL15 Safe position not valid	—	—	—	—	26
			HDSL15_acceleration_err	HDSL15 Fast channel / position error	—	—	—	—	27
			HDSL15_acc_thr_err	HDSL15 Fast channel / position threshold error	—	—	—	—	28
			HDSL15_encoding_err	HDSL15 DSL message encoding error	—	—	—	—	29
			HDSL15_dev_thr_err	HDSL15 Estimator deviation threshold reached	—	—	—	—	30

12.4.4 External Pin Interrupts

Selectable interrupt detection method and digital noise filter function are available in external pins like IRQ0 to IRQ15, SEI, and DREQ. Low level, rising edge, falling edge, or both edge detection methods are selectable by S_PORTNF_MD register and NS_PORTNF_MD register each pin.

Digital noise filter has the following features:

- Sampling clock cycle is selectable from 64 divisions, 32 divisions, 8 divisions, or 1 division for PCLKM by S_PORTNF_CLKSEL and NS_PORTNF_CLKSEL register.
- Pulses with levels that only match once or twice are removed.

Digital noise filters are desirable for all signals as initial value. Set S_PORTNF_FLTSEL and NS_PORTNF_FLTSEL when using digital noise filter.

Figure 12.5 shows an example of digital noise filter operation.

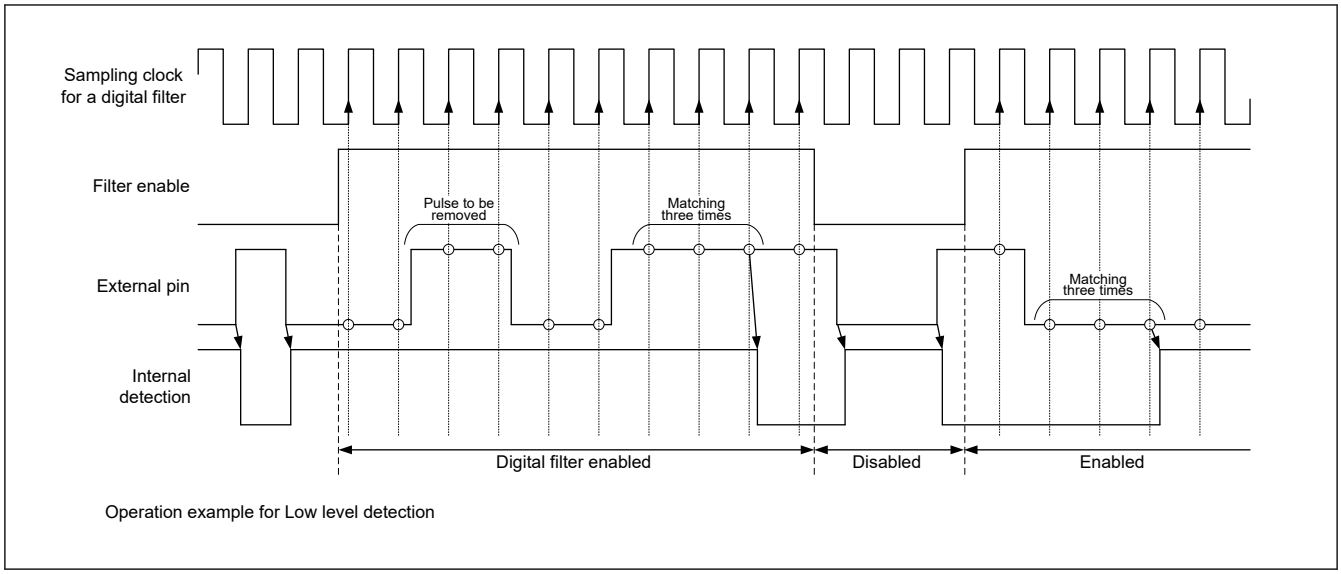


Figure 12.5 Digital noise filter operation example

13. Internal Buses

13.1 Overview

The LSI contains internal main bus A and R, low latency peripheral port bus, and multiple peripheral buses. Master MPUs are implemented for bus masters except the Cortex-R52 and Cortex-A55 so that access to the bus slaves can be controlled by the Cortex-R52 and Cortex-A55. Additionally, physical paths to the bus slaves are separated among some peripherals. It allows CPU access to avoid access conflict from bus masters except the Cortex-R52 and Cortex-A55 physically. TrustZone filters are implemented for bus masters and slaves. TrustZone Master (TZM) can control attribute of the transaction from the master. TrustZone Slave (TZS) can filter the transaction by security level. Table 13.1 lists the specifications of internal buses and Figure 13.1 shows the internal bus configuration.

Table 13.1 Internal buses specifications (1 of 2)

Internal bus type		Description
Internal main bus A	• Bus masters Cortex-A55 ACEM, GMAC (unit 1, 2), PCIE (unit 0, 1), USBHC, USBf, SDHI (unit 0, 1), LCDC, CoreSight ETR, internal main bus R (Cortex-R52 CPU0 and CPU1 AXIM, DMAC (unit 0 to 2), CoreSight AXI-AP)	• Operates in synchronization with PCLKAH • Bus protocol: AMBA AXI • Priority order decision: Round-robin
Internal main bus R	• Bus masters Cortex-A55 PP, Cortex-R52 CPU0 and CPU1 AXIM, DMAC (unit 0 to 2), GMAC (unit 0), SHOSTIF, CoreSight AXI-AP, internal main bus A (Cortex-A55 ACEM, GMAC (unit 1, 2), PCIE (unit 0, 1), USBHC, USBf, SDHI (unit 0, 1), CoreSight ETR)	• Operates in synchronization with PCLKH • Bus protocol: AMBA AXI • Priority order decision: Round-robin
Low latency peripheral port bus	• Bus masters Cortex-R52 CPU0 and CPU1 LLPP, internal main bus R (Cortex-A55 PP, DMAC (unit 0 to 2), PCIE (unit 0, 1), CoreSight AXI-AP)	• Operates in synchronization with PCLKH • Bus protocol: AMBA AXI • Priority order decision: Fixed priority LLPP0: Cortex-R52 CPU0 > Cortex-R52 CPU1 > internal main bus LLPP1: Cortex-R52 CPU1 > Cortex-R52 CPU0 > internal main bus
Peripheral bus R0	MTU3, POE3, GPT (unit 0 to 5), POEG (unit 0), TFU (unit 0), DSMIF (unit 0 to 5), ADC12 (unit 0, 1), ENCIFSS (ENDAT, BISS, HDSL, AFMT) (unit 0 to 11), ENCOUT, SCIE (unit 0 to 7)	• LLPP peripheral (LLPP0) • Operates in synchronization with PCLKH
Peripheral bus R1	GPT (unit 6 to 8), TFU (unit 1), DSMIF (unit 6 to 9), ADC12 (unit 0, 1), ENCIFSS (ENDAT, BISS, HDSL, AFMT) (unit 12 to 15), SCIE (unit 8 to 11)	• LLPP peripheral (LLPP1) • Operates in synchronization with PCLKH
Peripheral bus R2	GPT (unit 9), SCI (unit 0 to 4), SPI (unit 0 to 2), CRC (unit 0), CANFD	• Non-Safety peripheral (NONSAFETY) • Operates in synchronization with PCLKM
Peripheral bus R3	CMT, CMTW, WDT (unit 0 to 5), IIC (unit 0, 1), DOC, ADC12 (unit 2), TSU, POEG (unit 1)	• Non-Safety peripheral (NONSAFETY) • Operates in synchronization with PCLKL
Peripheral bus R4	GPT (unit 10), SCI (unit 5), SPI (unit 3), CRC (unit 1)	• Safety peripheral (SAFETY) • Operates in synchronization with PCLKM
Peripheral bus R5	IIC (unit 2), RTC, POEG (unit 2)	• Safety peripheral (SAFETY) • Operates in synchronization with PCLKL
Peripheral bus R6	DMAC (unit 0 to 2), GIC600	• Operates in synchronization with PCLKH
Peripheral bus R7	GMAC (unit 0), ETHSS, ETHSW, ESC	• Operates in synchronization with PCLKM
Cortex-R52 CPU0 AXIS	Cortex-R52 CPU0 ATCM, BTCM	• Operates in synchronization with PCLKH
Cortex-R52 CPU1 AXIS	Cortex-R52 CPU1 ATCM, BTCM	• Operates in synchronization with PCLKH
BSC	• External memory space • Related registers	• Operates in synchronization with PCLKH
xSPI	• External memory space • Related registers	• Operates in synchronization with PCLKH

Table 13.1 Internal buses specifications (2 of 2)

Internal bus type		Description
System SRAM	• System SRAM memory space • Related registers	• Operates in synchronization with PCLKH
System Control	ICU, ELC, clock controller, reset controller, master MPU, address expander, TZC-400	• Operates in synchronization with PCLKM
IO Port	Non-safety IO, Safety IO	• Operates in synchronization with PCLKM
USB	USBHC, USBf	• Operates in synchronization with PCLKAM
PCIE	• External memory space • Related registers	• Operates in synchronization with PCLKAH
DDR memory area	A0 I/F: For access from SDHI (unit 0, 1), LCDC A1 I/F: For access from GMAC (unit 1, 2), PCIE (unit 0, 1), USBHC, USBf, CoreSight ETR A4 I/F: For access from Cortex-A55 ACEM R2 I/F: For access from DMAC (unit 0 to 2), GMAC (unit 0), CoreSight AXI-AP R3 I/F: For access from Cortex-R52 CPU0 and CPU1 AXIM	• A0, A1, A4 I/F: Operates in synchronization with PCLKAH • R2, R3 I/F: Operates in synchronization with PCLKH
DDR Control		• Operates in synchronization with PCLKL
GMAC (unit 1, 2)		• Operates in synchronization with PCLKAM
SDHI		• Operates in synchronization with PCLKAM
LCDC		• Operates in synchronization with PCLKAL
SHOSTIF		• Operates in synchronization with PCLKH
MBXSEM		• Operates in synchronization with PCLKH
OTP		• Operates in synchronization with PCLKM
Security		• Operates in synchronization with PCLKH
CoreSight		• Operates in synchronization with PCLKH
Boot ROM		• Operates in synchronization with PCLKH

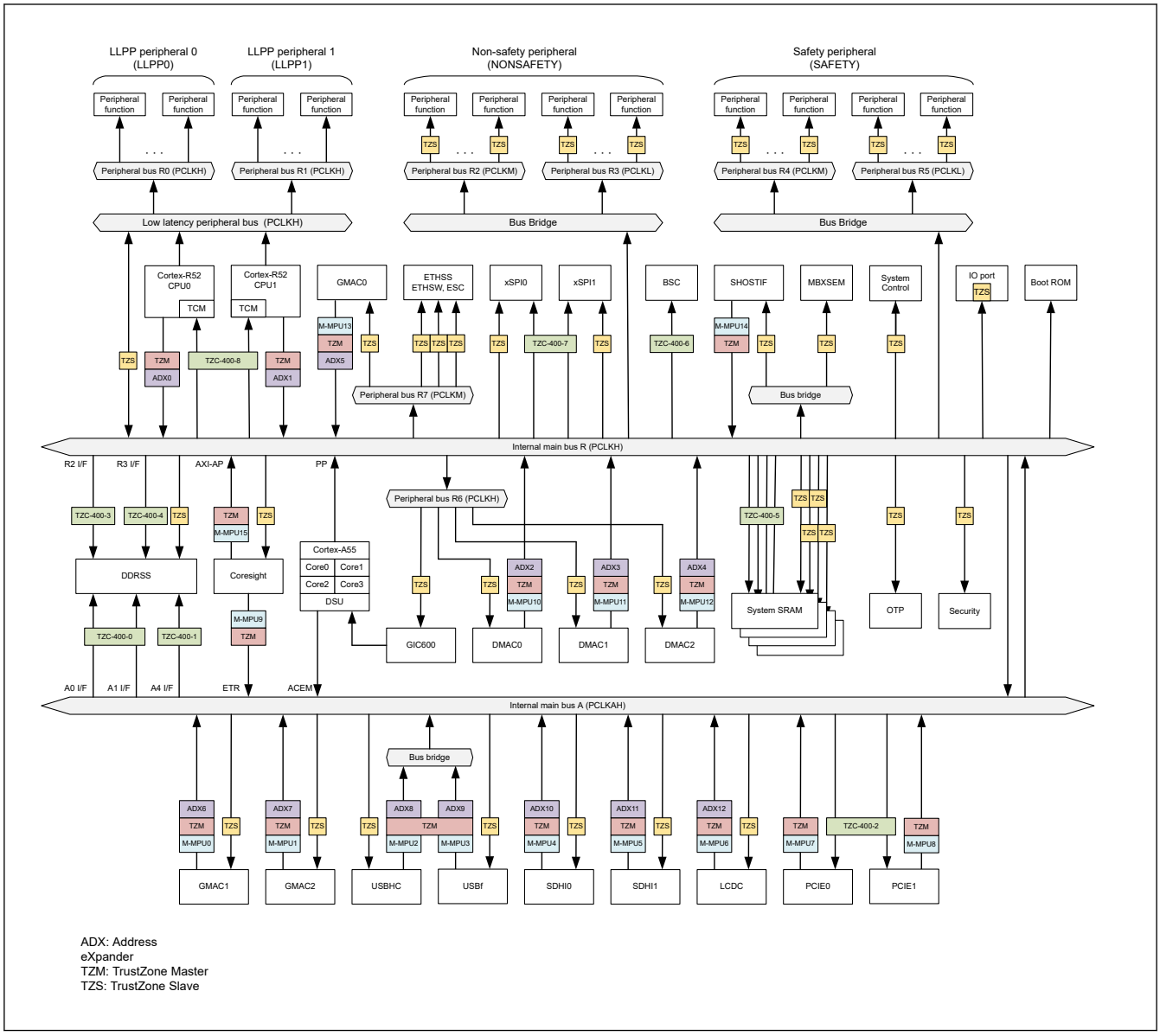


Figure 13.1 Bus configuration

13.2 Register Map

Table 13.2 Internal Bus register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x8129_1400 + 0x0100 × i + 0x08 × m (i = 0 to 9) 0x8129_0400 + 0x0100 × (i - 10) + 0x08 × m (i = 10 to 15)	STADDm	Master MPU Safety Region Start Address Register m (m = 0 to 15)	PRCRS.PRC3
0x8129_1404 + 0x0100 × i + 0x08 × m (i = 0 to 9) 0x8129_0404 + 0x0100 × (i - 10) + 0x08 × m (i = 10 to 15)	ENDADDm	Master MPU Safety Region End Address Register m (m = 0 to 15)	PRCRS.PRC3
0x8129_1480 + 0x0100 × i (i = 2, 3) 0x8129_0480 + 0x0100 × (i - 10) (i = 14)	ERRINF	Master MPU Error Information Register for AHB type	PRCRS.PRC3
ERRINF_R 0x8129_1480 + 0x0100 × i (i = 0, 1, 4 to 9) 0x8129_0480 + 0x0100 × (i - 10) (i = 10 to 13, and 15) ERRINF_W 0x8129_1484 + 0x0100 × i (i = 0, 1, 4 to 9) 0x8129_0484 + 0x0100 × (i - 10) (i = 10 to 13, and 15)	ERRINF_x	Master MPU Error Information Register for AXI type (x = R, W)	PRCRS.PRC3

Table 13.2 Internal Bus register map (2 of 2)

Address	Register symbol	Register name	Write protection
ERRINF1_R 0x8129_1488 + 0x0100 × i (i = 7 to 9) 0x8129_0488 + 0x0100 × (i - 10) (i = 15) ERRINF1_W 0x8129_148C + 0x0100 × i (i = 7 to 9) 0x8129_048C + 0x0100 × (i - 10) (i = 15)	ERRINF1_x	Master MPU Error Information Register for AXI type 1 (x = R, W)	—
0x8124_0000	CPU_CTRL	CPU Access Permission Control Register	—
0x8129_1200	SSTPCR0	Slave Stop Control Register 0	PRCRS.PRC3
0x8129_1204	SSTPCR1	Slave Stop Control Register 1	PRCRS.PRC3
0x8129_0200	SSTPCR4	Slave Stop Control Register 4	PRCRS.PRC3
0x8129_0204	SSTPCR5	Slave Stop Control Register 5	PRCRS.PRC3
0x8129_0208	SSTPCR6	Slave Stop Control Register 6	PRCRS.PRC3
0x8129_020C	SSTPCR7	Slave Stop Control Register 7	PRCRS.PRC3
0x8129_1300	MSTACCCTL0	Master Access Control Register 0	PRCRS.PRC3
0x8129_1304	MSTACCCTL1	Master Access Control Register 1	PRCRS.PRC3
0x8129_0300	MSTACCCTL2	Master Access Control Register 2	PRCRS.PRC3
0x8129_0304	MSTACCCTL3	Master Access Control Register 3	PRCRS.PRC3
0x8129_1310	SLVACCCTL0	Slave Access Control Register 0	PRCRS.PRC3
0x8129_1314	SLVACCCTL1	Slave Access Control Register 1	PRCRS.PRC3
0x8129_1318	SLVACCCTL2	Slave Access Control Register 2	PRCRS.PRC3
0x8129_1320	SLVACCCTL4	Slave Access Control Register 4	PRCRS.PRC3
0x8129_1324	SLVACCCTL5	Slave Access Control Register 5	PRCRS.PRC3
0x8129_0310	SLVACCCTL6	Slave Access Control Register 6	PRCRS.PRC3
0x8129_0314	SLVACCCTL7	Slave Access Control Register 7	PRCRS.PRC3
0x8129_0318	SLVACCCTL8	Slave Access Control Register 8	PRCRS.PRC3
0x8129_031C	SLVACCCTL9	Slave Access Control Register 9	PRCRS.PRC3
0x8129_0100 + 0x4 × n (n = 0 to 5) 0x8129_1100 + 0x4 × (n - 6) (n = 6 to 12)	ADXCTLn	Address Expander Control Register n (n = 0 to 12)	PRCRS.PRC3

Table 13.3 Internal Bus related system control register (1 of 2)

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
MPU0 registers (for GMAC1)	—	—	SLVACCCTL5.MPUGMAC1_SL
MPU1 registers (for GMAC2)	—	—	SLVACCCTL5.MPUGMAC2_SL
MPU2 registers (for USBHC)	—	—	SLVACCCTL5.MPUUSB_SL
MPU3 registers (for USBf)	—	—	SLVACCCTL5.MPUUSB_SL
MPU4 registers (for SDHI0)	—	—	SLVACCCTL5.MPUSDHI0_SL
MPU5 registers (for SDHI1)	—	—	SLVACCCTL5.MPUSDHI1_SL
MPU6 registers (for LCDC)	—	—	SLVACCCTL5.MPULCDC_SL
MPU7 registers (for PCIE0)	—	—	SLVACCCTL5.MPUPCIE0_SL
MPU8 registers (for PCIE1)	—	—	SLVACCCTL5.MPUPCIE1_SL
MPU9 registers (for CS ETR)	—	—	SLVACCCTL9.MPUCS_SL
MPU10 registers (for DMAC0)	—	—	SLVACCCTL9.MPUDMAC0_SL
MPU11 registers (for DMAC1)	—	—	SLVACCCTL9.MPUDMAC1_SL

Table 13.3 Internal Bus related system control register (2 of 2)

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
MPU12 registers (for DMAC2)	—	—	SLVACCCTL9.MPUDMAC2_SL
MPU13 registers (for GMAC0)	—	—	SLVACCCTL9.MPUGMAC0_SL
MPU14 registers (for SHOSTIF)	—	—	SLVACCCTL9.MPUSHOSTIF_SL
MPU15 registers (for CS AXI-AP)	—	—	SLVACCCTL9.MPUCS_SL
CPU_CTRL	—	—	Fixed to 2
SSTPRCRn (n = 0, 1, 4 to 7)	—	—	SLVACCCTL8.SYSCTRL_SL
MSTACCCTLn (n = 0 to 3)	—	—	Fixed to 3
SLVACCCTLn (n = 0 to 2, 4 to 9)	—	—	Fixed to 3
ADXCTLn (n = 0 to 12)	—	—	SLVACCCTL9.ADDEXP_SL

13.3 Register Descriptions

13.3.1 STADDm : Master MPU Safety Region Start Address Register m (m = 0 to 15)

Base address: MPUi = 0x8129_1400 + 0x0100 × i (i = 0 to 9)
MPUj = 0x8129_0400 + 0x0100 × (j - 10) (j = 10 to 15)

Offset address: 0x00 + 0x08 × m

Bit	Symbol	Function	R/W
0	RDPR	Enable read protection for region m of master MPU 0: Disable read protection. Read access to the safety region is permitted. 1: Enable read protection. Read access to the safety region is protected and error is generated.	R/W
1	WRPR	Enable write protection for region m of master MPU 0: Disable write protection. Write access to the safety region is permitted. 1: Enable write protection. Write access to the safety region is protected and error is generated.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	STADDR2[2:0]*2	Start address for MPU region Set upper 3 bits of the 35-bit start address of the safety region to be protected access. 35-bit address master (i = 7 to 9, and 15) only.	R/W
9:7	—	These bits are read as 0. The write value should be 0.	R/W
31:10	STADDR[21:0]	Start address for MPU region Set upper 22 bits of the 32-bit start address of the safety region to be protected access. Lower 10 bits of the 32-bit start address is set 0x000 automatically. For AXI type MPU (i = 0, 1, 4 to 13, and 15), STADDR[1:0] must be set 00b.	R/W

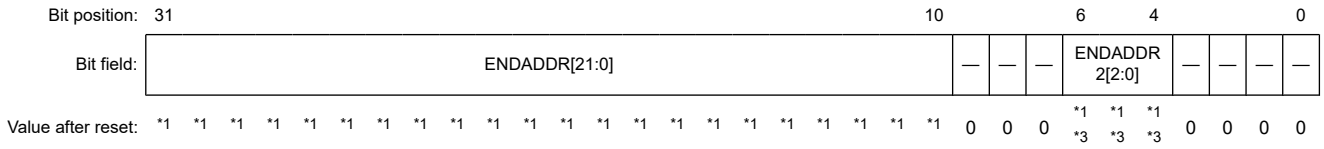
Note 1. Initial value depends on the number of m. The value is 1 for m = 0 and 0 for the others.
Note 2. i = 7 to 9, and 15 only. Reserved for the other.

The STADDm register is a 32-bit register that controls the start address of the safety region. Access from the master module to the safety region which is specified is protected by the master MPU. Up to 16 regions (m = 0 to 15) can be defined for each master MPU. Boundary and granularity of the start address to be set is 1 KB for AHB type MPU (i = 2, 3, and 14) and 4 KB for AXI type MPU (i = 0, 1, 4 to 13, and 15).

13.3.2 ENDADDm : Master MPU Safety Region End Address Register m (m = 0 to 15)

Base address: MPUi = 0x8129_1400 + 0x0100 × i (i = 0 to 9)
MPUj = 0x8129_0400 + 0x0100 × (j - 10) (j = 10 to 15)

Offset address: 0x04 + 0x08 × m



Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
6:4	ENDADDR2[2:0]*2	End address for MPU region Set upper 3 bits of the 35-bit end address of the safety region to be protected access. 35-bit address master (i = 7 to 9, and 15) only.	R/W
9:7	—	These bits are read as 0. The write value should be 0.	R/W
31:10	ENDADDR[21:0]	End address for MPU region Set upper 22 bits of the 32-bit end address of the safety region to be protected access. Lower 10 bits of the 32-bit end address is set 0x3FF automatically. For AXI type MPU (i = 0, 1, 4 to 13, and 15), ENDADDR[1:0] must be set 11b.	R/W

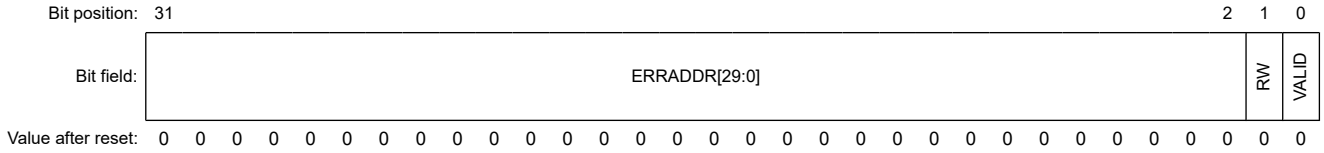
Note 1. Initial value depends on the number of m. The value is 1 for m = 0 and 0 for the others.
Note 2. The ENDADDR2[2:0] bits are available when the base address is MPU7 to MPU9 and MPU15. When the base address is other than MPU7 to MPU9 and MPU15, the ENDADDR2[2:0] bits are reserved.
Note 3. When the base address is MPU0 to MPU6 and MPU10 to MPU14, the initial value of the ENDADDR2[2:0] bits is 000b regardless of the number of m.

The ENDADDm register is a 32-bit register that controls the end address of the safety region. Access from the master module to the safety region which is specified is protected by the master MPU. Up to 16 regions (m = 0 to 15) can be defined for each master MPU. Boundary and granularity of the end address to be set is 1 KB for AHB type MPU (i = 2, 3, and 14) and 4 KB for AXI type MPU (i = 0, 1, 4 to 13, and 15).

13.3.3 ERRINF : Master MPU Error Information Register for AHB type

Base address: MPUi = 0x8129_1400 + 0x0100 × i (i = 2, 3)
MPU14 = 0x8129_0800

Offset address: 0x80



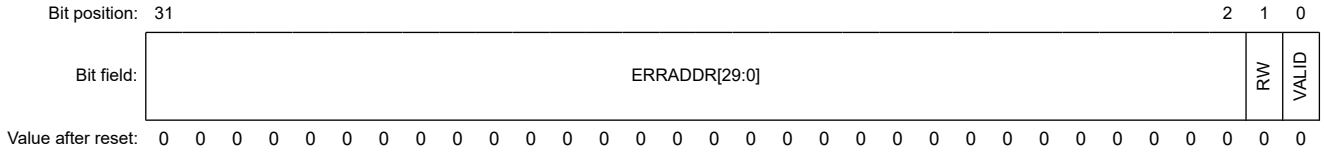
Bit	Symbol	Function	R/W
0	VALID	Validity of Access Error Information Cleared to 0 when 0 is written. 0: Error information is invalid 1: Error information is valid	R/W
1	RW	Access error type 0: Read access 1: Write access	R
31:2	ERRADDR[29:0]	Access Error Address Indicates upper 30 bits of the 32-bit address. Lower 2 bits are masked.	R

The ERRINF is a 32-bit register to store the safety region access error information by master module. AHB type MPUi (i = 2, 3, and 14) provides one error information for both read and write.

13.3.4 ERRINF_x : Master MPU Error Information Register for AXI type (x = R, W)

Base address: MPUi = 0x8129_1400 + 0x0100 × i (i = 0, 1, 4 to 9)
MPUj = 0x8129_0400 + 0x0100 × (j - 10) (j = 10 to 13)
MPU15 = 0x8129_0900

Offset address: 0x80 (ERRINF_R)
0x84 (ERRINF_W)



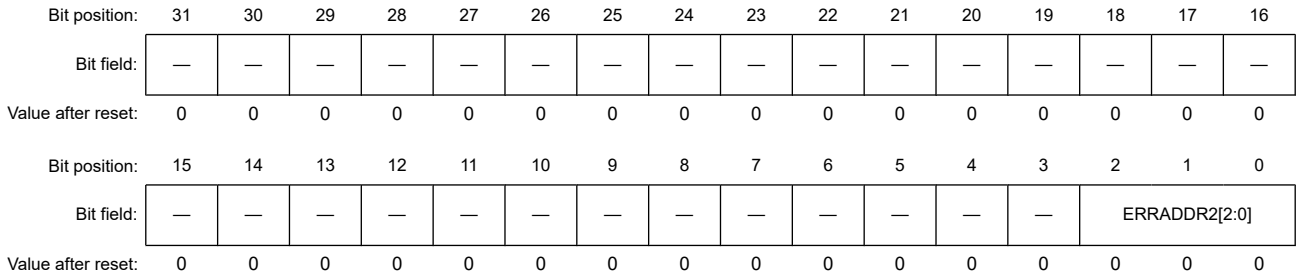
Bit	Symbol	Function	R/W
0	VALID	Validity of access error information Cleared to 0 when 0 is written. 0: Error information is invalid 1: Error information is valid	R/W
1	RW	Access error type 0: Read access 1: Write access	R
31:2	ERRADDR[29:0]	Access error address Indicates upper 30 bits of the 32-bit address. Lower 2 bits are masked.	R

The ERRINF_x register is a 32-bit register to store the safety region access error information by master module. AXI type MPUi (i = 0 to 2, and 6) provides error information for each read and write channel separately. ERRINF_R is for read channel and ERRINF_W is for write channel.

13.3.5 ERRINF1_x : Master MPU Error Information Register for AXI type 1 (x = R, W)

Base address: MPUi = 0x8129_1400 + 0x0100 × i (i = 7 to 9)
MPU15 = 0x8129_0900

Offset address: 0x88 (ERRINF1_R)
0x8C (ERRINF1_W)



Bit	Symbol	Function	R/W
2:0	ERRADDR2[2:0]	Access error address. Indicates upper 3 bits of the 35-bit address.	R
31:3	—	These bits are read as 0.	R

The ERRINF1_x register is a 32-bit register to store the safety region access error information by master module. AXI type MPUi (i = 7 to 9, and 15) provides upper 3 bits of 35-bit error information for each read and write channel separately. ERRINF1_R is for read channel and ERRINF1_W is for write channel.

13.3.6 CPU_CTRL : CPU Access Permission Control Register

Base address: MPU_AC = 0x8124_0000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CA553_CTRL	CA552_CTRL	CA551_CTRL	CA550_CTRL	—	—	—	—	—	—	—	CR521_CTRL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CR521_CTRL	Cortex-R52 CPU1 access permission control to Master MPU related registers 0: Access prohibited 1: Access permitted	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	CA550_CTRL	Cortex-A55 Core0 access permission control to Master MPU related registers 0: Access prohibited 1: Access permitted	R/W
9	CA551_CTRL	Cortex-A55 Core1 access permission control to Master MPU related registers 0: Access prohibited 1: Access permitted	R/W
10	CA552_CTRL	Cortex-A55 Core2 access permission control to Master MPU related registers 0: Access prohibited 1: Access permitted	R/W
11	CA553_CTRL	Cortex-A55 Core3 access permission control to Master MPU related registers 0: Access prohibited 1: Access permitted	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The CPU_CTRL register is used to control access from Cortex-R52 CPU1 and Cortex-A55 to Master MPU related registers.

13.3.7 SSTPCR0 : Slave Stop Control Register 0

Base address: SSC = 0x8129_0200

Offset address: 0x1000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DDRA4_ACK	DDRA4_REQ	—	—	DDRA1_ACK	DDRA1_REQ	—	—	DDRA0_ACK	DDRA0_REQ
Value after reset:	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1

Bit	Symbol	Function	R/W
0	DDRA0_REQ	DDRSS A0 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W

Bit	Symbol	Function	R/W
1	DDRA0_ACK	DDRSS A0 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DDRA1_REQ	DDRSS A1 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W
5	DDRA1_ACK	DDRSS A1 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	DDRA4_REQ	DDRSS A4 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W
9	DDRA4_ACK	DDRSS A4 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The SSTPCR0 register controls the slave bus stop state in main bus A. This register is available in Safety region.

13.3.8 SSTPCR1 : Slave Stop Control Register 1

Base address: SSC = 0x8129_0200

Offset address: 0x1004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	SDHI1_ACK	SDHI1_REQ	—	—	SDHI0_ACK	SDHI0_REQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	PCIE1_ACK	PCIE1_REQ	—	—	PCIE0_ACK	PCIE0_REQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

Bit	Symbol	Function	R/W
0	PCIE0_REQ	PCIE0 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W
1	PCIE0_ACK	PCIE0 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	PCIE1_REQ	PCIE1 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W
5	PCIE1_ACK	PCIE1 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
15:6	—	These bits are read as 0. The write value should be 0.	R/W
16	SDHI0_REQ	SDHI0 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W

Bit	Symbol	Function	R/W
17	SDHI0_ACK	SDHI0 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	SDHI1_REQ	SDHI1 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W
21	SDHI1_ACK	SDHI1 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
31:22	—	These bits are read as 0. The write value should be 0.	R/W

The SSTPCR1 register controls the slave bus stop state in main bus A. This register is available in Safety region.

13.3.9 SSTPCR4 : Slave Stop Control Register 4

Base address: SSC = 0x8129_0200

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DDRAPB	—	—	DDRR3_ACK	DDRR3_REQ	—	—	DDRR2_ACK	DDRR2_REQ
Value after reset:	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	1

Bit	Symbol	Function	R/W
0	DDRR2_REQ	DDRSS R2 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W
1	DDRR2_ACK	DDRSS R2 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DDRR3_REQ	DDRSS R3 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W
5	DDRR3_ACK	DDRSS R3 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	DDRAPB	DDR APB I/F Stop Control 0: Normal operation 1: Stop	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The SSTPCR4 register controls the slave bus stop state in main bus R. This register is available in Safety region.

13.3.10 SSTPCR5 : Slave Stop Control Register 5

Base address: SSC = 0x8129_0200

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GIC600_ACK	GIC600_REQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	GIC600_REQ	GIC600 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W
17	GIC600_ACK	GIC600 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The SSTPCR5 register controls the slave bus stop state in main bus R. This register is available in Safety region.

13.3.11 SSTPCR6 : Slave Stop Control Register 6

Base address: SSC = 0x8129_0200

Offset address: 0x0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	XSPI1_ACK	XSPI1_REQ	—	—	XSPI0_ACK	XSPI0_REQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

Bit	Symbol	Function	R/W
0	XSPI0_REQ	XSPI0 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W
1	XSPI0_ACK	XSPI0 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	XSPI1_REQ	XSPI1 I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W

Bit	Symbol	Function	R/W
5	XSPI1_ACK	XSPI1 I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The SSTPCR6 register controls the slave bus stop state in main bus R. This register is available in Safety region.

13.3.12 SSTPCR7 : Slave Stop Control Register 7

Base address: SSC = 0x8129_0200

Offset address: 0x000C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	AXIS1_ACK	AXIS1_REQ	—	—	AXIS0_ACK	AXIS0_REQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	Symbol	Function	R/W
0	AXIS0_REQ	Cortex-R52 CPU0 AXIS I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W
1	AXIS0_ACK	Cortex-R52 CPU0 AXIS I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	AXIS1_REQ	Cortex-R52 CPU1 AXIS I/F Bus Stop Request 0: Normal operation 1: Bus Stop Request	R/W
5	AXIS1_ACK	Cortex-R52 CPU1 AXIS I/F Bus Stop Acknowledge 0: Normal operation 1: Bus Stop Acknowledge	R
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The SSTPCR7 register controls the slave bus stop state in main bus R. This register is available in Safety region.

13.3.13 MSTACCCTL0 : Master Access Control Register 0

Base address: MSAC = 0x8129_0300

Offset address: 0x1000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GMAC2_SEL1x	GMAC2_SEL0	GMAC2_PROT1	GMAC2_PROT0	GMAC1_SEL1	GMAC1_SEL0	GMAC1_PROT1	GMAC1_PROT0	SDHI1_SEL1	SDHI1_SEL0	SDHI1_PROT1	SDHI1_PROT0	SDHI0_SEL1	SDHI0_SEL0	SDHI0_PROT1	SDHI0_PROT0
Value after reset:	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCIE1_SEL1	PCIE1_SEL0	PCIE1_PROT1	PCIE1_PROT0	PCIE0_SEL1	PCIE0_SEL0	PCIE0_PROT1	PCIE0_PROT0	LCDC_SEL1	LCDC_SEL0	LCDC_PROT1	LCDC_PROT0	CSETR_SEL1	CSETR_SEL0	CSETR_PROT1	CSETR_PROT0
Value after reset:	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0

Bit	Symbol	Function	R/W
0	CSETR_PROT0	CS ETR Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
1	CSETR_PROT1	CS ETR Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
2	CSETR_SEL0	CS ETR Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by CSETR_PROT0	R/W
3	CSETR_SEL1	CS ETR Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by CSETR_PROT1	R/W
4	LCDC_PROT0	LCDC Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
5	LCDC_PROT1	LCDC Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
6	LCDC_SEL0	LCDC Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by LCDC_PROT0	R/W
7	LCDC_SEL1	LCDC Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by LCDC_PROT1	R/W
8	PCIE0_PROT0	PCIE0 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
9	PCIE0_PROT1	PCIE0 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
10	PCIE0_SEL0	PCIE0 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by PCIE0_PROT0	R/W
11	PCIE0_SEL1	PCIE0 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by PCIE0_PROT1	R/W
12	PCIE1_PROT0	PCIE1 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
13	PCIE1_PROT1	PCIE1 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
14	PCIE1_SEL0	PCIE1 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by PCIE1_PROT0	R/W
15	PCIE1_SEL1	PCIE1 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by PCIE1_PROT1	R/W
16	SDHI0_PROT0	SDHI0 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
17	SDHI0_PROT1	SDHI0 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W

Bit	Symbol	Function	R/W
18	SDHI0_SEL0	SDHI0 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by SDHI0_PROT0	R/W
19	SDHI0_SEL1	SDHI0 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by SDHI0_PROT1	R/W
20	SDHI1_PROT0	SDHI1 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
21	SDHI1_PROT1	SDHI1 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
22	SDHI1_SEL0	SDHI1 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by SDHI1_PROT0	R/W
23	SDHI1_SEL1	SDHI1 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by SDHI1_PROT1	R/W
24	GMAC1_PROT0	GMAC1 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
25	GMAC1_PROT1	GMAC1 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
26	GMAC1_SEL0	GMAC1 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by GMAC1_PROT0	R/W
27	GMAC1_SEL1	GMAC1 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by GMAC1_PROT1	R/W
28	GMAC2_PROT0	GMAC2 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
29	GMAC2_PROT1	GMAC2 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
30	GMAC2_SEL0	GMAC2 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by GMAC2_PROT0	R/W
31	GMAC2_SEL1x	GMAC2 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by GMAC2_PROT1	R/W

The MSTACCCTL0 register is used to select the secure attribute and privilege attribute of transactions from master.

13.3.14 MSTACCCTL1 : Master Access Control Register 1

Base address: MSAC = 0x8129_0300

Offset address: 0x1004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	USB_SEL	—	USB_PROT1	USB_PROT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit	Symbol	Function	R/W
0	USB_PROT0	USB Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
1	USB_PROT1	USB Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
2	—	This bit is read as 1. The write value should be 1.	R/W
3	USB_SEL	USB Security Attribute (HNONSEC) and Privilege Attribute (HPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by USB_PROT[1:0]	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The MSTACCCTL1 register is used to select the secure attribute and privilege attribute of transactions from master.

13.3.15 MSTACCCTL2 : Master Access Control Register 2

Base address: MSAC = 0x8129_0300

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CR521_SEL1x	CR521_SEL0	CR521_PROT1	CR521_PROT0	CR520_SEL1	CR520_SEL0	CR520_PROT1	CR520_PROT0	DMAC2_SEL1	DMAC2_SEL0	DMAC2_PROT1	DMAC2_PROT0	DMAC1_SEL1	DMAC1_SEL0	DMAC1_PROT1	DMAC1_PROT0
Value after reset:	0	0	1	0	1	1	0	1	1	1	1	0	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DMAC0_SEL1	DMAC0_SEL0	DMAC0_PROT1	DMAC0_PROT0	—	—	—	—	GMAC0_SEL1	GMAC0_SEL0	GMAC0_PROT1	GMAC0_PROT0	CSAXIAP_SEL1	CSAXIAP_SEL0	CSAXIAP_PROT1	CSAXIAP_PROT0
Value after reset:	1	1	1	0	0	0	0	0	1	1	1	0	1	1	1	0

Bit	Symbol	Function	R/W
0	CSAXIAP_PROT0	CS AXI-AP Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
1	CSAXIAP_PROT1	CS AXI-AP Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W

Bit	Symbol	Function	R/W
2	CSAXIAP_SEL0	CS AXI-AP Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by CSAXIAP_PROT0	R/W
3	CSAXIAP_SEL1	CS AXI-AP Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by CSAXIAP_PROT1	R/W
4	GMAC0_PROT0	GMAC0 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
5	GMAC0_PROT1	GMAC0 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
6	GMAC0_SEL0	GMAC0 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by GMAC0_PROT0	R/W
7	GMAC0_SEL1	GMAC0 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by GMAC0_PROT1	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	DMAC0_PROT0	DMAC0 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
13	DMAC0_PROT1	DMAC0 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
14	DMAC0_SEL0	DMAC0 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by DMAC0_PROT0	R/W
15	DMAC0_SEL1	DMAC0 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by DMAC0_PROT1	R/W
16	DMAC1_PROT0	DMAC1 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
17	DMAC1_PROT1	DMAC1 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
18	DMAC1_SEL0	DMAC1 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by DMAC1_PROT0	R/W
19	DMAC1_SEL1	DMAC1 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by DMAC1_PROT1	R/W
20	DMAC2_PROT0	DMAC2 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
21	DMAC2_PROT1	DMAC2 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
22	DMAC2_SEL0	DMAC2 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by DMAC2_PROT0	R/W
23	DMAC2_SEL1	DMAC2 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by DMAC2_PROT1	R/W

Bit	Symbol	Function	R/W
24	CR520_PROT0	CR520 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
25	CR520_PROT1	CR520 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
26	CR520_SEL0	CR520 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by CR520_PROT0	R/W
27	CR520_SEL1	CR520 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by CR520_PROT1	R/W
28	CR521_PROT0	CR521 Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
29	CR521_PROT1	CR521 Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
30	CR521_SEL0	CR521 Privilege Attribute (AxPROT[0]) Overwrite Select 0: No overwrite 1: Overwritten by CR521_PROT0	R/W
31	CR521_SEL1x	CR521 Security Attribute (AxPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by CR521_PROT1	R/W

The MSTACCCTL2 register is used to select the secure attribute and privilege attribute of transactions from master.

13.3.16 MSTACCCTL3 : Master Access Control Register 3

Base address: MSAC = 0x8129_0300

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SHOSTIF_SEL	—	SHOSTIF_PROT1	SHOSTIF_PROT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit	Symbol	Function	R/W
0	SHOSTIF_PROT0	SHOSTIF Privilege Attribute Overwrite 0: Unprivileged 1: Privileged	R/W
1	SHOSTIF_PROT1	SHOSTIF Security Attribute Overwrite 0: Secure 1: Non-Secure	R/W
2	—	This bit is read as 1. The write value should be 1.	R/W
3	SHOSTIF_SEL	SHOSTIF Security Attribute (HNONSEC) and Privilege Attribute (HPROT[1]) Overwrite Select 0: No overwrite 1: Overwritten by SHOSTIF_PROT[1:0]	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The MSTACCCTL3 register is used to select the secure attribute and privilege attribute of transactions from master.

13.3.17 SLVACCCTL0 : Slave Access Control Register 0

Base address: MSAC = 0x8129_0300

Offset address: 0x1010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADC2_SL[1:0]		—	—	WDT5_SL[1:0]		WDT4_SL[1:0]		WDT3_SL[1:0]		WDT2_SL[1:0]		WDT1_SL[1:0]		WDT0_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	POEG1_SL[1:0]		GPT09_SL[1:0]		—	—	CMTW1_SL[1:0]		CMTW0_SL[1:0]		CMT2_SL[1:0]		CMT1_SL[1:0]		CMT0_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMT0_SL[1:0]	CMT Unit 0 Security Level	R/W
3:2	CMT1_SL[1:0]	CMT Unit 1 Security Level	R/W
5:4	CMT2_SL[1:0]	CMT Unit 2 Security Level	R/W
7:6	CMTW0_SL[1:0]	CMTW Unit 0 Security Level	R/W
9:8	CMTW1_SL[1:0]	CMTW Unit 1 Security Level	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	GPT09_SL[1:0]	GPT Unit 09 Security Level	R/W
15:14	POEG1_SL[1:0]	POEG Unit 1 Security Level	R/W
17:16	WDT0_SL[1:0]	WDT Unit 0 Security Level	R/W
19:18	WDT1_SL[1:0]	WDT Unit 1 Security Level	R/W
21:20	WDT2_SL[1:0]	WDT Unit 2 Security Level	R/W
23:22	WDT3_SL[1:0]	WDT Unit 3 Security Level	R/W
25:24	WDT4_SL[1:0]	WDT Unit 4 Security Level	R/W
27:26	WDT5_SL[1:0]	WDT Unit 5 Security Level	R/W
29:28	—	These bits are read as 0. The write value should be 0.	R/W
31:30	ADC2_SL[1:0]	ADC Unit 2 Security Level	R/W

The SLVACCCTL0 register is used to select the security level of the slaves.

13.3.18 SLVACCCTL1 : Slave Access Control Register 1

Base address: MSAC = 0x8129_0300

Offset address: 0x1014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TSU_SL[1:0]		CRC0_SL[1:0]		DOC_SL[1:0]		CANFD_SL[1:0]		—	—	SPI2_SL[1:0]		SPI1_SL[1:0]		SPI0_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IIC1_SL[1:0]		IIC0_SL[1:0]		—	—	SCI4_SL[1:0]		SCI3_SL[1:0]		SCI2_SL[1:0]		SCI1_SL[1:0]		SCI0_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SCI0_SL[1:0]	SCI Unit 0 Security Level	R/W
3:2	SCI1_SL[1:0]	SCI Unit 1 Security Level	R/W
5:4	SCI2_SL[1:0]	SCI Unit 2 Security Level	R/W
7:6	SCI3_SL[1:0]	SCI Unit 3 Security Level	R/W
9:8	SCI4_SL[1:0]	SCI Unit 4 Security Level	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	IIC0_SL[1:0]	IIC Unit 0 Security Level	R/W
15:14	IIC1_SL[1:0]	IIC Unit 1 Security Level	R/W
17:16	SPI0_SL[1:0]	SPI Unit 0 Security Level	R/W
19:18	SPI1_SL[1:0]	SPI Unit 1 Security Level	R/W
21:20	SPI2_SL[1:0]	SPI Unit 2 Security Level	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	CANFD_SL[1:0]	CANFD Security Level	R/W
27:26	DOC_SL[1:0]	DOC Security Level	R/W
29:28	CRC0_SL[1:0]	CRC Unit 0 Security Level	R/W
31:30	TSU_SL[1:0]	TSU Security Level	R/W

The SLVACCCTL1 register is used to select the security level of the slaves.

13.3.19 SLVACCCTL2 : Slave Access Control Register 2

Base address: MSAC = 0x8129_0300

Offset address: 0x1018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	SPI3_SL[1:0]		IIC2_SL[1:0]		SCI5_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CRC1_SL[1:0]		RTC_SL[1:0]		—	—	—	—	POEG2_SL[1:0]		GPT10_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	GPT10_SL[1:0]	GPT Unit 10 Security Level	R/W
3:2	POEG2_SL[1:0]	POEG Unit 2 Security Level	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
9:8	RTC_SL[1:0]	RTC Security Level	R/W
11:10	CRC1_SL[1:0]	CRC Unit 1 Security Level	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
17:16	SCI5_SL[1:0]	SCI Unit 5 Security Level	R/W
19:18	IIC2_SL[1:0]	IIC Unit 2 Security Level	R/W
21:20	SPI3_SL[1:0]	SPI Unit 3 Security Level	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

The SLVACCCTL2 register is used to select the security level of the slaves.

13.3.20 SLVACCCTL4 : Slave Access Control Register 4

Base address: MSAC = 0x8129_0300

Offset address: 0x1020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	GMAC2_SL[1:0]		GMAC1_SL[1:0]		—	—	—	—	SDHI1_SL[1:0]		SDHI0_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LCDC_SL[1:0]		—	—	—	—	—	—	USB_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	USB_SL[1:0]	USB (USBHC and USBf) Security Level	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	LCDC_SL[1:0]	LCDC Security Level	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	SDHI0_SL[1:0]	SDHI0 Unit 0 Security Level	R/W
19:18	SDHI1_SL[1:0]	SDHI1 Unit 1 Security Level	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GMAC1_SL[1:0]	GMAC Unit 1 Security Level	R/W
27:26	GMAC2_SL[1:0]	GMAC Unit 2 Security Level	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The SLVACCCTL4 register is used to select the security level of the slaves.

13.3.21 SLVACCCTL5 : Slave Access Control Register 5

Base address: MSAC = 0x8129_0300

Offset address: 0x1024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MPUGMAC2_SL[1:0]		MPUGMAC1_SL[1:0]		MPUSDHI1_SL[1:0]		MPUSDHI0_SL[1:0]		MPULCDC_SL[1:0]		MPUPCIE1_SL[1:0]		MPUPCIE0_SL[1:0]		MPUUSB_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TZCPCIE_SL[1:0]		TZCDDRA4_SL[1:0]		TZCDDRA0A1_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0

Bit	Symbol	Function	R/W
1:0	TZCDDRA0A1_SL[1:0]	DDR A0 and A1 I/F TZC-400 Security Level	R/W
3:2	TZCDDRA4_SL[1:0]	DDR A4 I/F TZC-400 Security Level	R/W
5:4	TZCPCIE_SL[1:0]	PCIE TZC-400 Security Level	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
17:16	MPUUSB_SL[1:0]	USB (USBHC and USBf) MPU Security Level	R/W
19:18	MPUPCIE0_SL[1:0]	PCIE Unit 0 MPU Security Level	R/W

Bit	Symbol	Function	R/W
21:20	MPUPCIE1_SL[1:0]	PCIE Unit 1 MPU Security Level	R/W
23:22	MPULCDC_SL[1:0]	LCDC MPU Security Level	R/W
25:24	MPUSDHI0_SL[1:0]	SDHI Unit 0 MPU Security Level	R/W
27:26	MPUSDHI1_SL[1:0]	SDHI Unit 1 MPU Security Level	R/W
29:28	MPUGMAC1_SL[1:0]	GMAC Unit 1 MPU Security Level	R/W
31:30	MPUGMAC2_SL[1:0]	GMAC Unit 2 MPU Security Level	R/W

The SLVACCCTL5 register is used to select the security level of the slaves.

13.3.22 SLVACCCTL6 : Slave Access Control Register 6

Base address: MSAC = 0x8129_0300

Offset address: 0x0010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SYSRAM3_SL[1:0]		SYSRAM2_SL[1:0]		SYSRAM1_SL[1:0]		SYSRAM0_SL[1:0]		—	—	DMAC2_SL[1:0]		DMAC1_SL[1:0]		DMAC0_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CR521_SL[1:0]		CR520_SL[1:0]		CA55_SL[1:0]		—	—	GMAC0_SL[1:0]		ESC_SL[1:0]		ETHSW_SL[1:0]		ETHSS_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ETHSS_SL[1:0]	ETHSS Security Level	R/W
3:2	ETHSW_SL[1:0]	ETHSW Security Level	R/W
5:4	ESC_SL[1:0]	ESC Security Level	R/W
7:6	GMAC0_SL[1:0]	GMAC Unit 0 Security Level	R/W
9:8	—	These bits are read as 0. The write value should be 0.	R/W
11:10	CA55_SL[1:0]	Cortex-A55 Auxiliary Register Security Level	R/W
13:12	CR520_SL[1:0]	Cortex-R52 CPU0 Auxiliary Register Security Level	R/W
15:14	CR521_SL[1:0]	Cortex-R52 CPU1 Auxiliary Register Security Level	R/W
17:16	DMAC0_SL[1:0]	DMAC Unit 0 Security Level	R/W
19:18	DMAC1_SL[1:0]	DMAC Unit 1 Security Level	R/W
21:20	DMAC2_SL[1:0]	DMAC Unit 2 Security Level	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	SYSRAM0_SL[1:0]	SYSRAM Unit 0 Auxiliary Register Security Level	R/W
27:26	SYSRAM1_SL[1:0]	SYSRAM Unit 1 Auxiliary Register Security Level	R/W
29:28	SYSRAM2_SL[1:0]	SYSRAM Unit 2 Auxiliary Register Security Level	R/W
31:30	SYSRAM3_SL[1:0]	SYSRAM Unit 3 Auxiliary Register Security Level	R/W

The SLVACCCTL6 register is used to select the security level of the slaves.

13.3.23 SLVACCCTL7 : Slave Access Control Register 7

Base address: MSAC = 0x8129_0300

Offset address: 0x0014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSDBGAPB_SL[1:0]		GIC600_SL[1:0]		OTP_SL[1:0]		RSIP_SL[1:0]		BUS_SL[1:0]		—	—	LLPP_SL[1:0]		—	—
Value after reset:	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SHOSTIF_SL[1:0]		MBXSEM_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	MBXSEM_SL[1:0]	MBXSEM Security Level	R/W
3:2	SHOSTIF_SL[1:0]	SHOSTIF Security Level	R/W
17:4	—	These bits are read as 0. The write value should be 0.	R/W
19:18	LLPP_SL[1:0]	LLPP Peripherals Security Level	R/W
21:20	—	These bits are read as 0. The write value should be 0.	R/W
23:22	BUS_SL[1:0]	Bus Security Level Always set to 11b.	R/W
25:24	RSIP_SL[1:0]	RSIP Security Level	R/W
27:26	OTP_SL[1:0]	OTP Security Level	R/W
29:28	GIC600_SL[1:0]	GIC600 Security Level	R/W
31:30	CSDBGAPB_SL[1:0]	CS Debug APB Security Level	R/W

The SLVACCCTL7 register is used to select the security level of the slaves.

13.3.24 SLVACCCTL8 : Slave Access Control Register 8

Base address: MSAC = 0x8129_0300

Offset address: 0x0018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	XSPI1_SL[1:0]		XSPI0_SL[1:0]		—	—	SYSCTRL_SL[1:0]		ICU_SL[1:0]		PRCR_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TZCBSC_SL[1:0]		TZCXSPI_SL[1:0]		TZCSYSRAM_SL[1:0]		TZCDDRR3_SL[1:0]		TZCDDRR2_SL[1:0]		TZCTCM_SL[1:0]		—	—	DDRCTRL_SL[1:0]	
Value after reset:	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	DDRCTRL_SL[1:0]	DDR CTRL Security Level	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	TZCTCM_SL[1:0]	Cortex-R52 TCM TZC-400 Security Level	R/W
7:6	TZCDDRR2_SL[1:0]	DDR R2 I/F TZC-400 Security Level	R/W
9:8	TZCDDRR3_SL[1:0]	DDR R3 I/F TZC-400 Security Level	R/W

Bit	Symbol	Function	R/W
11:10	TZCSYSRAM_SL[1:0]	SYSRAM TZC-400 Security Level	R/W
13:12	TZCXSPI_SL[1:0]	XSPI TZC-400 Security Level	R/W
15:14	TZCBSC_SL[1:0]	BSC TZC-400 Security Level	R/W
17:16	PRCR_SL[1:0]	PRCR Security Level	R/W
19:18	ICU_SL[1:0]	ICU Security Level	R/W
21:20	SYSCTRL_SL[1:0]	System Control (Reset, CGC, CLMA, Low Power) Security Level	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	XSPI0_SL[1:0]	XSPI Unit 0 Security Level	R/W
27:26	XSPI1_SL[1:0]	XSPI Unit 1 Security Level	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The SLVACCCTL8 register is used to select the security level of the slaves.

13.3.25 SLVACCCTL9 : Slave Access Control Register 9

Base address: MSAC = 0x8129_0300

Offset address: 0x001C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	MPUCS_SL[1:0]		MPUDMAC2_SL[1:0]		MPUDMAC1_SL[1:0]		MPUDMAC0_SL[1:0]		MPUGMAC0_SL[1:0]		MPUSHOSTIF_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADDEXP_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADDEXP_SL[1:0]	Address Expander Security Level	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	MPUSHOSTIF_SL[1:0]	SHOSTIF MPU Security Level	R/W
19:18	MPUGMAC0_SL[1:0]	GMAC Unit 0 MPU Security Level	R/W
21:20	MPUDMAC0_SL[1:0]	DMAC Unit 0 MPU Security Level	R/W
23:22	MPUDMAC1_SL[1:0]	DMAC Unit 1 MPU Security Level	R/W
25:24	MPUDMAC2_SL[1:0]	DMAC Unit 2 MPU Security Level	R/W
27:26	MPUCS_SL[1:0]	CS (ETR and AXI-AP) MPU Security Level	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The SLVACCCTL9 register is used to select the security level of the slaves.

13.3.26 ADXCTLn : Address Expander Control Register n (n = 0 to 12)

Base address: ADXC0 = 0x8129_0100 (ADXCTL0 to ADXCTL5)
ADXC1 = 0x8129_1100 (ADXCTL6 to ADXCTL12)

Offset address: 0x00 + 0x04 × n (ADXCTL0 to ADXCTL5)
0x00 + 0x04 × (n - 6) (ADXCTL6 to ADXCTL12)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	DDRMIR1[4:0]					—	—	—	DDRMIR0[4:0]				
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PCIE1MIR[4:0]					—	—	—	PCIE0MIR[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	PCIE0MIR[4:0]*1	PCIE0 Mirror Address Specify [32:28] bit of start address of any 256 MB in PCIE0 area (0x4_0000_0000 to 0x5_FFFF_FFFF).	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
12:8	PCIE1MIR[4:0]*1	PCIE1 Mirror Address Specify [32:28] bit of start address of any 256 MB in PCIE1 area (0x6_0000_0000 to 0x7_FFFF_FFFF).	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	DDRMIR0[4:0]	DDR Mirror Address 0 Specify [32:28] bit of start address of any 512 MB in DDR area (0x2_0000_0000 to 0x3_FFFF_FFFF). 0x1F is prohibited.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	DDRMIR1[4:0]	DDR Mirror Address 1 Specify [32:28] bit of start address of any 512 MB in DDR area (0x2_0000_0000 to 0x3_FFFF_FFFF). 0x1F is prohibited.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Valid for n = 0 to 4. Reserved for n = 5 to 12.

The ADXCTLn registers are used to expand address for 32-bit address master to access PCIE and DDR area.

13.4 Operation

13.4.1 Bus Connection

The internal main bus A, R, and low-latency peripheral bus of this LSI have a multiple-layer structure. If bus masters request to access different bus slaves respectively, multiple accesses are processed in parallel. If bus masters request to access the same bus slave, priority order decision is performed and accesses are processed sequentially according to the priority order. Arbitration scheme for internal main bus A and R is round-robin and for low-latency peripheral bus is fixed priority. Priority of access to slaves on low-latency peripheral bus 0 (LLPP0) is fixed to Cortex-R52 CPU0 > Cortex-R52 CPU1 > the other bus masters on internal main bus. Priority of access to slaves on low-latency peripheral bus 1 (LLPP1) is fixed to Cortex-R52 CPU1 > Cortex-R52 CPU0 > the other bus masters on internal main bus.

Table 13.4 lists the connection between the bus masters and the bus slaves.

Table 13.4 Connection between bus masters and bus slaves (A: Accessible, —: Inaccessible)

Bus slave	Bus master											
	Cortex-A55	Cortex-R52 CPU0	Cortex-R52 CPU1	SHOSTIF	DMAC unit 0 to 2	GMAC unit 0 to 2	PCIE	USBHC USBf	SDHI unit 0, 1	LCDC	CoreSight	
											AXI-AP	ETR
Peripheral bus R0, R1 (LLPP)	A*1	A	A	—	A*1	—	A*1	—	—	—	A*1	—
Peripheral bus R2, R3 (NONSAFETY)	A	A	A	—	A	—	A	—			A	—
Peripheral bus R4, R5 (SAFETY)	A	A	A	—	—	—	—	—	—	—	—	—
Peripheral bus R6 (DMAC, GIC600)	A*2	A	A	—	—	—	—	—	—	—	A	—
Peripheral bus R7	A	A	A	—	A*3	—	A	—	—	—	A	—
Cortex-R52 CPU0 AXIS (ATCM, BTCM)	A	—	A	—	A	A	A	A	—	—	A	A
Cortex-R52 CPU1 AXIS (ATCM, BTCM)	A	A	—	—	A	A	A	A	—	—	A	A
BSC	A	A	A	—	A*4	A*4	A*4	A*4	A*4	—	A	A*4
xSPI	A	A	A	—	A*4	A*4	A*4	A*4	A*4	—	A	A*4
System SRAM	A	A	A	A*4	A*4	A*4	A*4	A*4	A*4	A*4	A*4	A*4
System Control	A*5 *6	A	A*5 *6	—	—	—	—	—	—	—	A*5	—
IO Port (Non-safety IO)	A	A	A	—	A	—	A	—	—	—	A	—
IO Port (Safety IO)	A	A	A	—	—	—	—	—	—	—	—	—
USB	A	A	A	—	A	—	A	—	—	—	A	—
PCIE	A	A	A	—	A	—	—	—	—	—	A	A
DDR (memory)	A (A4 I/F)	A (R3 I/F)	A (R3 I/F)	—	A (R2 I/F)	A (R2 I/F for unit 0, A1 I/F for unit 1,2)	A (A1 I/F)	A (A1 I/F)	A (A0 I/F)	A (A0 I/F)	A (R2 I/F)	A (A1 I/F)
DDR (register)	A	A	A	—	—	—	—	—	—	—	A	—
GMAC1, 2	A	A	A	—	A	—	A	—	—	—	A	—
SDHI	A	A	A	—	A	—	A	—	—	—	A	—
LCDC	A	A	A	—	A	—	A	—	—	—	A	—
SHOSTIF, MBXSEM	A	A	A	A	—	—	—	—	—	—	A	—
OTP	A	A	A	—	—	—	—	—	—	—	—	—
RSIP	A	A	A	—	A	—	—	—	—	—	—	—
CoreSight	A	A	A	—	—	—	—	—	—	—	—	—

Note 1. Access through internal main bus is available. Priority order for these bus masters is round-robin.
Note 2. GIC600 register are accessible from Cortex-A55 only.
Note 3. Process data RAM on ESC only.
Note 4. Memory space only. Control registers are not accessible.
Note 5. CPU_CTRL register is not accessible.

Note 6. Master MPU related registers are accessible if CPU_CTRL register is set by Cortex-R52 CPU0.

13.4.2 Master MPU

Master MPU (MPUi, (i = 0 to 15)) is used to prevent master modules except the Cortex-R52 and Cortex-A55 from accessing the safety region that is specified by STADDm and ENDADDm registers. When the master MPU detects the access to the safety region, access is terminated with an error response from the bus. Up to 16 regions can be protected for each master module. Boundary of the safety region is set with 1 KB granularity for AHB type MPU and 4 KB for AXI type MPU.

Master MPU stores error address and access attribution (read or write) to the Master MPU Error Information registers (ERRINF, ERRINF_R, ERRINF_W) when access error is detected. Interrupt is also generated to inform the CPU of the error occurrence.

Master MPU registers (STADDm, ENDADDm, ERRINF, and ERRINF_x) can be controlled by Cortex-R52 CPU0, CPU1, and Cortex-A55. After the reset, only Cortex-R52 CPU0 can read and write these registers in case of Cortex-R52 CPU0 boot mode. By setting CPU_CTRL register by Cortex-R52 CPU0, Cortex-R52 CPU1 and Cortex-A55 can also read and write these registers. In case of Cortex-A55 Core0 boot mode, all the CPU can read and write these registers after the reset.

Table 13.5 lists the correspondence between MPUi and the master module. AXI type master MPU monitors access address and stores error address to the registers for read and write channels separately. On the other hand, AHB type master MPU monitors access address and stores error address to the same register for commonly read and write operation.

Table 13.5 Correspondence between master module and error information registers

Master module	Bus	Type	Address size	Error information registers	
				Base address symbol	Register symbol
GMAC Unit 1	A	AXI	32	MPU0	ERRINF_R, ERRINF_W
GMAC Unit 2	A	AXI	32	MPU1	ERRINF_R, ERRINF_W
USB Host	A	AHB	32	MPU2	ERRINF
USB function	A	AHB	32	MPU3	ERRINF
SDHI Unit 0	A	AXI	32	MPU4	ERRINF_R, ERRINF_W
SDHI Unit 1	A	AXI	32	MPU5	ERRINF_R, ERRINF_W
LCD Controller	A	AXI	32	MPU6	ERRINF_R, ERRINF_W
PCIE Unit 0	A	AXI	35	MPU7	ERRINF_R, ERRINF_W, ERRINF1_R, ERRINF1_W
PCIE Unit 1	A	AXI	35	MPU8	ERRINF_R, ERRINF_W, ERRINF1_R, ERRINF1_W
CoreSight ETR	A	AXI	35	MPU9	ERRINF_R, ERRINF_W, ERRINF1_R, ERRINF1_W
DMAC Unit 0	R	AXI	32	MPU10	ERRINF_R, ERRINF_W
DMAC Unit 1	R	AXI	32	MPU11	ERRINF_R, ERRINF_W
DMAC Unit 2	R	AXI	32	MPU12	ERRINF_R, ERRINF_W
GMAC Unit 0	R	AXI	32	MPU13	ERRINF_R, ERRINF_W
SHOSTIF	R	AHB	32	MPU14	ERRINF
CoreSight AXI-AP	R	AXI	35	MPU15	ERRINF_R, ERRINF_W, ERRINF1_R, ERRINF1_W

Safety region that is specified can be overlapped. If different attributions (read protection, write protection, or read/write protection) are set to the same area, combined protection is applied. Figure 13.2 shows an example of the safety region and attribution by the master MPU.

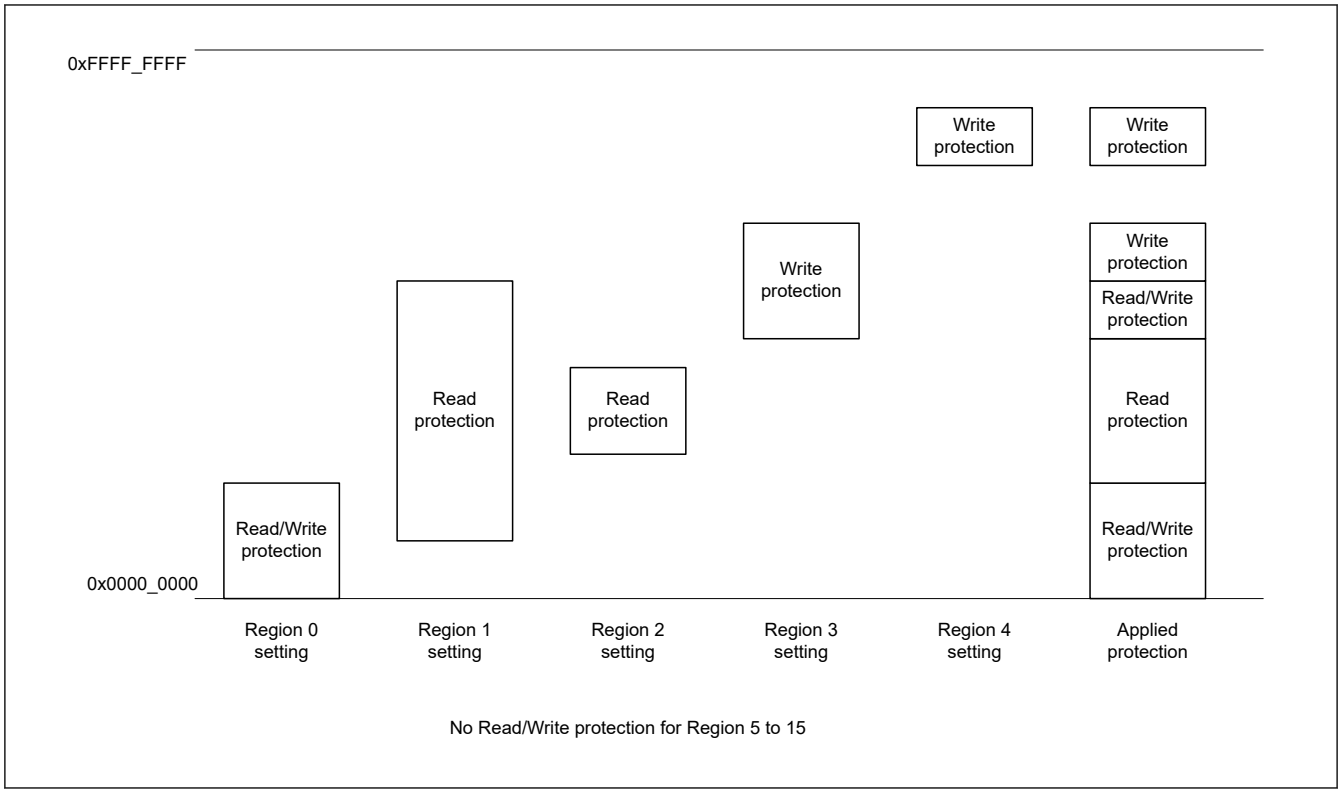


Figure 13.2 Safety region and attribution by master MPU

13.4.3 Slave Stop Function

The slave stop function is used to stop the access to the AXI-type slave I/F safely. Before changing module reset state or module stop state, the access to the slave I/F needs to be stopped by slave stop function. Control procedure is the following.

In case of release from the module reset or module stop state:

1. Release the target slave from the module reset or module stop state by changing corresponding module reset control register or module stop control register.
2. Set xxx_REQ bit in SSTPCRn register to 0. (xxx: target slave name)
3. Polling xxx_ACK bit in SSTPCRn register until it changes to 0.
4. After xxx_ACK bit in SSTPCRn register changes to 0, access to the module is possible.

In case of transition to the module reset or module stop state:

1. Stop operation of target module.
2. Set xxx_REQ bit in SSTPCRn register to 1. (xxx: target slave name)
3. Polling xxx_ACK bit in SSTPCRn register until it changes to 1.
4. After xxx_ACK bit in SSTPCRn register changes to 1, transit the target slave to the module reset or module stop state by changing corresponding module reset control register or module stop control register.

Table 13.6 lists AXI-type slaves supporting slave stop function.

Table 13.6 AXI-type Slaves supporting Slave Stop Function

Bus slave	I/F	Slave Stop Control Register			Related module reset and stop register	
		Register	ACK bit	REQ bit	Module reset	Module stop
DDRSS	A0 I/F	SSTPCR0	DDRA0_ACK	DDRA0_REQ	MRCTLM.[25:16]*1	MSTPCRM.[0]*1
	A1 I/F	SSTPCR0	DDRA1_ACK	DDRA1_REQ		
	A4 I/F	SSTPCR0	DDRA4_ACK	DDRA4_REQ		
	R2 I/F	SSTPCR4	DDRR2_ACK	DDRR2_REQ		
	R3 I/F	SSTPCR4	DDRR3_ACK	DDRR3_REQ		
PCIE unit 0	—	SSTPCR1	PCIE0_ACK	PCIE0_REQ	—	MSTPCRM.[8]*1
PCIE unit 1	—	SSTPCR1	PCIE1_ACK	PCIE1_REQ	—	
SDHI unit 0	—	SSTPCR1	SDHI0_ACK	SDHI0_REQ	—	MSTPCRM.[12]
SDHI unit 1	—	SSTPCR1	SDHI1_ACK	SDHI1_REQ	—	MSTPCRM.[13]
GIC600	—	SSTPCR5	GIC600_ACK	GIC600_REQ	SWR55C	—
XSPI unit 0	—	SSTPCR6	XSPI0_ACK	XSPI0_REQ	MRCTLA.[4]	MSTPCRA.[4]
XSPI unit 1	—	SSTPCR6	XSPI1_ACK	XSPI1_REQ	MRCTLA.[5]	MSTPCRA.[5]
Cortex-R52 CPU0 AXIS	—	SSTPCR7	AXIS0_ACK	AXIS0_REQ	SWRCPU0	MSTPCRN.[0]
Cortex-R52 CPU1 AXIS	—	SSTPCR7	AXIS1_ACK	AXIS1_REQ	SWRCPU1	MSTPCRN.[1]

Note 1. All the related bits of slave stop control register must be controlled to change this register.

13.4.4 TrustZone Access Control

Arm® TrustZone® technology is a system architecture to separate secure world and non-secure world by hardware and software. Cortex-A55 supports secure and non-secure transaction natively, but other bus masters and bus slaves also need to support handling of secure or non-secure transaction. This LSI implements TrustZone Master (TZM) module, TrustZone Slave (TZS) and TZC-400 to enable those bus masters and slaves support TrustZone in a whole chip.

In addition, this LSI can identify not only secure or non-secure transaction but also privileged or un-privileged transaction to protect illegal access. Figure 13.3 shows the overview of TrustZone access control in this LSI.

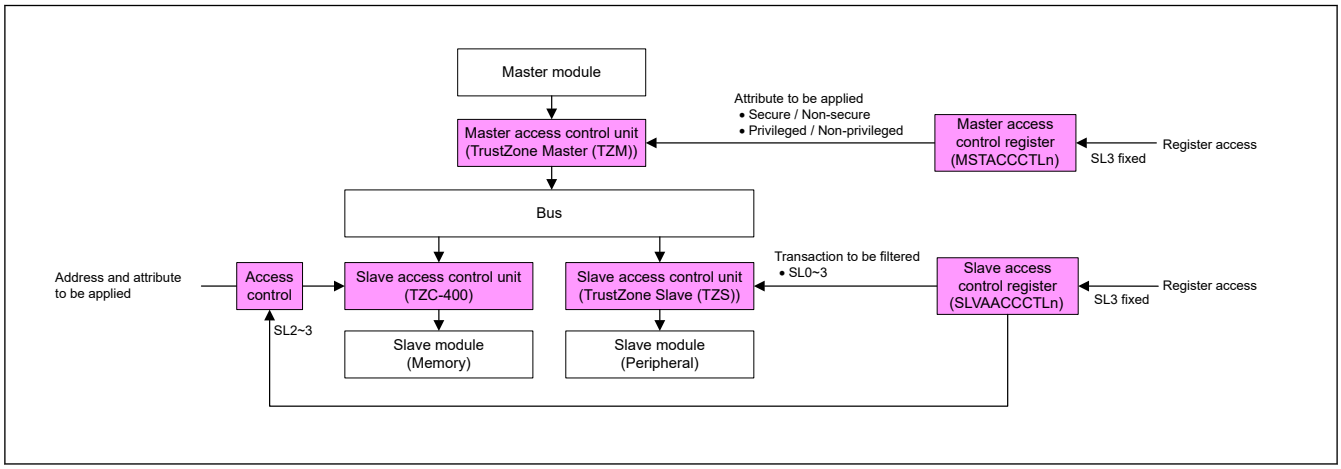


Figure 13.3 Overview of TrustZone access control

13.4.4.1 TrustZone Master (TZM)

TrustZone Master (TZM) provides the function to overwrite attribute of bus transaction issued by bus master according to the master access control register (MSTACCCTLn). xxx_PROT1 and xxx_PROT0 (xxx: master name) bits in MSTACCCTLn register indicate security attribute and privilege attribute to be overwritten, respectively. It is selectable by xxx_SEL1 and xxx_SEL0 bits in MSTACCCTLn register whether security attribute and privileged attribute are overwritten or not, respectively. Access to MSTACCCTLn register is allowed by secure and privileged transaction only. MSTACCCTLn register setting is applied to both read and write transaction.

Table 13.7 shows security and privilege attribute and Table 13.8 lists bus masters supporting TZM, and corresponding initial setting of MSTACCCTLn register. Some Bus Masters have own configurable registers. If overwrite is disabled, attribute can be also changed by those registers. For details, see register descriptions of each section.

Table 13.7 Security and Privilege Attribute

AXI signal	AHB signal	Description	
AxPROT[1]	HNONSEC	Security attribute	0: Secure 1: Non-secure
AxPROT[0]	HPORT[1]	Privilege attribute	0: Unprivileged 1: Privileged

Table 13.8 Bus Master supporting TZM and Initial Setting of Master Access Control Register (1 of 2)

Bus Master	Port	Master Access Control Register						Bus Master Information			
			Bit		Initial Setting			Initial Setting		Configurable Register	
		Register Name	SEL	PROT	Overwrite	Security attribute	Privilege attribute	Security attribute	Privilege attribute	Security attribute	Privilege attribute
Cortex-R52 CPU0	AXIM	MSTACCCTL2	27:26	25:24	Enabled both	Secure	Privileged	Non-secure	Depend on EL	—	—
	LLPP*1	— (No TZM)	—	—	—	—	—	Non-secure	Depend on EL	—	—
Cortex-R52 CPU1	AXIM	MSTACCCTL2	31:30	29:28	Disabled both	—	—	Non-secure	Depend on EL	—	—
	LLPP*1	— (No TZM)	—	—	—	—	—	Non-secure	Depend on EL	—	—
SHOSTIF	—	MSTACCCTL3	3	1:0	Enabled both	Non-secure	Non-privileged	Non-secure	Non-privileged	—	—
DMAC Unit 0	—	MSTACCCTL2	15:14	13:12	Enabled both	Non-secure	Non-privileged	Secure	Non-privileged	CHEXT_n	CHEXT_n
DMAC Unit 1	—	MSTACCCTL2	19:18	17:16	Enabled both	Non-secure	Non-privileged	Secure	Non-privileged	CHEXT_n	CHEXT_n
DMAC Unit 2	—	MSTACCCTL2	23:22	21:20	Enabled both	Non-secure	Non-privileged	Secure	Non-privileged	CHEXT_n	CHEXT_n
GMAC Unit 0	—	MSTACCCTL2	7:6	5:4	Enabled both	Non-secure	Non-privileged	Secure	Non-privileged	—	—
GMAC Unit 1	—	MSTACCCTL0	27:26	25:24	Enabled both	Non-secure	Non-privileged	Secure	Non-privileged	—	—
GMAC Unit 2	—	MSTACCCTL0	31:30	29:28	Enabled both	Non-secure	Non-privileged	Secure	Non-privileged	—	—
PCIE Unit 0	—	MSTACCCTL0	11:10	9:8	Enabled both	Non-secure	Non-privileged	Non-secure	Non-privileged	PCI_RC_MSET0, PCI_EP_MSET0	PCI_RC_MSET0, PCI_EP_MSET0
PCIE Unit 1	—	MSTACCCTL0	15:14	13:12	Enabled both	Non-secure	Non-privileged	Non-secure	Non-privileged	PCI_RC_MSET0, PCI_EP_MSET0	PCI_RC_MSET0, PCI_EP_MSET0
USBHC	—	MSTACCCTL1	3	1:0	Enabled both	Non-secure	Non-privileged	Non-secure	Non-privileged	—	AHBBUSCTR
USBf	—	MSTACCCTL1	3	1:0	Enabled both	Non-secure	Non-privileged	Non-secure	Non-privileged	—	CHEXT_n, DCTRL
SDHI Unit 0	—	MSTACCCTL0	19:18	17:16	Enabled both	Non-secure	Non-privileged	Non-secure	Non-privileged	—	—
SDHI Unit 1	—	MSTACCCTL0	23:22	21:20	Enabled both	Non-secure	Non-privileged	Non-secure	Non-privileged	—	—
LCDC	—	MSTACCCTL0	7:6	5:4	Enabled both	Non-secure	Non-privileged	Non-secure	Non-privileged	—	—

Table 13.8 Bus Master supporting TZM and Initial Setting of Master Access Control Register (2 of 2)

Bus Master	Port	Master Access Control Register						Bus Master Information			
			Bit		Initial Setting			Initial Setting		Configurable Register	
		Register Name	SEL	PROT	Overwrite	Security attribute	Privilege attribute	Security attribute	Privilege attribute	Security attribute	Privilege attribute
CoreSight	ETR	MSTACCCTL0	3:2	1:0	Enabled both	Non-secure	Non-privileged	Secure	Non-privileged	AXICTL	AXICTL
	AXI-AP	MSTACCCTL2	3:2	1:0	Enabled both	Non-secure	Non-privileged	Non-secure	Privileged	CSW	CSW

Note 1. Transaction from LLPP of Cortex-R52 is always non-secure. LLPP peripherals need to be used as non-security peripherals.

13.4.4.2 TrustZone Slave (TZS)

TrustZone Slave (TZS) provides the function to filter bus transaction by comparing its attribution with security level (SL) set in slave access control register (SLVACCCTLn). Combination of security level (SL) and transaction filtering rule are defined in Table 13.9. SL can be set in SLVACCCTLn register each slave and it is applied to both read and write transaction. Table 13.10 lists bus slave supporting TZS, and corresponding bit, initial setting and configurable range of SLVACCCTLn register. When transaction is not allowed by TZS, bus error reply is generated. TZS is used for peripheral register access basically.

Table 13.9 Transaction filtering rule for TrustZone Slave (TZS)

Security Level (SL)	Transaction			
	Non-secure		Secure	
	Unprivileged	Privileged	Unprivileged	Privileged
0	Allowed	Allowed	Allowed	Allowed
1	—	Allowed	Allowed	Allowed
2	—	—	Allowed	Allowed
3	—	—	—	Allowed

Note: When transaction is not allowed, bus error reply is generated.

Table 13.10 Bus Slave supporting TZS and Initial Setting and Configurable Range of Slave Access Control Register (1 of 4)

Bus Slave	Slave Access Control Register			
	Register Name	Bit	Initial Setting	Configurable range
Master Access Control Registers	—	—	3	3 fixed
Slave Access Control Registers	—	—	3	3 fixed
TZC400-0 (DDR A0, A1 I/F)	SLVACCCTL5	TZCDDRA0A1_SL	2	2 or 3
TZC400-1 (DDR A4 I/F)	SLVACCCTL5	TZCDDRA4_SL	2	2 or 3
TZC400-2 (PCIE)	SLVACCCTL5	TZCPCIE_SL	2	2 or 3
TZC400-3 (DDR R2 I/F)	SLVACCCTL8	TZCDDRR2_SL	2	2 or 3
TZC400-4 (DDR R3 I/F)	SLVACCCTL8	TZCDDRR3_SL	2	2 or 3
TZC400-5 (SYSRAM)	SLVACCCTL8	TZCSYSRAM_SL	2	2 or 3
TZC400-6 (BSC)	SLVACCCTL8	TZCBSC_SL	2	2 or 3
TZC400-7 (XSPI)	SLVACCCTL8	TZCXSPI_SL	2	2 or 3
TZC400-8 (TCM)	SLVACCCTL8	TZCTCM_SL	2	2 or 3
Cortex-A55 Auxiliary Register	SLVACCCTL6	CA55_SL	0	0 to 3
Cortex-R52 CPU0 Auxiliary Register	SLVACCCTL6	CR520_SL	0	0 to 3
Cortex-R52 CPU1 Auxiliary Register	SLVACCCTL6	CR521_SL	0	0 to 3
System Control (Reset, Clock, CLMA, Low Power)	SLVACCCTL8	SYSCTRL_SL	0	0 to 3

Table 13.10 Bus Slave supporting TZS and Initial Setting and Configurable Range of Slave Access Control Register (2 of 4)

Bus Slave	Slave Access Control Register			
	Register Name	Bit	Initial Setting	Configurable range
CoreSight	SLVACCCTL7	CSDBAPB_SL	0	0 to 3
Register Write Protection	SLVACCCTL8	PRCR_SL	0	0 to 3
ICU	SLVACCCTL8	ICU_SL	0	0 to 3
GIC600	SLVACCCTL7	GIC600_SL	0	0 to 3
Bus	SLVACCCTL7	BUS_SL	0	0 to 3
MPU0 (GMAC1)	SLVACCCTL5	MPUGMAC1_SL	0	0 to 3
MPU1 (GMAC2)	SLVACCCTL5	MPUGMAC2_SL	0	0 to 3
MPU2, MPU3 (USBHC, USBf)	SLVACCCTL5	MPUUSB_SL	0	0 to 3
MPU4 (SDHI0)	SLVACCCTL5	MPUSDHI0_SL	0	0 to 3
MPU5 (SDHI1)	SLVACCCTL5	MPUSDHI1_SL	0	0 to 3
MPU6 (LCDC)	SLVACCCTL5	MPULCDC_SL	0	0 to 3
MPU7 (PCIE0)	SLVACCCTL5	MPUPCIE0_SL	0	0 to 3
MPU8 (PCIE1)	SLVACCCTL5	MPUPCIE1_SL	0	0 to 3
MPU9, 15 (CoreSight AXI-AP, ETR)	SLVACCCTL9	MPUCS_SL	0	0 to 3
MPU10 (DMAC0)	SLVACCCTL9	MPUDMAC0_SL	0	0 to 3
MPU11 (DMAC1)	SLVACCCTL9	MPUDMAC1_SL	0	0 to 3
MPU12 (DMAC2)	SLVACCCTL9	MPUDMAC2_SL	0	0 to 3
MPU13 (GMAC0)	SLVACCCTL9	MPUGMAC0_SL	0	0 to 3
MPU14 (SHOSTIF)	SLVACCCTL9	MPUSHOSTIF_SL	0	0 to 3
CPU_CTRL	—	—	2	2 fixed
Address Expander	SLVACCCTL9	ADDEXP_SL	0	0 to 3
DMAC unit 0	SLVACCCTL6	DMAC0_SL	0	0 to 3
DMAC unit 1	SLVACCCTL6	DMAC1_SL	0	0 to 3
DMAC unit 2	SLVACCCTL6	DMAC2_SL	0	0 to 3
LLPP Peripherals (Main bus port)	SLVACCCTL7	LLPP_SL	0	0 to 3
GPT unit 9	SLVACCCTL0	GPT09_SL	0	0 to 3
GPT unit 10	SLVACCCTL2	GPT09_SL	0	0 to 3
POEG unit 1	SLVACCCTL0	POEG1_SL	0	0 to 3
POEG unit 2	SLVACCCTL2	POEG2_SL	0	0 to 3
CMT unit 0	SLVACCCTL0	CMT0_SL	0	0 to 3
CMT unit 1	SLVACCCTL0	CMT1_SL	0	0 to 3
CMT unit 2	SLVACCCTL0	CMT2_SL	0	0 to 3
CMTW unit 0	SLVACCCTL0	CMTW0_SL	0	0 to 3
CMTW unit 1	SLVACCCTL0	CMTW1_SL	0	0 to 3
WDT unit 0	SLVACCCTL0	WDT0_SL	0	0 to 3
WDT unit 1	SLVACCCTL0	WDT1_SL	0	0 to 3
WDT unit 2	SLVACCCTL0	WDT2_SL	0	0 to 3
WDT unit 3	SLVACCCTL0	WDT3_SL	0	0 to 3
WDT unit 4	SLVACCCTL0	WDT4_SL	0	0 to 3
WDT unit 5	SLVACCCTL0	WDT5_SL	0	0 to 3

Table 13.10 Bus Slave supporting TZS and Initial Setting and Configurable Range of Slave Access Control Register (3 of 4)

Bus Slave	Slave Access Control Register			
	Register Name	Bit	Initial Setting	Configurable range
RTC	SLVACCCTL2	RTC_SL	0	0 to 3
ETHSS	SLVACCCTL6	ETHSS_SL	0	0 to 3
GMAC unit 0	SLVACCCTL6	GMAC0_SL	0	0 to 3
GMAC unit 1	SLVACCCTL4	GMAC1_SL	0	0 to 3
GMAC unit 2	SLVACCCTL4	GMAC2_SL	0	0 to 3
ETHSW	SLVACCCTL6	ETHSW_SL	0	0 to 3
ESC	SLVACCCTL6	ESC_SL	0	0 to 3
USB	SLVACCCTL4	USB_SL	0	0 to 3
SCI unit 0	SLVACCCTL1	SCI0_SL	0	0 to 3
SCI unit 1	SLVACCCTL1	SCI1_SL	0	0 to 3
SCI unit 2	SLVACCCTL1	SCI2_SL	0	0 to 3
SCI unit 3	SLVACCCTL1	SCI3_SL	0	0 to 3
SCI unit 4	SLVACCCTL1	SCI4_SL	0	0 to 3
SCI unit 5	SLVACCCTL2	SCI5_SL	0	0 to 3
IIC unit 0	SLVACCCTL1	IIC0_SL	0	0 to 3
IIC unit 1	SLVACCCTL1	IIC1_SL	0	0 to 3
IIC unit 2	SLVACCCTL2	IIC2_SL	0	0 to 3
CANFD	SLVACCCTL1	CANFD_SL	0	0 to 3
SPI unit 0	SLVACCCTL1	SPI0_SL	0	0 to 3
SPI unit 1	SLVACCCTL1	SPI1_SL	0	0 to 3
SPI unit 2	SLVACCCTL1	SPI2_SL	0	0 to 3
SPI unit 3	SLVACCCTL2	SPI3_SL	0	0 to 3
xSPI unit 0	SLVACCCTL8	XSPI0_SL	0	0 to 3
xSPI unit 1	SLVACCCTL8	XSPI1_SL	0	0 to 3
CRC unit 0	SLVACCCTL1	CRC0_SL	0	0 to 3
CRC unit 1	SLVACCCTL2	CRC1_SL	0	0 to 3
ADC unit 2	SLVACCCTL0	ADC2_SL	0	0 to 3
TSU	SLVACCCTL1	TSU_SL	0	0 to 3
DOC	SLVACCCTL1	DOC_SL	0	0 to 3
SYSRAM unit 0	SLVACCCTL6	SYSRAM0_SL	0	0 to 3
SYSRAM unit 1	SLVACCCTL6	SYSRAM1_SL	0	0 to 3
SYSRAM unit 2	SLVACCCTL6	SYSRAM2_SL	0	0 to 3
SYSRAM unit 3	SLVACCCTL6	SYSRAM3_SL	0	0 to 3
RSIP	SLVACCCTL7	RSIP_SL	2	0 to 3
OTP	SLVACCCTL7	OTP_SL	2	0 to 3
SHOSTIF	SLVACCCTL7	SHOSTIF_SL	0	0 to 3
MBXSEM	SLVACCCTL7	MBXSEM_SL	0	0 to 3
LCDC	SLVACCCTL4	LCDC_SL	0	0 to 3
SDHI unit 0	SLVACCCTL4	SDHI0_SL	0	0 to 3
SDHI unit 1	SLVACCCTL4	SDHI1_SL	0	0 to 3

Table 13.10 Bus Slave supporting TZS and Initial Setting and Configurable Range of Slave Access Control Register (4 of 4)

Bus Slave	Slave Access Control Register			
	Register Name	Bit	Initial Setting	Configurable range
DDR Control	SLVACCCTL8	DDRCTRL_SL	0	0 to 3

Note: Corresponding slave access control register for some auxiliary registers may be different from the above table. For detail, see register map of each section.

13.4.4.3 TZC-400

TrustZone Address Space Controller (TZC-400) provides the function to realize secure memory access. It performs security checks on transactions to memory basically. Transactions must meet security requirements to access memory or peripherals. This is an IP ("CoreLink™ TrustZone Address Space Controller TZC-400") provided by Arm. Table 13.11 shows combination of region ID access register setting and transaction filtering rule. Table 13.12 lists TZC-400 modules in this LSI. After reset is released, TZC-400 starts up with rejection of all the transaction. Set the control register of TZC-400 to proper access permission by secure transaction so that master can access the memory area with proper access attribute. Since Cortex-R52 AXIS interface has own privilege setting, access level must be match between Cortex-R52 AXIS and TZC-400-8.

For details on the functions of TZC-400, see the relevant Technical Reference Manual.

Table 13.11 Transaction filtering rule for TZC-400

Region ID access register nsaid_wr_en[31:16] nsaid_rd_en[15:0]	Transaction			
	Non-secure		Secure	
	Unprivileged	Privileged	Unprivileged	Privileged
0x000F	Access allowed	Access allowed	Access allowed	Access allowed
0x000B	—	Access allowed	Access allowed	Access allowed
0x0003	—	—	Access allowed	Access allowed
0x0002	—	—	—	Access allowed

Setting other than values shown in the above table to the region ID access register is prohibited.

Table 13.12 TZC-400 Modules (1 of 2)

Symbol	Filter unit number	Target memory area	Base address of control register	Size
TZC-400-0	0	DDR SDRAM A0 I/F	0x8110_0000	4 KB
	1	DDR SDRAM A1 I/F		
TZC-400-1	0	DDR SDRAM A4 I/F	0x8110_1000	4 KB
TZC-400-2	0	PCIE Unit 0	0x8110_2000	4 KB
	1	PCIE Unit 1		
TZC-400-3	0	DDR SDRAM R2 I/F	0x8110_3000	4 KB
TZC-400-4	0	DDR SDRAM R3 I/F	0x8110_4000	4 KB
TZC-400-5	0	SYSRAM Unit 0	0x8110_5000	4 KB
	1	SYSRAM Unit 1		
	2	SYSRAM Unit 2		
	3	SYSRAM Unit 3		
TZC-400-6	0	BSC	0x8110_6000	4 KB
TZC-400-7	0	xSPI Unit 0	0x8110_7000	4 KB
	1	xSPI Unit 1		

Table 13.12 TZC-400 Modules (2 of 2)

Symbol	Filter unit number	Target memory area	Base address of control register	Size
TZC-400-8	0	Cortex-R52 CPU0 AXIS (TCM)	0x8110_8000	4 KB
	1	Cortex-R52 CPU1 AXIS (TCM)		

13.4.5 Address Expander

This LSI supports 35-bit address to access 32 GB area. Bus masters except Cortex-A55, CoreSight, and PCIE can handle 32-bit address space only. Address expander enables these 32-bit addressing bus masters to access over 4 GB area.

Over 4 GB area are DDR (0x2_0000_0000 to 0x3_FFFF_FFFF), PCIE0 (0x4_0000_0000 to 0x5_FFFF_FFFF), and PCIE1 (0x6_0000_0000 to 0x7_FFFF_FFFF). Mirror areas of these areas are available within 32-bit address space. Any 256 MB area from PCIE0 area, any 256 MB area from PCIE1 area, and any two 512 MB area can be mirrored to PCIE0 mirror (0xA000_0000 to 0xAFFF_FFFF), PCIE1 mirror (0xB000_0000 to 0xBFFF_FFFF), and DDR mirror 0 and 1 (0xC000_0000 to 0xDFFF_FFFF and 0xE000_0000 to 0xFFFF_FFFF) respectively by ADXCTLn register each 32-bit bus master. All the bus masters can access DDR area but only Cortex-R52 CPU0, CPU1 and DMAC (unit 0 to 2) can access PCIE area. Address boundary of selected area is 256 MB.

Set [32:28] bit of start address of the area to be mirrored to target bits of the ADXCTLn register each 32-bit bus master. Figure 13.4 shows the mirroring image when DDRMIR0[4:0] bits in ADXCTLn register is set to 10111b.

Note that Cortex-A55, CoreSight and PCIE can access DDR, PCIE0 and PCIE1 via not only over 4 GB area but also mirror area in 32-bit address space whose reference address is fixed, not configurable by address expander.

Table 13.13 Correspondence between master module and Address Expander

Symbol	Master module	PCIE0 mirror (256 MB) (0xA000_0000 to 0xAFFF_FFFF)	PCIE1 mirror (256 MB) (0xB000_0000 to 0xBFFF_FFFF)	DDR mirror 0 (512 MB) (0xC000_0000 to 0xDFFF_FFFF)	DDR mirror 1 (512 MB) (0xE000_0000 to 0xFFFF_FFFF)
ADX0	Cortex-R52 CPU0	Available	Available	Available	Available
ADX1	Cortex-R52 CPU1	Available	Available	Available	Available
ADX2	DMAC unit 0	Available	Available	Available	Available
ADX3	DMAC unit 1	Available	Available	Available	Available
ADX4	DMAC unit 2	Available	Available	Available	Available
ADX5	GMAC unit 0	—	—	Available	Available
ADX6	GMAC unit 1	—	—	Available	Available
ADX7	GMAC unit 2	—	—	Available	Available
ADX8	USBHC	—	—	Available	Available
ADX9	USBf	—	—	Available	Available
ADX10	SDHI unit 0	—	—	Available	Available
ADX11	SDHI unit 1	—	—	Available	Available
ADX12	LCDC	—	—	Available	Available
—	Cortex-A55	Available*1	Available*2	Available*3	
—	PCIE unit 0	—	—	Available*3	
—	PCIE unit 1	—	—	Available*3	
—	CoreSight (AXI-AP)	Available*1	Available*2	Available*3	
—	CoreSight (ETR)	—	—	—	—

Note 1. Reference address is not configurable. Fixed to 0x4_0000_0000 to 0x4_0FFF_FFFF.
Note 2. Reference address is not configurable. Fixed to 0x6_0000_0000 to 0x6_0FFF_FFFF.
Note 3. Reference address is not configurable. Fixed to 0x2_0000_0000 to 0x2_3FFF_FFFF.

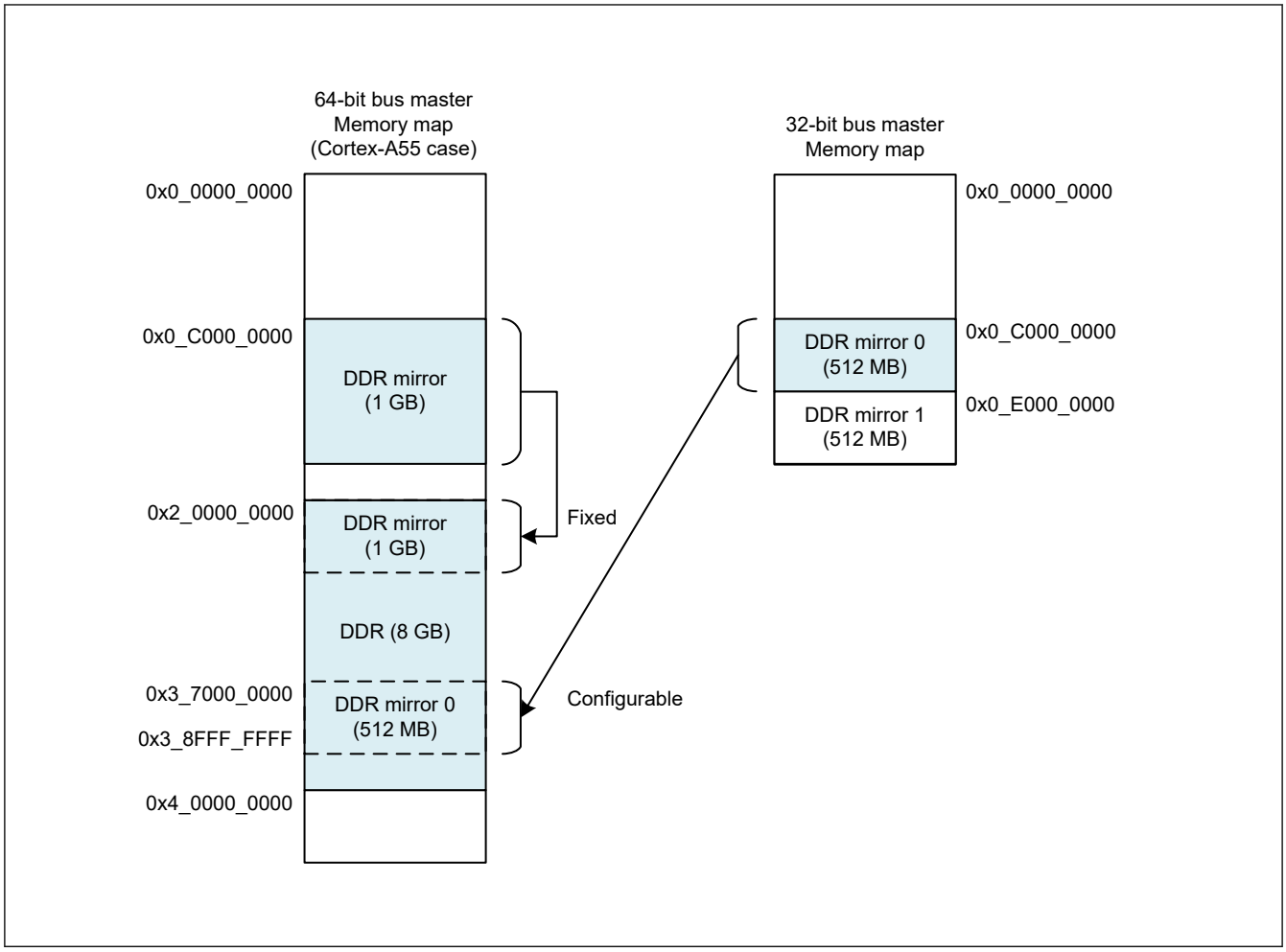


Figure 13.4 Mirroring image of the DDR area to 32-bit bus master

13.4.6 Interrupt

When a bus error, an MPU error, or TrustZone error occurs by reading or writing to the protected region, interrupt can be output. Table 13.14 lists the internal bus, master MPU, and TrustZone error interrupt sources. These interrupts are combined with peripheral error interrupts (PERI_ERR0 and PERI_ERR1) as error events with other error interrupts.

Table 13.14 Internal bus, master MPU, and TrustZone error interrupt sources (1 of 2)

Name	Interrupt sources
MPU_DMACR0	Master MPU10 Read channel for DMAC0 error
MPU_DMACW0	Master MPU10 Write channel for DMAC0 error
MPU_DMACR1	Master MPU11 Read channel for DMAC1 error
MPU_DMACW1	Master MPU11 Write channel for DMAC1 error
MPU_DMACR2	Master MPU12 Read channel for DMAC2 error
MPU_DMACW2	Master MPU12 Write channel for DMAC2 error
MPU_GMACR0	Master MPU13 Read channel for GMAC0 error
MPU_GMACW0	Master MPU13 Write channel for GMAC0 error
MPU_GMACR1	Master MPU0 Read channel for GMAC1 error
MPU_GMACW1	Master MPU0 Write channel for GMAC1 error
MPU_GMACR2	Master MPU1 Read channel for GMAC2 error
MPU_GMACW2	Master MPU1 Write channel for GMAC2 error
MPU_USBH	Master MPU2 for USB Host error

Table 13.14 Internal bus, master MPU, and TrustZone error interrupt sources (2 of 2)

Name	Interrupt sources
MPU_USBF	Master MPU3 for USB Function error
MPU_DBGR	Master MPU15 Read channel for AXI-AP error
MPU_DBGW	Master MPU15 Write channel for AXI-AP error
MPU_ETRR	Master MPU9 Read channel for ETR error
MPU_ETRW	Master MPU9 Write channel for ETR error
MPU_SHOSTIF	Master MPU14 for Serial Host I/F error
MPU_LCDCR	Master MPU6 Read channel for LCDC error
MPU_LCDCW	Master MPU6 Write channel for LCDC error
MPU_SDHIR0	Master MPU4 Read channel for SDHI0 error
MPU_SDHIW0	Master MPU4 Write channel for SDHI0 error
MPU_SDHIR1	Master MPU5 Read channel for SDHI1 error
MPU_SDHIW1	Master MPU5 Write channel for SDHI1 error
MPU_PCIER0	Master MPU7 Read channel for PCIE0 error
MPU_PCIEW0	Master MPU7 Write channel for PCIE0 error
MPU_PCIER1	Master MPU8 Read channel for PCIE1 error
MPU_PCIEW1	Master MPU8 Write channel for PCIE1 error
CA55MPUR	Cortex-A55 read error to Master MPU registers
CA55MPUW	Cortex-A55 write error to Master MPU registers
CR521MPUR	Cortex-R52 CPU1 read error to Master MPU registers
CR521MPUW	Cortex-R52 CPU1 write error to Master MPU registers
MAINBUSA_ERRINT	Internal main bus A error
MAINBUSR_ERRINT	Internal main bus R error
LLPPBUS_ERRINT	LLPP bus error
TZC_DDR_A0A1	TZC-400-0 for DDR A0 and A1 I/F error
TZC_DDR_A4	TZC-400-1 for DDR A4 I/F error
TZC_PCIE	TZC-400-2 for PCIE error
TZC_DDR_R2	TZC-400-3 for DDR R2 I/F error
TZC_DDR_R3	TZC-400-4 for DDR R3 I/F error
TZC_SYSRAM	TZC-400-5 for SYSRAM error
TZC_BSC	TZC-400-6 for BSC error
TZC_XSPI	TZC-400-7 for XSPI error
TZC_TCM	TZC-400-8 for TCM error
MAINBUSA_SEC_ERRINT	Internal main bus A error (TZM and TZS)
MAINBUSR_SEC_ERRINT	Internal main bus R error (TZM and TZS)
PERI_SEC_ERRINT	LLPP0, LLPP1, NONSAFETY, SAFETY Peripheral error (TZS)

Note: Setting of the Action register in TZC-400 is needed to generate TZC-400 error interrupt.

13.5 Usage Notes

13.5.1 Release of Register Write Protection

The Master MPU registers are write protected. To release write protection, follow the steps below.

1. Write 0x0000A508 to PRCRS register (release write protection using PRCS3 bit).

2. Write value to the STADDm, ENDADDm, ERRINF, and ERRINF_x registers.
3. Write 0x0000A500 to PRCRS register (set write protection with PRCS3 bit).

13.5.2 Initial Setting of Master MPU

The initial setting of region 0 specifies all the area of memory map with read/write protection for every master MPU. This means the master modules cannot access any of the bus slaves. The Master MPU register must be changed from Cortex-R52 CPU0 before operating master modules. By setting CPU_CTRL register by Cortex-R52 CPU0, Cortex-A55 and Cortex-R52 CPU1 can also change the Master MPU registers in case of Cortex-R52 CPU0 boot mode.

13.5.3 New MPU Error Detection While Preserving Valid Error Information

When master MPU detects new error access while preserving valid error information (VALID bit = 1 in Master MPU error information registers), the error information is not updated to the new one unless the VALID bit is cleared to 0 by writing 0. Nevertheless, access to the safety region is protected. Interrupt is also not regenerated while the VALID bit is 1.

13.5.4 TCM Access via AXIS Interface of Cortex-R52

Bus masters can access to TCMs of Cortex-R52 via AXIS interface in Cortex-R52. AXIS has some limitations for data size and address alignment. Table 13.15 shows the limitation of TCM access via AXIS interface each bus master and Table 13.16 shows register setting to access TCM under limitation.

Table 13.15 Limitation of TCM access via AXIS interface

Bus master	Description
Cortex-A55 via ACEM Cortex-R52 CPU0 via AXIM Cortex-R52 CPU1 via AXIM	Instruction fetch is not possible. Data access Read ● Possible if region attribution is set to Device and LD instruction (8, 16, or 32 bits) is used to access. Write ● Possible if region attribution is set to Device and ST instruction (8, 16, or 32 bits, or 64 bits with 64-bit address alignment) is used to access.
DMAC0	Descriptor access is not possible. Data access Read ● Possible if transfer size is 8, 16, 32, 64, 128, or 512 bits. Write ● Possible if transfer size is 8, 16, 32, 64, or 128 bits. ● Possible if transfer size is 512 bits and access address is 64-byte aligned. 256-bits transfer is not possible.
DMAC1, 2	Descriptor access is not possible. Data access Read and Write ● Possible if transfer size is 8, 16, 32, 64, or 128 bits. 256-bit transfer is not possible.
GMAC0, 1, 2	Read and Write ● Possible if transfer is 128-bit ×1 beat and access address is 4-byte aligned. Other combination transfer is not possible.
USBHC USBf	Descriptor access is not possible. Data access Read and Write ● Possible if transfer size is 128 bits and access addresses is 16-byte aligned. Other combination transfer is not possible.
PCIE0, 1	Read and Write ● Possible if burst length is 1.
CoreSight (AXI-AP)	No limitation in the supported access.
CoreSight (ETR)	Read ● Possible if access address is 4-byte or 16-byte aligned. Write ● Possible if burst length is 1.

Table 13.16 Register setting to access TCM from each bus master

Bus master	Register	Bit setting
DMAC0	CHCFG_n	DDS = 0x0 (8-bit), 0x1 (16-bit), 0x2 (32-bit), 0x3 (64-bit), 0x4 (128-bit), or 0x6 (512-bit) SDS = 0x0 (8-bit), 0x1 (16-bit), 0x2 (32-bit), 0x3 (64-bit), 0x4 (128-bit), or 0x6 (512-bit)
	CHEXT_n	DCA = 0x2 if CHCFG_n.DDS = 0x4 or 0x6 and destination is TCM SCA = 0x2 if CHCFG_n.SDS = 0x4 or 0x6 and source is TCM
DMAC1, 2	CHCFG_n	DDS = 0x0 (8-bit), 0x1 (16-bit), 0x2 (32-bit), 0x3 (64-bit), or 0x4 (128-bit) SDS = 0x0 (8-bit), 0x1 (16-bit), 0x2 (32-bit), 0x3 (64-bit), or 0x4 (128-bit)
	CHEXT_n	DCA = 0x2 if CHCFG_n.DDS = 0x4 and destination is TCM SCA = 0x2 if CHCFG_n.SDS = 0x4 and source is TCM
GMAC0, 1, 2	DMA_CHn_TX_CONTROL	TxPBL = 0x1 IPBL = 0
	DMA_CHn_RX_CONTROL	RxPBL = 0x1
	DMA_SysBus_Mode	AAL = 0x1 BLEN16 = 0 BLEN8 = 0 BLEN4 = 0 FB = 0x1
USBHC	AHB_BUS_CTR	PROT_TYPE = 1xxxb ALIGN_ADDRESS = 11b
USBf	CHCFG_n	DDS = 0x4 (128-bit) SDS = 0x4 (128-bit)
	CHEXT_n	DPR = 1xxxb SPR = 1xxxb
PCIE	PCI_RC_MSET1 PCI_EP_MSET1	AXIMMB = 0

13.5.5 Exclusive Access

Exclusive access instructions for synchronization and semaphores, and atomic instructions in Cortex-R52 and Cortex-A55 are not supported in the sharable areas. Use mailbox and semaphore (MBXSEM) for exclusive control.

For details of the target instructions, see the below sections in Arm Architecture Reference Manual for A-profile architecture (DDI 0487).

- For AArch64
  - B2.17 Synchronization and semaphores
  - C3.2.12 Atomic instructions
- For AArch32
  - E2.10 Synchronization and semaphores

14. Bus State Controller

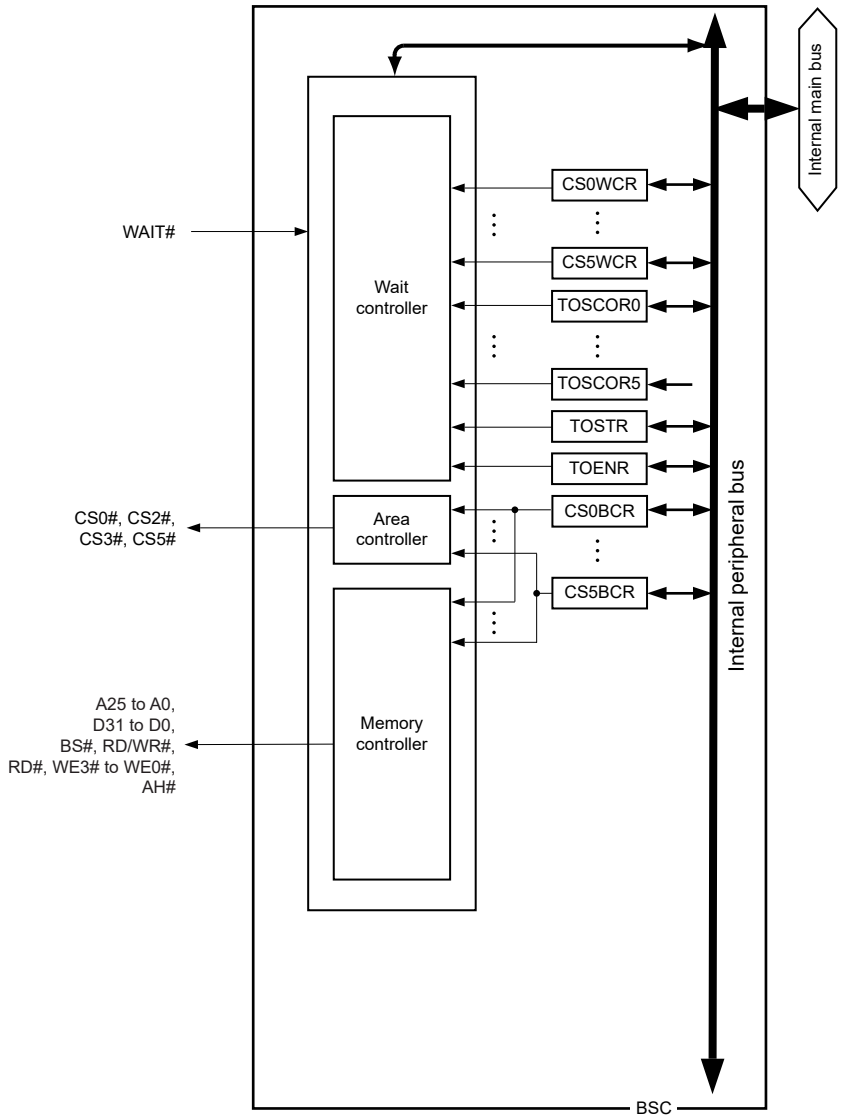
14.1 Overview

The bus state controller outputs control signals for various types of memory and external devices that are connected to the external address space. The functions of this module enable this LSI to connect directly with SRAM, other memory storage devices, and external devices.

Table 14.1 Bus state controller specifications

Item	Description
External address space	• A maximum of 64 MB for each of areas CS0, CS2, CS3, and CS5 • Can specify the following for each CS space: SRAM interface, SRAM interface with byte selection, burst ROM (clocked synchronous or asynchronous), and MPX-I/O. – Data bus width (8, 16, or 32 bits) Insertion of wait cycles for each read access and write access. • Can set independent idle cycles for the following five cases: Read-write (in same space/different spaces), read-read (in same space/different spaces), and the first cycle is a write access.
Various interfaces	• SRAM interface Supports the interface that can directly connect to the SRAM.
	• Burst ROM interface (clocked asynchronous) High-speed access to the ROM that has the page mode function.
	• MPX-I/O interface Can directly connect to a peripheral LSI that needs an address/data multiplexing.
	• SRAM interface with byte selection Can connect directly to an SRAM with byte selection.
	• Burst ROM interface (clocked synchronous) Can connect directly to a burst ROM of the clocked synchronous type.
External WAIT# pin	• Detection of long wait state for access by the signal on the external WAIT# pin • A timeout detection condition is specifiable per CS space • Once timeout is detected, the external WAIT function is disabled and an external wait timeout detection interrupt is issued

Figure 14.1 is a block diagram of the bus state controller.



Note: CSnWCR: CSn Space Wait Control Register (n = 0, 2, 3, 5)
CSnBCR: CSn Space Bus Control Register (n = 0, 2, 3, 5)
TOSCORn: Timeout Cycle Constant Register (n = 0, 2, 3, 5)
TOSTR: Timeout Status Register
TOENR: Timeout Enable Register

Figure 14.1 Block diagram of bus state controller

Table 14.2 lists the input/output pins of the bus state controller.

Table 14.2 Input/output pins of the bus state controller (1 of 2)

Name	I/O	Function
A25 to A0	Output	Address output pins
D31 to D0	I/O	Data input/output pins
BS#	Output	Status signal output pin that indicates the start of the bus cycle
CS0#, CS2#, CS3#, CS5#	Output	Chip select signal output pins
RD/WR#	Output	Strobe signal output pin that indicates a read or write access Connect this pin to WE# pins when SRAM with byte selection is connected.

Table 14.2 Input/output pins of the bus state controller (2 of 2)

Name	I/O	Function
RD#	Output	Strobe signal output pin that indicates a read signal (read data output enable signal)
WE3#/AH#	Output	Write strobe signal output pin for D31 to D24 Connect this pin to the byte select pin when SRAM with byte selection is connected. Functions as the address hold signal output pin when the MPX-I/O interface is used.
WE2#	Output	Write strobe signal output pin for D23 to D16 Connect this pin to the byte select pin when an SRAM with byte selection is connected.
WE1#	Output	Write strobe signal output pin for D15 to D8 Connect this pin to the byte select pin when SRAM with byte selection is connected.
WE0#	Output	Write strobe signal output pin for D7 to D0 Connect this pin to the byte select pin when SRAM with byte selection is connected.
WAIT#	Input	External wait control signal input pin used to insert a wait state into a bus cycle

14.2 Area Overview

14.2.1 Address Map

The kind of memory to be connected to the external address space and the data bus width are specified in each of CS0, CS2, CS3, and CS5 spaces. The address map for the external address space is listed below.

Table 14.3 Address map

Internal address	Space	Memory to be connected
0x7000_0000 to 0x73FF_FFFF	CS0	SRAM interface, SRAM with byte selection, burst ROM (asynchronous or synchronous)
0x7400_0000 to 0x77FF_FFFF	CS2	SRAM interface, SRAM with byte selection
0x7800_0000 to 0x7BFF_FFFF	CS3	SRAM interface, SRAM with byte selection
0x7C00_0000 to 0x7FFF_FFFF	CS5	SRAM interface, SRAM with byte selection, MPX-I/O

14.3 Register Map

Table 14.4 BSC register map

Address	Register symbol	Register name	Write protection
0x8021_0004 + 0x4 × n	CSnBCR	CSn Space Bus Control Register (n = 0, 2, 3, 5)	—
0x8021_0028 + 0x4 × n	CSnWCR	CSn Space Wait Control Register (n = 0, 2, 3, 5)	—
0x8021_0060 + 0x4 × n	TOSCORn	Timeout Cycle Constant Register n (n = 0, 2, 3, 5)	—
0x8021_0080	TOSTR	Timeout Status Register	—
0x8021_0084	TOENR	Timeout Enable Register	—

Table 14.5 BSC related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
All the registers and memory area	—	MSTPCRA.MSTPCRA00	SLVACCCTL8.TZCBSC_SL

14.4 Register Descriptions

14.4.1 CSnBCR : CSn Space Bus Control Register (n = 0, 2, 3, 5)

Base address: BSC = 0x8021_0000

Offset address: 0x004 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2:0]			IWRRD[2:0]			IWRRS[2:0]		
Value after reset:	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	TYPE[2:0]			—	BSZ[1:0]		—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	—	These bits are read as 0. The write value should be 0.	R/W
10:9	BSZ[1:0]	Data Bus Width Specification Specify the data bus width of CS spaces. For MPX-I/O, selects bus width by address. ● If area 5 is specified as MPX-I/O, setting these bits to 11b selects 8-bit or 16-bit bus width according to the address settings of the SZSEL bit in CS5WCR. Setting these bits to 01b or 10b selects the fixed bus width of 8-bit or 16-bit, respectively. ● If area 0 is specified as clocked synchronous burst ROM space, the bus width can be specified as either 16 bits or 32 bits. 0 0: Setting prohibited 0 1: 8-bit size 1 0: 16-bit size 1 1: 32-bit size	R/W
11	—	This bit is read as 1. The write value should be 1.	R/W
14:12	TYPE[2:0]	Memory Connected to a Space Specify the type of memory connected to a CS space. For details for memory type in each area, see Table 14.3. 0 0 0: Normal space 0 0 1: Burst ROM (clock asynchronous) 0 1 0: MPX-I/O 0 1 1: SRAM with byte selection 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Burst ROM (clock synchronous)	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	IWRRS[2:0]	Idle State Insertion between Read-Read Cycles in the Same CS Space Specify the number of idle cycles to be inserted after the access to an external memory that is connected to the CS space. The target cycle is a read-read cycle of which continuous access cycles are for the same CS space. 0 0 0: No idle cycle inserted 0 0 1: 1 idle cycle inserted 0 1 0: 2 idle cycles inserted 0 1 1: 4 idle cycles inserted 1 0 0: 6 idle cycles inserted 1 0 1: 8 idle cycles inserted 1 1 0: 10 idle cycles inserted 1 1 1: 12 idle cycles inserted	R/W

Bit	Symbol	Function	R/W
21:19	IWRRD[2:0]	Idle State Insertion between Read-Read Cycles in Different CS Spaces Specify the number of idle cycles to be inserted after the access to an external memory that is connected to the CS space. The target cycle is a read-read cycle of which continuous access cycles switch between different CS spaces. 0 0 0: No idle cycle inserted 0 0 1: 1 idle cycle inserted 0 1 0: 2 idle cycles inserted 0 1 1: 4 idle cycles inserted 1 0 0: 6 idle cycles inserted 1 0 1: 8 idle cycles inserted 1 1 0: 10 idle cycles inserted 1 1 1: 12 idle cycles inserted	R/W
24:22	IWRWS[2:0]	Idle State Insertion between Read-Write Cycles in the Same CS Space Specify the number of idle cycles to be inserted after the access to an external memory that is connected to the CS space. The target cycle is a read-write cycle of which continuous access cycles are for the same CS space. 0 0 0: No idle cycle inserted 0 0 1: 1 idle cycle inserted 0 1 0: 2 idle cycles inserted 0 1 1: 4 idle cycles inserted 1 0 0: 6 idle cycles inserted 1 0 1: 8 idle cycles inserted 1 1 0: 10 idle cycles inserted 1 1 1: 12 idle cycles inserted	R/W
27:25	IWRWD[2:0]	Idle State Insertion between Read-Write Cycles in Different CS Spaces Specify the number of idle cycles to be inserted after the access to an external memory that is connected to the CS space. The target access cycle is a read-write one in which continuous access cycles switch between different CS spaces. 0 0 0: No idle cycle inserted 0 0 1: 1 idle cycle inserted 0 1 0: 2 idle cycles inserted 0 1 1: 4 idle cycles inserted 1 0 0: 6 idle cycles inserted 1 0 1: 8 idle cycles inserted 1 1 0: 10 idle cycles inserted 1 1 1: 12 idle cycles inserted	R/W
30:28	IWW[2:0]	Idle Cycles between Write-Read Cycles and Write-Write Cycles These bits specify the number of idle cycles to be inserted after the access to an external memory that is connected to the CS space. The target access cycles are the write-read cycle and write-write cycle. 0 0 0: No idle cycle inserted 0 0 1: 1 idle cycle inserted 0 1 0: 2 idle cycles inserted 0 1 1: 4 idle cycles inserted 1 0 0: 6 idle cycles inserted 1 0 1: 8 idle cycles inserted 1 1 0: 10 idle cycles inserted 1 1 1: 12 idle cycles inserted	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The CSnBCR register specifies the memory connected to each CS space, the number of idle cycles between bus cycles, and the bus width.

Do not access external memory for the corresponding area until CSnBCR initial setting and pin setting that are required for accessing the external memory are completed.

Idle cycles may be inserted even when they are not specified. For details, see section 14.5.9. Wait between Access Cycles.

14.4.2 CSn Space Wait Control Register (CSnWCR) (n = 0, 2, 3, 5)

The CSnWCR register specifies various wait cycles for external memory access. The bit configuration of this register varies as shown below according to the memory type (TYPE2 to TYPE0) specified by the CSn space bus control register (CSnBCR). Specify CSnWCR before accessing the target area. Specify CSnBCR first, then specify CSnWCR.

14.4.2.1 CS0WCR_0 : CS0 Space Wait Control Register for Normal Space, SRAM with Byte Selection

Base address: BSC = 0x8021_0000

Offset address: 0x028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	BAS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SW[1:0]		WR[3:0]				WM	—	—	—	—	HW[1:0]	
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	HW[1:0]	Delay States from RD#, WEn# Negation to Address, CS0# Negation Specify the number of delay states from RD# and WEn# inactive to address and CS0# inactive. 0 0: 0.5 cycles 0 1: 1.5 cycles 1 0: 2.5 cycles 1 1: 3.5 cycles	R/W
5:2	—	These bits are read as 0. The write value should be 0.	R/W
6	WM	External Wait Mask Specification Specifies whether the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored	R/W
10:7	WR[3:0]	Number of Access Waits Specify the number of wait insertions that are necessary for read/write access. 0x0: No wait insertion 0x1: 1 cycle 0x2: 2 cycles 0x3: 3 cycles 0x4: 4 cycles 0x5: 5 cycles 0x6: 6 cycles 0x7: 8 cycles 0x8: 10 cycles 0x9: 12 cycles 0xA: 14 cycles 0xB: 18 cycles 0xC: 24 cycles Others: Setting prohibited	R/W
12:11	SW[1:0]	Number of Delay Cycles from Address, CSn# Assertion to RD#, WEn# Assertion Specify the number of delay cycle states from address and CS0# active to RD# and WEn# active. 0 0: 0.5 cycles 0 1: 1.5 cycles 1 0: 2.5 cycles 1 1: 3.5 cycles	R/W
19:13	—	These bits are read as 0. The write value should be 0.	R/W
20	BAS*1	SRAM with Byte Selection Byte Access Select Specifies the WE# and RD/WR# signal timing when the SRAM interface with byte selection is used. 0: Activates the WEn# signal at the read/write timing and activates the RD/WR# signal during the write access cycle. 1: Activates the WEn# signal during the read/write access cycle and activates the RD/WR# signal at the write timing.	R/W

Bit	Symbol	Function	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When BAS = 0, the write access timing of the SRAM interface with byte selection is the same as that of the SRAM interface. In write operation, data is written to the memory when the byte select pins (WEn# (n = 0 to 3)) are active. In read operation, the active timing of the WEn# (n = 0 to 3) pins are different from that of the SRAM interface and the byte selection signal is output from the WEn# (n = 0 to 3) pins. When BAS = 1, the active timing of the WEn# (n = 0 to 3) and RD/WR# pins varies. In write operation, data is written to the memory when the write enable pin (RD/WR#) is active. Set HW[1:0] for the write data hold time to the RD/WR# inactive timing to ensure the hold time. For details of access timing, see section 14.5.7. SRAM Interface with Byte Selection.

14.4.2.2 CSnWCR_0 : CSn Space Wait Control Register for Normal Space, SRAM with Byte Selection (n = 2, 3)

Base address: BSC = 0x8021_0000

Offset address: 0x030 + 0x4 × (n - 2)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	BAS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	WR[3:0]				WM	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	WM	External Wait Mask Specification Specifies whether the external wait input is valid. The specification by this bit is valid even when the number of access waits is 0. 0: External wait input is valid 1: External wait input is ignored	R/W
10:7	WR[3:0]	Number of Access Waits Specify the number of waits that are necessary for read/write access. 0x0: No wait insertion 0x1: 1 cycle 0x2: 2 cycles 0x3: 3 cycles 0x4: 4 cycles 0x5: 5 cycles 0x6: 6 cycles 0x7: 8 cycles 0x8: 10 cycles 0x9: 12 cycles 0xA: 14 cycles 0xB: 18 cycles 0xC: 24 cycles Others: Setting prohibited	R/W
19:11	—	These bits are read as 0. The write value should be 0.	R/W
20	BAS	SRAM with Byte Selection Byte Access Select Specifies the WEn# and RD/WR# signal timing when the SRAM interface with byte selection is used. 0: Activates the WEn# signal at the read/write timing and activates the RD/WR# signal during the write access cycle. 1: Activates the WEn# signal during the read/write access cycle and activates the RD/WR# signal at the write timing.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

14.4.2.3 CS5WCR : CS5 Space Wait Control Register for Normal Space, SRAM with Byte Selection, and MPX-I/O

Base address: BSC = 0x8021_0000

Offset address: 0x03C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	SZSEL	MPXW SBAS	—	WW[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SW[1:0]		WR[3:0]				WM	—	—	—	—	HW[1:0]	
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	HW[1:0]	Delay Cycles from RD#, WEn# to Address, CS5# Specify the number of delay states from RD# and WEn# inactive to address and CS5# inactive when area 5 is specified as the SRAM interface or the SRAM interface with byte selection. When area 5 is specified as the MPX-I/O interface, these bits specify the number of delay states from RD# and WEn# inactive to CS5# inactive. 0 0: 0.5 states 0 1: 1.5 states 1 0: 2.5 states 1 1: 3.5 states	R/W
5:2	—	These bits are read as 0. The write value should be 0.	R/W
6	WM	External Wait Mask Specification Specifies whether the external wait input is valid. The specification by this bit is valid even when the number of access wait is 0. 0: External wait input is valid 1: External wait input is ignored	R/W
10:7	WR[3:0]	Number of Read Access Waits Specify the number of waits that are necessary for read access. 0x0: No wait insertion 0x1: 1 wait 0x2: 2 waits 0x3: 3 waits 0x4: 4 waits 0x5: 5 waits 0x6: 6 waits 0x7: 8 waits 0x8: 10 waits 0x9: 12 waits 0xA: 14 waits 0xB: 18 waits 0xC: 24 waits Others: Setting prohibited	R/W
12:11	SW[1:0]	Number of Delay Cycles from Address, CS5# Assertion to RD#, WEn# Assertion Specify the number of delay states from address and CS5# active to RD# and WEn# active when area 5 is specified as the SRAM interface or the SRAM interface with byte selection. When area 5 is specified as the MPX-I/O interface, these bits specify the number of delay states from the end of the address cycle (Ta3) to RD# and WEn# active. 0 0: 0.5 states 0 1: 1.5 states 1 0: 2.5 states 1 1: 3.5 states	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
18:16	WW[2:0]	Number of Write Access Waits Specify the number of wait insertions that are necessary for write access. 0 0 0: The same as WR[3:0] setting (number of read access waits) 0 0 1: No wait insertion 0 1 0: 1 wait 0 1 1: 2 waits 1 0 0: 3 waits 1 0 1: 4 waits 1 1 0: 5 waits 1 1 1: 6 waits	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	MPXWSBAS	MPX-I/O Interface Address Cycle Wait and SRAM with Byte Selection Byte Access Select CS5BCR.TYPE[2:0] = 010b specifies the address cycle wait for MPX-I/O interface. This bit (MPXW) setting is valid only when area 5 is specified as MPX-I/O. CS5BCR.TYPE[2:0] = 011b specifies the WEn# and RD/WR# signal timing when the SRAM interface with byte selection is used. This bit (BAS) setting is valid only when area 5 is specified as SRAM with byte selection. 0: CS5BCR.TYPE[2:0] = 010b: Inserts no wait cycle CS5BCR.TYPE[2:0] = 011b: Activates the WEn# signal at the read/write timing and activates the RD/WR# signal during the write access cycle. 1: CS5BCR.TYPE[2:0] = 010b: Inserts 1 wait cycle CS5BCR.TYPE[2:0] = 011b: Activates the WEn# signal during the read/write access cycle and activates the RD/WR# signal at the write timing.	R/W
21	SZSEL	MPX-I/O Interface Bus Width Specification Specifies an address to select the bus width when the BSZ[1:0] of the CS5BCR register are specified as 11b. This bit is valid only when area 5 is specified as MPX-I/O. The relationship between the SZSEL bit and bus width selected by A14 or A21 are summarized in the following table.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

SZSEL	A14	A21	Bus Width
0	0	Not affected	8 bits
	1	Not affected	16 bits
1	Not affected	0	8 bits
	Not affected	1	16 bits

14.4.2.4 CS0WCR_1 : CS0 Space Wait Control Register for Burst ROM with Clocked Asynchronous

Base address: BSC = 0x8021_0000

Offset address: 0x028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	BST[1:0]		—	—	BW[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	W[3:0]				WM	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	WM	External Wait Mask Specification Specifies whether the external wait input is valid. The specification by this bit is valid even when the number of access waits is 0. 0: External wait input is valid 1: External wait input is ignored	R/W
10:7	W[3:0]	Number of Access Waits Specify the number of waits to be inserted in the first access cycle. 0x0: No wait insertion 0x1: 1 wait 0x2: 2 waits 0x3: 3 waits 0x4: 4 waits 0x5: 5 waits 0x6: 6 waits 0x7: 8 waits 0x8: 10 waits 0x9: 12 waits 0xA: 14 waits 0xB: 18 waits 0xC: 24 waits Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
17:16	BW[1:0]	Number of Waits during Burst Access Specify the number of waits to be inserted between the second or subsequent access cycles in burst access. 0 0: No wait insertion 0 1: 1 wait 1 0: 2 waits 1 1: 3 waits	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
21:20	BST[1:0]	Burst Count Specification Specify the burst count for 16-byte or more access. Do not set BST[1:0] to 11b. For details, see Table 14.9.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Bus width	BST[1:0]	Burst count (16-byte access)
8 bits	00b	16 burst × 1 time
	01b	4 burst × 4 times
16 bits	00b	8 burst × 1 time
	01b	2 burst × 4 times
	10b	4-4 or 2-4-2 burst
32 bits	xx	4 burst × 1 time

14.4.2.5 CS0WCR_2 : CS0 Space Wait Control Register for Burst ROM with Clocked Synchronous

Base address: BSC = 0x8021_0000

Offset address: 0x028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BW[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	W[3:0]				WM	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	WM	External Wait Mask Specification Specifies whether the external wait input is valid. The specification by this bit is valid even when the number of access waits is 0. 0: External wait input is valid 1: External wait input is ignored	R/W
10:7	W[3:0]	Number of Access Waits Specify the number of waits to be inserted in the first access cycle. 0x0: No wait insertion 0x1: 1 wait 0x2: 2 waits 0x3: 3 waits 0x4: 4 waits 0x5: 5 waits 0x6: 6 waits 0x7: 8 waits 0x8: 10 waits 0x9: 12 waits 0xA: 14 waits 0xB: 18 waits 0xC: 24 waits Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
17:16	BW[1:0]	Number of Burst Wait Cycles Specify the number of waits to be inserted between the second or subsequent access cycles in burst access. 0 0: No wait insertion 0 1: 1 wait 1 0: 2 waits 1 1: 3 waits	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

14.4.3 TOSCORn : Timeout Cycle Constant Register n (n = 0, 2, 3, 5)

Base address: BSC = 0x8021_0000

Offset address: 0x060 + 0x4 × n

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TOCNUM[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TOCNUM[15:0]	Timeout Cycle Number When the number of external wait cycles matches the setting value of these bits, the external wait timeout detection interrupt occurs and the wait state is forcibly interrupted to end the cycle of access. 0x0000 65536 (number of external wait cycles) : 0x0001 1 : ⋮ 0xFFFF 65535 :	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The TOSCORn register sets the timeout to forcibly interrupt the wait state and end the access cycle of the CSn space. This register is enabled when the WM bit of the CSn Space Wait Control Register (CSnWCR) is 0 and the corresponding bit of the Timeout Enable Register (TOENR) is 1. When the number of wait cycles due to the signal on the external wait input pin matches the setting of TOSCORn, the wait state is forcibly interrupted (external wait input is disabled) to end the cycle of access, the Timeout Status flag for the corresponding space in the Timeout Status Register (TOSTR) is set, and an external wait timeout detection interrupt request from the external WAIT# pin is generated. The external wait timeout detection interrupt request is retained until the corresponding bit in the TOENR register is set to 0, or 0 is written to the Timeout Status flag for the corresponding space. Note that timeout detection is enabled even while the Timeout Status flag for the corresponding space in the TOSTR register is 1, and the wait state is forcibly interrupted (external wait input is disabled) to end the cycle of access in response to a further timeout.

14.4.4 TOSTR : Timeout Status Register

Base address: BSC = 0x8021_0000

Offset address: 0x080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	CS5TOSTF	—	CS3TOSTF	CS2TOSTF	—	CS0TOSTF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CS0TOSTF	CS0 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS0 space has matched the setting of the CS0 space timeout cycle constant register (TOSCOR0). This bit is set or cleared in the following conditions. 0: Clearing condition When 0 is written in CS0TOSTF. 1: Setting condition When the WM bit in the CS0 space wait control register (CS0WCR) is 0 and the CS0TOEN bit in the timeout enable register (TOENR) is 1, the number of cycles of waiting due to the input on the external wait pin during access to the CS0 space has matched the setting of the TOSCOR0 register.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	CS2TOSTF	CS2 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS2 space has matched the setting of the CS2 space timeout cycle constant register (TOSCOR2). For the condition to set or clear this bit, see the description of CS0TOSTF.	R/W

Bit	Symbol	Function	R/W
3	CS3TOSTF	CS3 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS3 space has matched the setting of the CS3 space timeout cycle constant register (TOSCOR3). For the condition to set or clear this bit, see the description of CS0TOSTF.	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	CS5TOSTF	CS5 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS5 space has matched the setting of the CS5 space timeout cycle constant register (TOSCOR5). For the condition to set or clear this bit, see the description of CS0TOSTF.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The TOSTR register is used for status of the external wait input. When the WM bit in the CSn space wait control register (CSnWCR) is 0 and the corresponding bit in the timeout enable register (TOENR) is 1 and the number of waits in response to the signal on the external wait input matches the setting of TOSCORn, the timeout status flag for the corresponding space is set and an external wait timeout detection interrupt request is generated. The only writable value for the timeout status flags is 0, which clears the flag. Writing 1 to a flag is ignored.

14.4.5 TOENR : Timeout Enable Register

Base address: BSC = 0x8021_0000

Offset address: 0x084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	CS5TOEN	—	CS3TOEN	CS2TOEN	—	CS0TOEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CS0TOEN	CS0 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS0 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	CS2TOEN	CS2 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS2 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.	R/W
3	CS3TOEN	CS3 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS3 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	CS5TOEN	CS5 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS5 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The TOENR register is used to enable or disable timeout detection function for each space.

14.5 Operation

14.5.1 Access Size and Data Alignment

This LSI supports little endian, in which the least significant byte (LSB) is that in the direction of the 0th address. Data bus width can be selected from 8 bits, 16 bits, and 32 bits for the normal memory and SRAM with byte selection. For MPX-I/O, the data bus width is fixed to either 8 or 16 bits, or made selectable as 8 bits or 16 bits by one of the address lines.

Data alignment is performed in accordance with the data bus width selected for the device. This also means that four read operations are required to read 32-bit data from a byte-width device. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces.

Table 14.6 to Table 14.8 show the relationship between device data width and access unit.

Table 14.6 32-bit external device access and data alignment in little endian

Operation	Data bus				Strobe signals			
	D31 to D24	D23 to D16	D15 to D8	D7 to D0	WE3#	WE2#	WE1#	WE0#
8-bit access at address 0	—	—	—	Data 7 to 0	—	—	—	Active
8-bit access at address 1	—	—	Data 7 to 0	—	—	—	Active	—
8-bit access at address 2	—	Data 7 to 0	—	—	—	Active	—	—
8-bit access at address 3	Data 7 to 0	—	—	—	Active	—	—	—
16-bit access at address 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Active	Active
16-bit access at address 2	Data 15 to 8	Data 7 to 0	—	—	Active	Active	—	—
32-bit access at address 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Active	Active	Active	Active

Table 14.7 16-bit external device access and data alignment in little endian

Operation		Data bus				Strobe signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	WE3#	WE2#	WE1#	WE0#
8-bit access at address 0		—	—	—	Data 7 to 0	—	—	—	Active
8-bit access at address 1		—	—	Data 7 to 0	—	—	—	Active	—
8-bit access at address 2		—	—	—	Data 7 to 0	—	—	—	Active
8-bit access at address 3		—	—	Data 7 to 0	—	—	—	Active	—
16-bit access at address 0		—	—	Data 15 to 8	Data 7 to 0	—	—	Active	Active
16-bit access at address 2		—	—	Data 15 to 8	Data 7 to 0	—	—	Active	Active
32-bit access at address 0	1st access at address 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Active	Active
	2nd access at address 2	—	—	Data 31 to 24	Data 23 to 16	—	—	Active	Active

Table 14.8 8-bit external device access and data alignment in little endian (1 of 2)

Operation		Data bus				Strobe signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	WE3#	WE2#	WE1#	WE0#
8-bit access at address 0		—	—	—	Data 7 to 0	—	—	—	Active
8-bit access at address 1		—	—	—	Data 7 to 0	—	—	—	Active
8-bit access at address 2		—	—	—	Data 7 to 0	—	—	—	Active
8-bit access at address 3		—	—	—	Data 7 to 0	—	—	—	Active
16-bit access at address 0	1st access at address 0	—	—	—	Data 7 to 0	—	—	—	Active
	2nd access at address 1	—	—	—	Data 15 to 8	—	—	—	Active

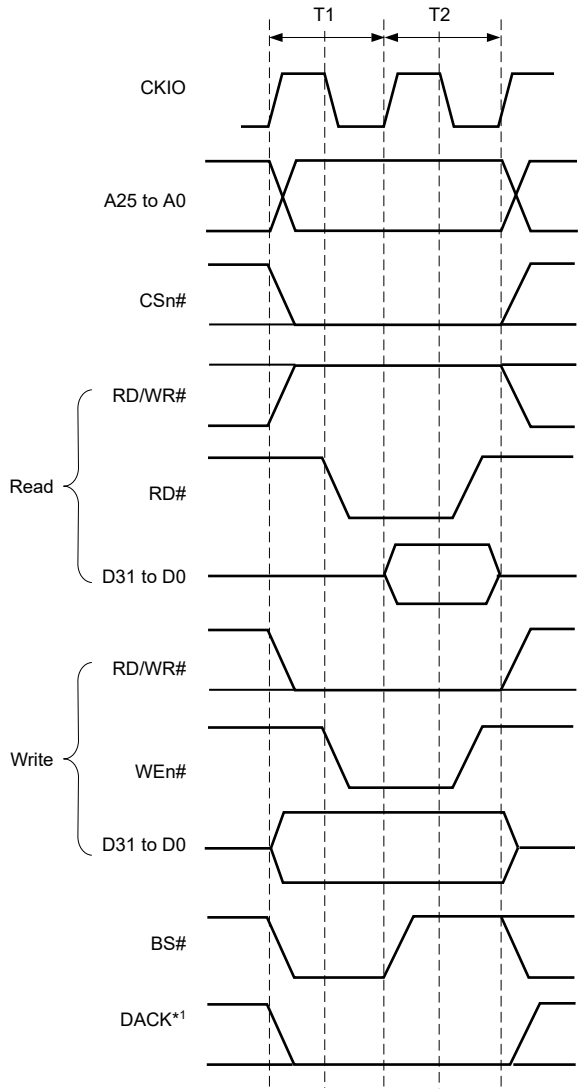
Table 14.8 8-bit external device access and data alignment in little endian (2 of 2)

Operation		Data bus				Strobe signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	WE3#	WE2#	WE1#	WE0#
16-bit access at address 2	1st access at address 0	—	—	—	Data 7 to 0	—	—	—	Active
	2nd access at address 1	—	—	—	Data 15 to 8	—	—	—	Active
32-bit access at address 0	1st access at address 0	—	—	—	Data 7 to 0	—	—	—	Active
	2nd access at address 1	—	—	—	Data 15 to 8	—	—	—	Active
	3rd access at address 2	—	—	—	Data 23 to 16	—	—	—	Active
	4th access at address 3	—	—	—	Data 31 to 24	—	—	—	Active

14.5.2 SRAM Interface

(1) Basic Timing

For access to an SRAM interface, this LSI uses strobe signal output in consideration of the fact that mainly static RAM will be directly connected. When using SRAM with a byte-selection pin, see section 14.5.7. SRAM Interface with Byte Selection. Figure 14.2 shows the basic timings of SRAM interface access. A no-wait normal access is completed in two cycles. The BS# signal is activated for one state to indicate the start of a bus cycle.



Note 1. The waveform for DACKn is when active-low is specified.

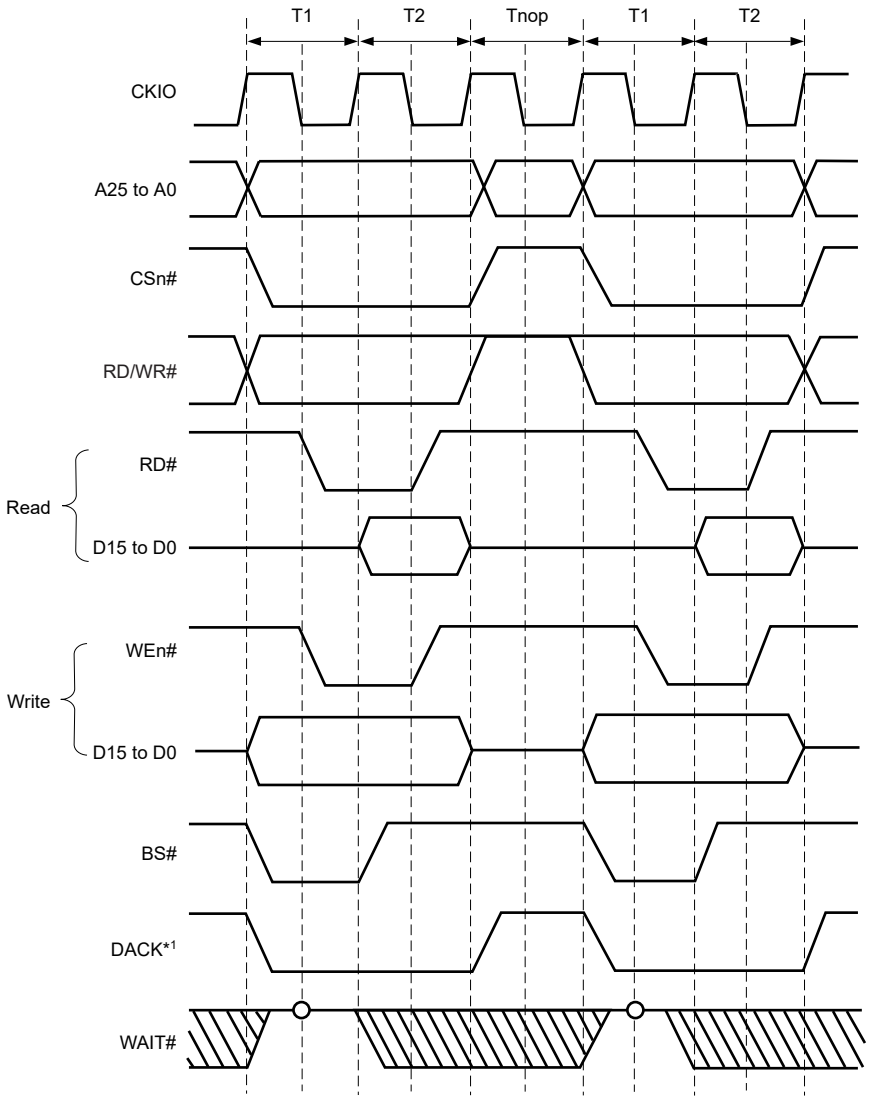
Figure 14.2 SRAM interface basic access timing (access wait 0)

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 32 bits are always read in case of a 32-bit device. 16 bits are always read in case of a 16-bit device. When writing, only the WEn# signal for the byte to be written is activated.

It is necessary to output the data that has been read using RD# when a buffer is established in the data bus. The RD/WR# signal is in a read state (high output) when no access has been carried out. Therefore, care must be taken when controlling the external data buffer with this signal, to avoid output collision.

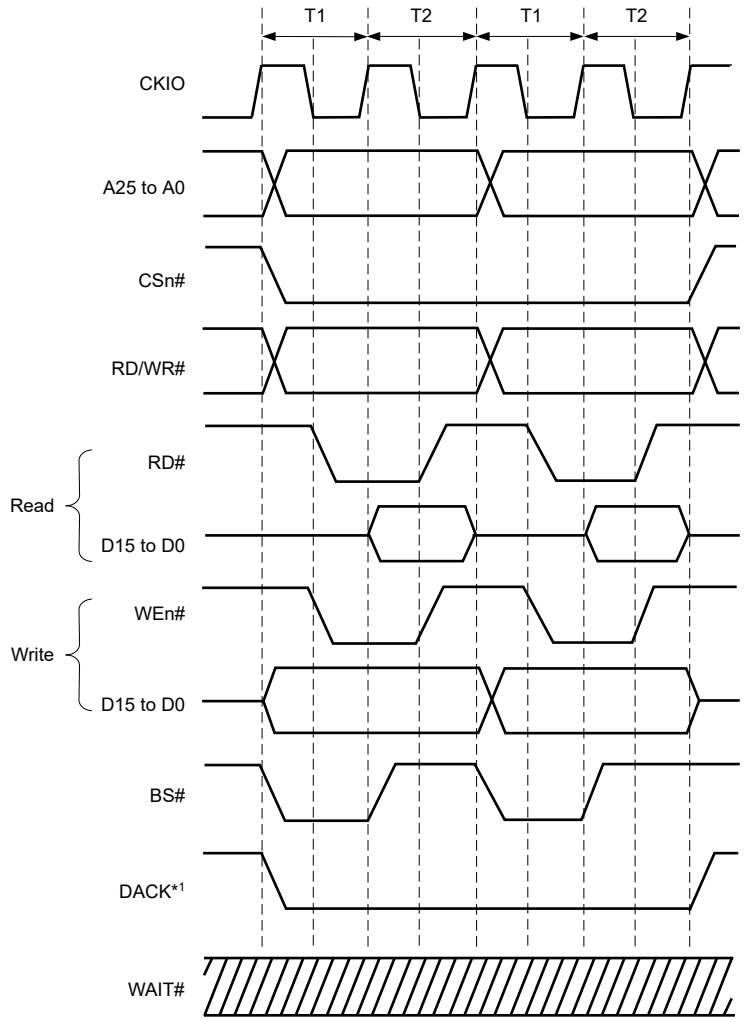
Figure 14.3 and Figure 14.4 show the basic timings in continuous access to the SRAM interface. If the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted after the CSn space access to sample the external wait (Figure 14.3).

If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (Figure 14.4).



Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.3 Continuous access to the SRAM interface (1) (bus width = 16 bits, 32-bit access, CSnWCR.WM bit = 0 (access wait = 0, cycle wait = 0))



Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.4 Continuous access to the SRAM interface (2) (bus width = 16 bits, 32-bit access, CSnWCR.WM bit = 1 (access wait = 0, cycle wait = 0))

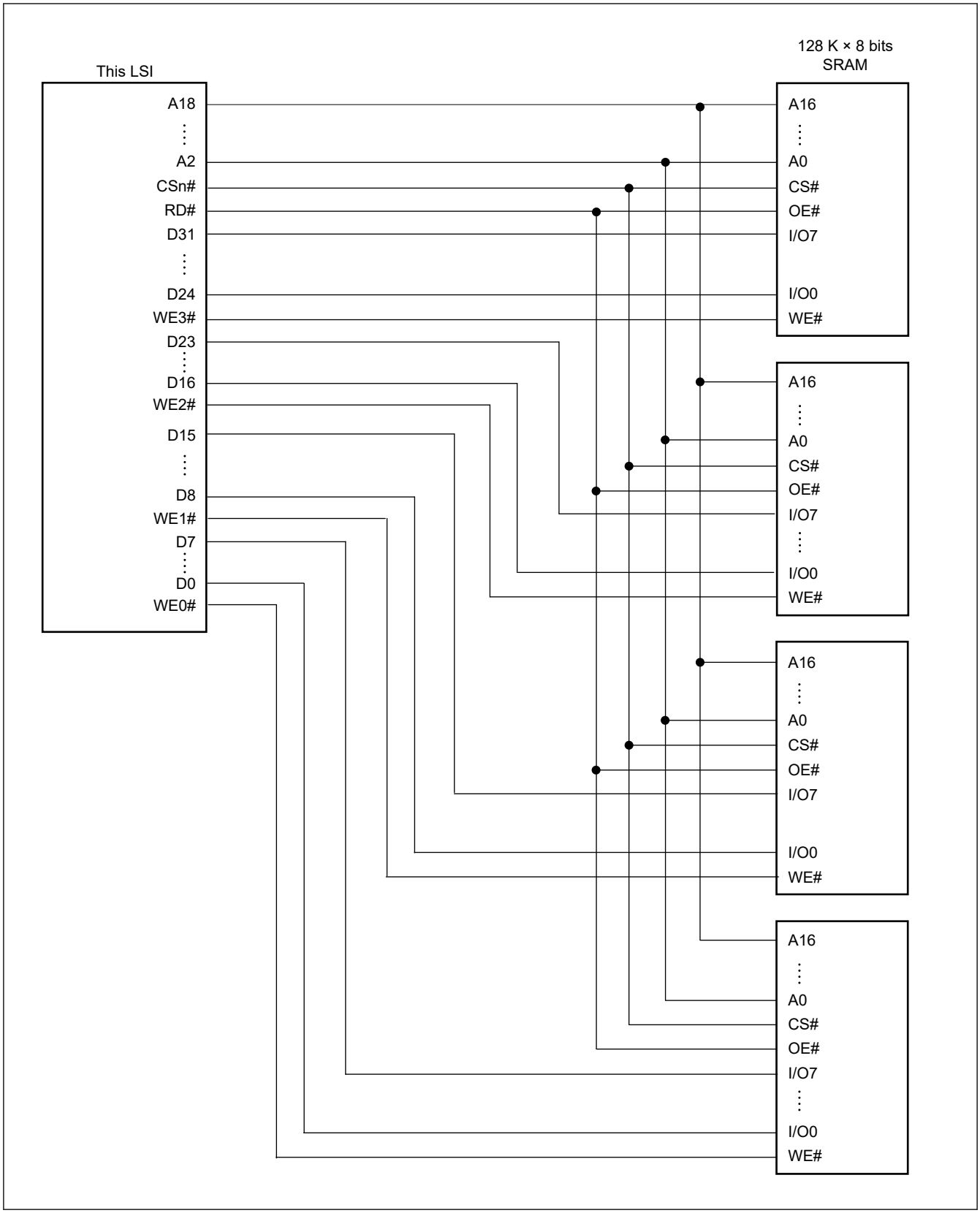


Figure 14.5 Example of 32-bit data-width SRAM connection

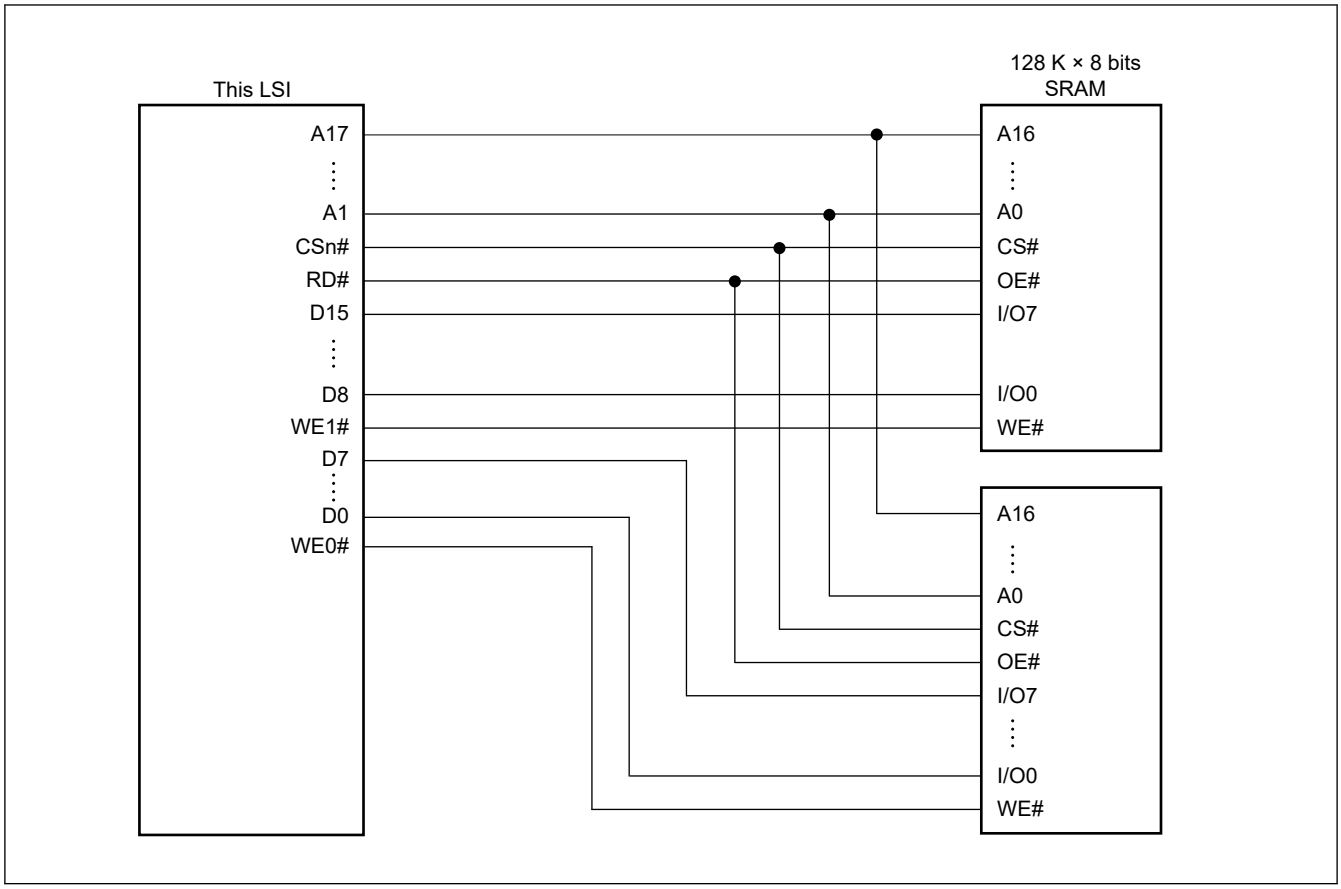


Figure 14.6 Example of 16-bit data-width SRAM connection

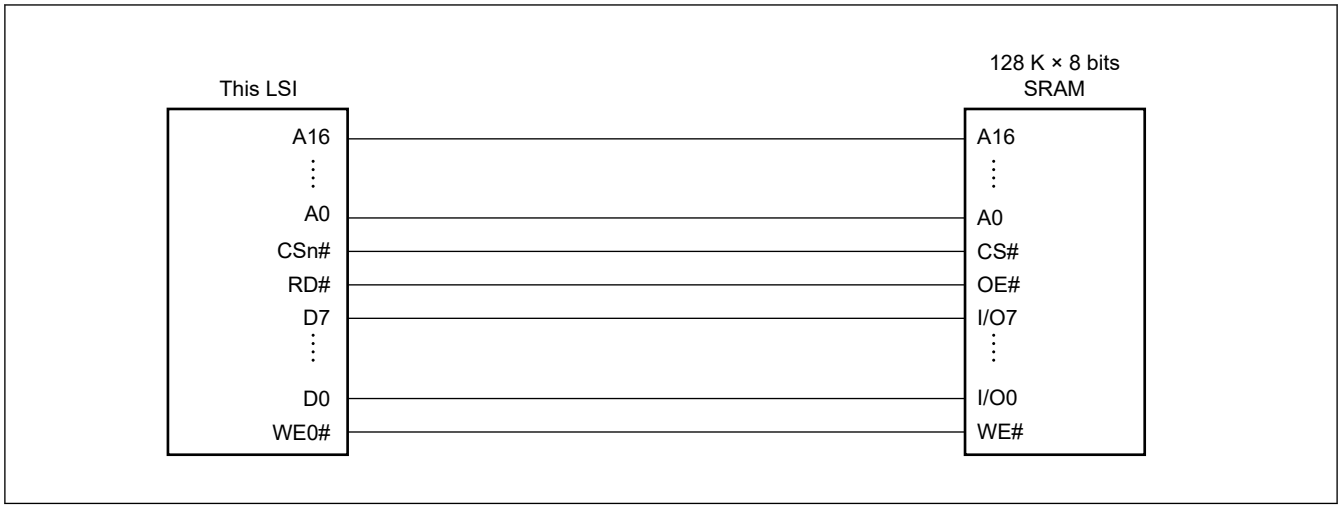
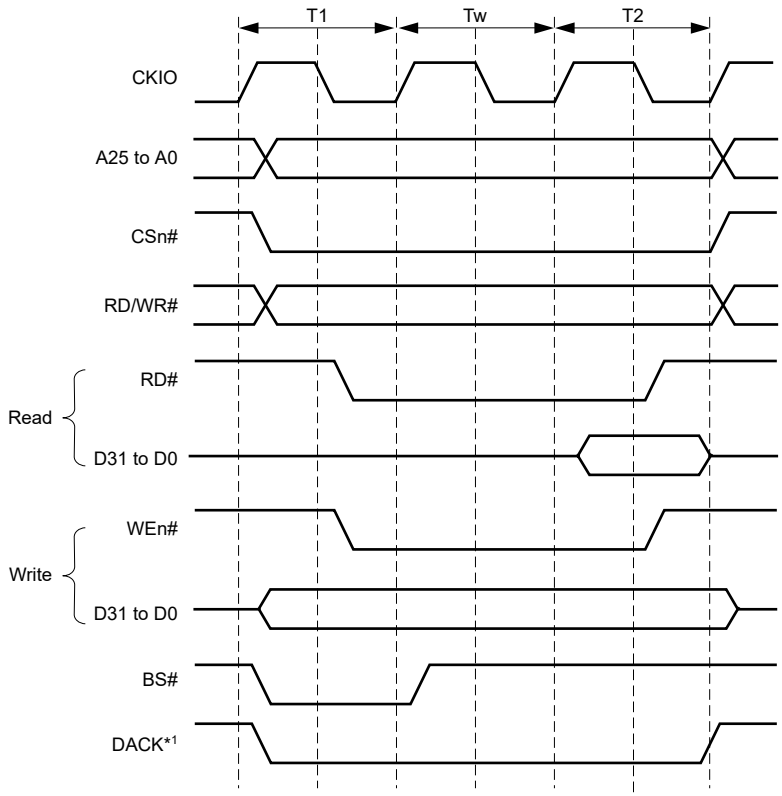


Figure 14.7 Example of 8-bit data-width SRAM connection

14.5.3 Access Wait Control

Wait insertion on an SRAM interface access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible for area 5 to insert waits independently in read access and in write access. Areas 0, 2, and 3 have common access wait for read cycle and write cycle. The specified number of Tw states are inserted as waits in an SRAM interface access shown in Figure 14.8.

Example: Insertion of 1 cycle of waiting by a software wait
(WR[3:0] = 0001b, WW[2:0] = 000b)

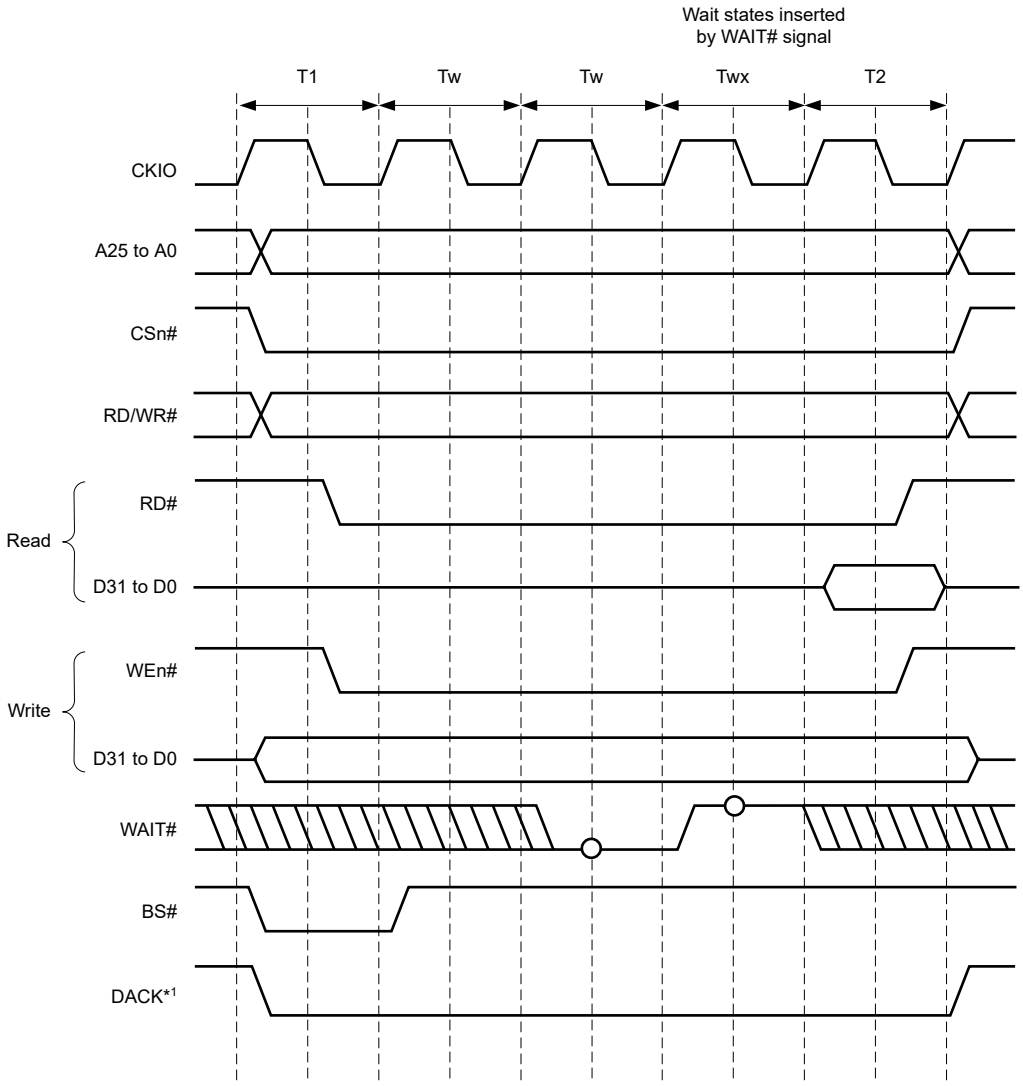


Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.8 Wait timing for SRAM interface access (software wait only)

When the WM bit in the CSnWCR register is cleared to 0, the signal on the external WAIT# pin is also sampled. WAIT# pin sampling is shown in Figure 14.9. Two waits are specified as software waits. The WAIT# signal is sampled on the falling edge of the CKIO signal at the transition from the T1 or Tw cycle to the T2 cycle.

Example: Insertion of 2 cycles of waiting by a software wait and 1 cycle of waiting by the WAIT# signal (WR[3:0] = 0010b, WW[2:0] = 000b, WM = 0)



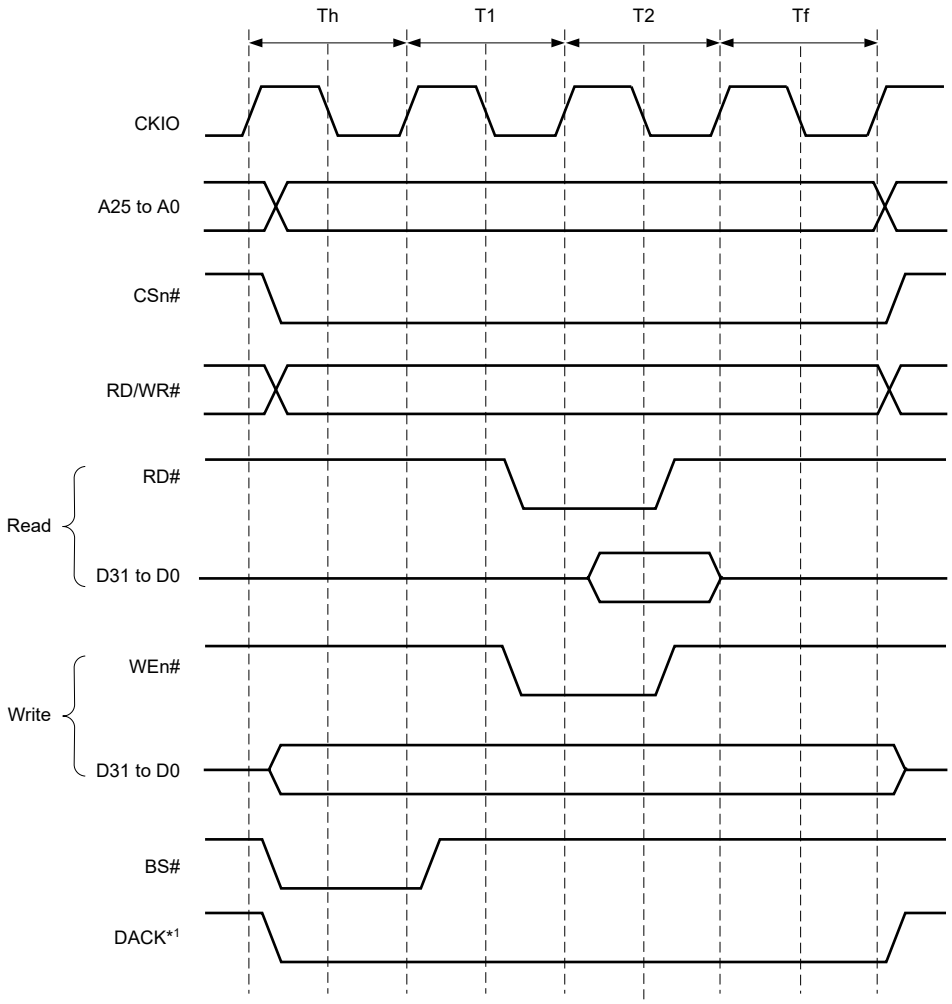
Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.9 Wait cycle timing for SRAM interface access (wait cycle insertion using WAIT# signal)

14.5.4 CSn# Assert Period Expansion

The number of states from CSn# active to RD#, WEn# active can be specified by setting bits SW1 and SW0 in CSnWCR. The number of states from the inactive state of the RD# and WEn# signals to the inactive state of the CSn# signal can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 14.10 shows an example. A Th state and a Tf state are added before and after an ordinary cycle, respectively. In these states, the RD# and WEn# signals are not activated, while other signals are activated. The data output is prolonged to the Tf cycle, and this prolongation is useful for devices with slow writing operations.

Example: SW[1:0] = 01b (delay of 1.5 cycles) and HW[1:0] = 01b (delay of 1.5 cycles)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.10 Active period expansion for the CSn signal

14.5.5 MPX-I/O Interface

Access timing for the MPX space is shown below. In the MPX space, CS5#, AH#, RD#, and WEn# signals control the accessing. The basic access for the MPX space consists of 2 cycles of address output followed by an access to an SRAM interface. The bus width for the address output cycle or the data input/output cycle is fixed to 8 bits or 16 bits.

Alternatively, it can be 8 bits or 16 bits depending on the address to be accessed.

Output of the addresses D15 to D0 or D7 to D0 is performed from state Ta2 to state Ta3. Because state Ta1 has a high-impedance state, collisions of addresses and data can be avoided without inserting idle cycles, even in continuous access cycles. Address output is increased to 3 cycles by setting the MPXW bit in the CS5WCR register to 1.

The RD/WR# signal is output at the same time as the CS5# signal; it is high in the read cycle and low in the write cycle.

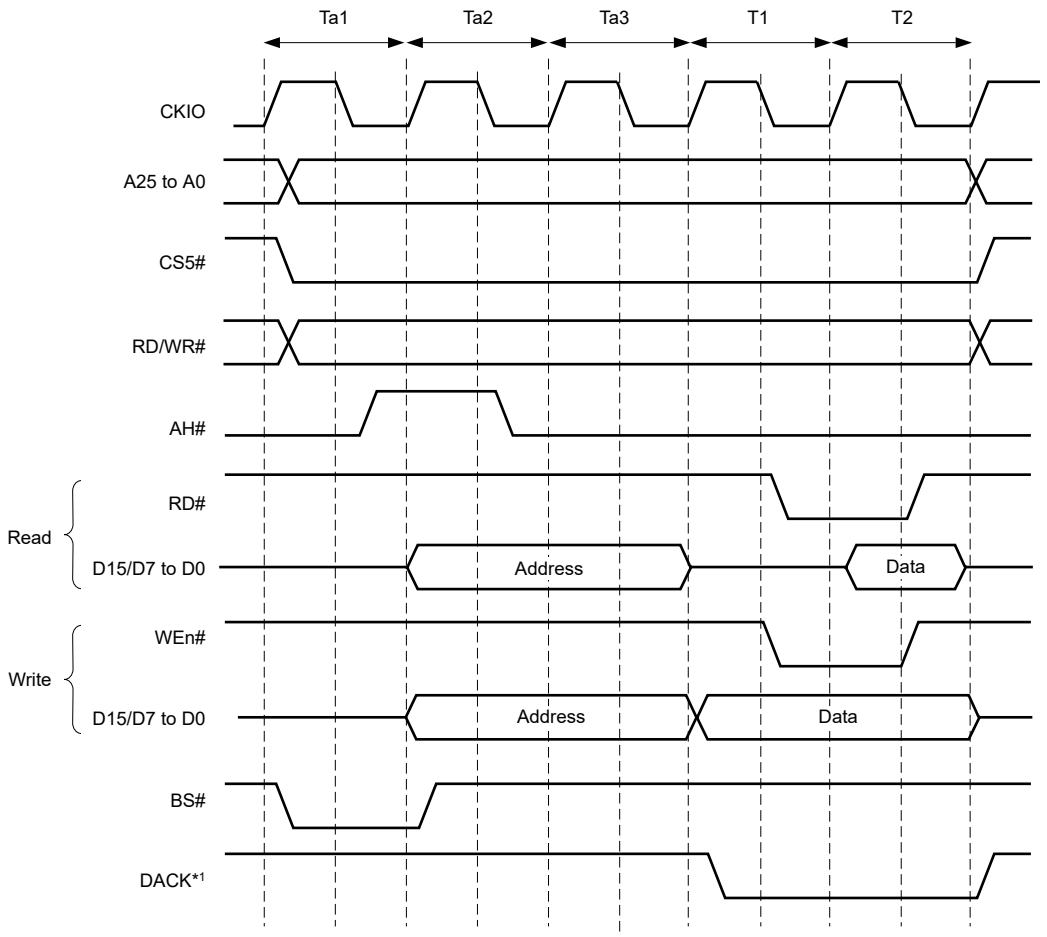
The data cycle is the same as that in an SRAM interface access.

Delay cycles specified in the SW[1:0] bits of the CS5WCR register are inserted between the Ta3 state and the T1 state.

Delay cycles specified in the HW[1:0] bits are added after the T2 state.

Timing charts are shown in Figure 14.11 to Figure 14.14.

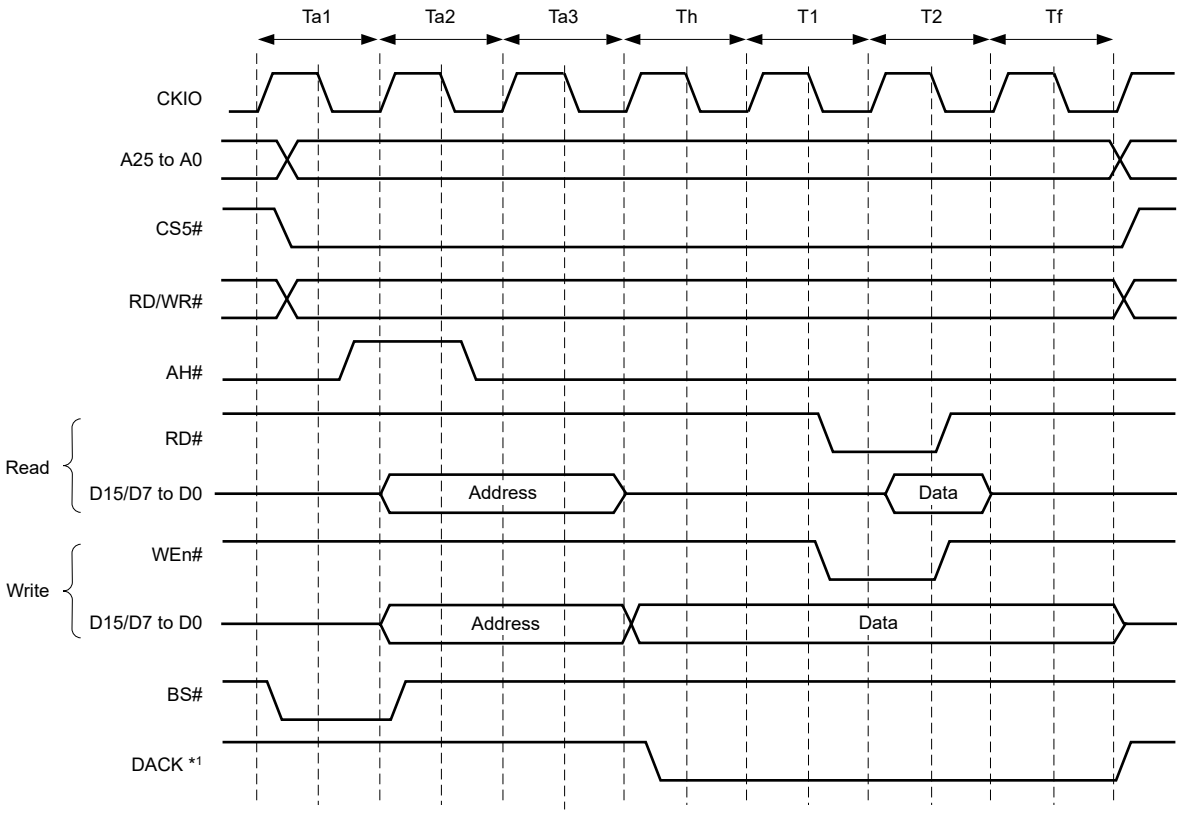
Example: MPXW = 0 (no insertion of wait cycles)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.11 Access timing for MPX space (1) (address cycle no wait, data cycle no wait)

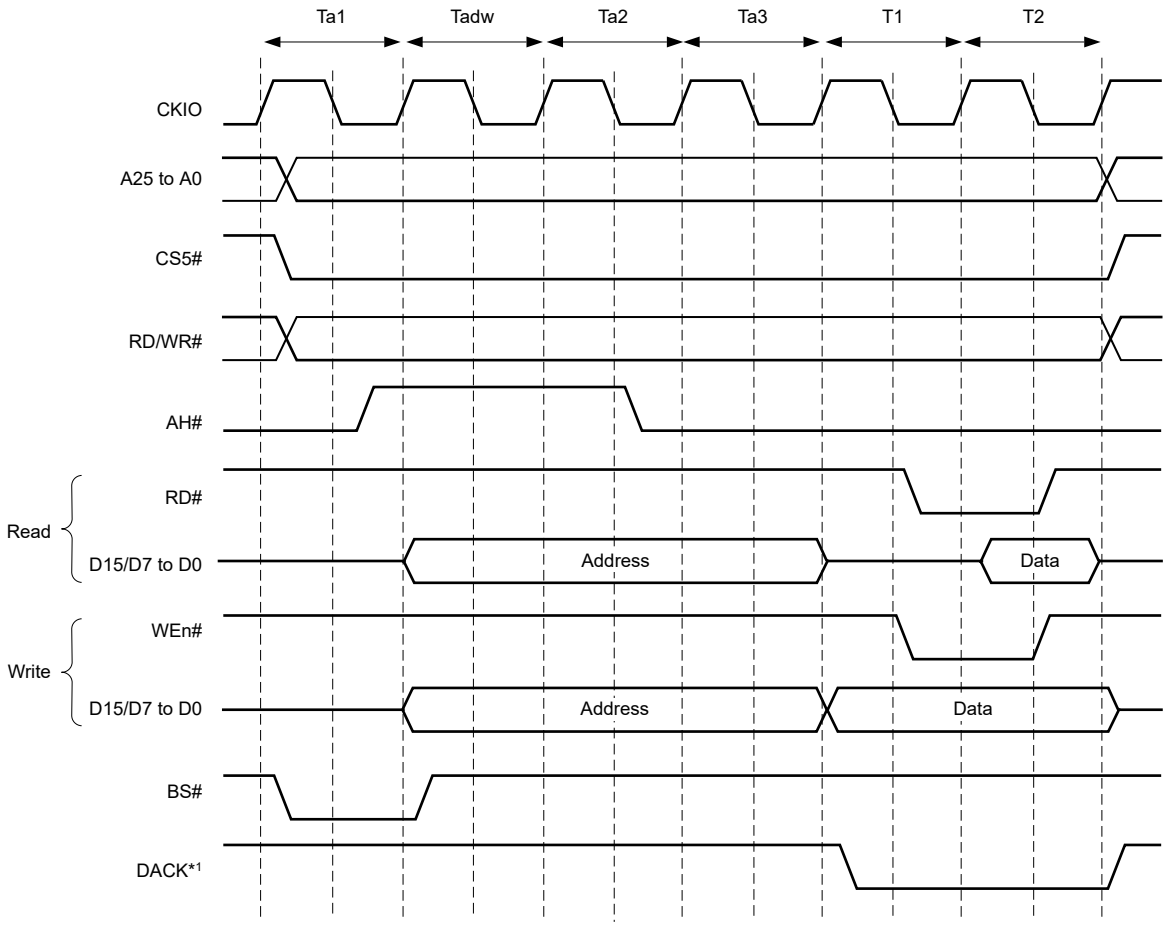
Example: MPXW = 0 (no insertion of wait cycles), SW[1:0] = 01b (delay of 1.5 cycles), HW[1:0] = 01b (delay of 1.5 cycles)



Note 1. The waveform for DACKn is when active-low is specified.

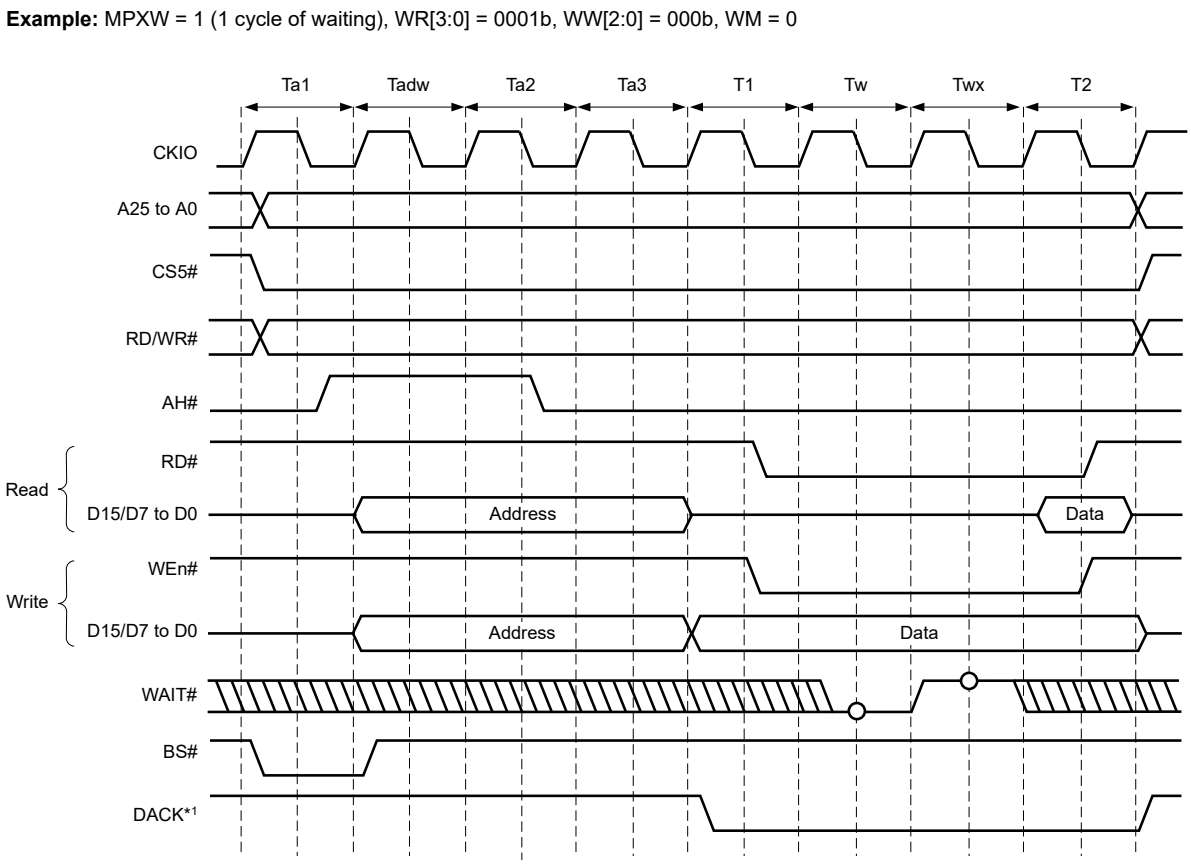
Figure 14.12 Access timing for MPX space (2) (address cycle no wait, 1.5 aassertion delay states, data cycle no wait, 1.5 delay states)

Example: MPXW = 1 (1 cycle of waiting)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.13 Access timing for MPX space (3) (address cycle wait 1, data cycle no wait)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.14 Access timing for MPX space (4) (address cycle access wait 1, data cycle wait 1, external wait 1)

14.5.6 Burst ROM (Clocked Asynchronous) Interface

The burst ROM (clocked asynchronous) interface is used to access a memory with a high-speed read function using a method of address switching called the burst mode or page mode. In a burst ROM (clocked asynchronous) interface, basically, the same access as the SRAM interface is performed, but the 2nd and subsequent access cycles are performed only by changing the address, without inactivating the RD# signal at the end of the 1st cycle. In the 2nd and subsequent access cycles, addresses are changed at the falling edge of the CKIO signal.

For the 1st access cycle, the number of waits specified by the W3 to W0 bits in the CS0WCR_1 register is inserted. For the 2nd and subsequent access cycles, the number of wait cycles specified by the BW1 and BW0 bits in the CS0WCR_1 register is inserted.

In the access to the burst ROM (clocked asynchronous), the BS# signal is activated only to the first access cycle. An external wait input is valid only to the first access cycle.

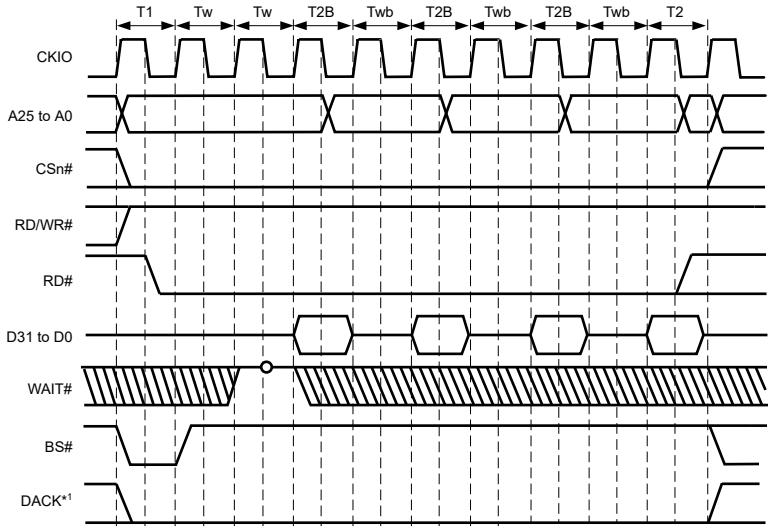
In the single access or write access that does not perform the burst operation in the burst ROM (clocked asynchronous) interface, access timing is the same as an SRAM interface.

Table 14.9 lists a relationship between bus width, access size, and the number of bursts. Figure 14.15 shows a timing chart.

Table 14.9 Relationship between bus width, access size, and number of bursts

Bus width	Access size	CSnWCR.BST[1:0] bits	Number of bursts	Access count
8 bits	8 bits	Not affected	1	1
	16 bits	Not affected	2	1
	32 bits	Not affected	4	1
	16 bytes	00	16	1
		01	4	4
	32 bytes	00	16	2
		01	4	8
	64 bytes	00	16	4
		01	4	16
16 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	2	1
	16 bytes	00	8	1
		01	2	4
		10*1	4	2
			2, 4, 2	3
	32 bytes	00	8	2
		01	2	8
		10*1	4	4
			2, 4, 2	6
	64 bytes	00	8	4
		01	2	16
		10*1	4	8
			2, 4, 2	12
32 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	1	1
	16 bytes	Not affected	4	1
	32 bytes	Not affected	4	2
	64 bytes	Not affected	4	4

Note 1. When the bus width is 16 bits, the access size is 16 bits or more, and the BST[1:0] bits in CS0WCR_1 are 10b, the number of bursts and access count depends on the access start address. At address 0x***0 or 0x***8, 4-4 burst access is performed (burst numbers: 4, 4 for a total of 2 rounds of access). At address 0x***4 or 0x***C, 2-4-2 burst access is performed (burst numbers: 2, 4, 2 for a total of 3 rounds of access).



Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.15 Burst ROM access timing (clocked asynchronous) (bus width = 32 bits, 16-byte transfer (number of burst 4), wait cycles inserted in first access = 2, wait cycles inserted in second and subsequent access cycles = 1)

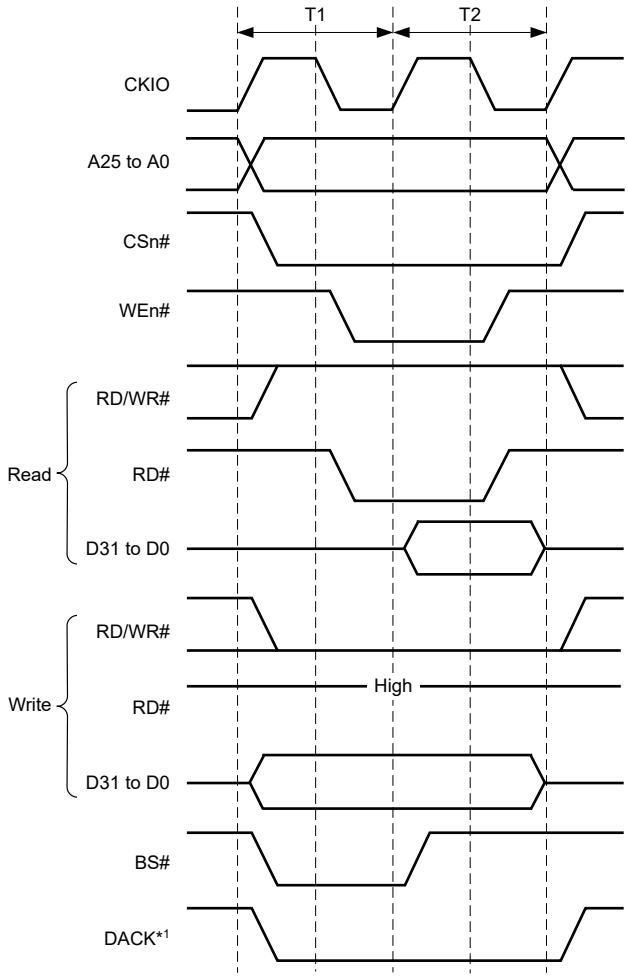
14.5.7 SRAM Interface with Byte Selection

The SRAM interface with byte selection is a memory interface that outputs the byte selection signal (WEn#) in both read and write bus cycles. This interface has 16-bit data pins and accesses SRAMs having upper and lower byte selection pins, such as UB# and LB#.

When the BAS bit in CSnWCR is cleared to 0 (value after reset), the write access timing of the SRAM interface with byte selection is the same as that for the SRAM interface. While in read access of a byte-selection SRAM interface, the byte-selection signal is output from the WEn# pin, which is different from that for the SRAM interface. The basic access timing is shown in Figure 14.16. In write access, data is written to the memory according to the timing of the byte-selection pin (WEn#). For details, please see the Data Sheet for the corresponding memory.

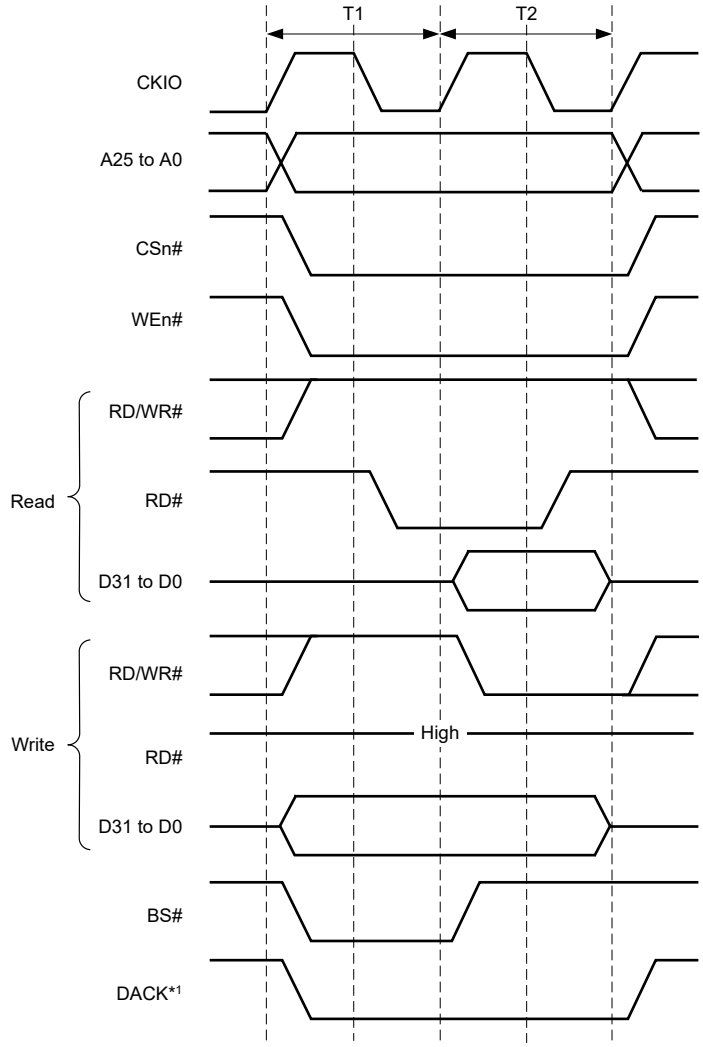
If the BAS bit in CSnWCR is set to 1, the WEn# pin and RD/WR# pin timings change. Figure 14.17 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin (RD/WR#).

The data hold of write data at the timing of RD/WR# inactive must be acquired by setting the HW1 and HW0 bits in the CSnWCR register. Figure 14.18 shows the access timing when a software wait is specified.



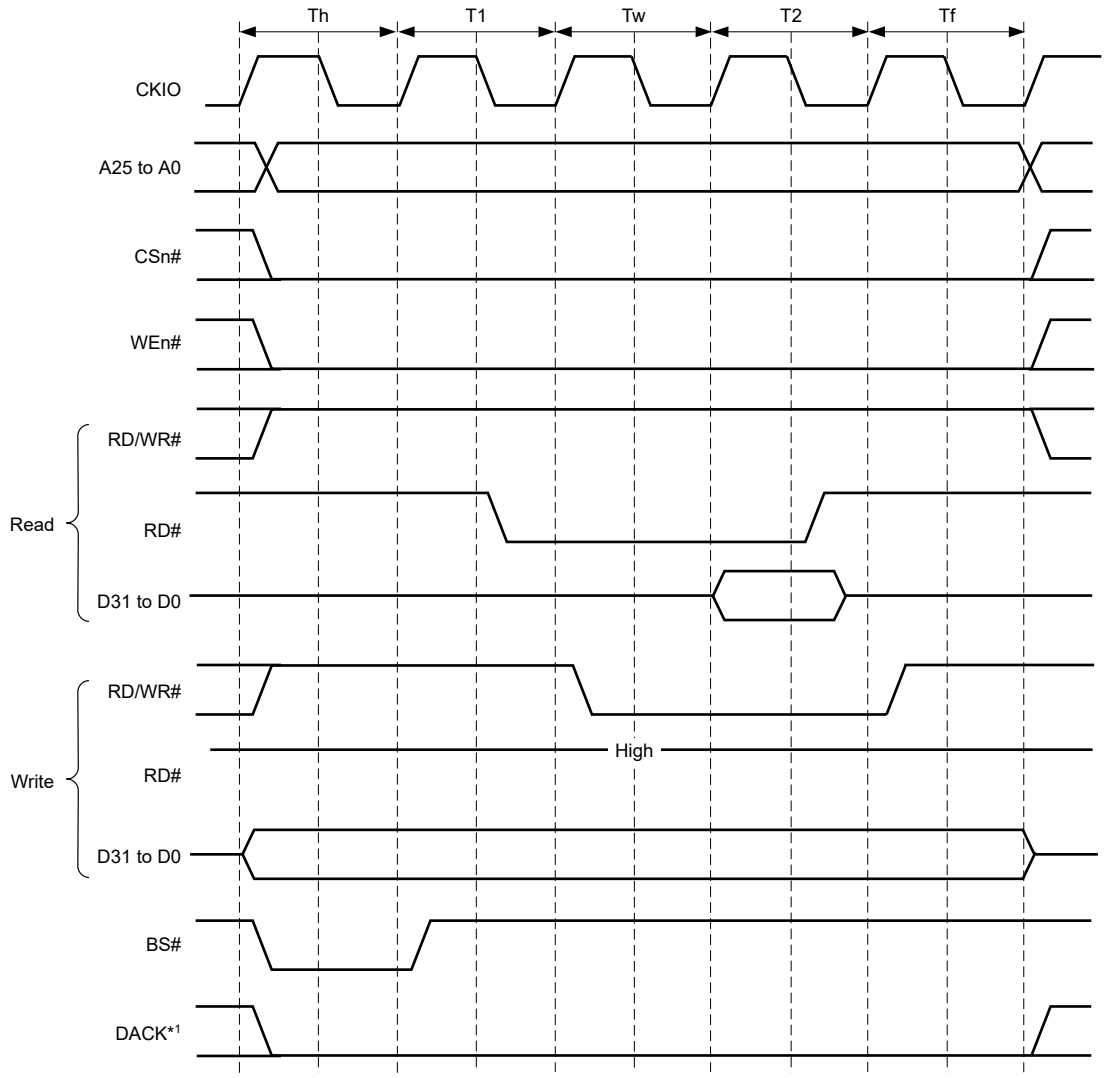
Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.16 Basic access timing for SRAM with byte selection (BAS = 0)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.17 Basic access timing for SRAM with byte selection (BAS = 1)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.18 Wait timing for SRAM with byte selection (BAS = 1) (SW[1:0] = 01b, WR[3:0] = 0001b, HW[1:0] = 01b)

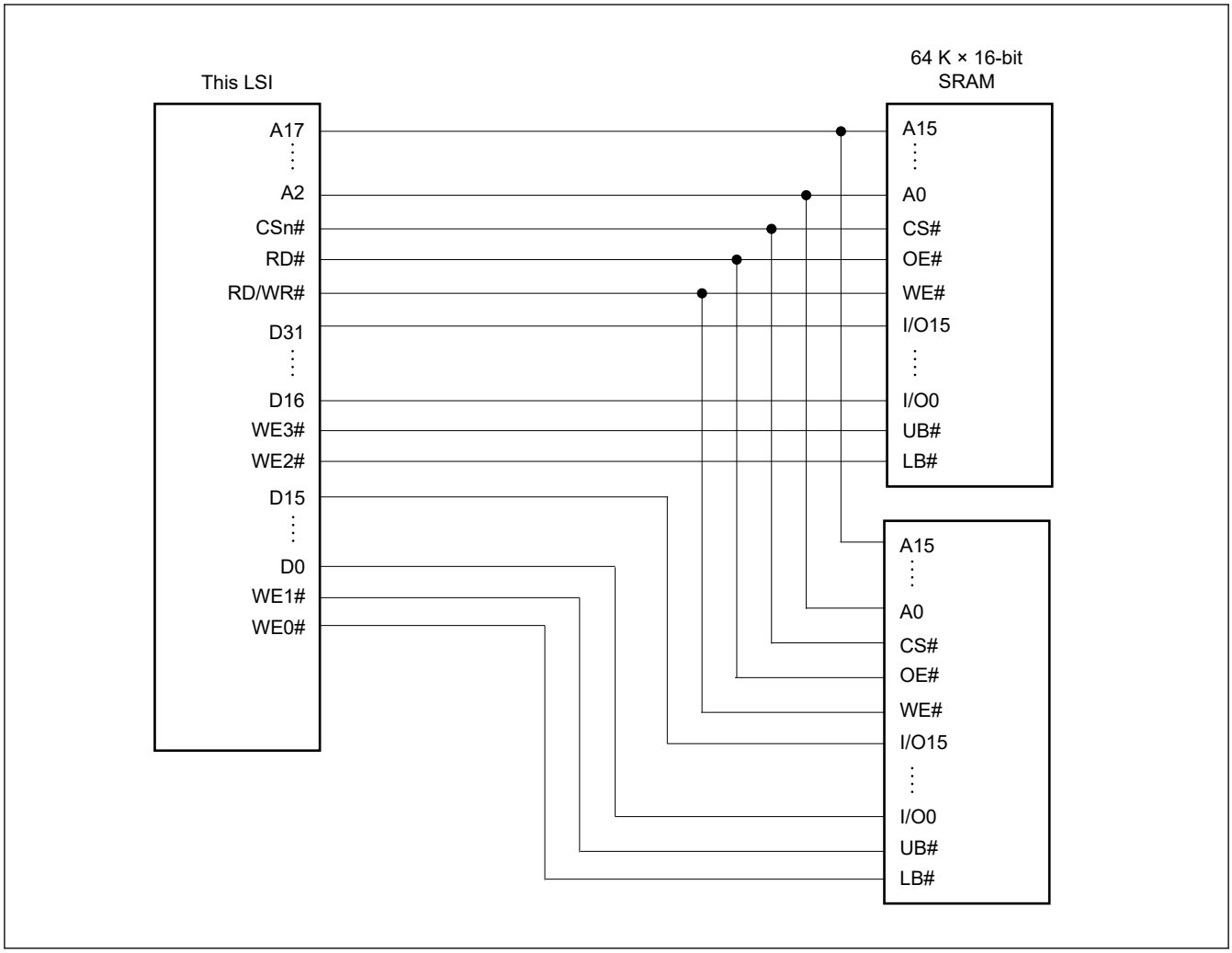


Figure 14.19 Example of connection with 32-bit data-width SRAM with byte selection

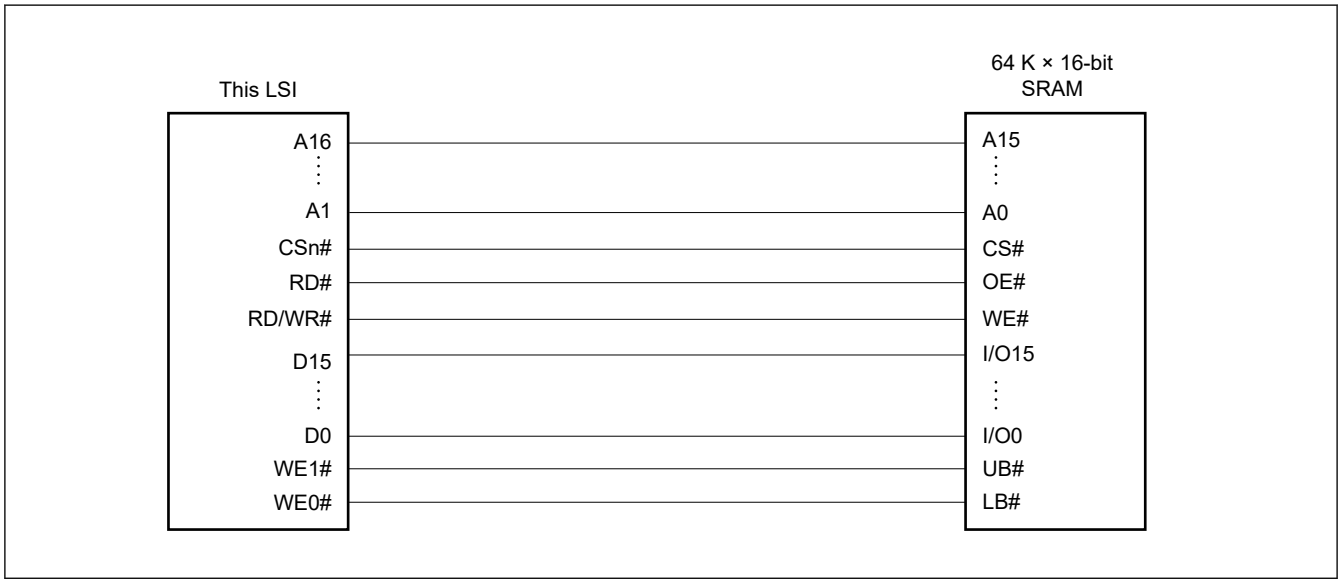


Figure 14.20 Example of connection with 16-bit data-width SRAM with byte selection

14.5.8 Burst ROM (Clocked Synchronous) Interface

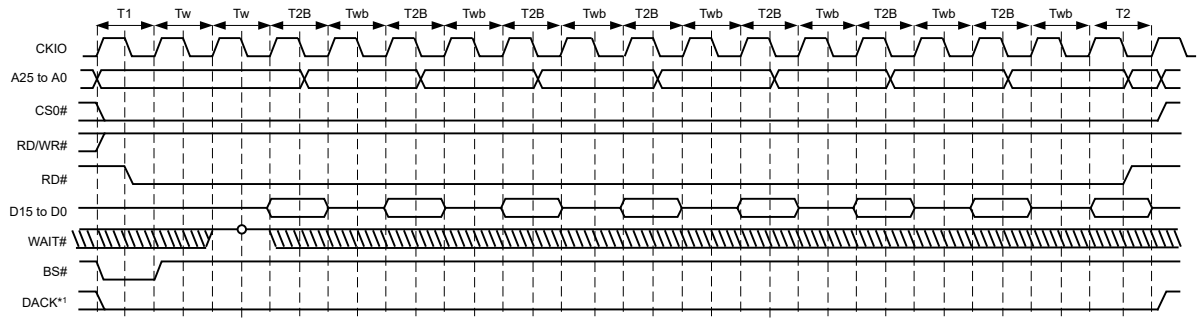
The burst ROM (clocked synchronous) interface is supported to access a ROM with a synchronous burst function at high speed. The burst ROM interface accesses the burst ROM in the same way as an SRAM interface. This interface is valid only for area 0.

In the first access cycle, wait cycles are inserted. In this case, the number of wait cycles to be inserted is specified by the W3 to W0 bits in CS0WCR. In the second and subsequent cycles, the number of wait cycles to be inserted is specified by the BW1 and BW0 bits in CS0WCR.

While the burst ROM (clocked synchronous) is accessed, the BS# signal is activated only for the first access cycle and an external wait input is also valid for the first access cycle.

When the bus width is 16 bits, the burst length must be specified as 8. When the bus width is 32 bits, the burst length must be specified as 4. The burst ROM interface does not support the 8-bit bus width for the burst ROM.

The burst ROM interface performs burst operations for all read access. For example, in a 32-bit access over a 16-bit bus, valid 16-bit data is read two times and invalid 16-bit data is read six times. These invalid data read cycles increase the memory access time and degrade the program execution speed and DMA transfer speed. To prevent this problem, it is recommended using a read in a 16-byte or more access size. The burst ROM interface performs write access in the same way as SRAM interface access.



Note 1. The waveform for DACKn is when active-low is specified.

Figure 14.21 Burst ROM access timing (clocked synchronous) (burst length = 8, wait cycles inserted in first access = 2, wait cycles inserted in second and subsequent access cycles = 1)

14.5.9 Wait between Access Cycles

As the operating frequency of LSIs becomes higher, the off-operation of the output data buffer for devices with slow access speed often collides with the data output for the next cycle. As a result of these collisions, the reliability of the device might be lowered or malfunctions might occur. Data collisions can be avoided by inserting idle (wait) cycles between continuous access cycles.

The number of idle states between access cycles can be set by the WM bit in CSnWCR, bits IWW2 to IWW0, IWRWD2 to IWRWD0, IWRWS2 to IWRWS0, IWRRD2 to IWRRD0, and IWRRS2 to IWRRS 0 in CSnBCR. The conditions for setting the idle cycles between access cycles are shown below.

1. Continuous access cycles are write-read or write-write
2. Continuous access cycles are read-write for different spaces
3. Continuous access cycles are read-write for the same space
4. Continuous access cycles are read-read for different spaces
5. Continuous access cycles are read-read for the same space

For the specification of the number of idle cycles between access cycles described above, see the description of each register.

Besides the idle states between access cycles specified by the registers, idle cycles must be inserted to interface with the internal bus or to obtain the minimum pulse width for a multiplexed pin (WEn#). The following gives detailed information about the idle cycles and describes how to estimate the number of idle cycles.

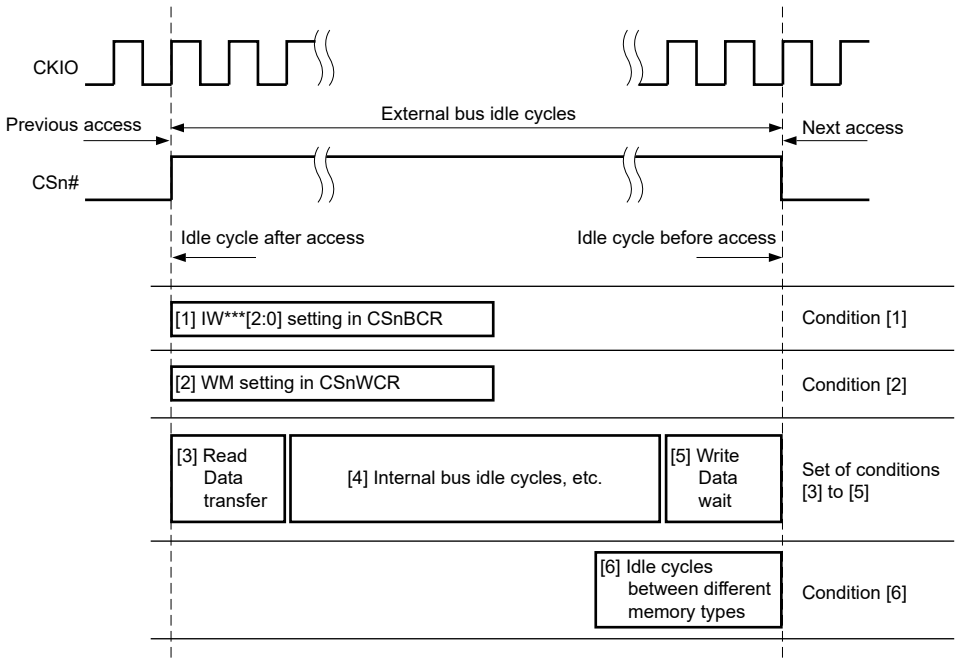
The number of idle cycles on the external bus from the inactive state of the CSn# signal to the active state of CSn# is described below.

There are seven conditions that determine the number of idle cycles on the external bus as shown in Table 14.10. The effects of these conditions are shown in Figure 14.22.

Table 14.10 Conditions for determining number of idle cycles

No.	Condition	Description	Range	Note
[1]	IW***[2:0] in CSnBCR	These bits specify the number of idle cycles for access. The number of idle cycles can be specified independently for each combination of the previous and next cycles. For example, in the case where reading CS2 space followed by reading other CS space, the bits IWRRD[2:0] in CS2BCR should be set to 100b to specify six or more idle cycles. This condition is effective only for access states other than single address transfer and generates idle states after the access is completed.	0 to 12	Do not set 0 for the number of idle cycles between memory types which are not allowed to be accessed successively.
[2]	WM in CSnWCR	This bit enables or disables external WAIT# pin input. When this bit is cleared to 0 (external WAIT# enabled), one idle state is inserted to check the external WAIT# pin input after the access is completed. When this bit is set to 1 (disabled), no idle state is generated.	0 or 1	—
[3]	Read data transfer cycle	One idle state is inserted after a read access is completed. This idle state is not generated for the first or middle cycles in divided access cycles. This is neither generated when the HW[1:0] bits in CSnWCR are not 00b.	0 or 1	—
[4]	Internal bus idle cycles, etc.	External bus access requests from the CPU or the direct memory access controller (DMA) and their results are passed through the internal bus. The external bus enters idle state during internal bus idle cycles or while a bus other than the external bus is being accessed. This condition is not effective for divided access cycles, which are generated by the bus state controller when the access size is larger than the external data bus width.	0 or larger	The number of internal bus idle cycles may not become 0 depending on the CPU: internal bus: CKIO.
[5]	Write data wait cycles	During write access, a write cycle is executed on the external bus only after the write data becomes ready. This write data wait period generates idle cycles before the write cycle. Note that when the previous cycle is a write cycle and the internal bus idle cycles are shorter than the previous write cycle, write data can be prepared in parallel with the previous write cycle and therefore, no idle cycle is generated (write buffer effect).	0 or 1	For write → write or write → read access cycles, successive access cycles without idle cycles may be available due to the write buffer effect described in the left column. If successive access cycles without idle cycles are not allowed, specify the minimum number of idle cycles between access cycles through the CSnBCR register.
[6]	Idle cycles between different memory types	To ensure the minimum pulse width on the multi-use pins, idle cycles may be inserted before access after memories are switched. For some memory types, idle cycles are inserted even when memory types are not switched.	0 to 2	The number of idle cycles depends on the target memory types. See Table 14.11.

In the above conditions, a total of four conditions, that is, condition [1], condition [2], a set of conditions [3] to [5] (these are generated successively, and therefore the sum of them should be taken as one set of idle cycles), and condition [7] are generated at the same time. The maximum number of idle cycles among these four conditions becomes the number of idle cycles on the external bus. To ensure the minimum idle cycles, be sure to make register settings for condition [1].



Note: A total of four conditions (condition [1], condition [2], a set of conditions [3] to [5], and condition [6]) generate idle cycle at the same time. Accordingly, the maximum number of cycles among these four conditions become the number of idle cycles.

Figure 14.22 Idle cycle conditions

Table 14.11 Number of idle cycles inserted between access cycles to different memory types

Previous cycle	Next Cycle					
	SRAM	Burst ROM (asynchronous)	MPX-I/O	Byte SRAM (BAS = 0)	Byte SRAM (BAS = 1)	Burst ROM (synchronous)
SRAM	0	0	1	0	0/1*1	0
Burst ROM (asynchronous)	0	0	1	0	0/1*1	0
MPX-I/O	1	1	0	1	1	1
Byte SRAM (BAS = 0)	0	0	1	0	0/1*1	0
Byte SRAM (BAS = 1)	0/1*1	0/1*1	1/2*1	0/1*1	0	0/1*1
Burst ROM (synchronous)	0	0	1	0	1	0

Note 1. The number of idle cycles depends on the setting of the CSnWCR.HW[1:0] bits in the previous cycle. When HW[1:0] ≠ 00b, the number of idle cycles is the value on the left; when HM[1:0] = 00b, the number of idle cycles is the value on the right. If the CSnWCR.HW[1:0] bits were for a CSn space that does not exist in the previous cycle, the number of idle cycles is the value on the right.

14.5.10 Others

(1) Reset

This module can be initialized completely only at reset of the entire chip (pin reset, software reset, and error reset).

When reset of the entire chip occurs, all signals are inactivated and data output buffers are turned off regardless of the bus cycle state after the internal reset is synchronized with the internal clock. All control registers are initialized. In the module stop state, control registers of the bus state controller are not initialized.

(2) Caution on Write Buffer

Since the bus state controller incorporates a one-stage write buffer, it can execute an access via the internal bus before the previous external bus cycle is completed in a write cycle. If the on-chip module is read or written after the external low-speed memory is written, the on-chip module can be accessed before the completion of the external low-speed memory write cycle.

In read cycles, the CPU is placed in the wait state until read operation has been completed. To continue the process after the data write to the device has been completed, perform a dummy read to the same address to check for completion of the write before the next process to be executed.

The write buffer of the bus state controller functions in the same way for an access by a bus master other than the CPU such as the direct memory access controller (DMA). Accordingly, to perform DMA transfers, the next read cycle is initiated before the previous write cycle is completed. Note, however, that if both the DMA source and destination addresses exist in external memory space, the next read cycle will not be initiated until the previous write cycle is completed.

Changing the registers in this module while the write buffer is operating may disrupt correct write access. Therefore, do not change the registers in this module immediately after a write access. If this change becomes necessary, do it after executing a dummy read of the write data.

15. DMA Controller (DMAC)

15.1 Overview

The LSI contains Direct Memory Access Controller (DMAC) consisting of three units, DMAC0 to DMAC2.

DMAC transfers data without using the CPU. When a transfer is requested, DMAC transfers data stored at the transfer source address to the transfer destination address.

Table 15.1 lists the specifications of DMAC.

Table 15.1 DMAC specifications

Parameter		Description	
		DMAC0	DMAC1, DMAC2
Number of channels		16 channels	16 channels
Address space		4 GB	
DMAC activation source		• External request (DREQ) • External interrupts (IRQ0 to IRQ13) • On-chip peripheral module requests/software requests*1	
Channel priority		• Priority among channels: Fixed priority levels within the group of channels 0 to 7 and within the group of channels 8 to 15, respectively, or round-robin scheduling for all channels. • Priority between channel groups: Round-robin scheduling between the group of channels 0 to 7 and the group of channels 8 to 15.	
Transfer data unit		8, 16, 32, 64, 128, 256, and 512 bits	8, 16, 32, 64, 128, and 256 bits
Maximum transfer size		2^{32} - 1 bytes	
Transfer mode	Single transfer	Performs DMA transfer for each DMA transfer request.	
	Block transfer	Performs DMA transfer of the specified transfer size for a DMAC activation request.	
DMA mode	Register mode	• DMA transfer setting value: Control register value within the DMA controller • DMA transfer from the source to the destination specified by a register.	
	Link mode	• DMA transfer setting value: Descriptors in the internal RAM or an external memory • Various DMA transfers specified by descriptors can be performed (responsiveness: register mode > link mode).	
Interval function		The DMA transfer interval can be specified (bus occupation ratio can be adjusted).	
Suspending function		Current DMA transfer can be paused.	
Buffer flush function		When DMAC is stopped forcibly, data in the buffer can be flushed.	
Interrupt request		Each channel has the following interrupt requests: • Transfer completion: Indicates completion of transfer of the specified size; each channel has this source. • Transfer error: Indicates a bus error; total of three sources, one for each unit.	

Note 1. A software request is output as a source of an on-chip peripheral module request from the interrupt controller. For how to set a software request, see section 12, Interrupt Controller (ICU).

Table 15.2 lists the input/output pins of the DMAC.

Table 15.2 Pin configuration of DMAC

Pin name	I/O	Description
DREQ	Input	DMA transfer request input from an external device to DMAC
DACK	Output	Output of DMAC transfer request acceptance from DMAC to an external device
TEND	Output	Output of transfer completion from DMAC to an external device

Note: For details on active levels of DACK and TEND, see section 15.3.12. CMNCR : Common Control Register.

15.2 Register Map

Table 15.3 DMAC register map

Address	Register symbol	Register name	Write protection
0x800C_0x000 + 0x1000 × i + 0x40 × n (m = 0) (n = 0 to 7) 0x800C_0x400 + 0x1000 × i + 0x40 × (n - 8) (m = 0) (n = 8 to 15) 0x800C_0x00C + 0x1000 × i + 0x40 × n (m = 1) (n = 0 to 7) 0x800C_0x40C + 0x1000 × i + 0x40 × (n - 8) (m = 1) (n = 8 to 15)	NmSA_n	Nextm Source Address Register n (m = 0, 1) (n = 0 to 15)	—
0x800C_0x004 + 0x1000 × i + 0x40 × n (m = 0) (n = 0 to 7) 0x800C_0x404 + 0x1000 × i + 0x40 × (n - 8) (m = 0) (n = 8 to 15) 0x800C_0x010 + 0x1000 × i + 0x40 × n (m = 1) (n = 0 to 7) 0x800C_0x410 + 0x1000 × i + 0x40 × (n - 8) (m = 1) (n = 8 to 15)	NmDA_n	Nextm Destination Address Register n (m = 0, 1) (n = 0 to 15)	—
0x800C_0x008 + 0x1000 × i + 0x40 × n (m = 0) (n = 0 to 7) 0x800C_0x408 + 0x1000 × i + 0x40 × (n - 8) (m = 0) (n = 8 to 15) 0x800C_0x014 + 0x1000 × i + 0x40 × n (m = 1) (n = 0 to 7) 0x800C_0x414 + 0x1000 × i + 0x40 × (n - 8) (m = 1) (n = 8 to 15)	NmTB_n	Nextm Transaction Byte Register n (m = 0, 1) (n = 0 to 15)	—
0x800C_0x018 + 0x1000 × i + 0x40 × n (n = 0 to 7) 0x800C_0x418 + 0x1000 × i + 0x40 × (n - 8) (n = 8 to 15)	CRSA_n	Current Source Address Register n (n = 0 to 15)	—
0x800C_0x01C + 0x1000 × i + 0x40 × n (n = 0 to 7) 0x800C_0x41C + 0x1000 × i + 0x40 × (n - 8) (n = 8 to 15)	CRDA_n	Current Destination Address Register n (n = 0 to 15)	—
0x800C_0x020 + 0x1000 × i + 0x40 × n (n = 0 to 7) 0x800C_0x420 + 0x1000 × i + 0x40 × (n - 8) (n = 8 to 15)	CRTB_n	Current Transaction Byte Register n (n = 0 to 15)	—
0x800C_0x024 + 0x1000 × i + 0x40 × n (n = 0 to 7) 0x800C_0x424 + 0x1000 × i + 0x40 × (n - 8) (n = 8 to 15)	CHSTAT_n	Channel Status Register n (n = 0 to 15)	—
0x800C_0x034 + 0x1000 × i + 0x40 × n (n = 0 to 7) 0x800C_0x434 + 0x1000 × i + 0x40 × (n - 8) (n = 8 to 15)	CHEXT_n	Channel Extension Register n (n = 0 to 15)	—
0x802A_07D0 + 0x04 × n	DMAC0_RSSELn	DMAC Unit 0 Resource Select Register n (n = 0 to 5)	—
0x802A_07E8 + 0x04 × n	DMAC1_RSSELn	DMAC Unit 1 Resource Select Register n (n = 0 to 5)	—
0x802A_0800 + 0x04 × n	DMAC2_RSSELn	DMAC Unit 2 Resource Select Register n (n = 0 to 5)	—
0x8021_0000	CMNCR	Common Control Register	—
0x800C_0x028 + 0x1000 × i + 0x40 × n (n = 0 to 7) 0x800C_0x428 + 0x1000 × i + 0x40 × (n - 8) (n = 8 to 15)	CHCTRL_n	Channel Control Register n (n = 0 to 15)	—
0x800C_0x02C + 0x1000 × i + 0x40 × n (n = 0 to 7) 0x800C_0x42C + 0x1000 × i + 0x40 × (n - 8) (n = 8 to 15)	CHCFG_n	Channel Configuration Register n (n = 0 to 15)	—
0x800C_0x030 + 0x1000 × i + 0x40 × n (n = 0 to 7) 0x800C_0x430 + 0x1000 × i + 0x40 × (n - 8) (n = 8 to 15)	CHITVL_n	Channel Interval Register n (n = 0 to 15)	—
0x800C_0x038 + 0x1000 × i + 0x40 × n (n = 0 to 7) 0x800C_0x438 + 0x1000 × i + 0x40 × (n - 8) (n = 8 to 15)	NXLA_n	Next Link Address Register n (n = 0 to 15)	—
0x800C_0x03C + 0x1000 × i + 0x40 × n (n = 0 to 7) 0x800C_0x43C + 0x1000 × i + 0x40 × (n - 8) (n = 8 to 15)	CRLA_n	Current Link Address Register n (n = 0 to 15)	—
0x800C_0x300 + 0x1000 × i (DCTRL_A) 0x800C_0x700 + 0x1000 × i (DCTRL_B)	DCTRL_x	DMA Control Register x (x = A, B)	—
0x800C_0x310 + 0x1000 × i (DSTAT_EN_A) 0x800C_0x710 + 0x1000 × i (DSTAT_EN_B)	DSTAT_EN_x	DMA Status EN Register x (x = A, B)	—
0x800C_0x314 + 0x1000 × i (DSTAT_ER_A) 0x800C_0x714 + 0x1000 × i (DSTAT_ER_B)	DSTAT_ER_x	DMA Status ER Register x (x = A, B)	—
0x800C_0x318 + 0x1000 × i (DSTAT_END_A) 0x800C_0x718 + 0x1000 × i (DSTAT_END_B)	DSTAT_END_x	DMA Status END Register x (x = A, B)	—
0x800C_0x320 + 0x1000 × i (DSTAT_SUS_A) 0x800C_0x720 + 0x1000 × i (DSTAT_SUS_B)	DSTAT_SUS_x	DMA Status SUS Register x (x = A, B)	—

Note: i = 0 to 2

Table 15.4 DMAC related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
DMAC0_RSSELn DMAC1_RSSELn DMAC2_RSSELn	—	—	SLVACCCTL8.ICU_SL
Other than the above in unit 0	—	—	SLVACCCTL6.DMAC0_SL
Other than the above in unit 1	—	—	SLVACCCTL6.DMAC1_SL
Other than the above in unit 2	—	—	SLVACCCTL6.DMAC2_SL

15.3 Register Descriptions

15.3.1 NmSA_n : Nextm Source Address Register n (m = 0, 1) (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x000 + 0x40 × n (m = 0) (n = 0 to 7)
0x400 + 0x40 × (n - 8) (m = 0) (n = 8 to 15)
0x00C + 0x40 × n (m = 1) (n = 0 to 7)
0x40C + 0x40 × (n - 8) (m = 1) (n = 8 to 15)

Bit position:	31																													0
Bit field:																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Source Address Sets the start address of the DMA transfer source.	R/W

Note: During link mode transfer, descriptor read data is set to the N0SA_n register automatically.

The N0SA_n and N1SA_n registers set the DMA transfer source address of DMA channel n (n = 0 to 15). The N0SA_n register is for Next0 Register Set, and the N1SA_n register is for Next1 Register Set.

15.3.2 NmDA_n : Nextm Destination Address Register n (m = 0, 1) (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x004 + 0x40 × n (m = 0) (n = 0 to 7)
0x404 + 0x40 × (n - 8) (m = 0) (n = 8 to 15)
0x010 + 0x40 × n (m = 1) (n = 0 to 7)
0x410 + 0x40 × (n - 8) (m = 1) (n = 8 to 15)

Bit position:	31																													0
Bit field:																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Destination Address Sets the start address of the DMA transfer destination.	R/W

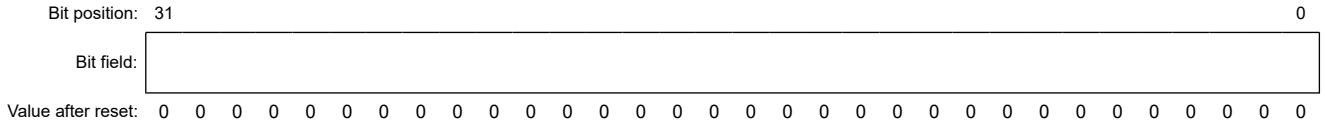
Note: During link mode transfer, descriptor read data is set to the N0DA_n register automatically.

The N0DA_n and N1DA_n registers set the DMA transfer destination address of DMA channel n (n = 0 to 15). The N0DA_n register is for Next0 Register Set, and the N1DA_n register is for the Next1 Register Set.

15.3.3 NmTB_n : Nextm Transaction Byte Register n (m = 0, 1) (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x008 + 0x40 × n (m = 0) (n = 0 to 7)
0x408 + 0x40 × (n - 8) (m = 0) (n = 8 to 15)
0x014 + 0x40 × n (m = 1) (n = 0 to 7)
0x414 + 0x40 × (n - 8) (m = 1) (n = 8 to 15)



Bit	Symbol	Function	R/W
31:0	n/a	Transaction Byte Sets the total number of transfer bytes. Do not start DMA transfer when 0 is set.	R/W

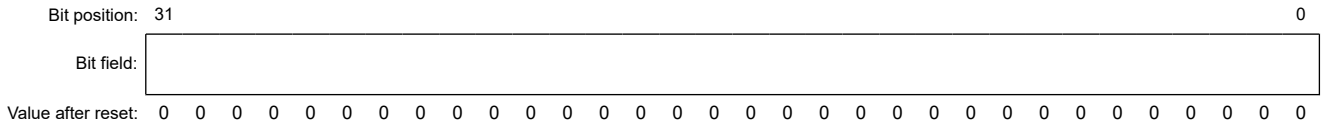
Note: During link mode transfer, descriptor read data is set to the N0TB_n register automatically.

The N0TB_n and N1TB_n registers set the total number of transfer bytes of DMA channel n (n = 0 to 15). The N0TB_n register is for Next0 Register Set, and the N1TB_n register is for Next1 Register Set.

15.3.4 CRSA_n : Current Source Address Register n (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x018 + 0x40 × n (n = 0 to 7)
0x418 + 0x40 × (n - 8) (n = 8 to 15)



Bit	Symbol	Function	R/W
31:0	n/a	Current Source Address Displays the read address of the next DMA transfer.	R

The CRSA_n register is a register that displays the DMA transfer source address of DMA channel n (n = 0 to 15). During DMA transfer, the value is incremented automatically (fixed when CHCFG_n.SAD = 1).

The value after a reset is loaded from the following registers:

- In register mode:
  Loads the transfer source address from the Next0/1 register.
- In link mode:
  Loads the transfer source address from the descriptor read data (the hardware inputs the descriptor read data to the N0SA_n register automatically, and loads it to the CRSA_n register when transfer starts).

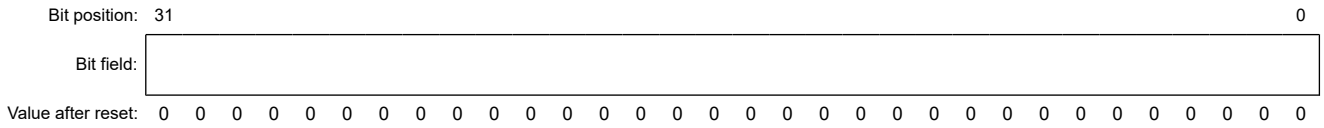
The value is incremented when reading data from the transfer source completes.

Read this register after DMA stops (when CHSTAT_n.EN = 0). Handle the value during DMA operation as a reference value.

15.3.5 CRDA_n : Current Destination Address Register n (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x01C + 0x40 × n (n = 0 to 7)
0x41C + 0x40 × (n - 8) (n = 8 to 15)



Bit	Symbol	Function	R/W
31:0	n/a	Current Destination Address Displays the write address of the next DMA transfer.	R

The CRDA_n register displays the DMA transfer destination address of DMA channel n (n = 0 to 15).

During DMA transfer, the value is incremented automatically (fixed when CHCFG_n.DAD = 1).

The value after a reset is loaded from the following registers:

- In register mode:
  Loads the transfer destination address from the Next0/1 register.
- In link mode:
  Loads the transfer destination address from the descriptor read data (the hardware inputs the descriptor read data to the N0DA_n register automatically, and loads it to the CRDA_n register when transfer starts).

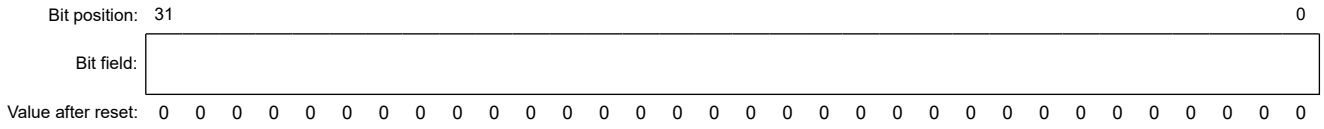
The value is incremented when writing data to the transfer destination completes.

Read this register after DMA stops (when CHSTAT_n.EN = 0). Handle the value during DMA operation as a reference value.

15.3.6 CRTB_n : Current Transaction Byte Register n (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x020 + 0x40 × n (n = 0 to 7)
0x420 + 0x40 × (n - 8) (n = 8 to 15)



Bit	Symbol	Function	R/W
31:0	n/a	Current Transaction Byte Displays the remaining number of transfer bytes that is currently performed.	R

The CRTB_n register displays the total number of transfer bytes of DMA channel n (n = 0 to 15). The value is cleared to 0 when transfer completes.

During DMA transfer, the value is decremented automatically.

The value after a reset is loaded from the following registers:

- In register mode:
  Loads the number of transfer bytes from the Next0/1 register.
- In link mode:
  Loads the number of transfer bytes from descriptor read data (the hardware inputs the descriptor read data to the N0TB_n register automatically and loads it to the CRTB_n register when transfer starts).

The value is decremented when writing data to the transfer destination completes.

Read this register after DMA stops (when CHSTAT_n.EN = 0). Handle the value during DMA operation as a reference value.

15.3.7 CHSTAT_n : Channel Status Register n (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x024 + 0x40 × n (n = 0 to 7)
0x424 + 0x40 × (n - 8) (n = 8 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INTM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MODE	DER	DW	DL	SR	TC	END	ER	SUS	TACT	RQST	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN	DMA Activation Enable Displays the operation status of DMA channel n. [Setting condition] ● Writing 1 to the SETEN bit in the CHCTRL_n register [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the CLREN bit in the CHCTRL_n register ● Writing 1 to the SWRST bit in the CHCTRL_n register ● When a bus error is reported during transfer ● When all DMA transfers complete in register mode (transfers complete when the REN bit in the CHCFG_n register = 0) ● In link mode, when DMA transfer (write back when WBD = 0) of the last descriptor (LE = 1) completes ● When reading the descriptor in link mode stopped (when LV = 0) 0: Operation disabled 1: Operation enabled	R
1	RQST	DMA Transfer Request This bit indicates that the transfer request is accepted. [Setting condition] ● The transfer request is accepted. [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the SWRST bit in the CHCTRL_n register ● Writing 1 to the STG bit in the CHCTRL_n register ● Writing 1 to the CLRRQ bit in the CHCTRL_n register ● In single transfer (the TM bit in the CHCFG_n register = 0) mode, the transfer specified in the REQD bit in the CHCFG_n register is performed ● When all DMA transfers complete in register mode (transfer completes when the REN bit in the CHCFG_n register = 0) ● In link mode, DMA transfer of the last descriptor (LE = 1) completes ● In link mode, reading the descriptor is disabled (when LV = 0) ● In link mode, the DEM bit in the CHCFG_n register = 0, and DMA transfer completes ● When a bus error is reported to the master interface 0: DMA transfer request not accepted 1: DMA transfer request accepted	R

Bit	Symbol	Function	R/W
2	TACT	DMAC Operating Status This bit indicates that DMAC is running. This bit is used to ensure the channel stops completely. For details, see section 15.4.9. DMA Transfer Status. [Setting condition] ● Writing 1 to the SETEN bit in the CHCTRL_n register (reading the descriptor starts, or the DMA request is being serviced). [Clearing condition] ● When the internal state is idling (the EN bit in the CHSTAT_n register is cleared to 0, and all DMA transfer completes). 0: DMA in Channel_n stops 1: DMA in Channel_n is running	R
3	SUS	Suspend This bit indicates that the channel is suspended. For details, see section 15.4.10. Suspending a Transfer. [Setting condition] ● Writing 1 to the SETSUS bit in the CHCTRL_n register during DMA transfer of Channel_n, and the internal state is changed to the suspended state [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the CLRSUS bit in the CHCTRL_n register ● Writing 1 to the CLREN bit in the CHCTRL_n register ● Condition for clearing the EN bit in the CHSTAT_n register 0: Channel_n not suspended 1: Channel_n suspended	R
4	ER	DMA Error This bit indicates that a DMA error interrupt is generated as a result of the bus error during DMA transfer. [Setting condition] ● A bus error is reported to the bus cycle [Clearing condition] ● Writing 1 to the SWRST bit in the CHCTRL_n register 0: No bus error occurred 1: A bus error occurred	R
5	END	DMA Transfer Completion Interrupt This bit indicates that DMA transfer completes, and a DMA interrupt is generated. [Setting conditions] When any of the following conditions is met: ● When the TC bit setting condition is met while DEM bit in the CHCFG_n register = 0 ● When the descriptor is read in link mode, LV of header = 0, and DIM in the header register = 0 [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the CLREND bit in the CHCTRL_n register ● Writing 1 to the SWRST bit in the CHCTRL_n register 0: DMA transfer not completed 1: DMA transfer completed	R
6	TC	DMA Transfer Completion (total number of data bytes for transaction) This bit indicates that DMA transfer is completed. [Setting conditions] When any of the following conditions is met: ● When a transfer for the total number of data bytes for transaction set in the CRTB_n register is completed in register mode ● When the header bit WBD = 1 in link mode, and a transfer for the total number of data bytes for transaction set in the CRTB_n register is completed ● When the header bit WBD = 0 in link mode, and write-back to the descriptor is completed [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the CLRTC bit in the CHCTRL_n register ● Writing 1 to the SWRST bit in the CHCTRL_n register 0: DMA transfer not completed 1: DMA transfer completed	R

Bit	Symbol	Function	R/W
7	SR	Next Register Select In register mode, this bit indicates the selected register set. [Setting condition] ● Setting 1 to the RSEL bit in the CHCFG_n register [Clearing condition] ● Clearing the RSEL bit in the CHCFG_n register to 0 0: Next0 register set 1: Next1 register set	R
8	DL	Descriptor Load This bit indicates that the descriptor is being read. In addition, if a bus error is reported while the descriptor is being read, 1 is retained. [Setting condition] ● Reading the descriptor in link mode is started [Clearing conditions] When any of the following conditions is met: ● Reading the descriptor in link mode completes with the OK response ● Writing 1 to the SWRST bit in the CHCTRL_n register (if 1 is retained for the bus error, it can be cleared to 0 only by the SWRST bit) 0: The descriptor is not being read 1: When ER = 0 The descriptor is being read in link mode When ER = 1 A bus error occurred while the descriptor is being read in link mode	R
9	DW	Descriptor Write Back This bit indicates that the descriptor is being written back. In addition, if a bus error is reported while the descriptor is written back, 1 is retained. [Setting condition] ● Writing back of header in link mode is started [Clearing conditions] ● Writing back of header in link mode completes with the OK response ● Writing 1 to the SWRST bit in the CHCTRL_n register (if 1 is retained due to the bus error, it can be cleared to 0 only by the SWRST bit) 0: Header is not written back in link mode 1: The ER bit in the CHSTAT_n register = 0 Header is written back in link mode The ER bit in the CHSTAT_n register = 1 A bus error occurred when header is written back in link mode	R
10	DER	Descriptor Error This bit indicates that the read descriptor is invalid (LV = 0) (does not depend on the value of the DIM bit in the header register). [Setting condition] ● In link mode, the LV of the read descriptor is 0 [Clearing conditions] When the following condition is met: ● Writing 1 to the SWRST bit in the CHCTRL_n register 0: No descriptor error occurred 1: Descriptor error occurred	R
11	MODE	DMA Mode This bit indicates DMA mode and displays the setting value of the DMS bit in the CHCFG_n register. 0: Register mode 1: Link mode	R
15:12	—	These bits are read as 0.	R
16	INTM	DMA Transfer Completion Interrupt Request Mask This bit displays the temporary mask status of DMA interrupt output. [Setting condition] ● Writing 1 to the SETINTM bit in the CHCTRL_n register [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the CLRINTM bit in the CHCTRL_n register ● Writing 1 to the SWRST bit in the CHCTRL_n register 0: The temporary mask status is cleared 1: Temporary mask status	R
31:17	—	These bits are read as 0.	R

Note:
- Handle the transfer as invalid when CHSTAT_n.ER is set to 1.
- To suspend DMA transfer, mask or clear the transfer request, or clear CHSTAT_n.EN. See section 15.4.11. Aborting a Transfer for the procedure).
- To request a transfer by software, make sure that the previously requested DMA transfer operation completes (by checking the current register), set the STG bit in the Channel Control Register n (CHCTRL_n), then activate DMA.
- When DMA transfer for the same channel is requested by both hardware and software, the DMA transfer request cannot be identified for which. Therefore, set the DMA transfer request on your system to either hardware or software.

The CHSTAT_n register displays the status of DMA channel n (n = 0 to 15).

15.3.8 CHEXT_n : Channel Extension Register n (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x034 + 0x40 × n (n = 0 to 7)
0x434 + 0x40 × (n - 8) (n = 8 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DCA[3:0]				—	DPR[2:0]			SCA[3:0]				—	SPR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SPR[2:0]	Source PROT These bits set the value to be output to the PROT[2:0] pins during DMA read transfer.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
7:4	SCA[3:0]	Source CACHE These bits set the value to be output to the CACHE[3:0] pins during DMA read transfer.	R/W
10:8	DPR[2:0]	Destination PROT These bits set the value to be output to the PROT[2:0] pins during DMA write transfer.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
15:12	DCA[3:0]	Destination CACHE These bits set the value to be output to the CACHE[3:0] pins during DMA write transfer.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The CHEXT_n register is an extension register for DMA channel n.

15.3.9 DMAC0_RSSELi : DMAC Unit 0 Resource Select Register i (i = 0 to 5)

Base address: DMA = 0x802A_07D0

Offset address: 0x00 + 0x04 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	REQ_SELC[9:0]										REQ_SELB[9:0]										REQ_SELA[9:0]									
Value after reset:	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
9:0	REQ_SELA[9:0]	DMA Resource Select for Channel n Select the trigger source of the DMA channel. 0x3FF: No resource assignment 0x000: ⋮ 0x3FE: DMA trigger source is assigned (event number)	R/W

Bit	Symbol	Function	R/W
19:10	REQ_SELB[9:0]	DMA Resource Select for Channel n + 1 Select the trigger source of the DMA channel. When i = 5, the REQ_SELB[9:0] bits are invalid. 0x3FF: No resource assignment 0x000: ⋮ 0x3FE: DMA trigger source is assigned (event number)	R/W
29:20	REQ_SELC[9:0]	DMA Resource Select for Channel n + 2 Select the trigger source of the DMA channel. When i = 5, the REQ_SELC[9:0] bits are invalid. 0x3FF: No resource assignment 0x000: ⋮ 0x3FE: DMA trigger source is assigned (event number)	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note:
- Relationship between register number and channel number is n = i × 3.
- The [29:10] bits in the DMAC0_RSSEL5 register are read as 0. The write value should be 0.

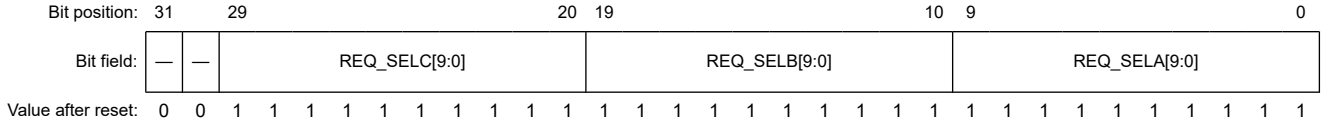
The DMAC0_RSSELi register selects the activation trigger source for DMAC unit 0. For details on numbers that are selected by this source selection, see the event numbers listed in section xx.xx.xx Event Table.

In addition, do not set the same source for multiple DMAC0_RSSELi, DMAC1_RSSELi, and DMAC2_RSSELi registers. If the same source is set, the operation of this LSI cannot be guaranteed.

15.3.10 DMAC1_RSSELi : DMAC Unit 1 Resource Select Register i (i = 0 to 5)

Base address: DMA = 0x802A_07D0

Offset address: 0x18 + 0x04 × i



Bit	Symbol	Function	R/W
9:0	REQ_SELA[9:0]	DMA Resource Select for Channel n Select the trigger source of the DMA channel. 0x3FF: No resource assignment 0x000: ⋮ 0x3FE: DMA trigger source is assigned (event number)	R/W
19:10	REQ_SELB[9:0]	DMA Resource Select for Channel n + 1 Select the trigger source of the DMA channel. When i = 5, the REQ_SELB[9:0] bits are invalid. 0x3FF: No resource assignment 0x000: ⋮ 0x3FE: DMA trigger source is assigned (event number)	R/W
29:20	REQ_SELC[9:0]	DMA Resource Select for Channel n + 2 Select the trigger source of the DMA channel. When i = 5, the REQ_SELC[9:0] bits are invalid. 0x3FF: No resource assignment 0x000: ⋮ 0x3FE: DMA trigger source is assigned (event number)	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note:
- Relationship between register number and channel number is n = i × 3.
- The [29:10] bits in the DMAC1_RSSEL5 register are read as 0. The write value should be 0.

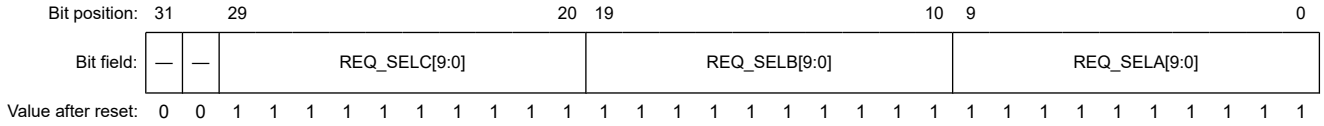
The DMAC1_RSSELi register selects the activation trigger source for DMAC unit 1. For details on numbers that are selected by this source selection, see the event numbers listed in section xx.xx.xx Event Table.

In addition, do not set the same source for multiple DMAC0_RSSELi, DMAC1_RSSELi, and DMAC2_RSSELi registers. If the same source is set, the operation of this LSI cannot be guaranteed.

15.3.11 DMAC2_RSSELi : DMAC Unit 2 Resource Select Register i (i = 0 to 5)

Base address: DMA = 0x802A_07D0

Offset address: 0x30 + 0x04 × i



Bit	Symbol	Function	R/W
9:0	REQ_SELA[9:0]	DMA Resource Select for Channel n Select the trigger source of the DMA channel. 0x3FF: No resource assignment 0x000: ⋮ 0x3FE: DMA trigger source is assigned (event number)	R/W
19:10	REQ_SELB[9:0]	DMA Resource Select for Channel n + 1 Select the trigger source of the DMA channel. When i = 5, the REQ_SELB[9:0] bits are invalid. 0x3FF: No resource assignment 0x000: ⋮ 0x3FE: DMA trigger source is assigned (event number)	R/W
29:20	REQ_SELC[9:0]	DMA Resource Select for Channel n + 2 Select the trigger source of the DMA channel. When i = 5, the REQ_SELC[9:0] bits are invalid. 0x3FF: No resource assignment 0x000: ⋮ 0x3FE: DMA trigger source is assigned (event number)	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note:
- Relationship between register number and channel number is n = i × 3.
- The [29:10] bits in the DMAC1_RSSEL5 register are read as 0. The write value should be 0.

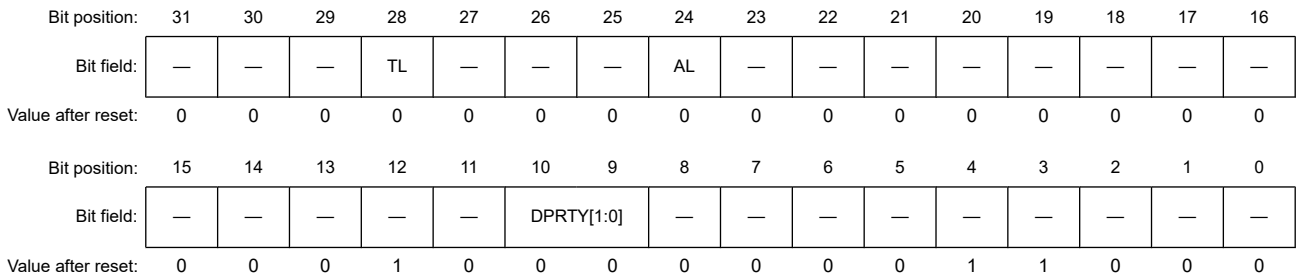
The DMAC2_RSSELi register selects the activation trigger source for DMAC unit 2. For details on numbers that are selected by this source selection, see the event numbers listed in section xx.xx.xx Event Table.

In addition, do not set the same source for multiple DMAC0_RSSELi, DMAC1_RSSELi, and DMAC2_RSSELi registers. If the same source is set, the operation of this LSI cannot be guaranteed.

15.3.12 CMNCR : Common Control Register

Base address: BSC = 0x8021_0000

Offset address: 0x00



Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
4:3	—	These bits are read as 1. The write value should be 1.	R/W
8:5	—	These bits are read as 0. The write value should be 0.	R/W
10:9	DPRTY[1:0]	DMA Burst Transfer Priority 0 x: Accept a refresh request during DMA burst transfer 1 0: Do not accept a refresh request during DMA burst transfer 1 1: Setting prohibited	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	—	This bit is read as 1. The write value should be 1.	R/W
23:13	—	These bits are read as 0. The write value should be 0.	R/W
24	AL	Acknowledge Level Specify the active level of the DACK signal 0: Active-low output from DACK 1: Active-high output from DACK	R/W
27:25	—	These bits are read as 0. The write value should be 0.	R/W
28	TL	Transfer End Level Specify the active level of the TEND signal 0: Active-low output from TEND 1: Active-high output from TEND	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

The CMNCR register is a 32-bit register that controls the common items from each area.

15.3.13 CHCTRL_n : Channel Control Register n (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x028 + 0x40 × n (n = 0 to 7)
0x428 + 0x40 × (n - 8) (n = 8 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRINTM	SETINTM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CLRSUS	SETSUS	—	CLRTC	CLRENDQ	CLRRQ	SWRST	STG	CLREN	SETEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SETEN	DMA Activation Enable Enable DMA transfer of DMA channel n. When this bit is set with the SWRST bit, clearing by the SWRST bit takes precedence, and a transfer does not start. This bit is always read as 0. To reset the DMA register, stop the ongoing DMA transfer by setting the CLREN bit and then set the SETEN bit. 0: Operation is not affected 1: DMA transfer is enabled (the EN bit in the CHSTAT_n register is set)	R/W
1	CLREN	DMA Activation Enable Clear Clear the EN bit in the CHSTAT_n register (for details, see section 15.4.11. Aborting a Transfer). This bit is always read as 0. 0: Operation is not affected 1: DMA transfer is disabled (the EN bit in the CHSTAT_n register is cleared)	R/W

Bit	Symbol	Function	R/W
2	STG	Software Trigger Set a transfer request by software. When this bit is set at the same time as the SWRST bit is being set, clearing by the SWRST bit takes precedence. This bit is always read as 0. 0: Operation is not affected 1: A transfer request is set by software (the RQST bit in the CHSTAT_n register is set)	R/W
3	SWRST	Software Reset Clear each bit in the CHSTAT_n register (for details on the bit to be cleared, see the descriptions of the applicable bit). Set this bit to 0 when the EN bit and the TACT bit are cleared to 0. This bit is always read as 0. 0: Operation is not affected 1: Clear each bit in the CHSTAT_n register	R/W
4	CLRRQ	DMA Transfer Request Clear Clear the RQST bit in the CHSTAT_n register to 0. This bit is always read as 0. 0: Operation is not affected 1: Clear the RQST bit in the CHSTAT_n register	R/W
5	CLREND	END Clear Clear the END bit in the CHSTAT_n register to 0. This bit is always read as 0. 0: Operation is not affected 1: Clear the END bit	R/W
6	CLRTC	TC Clear Clear the TC bit in the CHSTAT_n register to 0. This bit is always read as 0. 0: Operation is not affected 1: Clear the TC bit in the CHSTAT_n register	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	SETSUS	Suspend Request If 1 is set to this bit when the EN bit in the CHSTAT_n register is 1, the current DMA transfer is suspended. This bit is always read as 0. 0: Operation is not affected 1: Suspends the current DMA transfer	R/W
9	CLRSUS	Suspend Clear If 1 is set to this bit when the SUS bit in the CHSTAT_n register is 1, the temporary suspend status is cleared. This bit is always read as 0. 0: Operation is not affected 1: Clear the temporary suspend status of the current DMA transfer	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	SETINTM	DMA Transfer Completion Interrupt Request Mask Mask DMA transfer completion interrupt output temporarily. Additionally, the INTM bit in the CHSTATn register is set to 1. This bit is always read as 0. 0: Operation is not affected 1: Mask a DMA transfer completion interrupt temporarily	R/W
17	CLRINTM	DMA Transfer Completion Interrupt Request Mask Clear Clear the mask status of DMA transfer completion interrupt output. Additionally, the INTM bit in the CHSTATn register is cleared to 0. If the mask state is cleared when the LVINT bit in the DCTRL register = 1 and the END bit in the CHSTAT_n register = 1, DMA transfer completion interrupt is deactivated. It is not activated when LVINT = 0. This bit is always read as 0. 0: Operation is not affected 1: Clear the mask state set with the SETINTM bit	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The CHCTRL_n register controls DMA transfer of DMA channel n (n = 0 to 15).

15.3.14 CHCFG_n : Channel Configuration Register n (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x02C + 0x40 × n (n = 0 to 7)
0x42C + 0x40 × (n - 8) (n = 8 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMS	REN	RSW	RSEL	SBE	—	TCM	DEM	—	TM	DAD	SAD	DDS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SDS[3:0]				—	AM[2:0]			—	LVL	HIEN	LOEN	REQD	SEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SEL[2:0]	Pin Select Set channels of DMAC. Set the following values so that the channels of CHCFG_n (n = 0 to 15) and the channel set by SEL become the same. For example, set 001b to the SEL bits in CHCFG_1. 0 0 0: Channel 0/8 0 0 1: Channel 1/9 0 1 0: Channel 2/10 0 1 1: Channel 3/11 1 0 0: Channel 4/12 1 0 1: Channel 5/13 1 1 0: Channel 6/14 1 1 1: Channel 7/15	R/W
3	REQD	DMA Activation Request Source Select Specifies the DMA transfer request input source (internal modules or external devices that require DACK). The timing when DACK becomes active is determined by the setting of this bit. 0: Requested by a transfer source module. DACK output is activated when it is read (a value after a reset). 1: Requested by a transfer destination module. DACK output is activated when it is written.	R/W
4	LOEN	L Detection Enable Specifies the detection method of the DMA request signal. When LVL = 0 LOEN = 1: If DMA transfer request input detects a falling edge, it is regarded as requested. LOEN = 0: If the DMA transfer request input falls, the request is not recognized (a value after a reset). When LVL = 1 LOEN = 1: If DMA transfer request input detects a low level, it is regarded as requested. LOEN = 0: If the DMA transfer request input is a low level, the request is not recognized (a value after reset).	R/W
5	HIEN	H Detection Enable Specifies the detection method of the DMA request signal. When LVL = 0 HIEN = 1: If DMA transfer request input detects a rising edge, it is regarded as requested. HIEN = 0: If the DMA transfer request input rises, the request is not recognized (a value after reset). When LVL = 1 HIEN = 1: If DMA transfer request input detects a high level, it is regarded as requested. HIEN = 0: If the DMA transfer request input is a high level, the request is not recognized (a value after reset).	R/W

Bit	Symbol	Function	R/W
6	LVL	Level Detection Enable Specifies the detection method of the DMA request signal. 0: Detects the edge (a value after reset) 1: Detects the level	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	AM[2:0]	ACK Mode These bits specify the DACK output mode. For the conditions of outputting the DACK and TEND signals, see section 15.4.5. DMA Acknowledge Output/DMA Transaction Completion Output Function. 0 0 0: Setting is prohibited (a value after reset). At the initial setup, set a value other than 000b. 0 0 1: Level mode (active until DMA transfer request input becomes inactive) 0 1 x: Bus cycle mode (active between DMA transfer cycles (active if internal resources are the target)) 1 x x: Mask DACK output	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
15:12	SDS[3:0]	Source Data Size Set the size of data in the transfer source to be transferred at a time. 0x0: 8 bits (a value after reset) 0x1: 16 bits 0x2: 32 bits 0x3: 64 bits 0x4: 128 bits*1 0x5: 256 bits*2 0x6: 512 bits (can be set only for DMAC0)*1 Others: Setting prohibited	R/W
19:16	DDS[3:0]	Destination Data Size Set the size of data in the transfer destination to be transferred at a time. 0x0: 8 bits (a value after reset) 0x1: 16 bits 0x2: 32 bits 0x3: 64 bits 0x4: 128 bits*1 0x5: 256 bits*2 0x6: 512 bits (can be set only for DMAC0)*1 Others: Setting prohibited	R/W
20	SAD	Source Address Count Direction Set the count direction of the transfer source address of DMA channel n. When the transfer source address is beat-unaligned, specify SAD = 0 (increment). When the transfer source address is aligned with a boundary of the size specified by the SDS[3:0] bits, SAD can be set to 0 or 1. 0: Increment (a value after reset) 1: Fixed	R/W
21	DAD	Destination Address Count Direction Set the count direction of the transfer destination address of DMA channel n. When the transfer destination address is beat-unaligned, specify DAD = 0 (increment). When the transfer destination address is aligned with a boundary of the size specified by the DDS[3:0] bits, DAD can be set to 0 or 1. 0: Increment (a value after reset) 1: Fixed	R/W
22	TM	Transfer Mode Set DMA transfer mode. 0: Single transfer mode (a value after reset) 1: Block transfer mode	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
24	DEM	DMA Transfer Completion Interrupt Mask Mask DMA transfer completion interrupt detection. If this bit is set to 1 when DMA transfer completion interrupt is output, DMA transfer completion interrupt is not activated. At this time, the DEM bit is automatically cleared to 0. [Clearing condition] ● When DEM = 1 and DMA transfer completes 0: Does not mask (a value after reset) 1: Mask	R/W
25	TCM	TEND Mask Mask the TEND output. If this bit is 1 at the TEND output timing, TEND is not asserted. At this time, the TCM bit is automatically cleared to 0. Use this bit to control DMA transfers by software. [Clearing condition] ● When TCM = 1 and DMA transfer completes 0: Does not mask (a value after reset) 1: Mask	R/W
26	—	This bit is read as 0. The write value should be 0.	R/W
27	SBE	Buffer Flush Enable If the EN bit in the CHSTAT_n register is cleared to 0 during DMA transfer, select whether to stop flushing (writing) data that is already read and stored in the buffer. Only when REQD = 0, flush mode can be used. 0: Stop transfer without flushing data in the buffer (a value after reset) 1: Stop transfer after flushing data in the buffer	R/W
28	RSEL	Next Register Select Select the Next register set to be executed next. This bit is valid in register mode only. When RSW = 1, the value is reversed automatically (0 is reversed to 1, 1 is reversed to 0) after DMA transfer completes. [Transition condition] ● When RSW = 1 and DMA transfer completes 0: Execute Next0 Register Set (a value after reset) 1: Execute Next1 Register Set	R/W
29	RSW	RSEL Reverse When DMA transfer completes, the RSEL bit is automatically reversed (0 is reversed to 1, 1 is reversed to 0). This bit is valid in register mode only. 0: Does not reverse the RSEL bit when DMA transfer completes (a value after reset) 1: Reverse the RSEL bit when DMA transfer completes	R/W
30	REN	Register Set Enable When DMA transfer completes, performs DMA transfer of the Next register set selected by the RSEL bit accordingly. This bit is valid in register mode only. [Setting condition] ● Writing 1 to this bit [Clearing conditions] When any of the following conditions is met: ● Writing 0 to this bit ● When REN = 1 and DMA transfer completes 0: Does not perform DMA transfer accordingly 1: Performs DMA transfer accordingly	R/W
31	DMS	DMA Mode Select Set DMA mode. 0: Register mode (a value after reset). 1: Link mode	R/W

Note 1. The SCA[3:0] bits in the CHEXT_n register should be set to 0x2 if source is TCM. The DCA[3:0] bits in the CHEXT_n register should be set to 0x2 if destination is TCM.
Note 2. This setting is prohibited when DMAC accesses TCM.

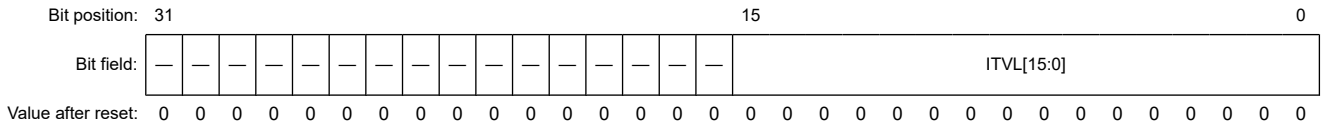
The CHCFG_n register controls DMA transfer of DMA channel n (n = 0 to 15).

Specify the detection method depending on the DMA transfer source to be used. For details on the DMA request signals, see section 15.4.4.1. Specifying Detection Operation of DMA Transfer Requests for Each Source.

15.3.15 CHITVL_n : Channel Interval Register n (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x030 + 0x40 × n (n = 0 to 7)
0x430 + 0x40 × (n - 8) (n = 8 to 15)



Bit	Symbol	Function	R/W
15:0	ITVL[15:0]	Interval These bits set the DMA transfer internal.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

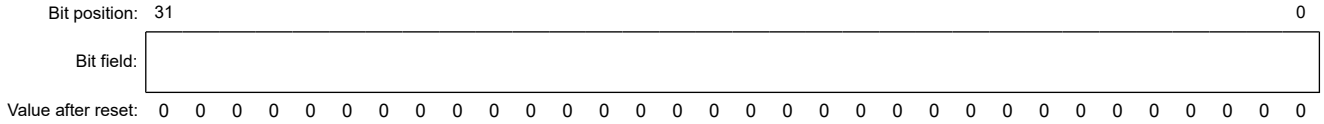
The CHITVL_n register sets the DMA transfer interval of DMA channel n (n = 0 to 15).

For details, see section 15.4.7. Interval Count Function.

15.3.16 NXLA_n : Next Link Address Register n (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x038 + 0x40 × n (n = 0 to 7)
0x438 + 0x40 × (n - 8) (n = 8 to 15)



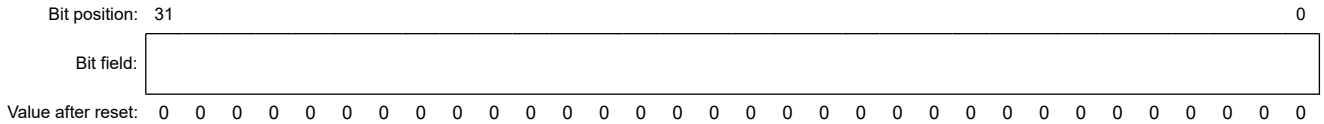
Bit	Symbol	Function	R/W
31:0	n/a	Next Link Address Sets the address of the link destination. Because the 2 lower-order bits are fixed to 0, only word-aligned addresses can be set.	R/W

The NXLA_n register sets the link address of DMA channel n (n = 0 to 15).

15.3.17 CRLA_n : Current Link Address Register n (n = 0 to 15)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x03C + 0x40 × n (n = 0 to 7)
0x43C + 0x40 × (n - 8) (n = 8 to 15)



Bit	Symbol	Function	R/W
31:0	n/a	Current Link Address Displays the address of the descriptor that is being executed.	R

The CRLA_n register sets the link address of DMA channel n (n = 0 to 15).

15.3.18 DCTRL_x : DMA Control Register x (x = A, B)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x300 (DCTRL_A)
0x700 (DCTRL_B)

Bit position:	31																												1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LVINT	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PR	Priority Control Select Sets the transfer priority control mode (see section 15.4.3. DMA Channel Priority Control. 0: Fixed priority mode 1: Round-robin mode	R/W
1	LVINT	DMA Interrupt Output Select Sets the interrupt output mode. 0: Pulse output 1: Level output	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The DCTRL_x register sets the arbitration between channels in all channels (DCTRL_A = channels 0 to 7, DCTRL_B = channels 8 to 15).

15.3.19 DSTAT_EN_x : DMA Status EN Register x (x = A, B)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x310 (DSTAT_EN_A)
0x710 (DSTAT_EN_B)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	EN0715	EN0614	EN0513	EN0412	EN0311	EN0210	EN0109	EN0008
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN0008	Channel 0/8 EN Displays the status for the EN bit of DMA channel 0/8.	R
1	EN0109	Channel 1/9 EN Displays the status for the EN bit of DMA channel 1/9.	R
2	EN0210	Channel 2/10 EN Displays the status for the EN bit of DMA channel 2/10.	R
3	EN0311	Channel 3/11 EN Displays the status for the EN bit of DMA channel 3/11.	R
4	EN0412	Channel 4/12 EN Displays the status for the EN bit of DMA channel 4/12.	R
5	EN0513	Channel 5/13 EN Displays the status for the EN bit of DMA channel 5/13.	R
6	EN0614	Channel 6/14 EN Displays the status for the EN bit of DMA channel 6/14.	R

Bit	Symbol	Function	R/W
7	EN0715	Channel 7/15 EN Displays the status for the EN bit of DMA channel 7/15.	R
31:8	—	These bits are read as 0.	R

The DSTAT_EN_x register displays the status for the EN bit of all channels (DSTAT_EN_A = channels 0 to 7, DSTAT_EN_B = channels 8 to 15).

Writing data to this register does not change the value of each bit.

15.3.20 DSTAT_ER_x : DMA Status ER Register x (x = A, B)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x314 (DSTAT_ER_A)
0x714 (DSTAT_ER_B)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ER0715	ER0614	ER0513	ER0412	ER0311	ER0210	ER0109	ER0008
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER0008	Channel 0/8 ER Displays the status for the ER bit of DMA channel 0/8.	R
1	ER0109	Channel 1/9 ER Displays the status for the ER bit of DMA channel 1/9.	R
2	ER0210	Channel 2/10 ER Displays the status for the ER bit of DMA channel 2/10.	R
3	ER0311	Channel 3/11 ER Displays the status for the ER bit of DMA channel 3/11.	R
4	ER0412	Channel 4/12 ER Displays the status for the ER bit of DMA channel 4/12.	R
5	ER0513	Channel 5/13 ER Displays the status for the ER bit of DMA channel 5/13.	R
6	ER0614	Channel 6/14 ER Displays the status for the ER bit of DMA channel 6/14.	R
7	ER0715	Channel 7/15 ER Displays the status for the ER bit of DMA channel 7/15.	R
31:8	—	These bits are read as 0.	R

The DSTAT_ER_x register displays the status for the ER bit of all channels (DSTAT_ER_A = channels 0 to 7, DSTAT_ER_B = channels 8 to 15).

Writing data to this register does not change the value of each bit.

15.3.21 DSTAT_END_x : DMA Status END Register x (x = A, B)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x318 (DSTAT_END_A)
0x718 (DSTAT_END_B)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	END0715	END0614	END0513	END0412	END0311	END0210	END0109	END0008
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	END0008	Channel 0/8 END Displays the status for the END bit of DMA channel 0/8.	R
1	END0109	Channel 1/9 END Displays the status for the END bit of DMA channel 1/9.	R
2	END0210	Channel 2/10 END Displays the status for the END bit of DMA channel 2/10.	R
3	END0311	Channel 3/11 END Displays the status for the END bit of DMA channel 3/11.	R
4	END0412	Channel 4/12 END Displays the status for the END bit of DMA channel 4/12.	R
5	END0513	Channel 5/13 END Displays the status for the END bit of DMA channel 5/13.	R
6	END0614	Channel 6/14 END Displays the status for the END bit of DMA channel 6/14.	R
7	END0715	Channel 7/15 END Displays the status for the END bit of DMA channel 7/15.	R
31:8	—	These bits are read as 0.	R

The DSTAT_END_x register displays the status for the END bit of all channels (DSTAT_END_A = channels 0 to 7, DSTAT_END_B = channels 8 to 15).

Writing data to this register does not change the value of each bit.

15.3.22 DSTAT_SUS_x : DMA Status SUS Register x (x = A, B)

Base address: DMACi = 0x800C_0000 + 0x1000 × i (i = 0 to 2)

Offset address: 0x320 (DSTAT_SUS_A)
0x720 (DSTAT_SUS_B)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SUS0715	SUS0614	SUS0513	SUS0412	SUS0311	SUS0210	SUS0109	SUS0008
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SUS0008	Channel 0/8 SUS Displays the status for the SUS bit of DMA channel 0/8.	R
1	SUS0109	Channel 1/9 SUS Displays the status for the SUS bit of DMA channel 1/9.	R
2	SUS0210	Channel 2/10 SUS Displays the status for the SUS bit of DMA channel 2/10.	R
3	SUS0311	Channel 3/11 SUS Displays the status for the SUS bit of DMA channel 3/11.	R
4	SUS0412	Channel 4/12 SUS Displays the status for the SUS bit of DMA channel 4/12.	R
5	SUS0513	Channel 5/13 SUS Displays the status for the SUS bit of DMA channel 5/13.	R
6	SUS0614	Channel 6/14 SUS Displays the status for the SUS bit of DMA channel 6/14.	R
7	SUS0715	Channel 7/15 SUS Displays the status for the SUS bit of DMA channel 7/15.	R
31:8	—	These bits are read as 0.	R

The DSTAT_SUS_x register displays the status for the SUS bit of all channels (DSTAT_SUS_A = channels 0 to 7, DSTAT_SUS_B = channels 8 to 15).

Writing data to this register does not change the value of each bit.

15.4 Operation

15.4.1 DMA Mode

With the DMS bit in the CHCFG_n register, DMA mode can be switched between register mode and link mode.

Table 15.5 DMA mode setting

DMS (CHCFG_n)	Description	Applications
0	Register mode	With the values set for the next register set, performs a DMA transfer.
1	Link mode	Accesses the descriptor area, and performs a DMA transfer with the value set for the descriptor. Repeats descriptor read and DMA transfer unless you set the descriptor or use the control register to stop them.

15.4.1.1 Register Mode

In register mode, you can perform a DMA transfer with the value set in the internal register. You can set two sets (Next0 Register Set and Next1 Register Set) of the transfer source addresses, transfer destination addresses, and numbers of transfer bytes. You can select the next register set to perform a transfer, or transfer two next register sets successively.

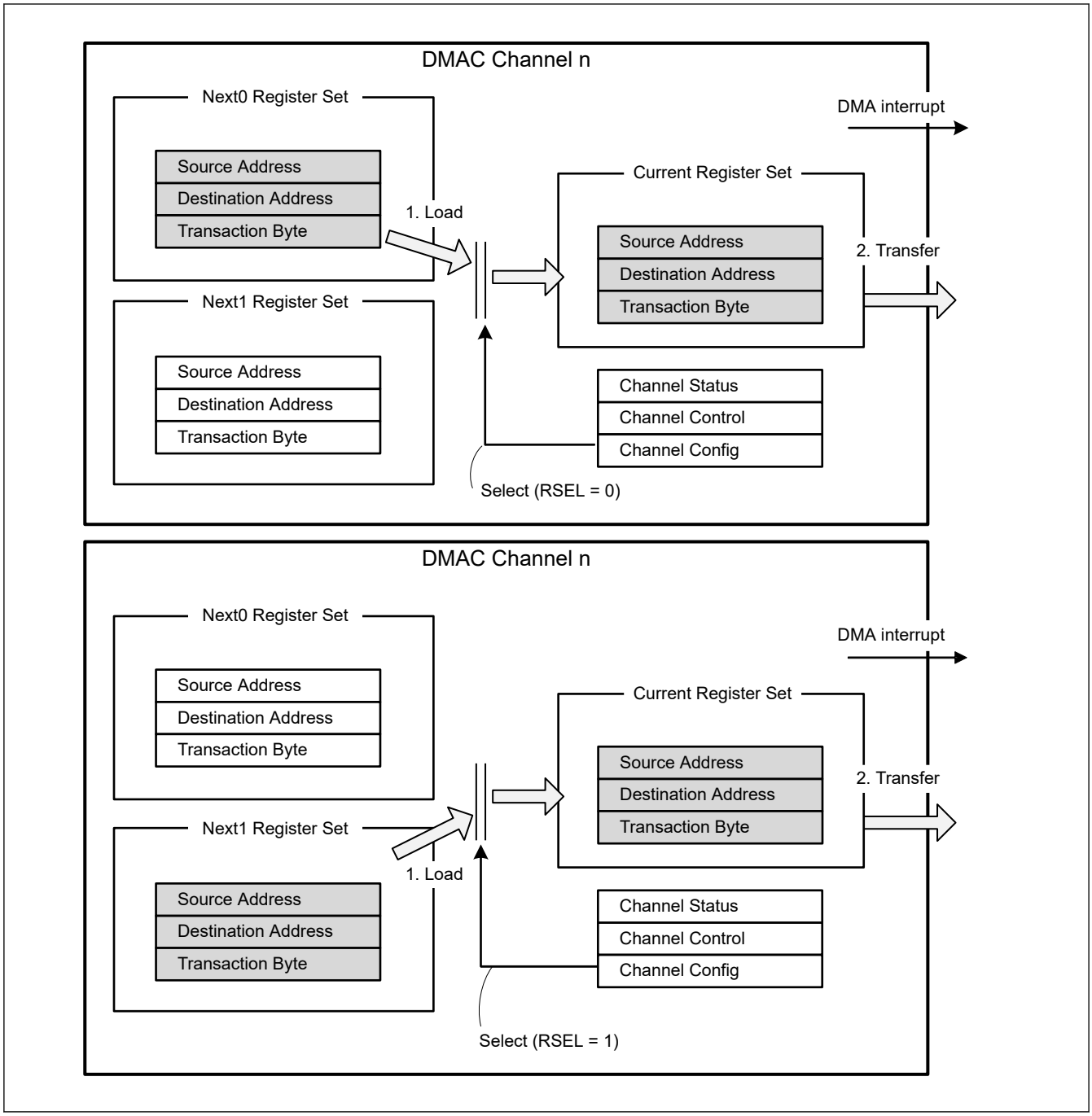


Figure 15.1 Overview of register normal mode

Figure 15.1 shows operations when Next0 Register Set is executed above and when Next1 Register Set is executed below.

(1) Operation Flow in Register Mode

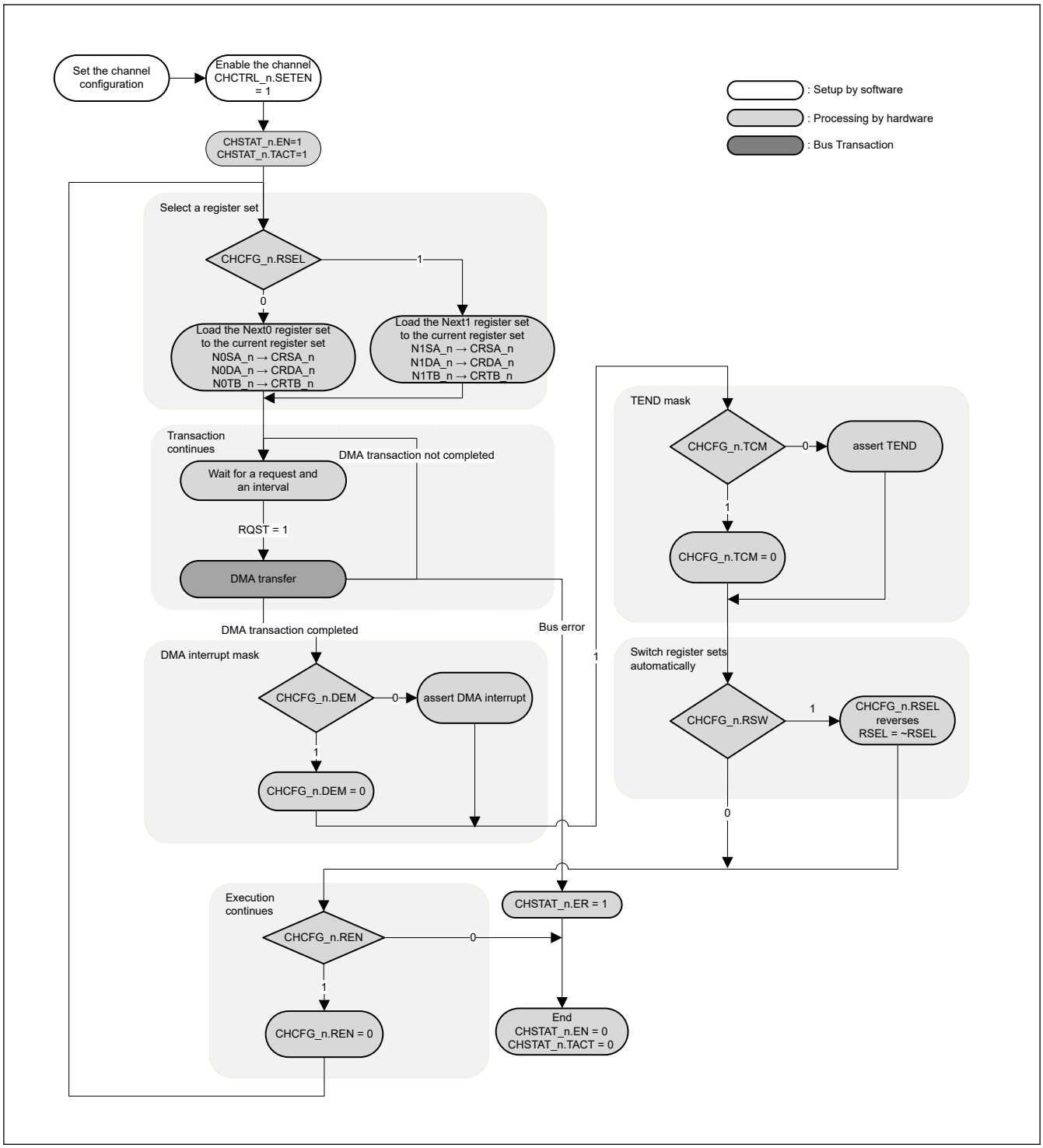


Figure 15.2 Register mode flow

(2) Register Mode Flow Description

1. Channel setting
   Set Next0 or Next1 Register Set (the transfer destination address, the transfer source address, and the total number of transfer bytes).
   In addition, use setting registers CHCTRL_n or CHCFG_n for each channel to set the detection method of the DMA transfer request, the output mode of the DACK/TEND signal, and the amount of data for a transfer.
2. Selecting the register set

When 1 is written to the SETEN bit in the CHCTRL_n register, the EN bit and the TACT bit in the CHSTAT_n register are set to 1, and the setting value of the next register set selected by the RSEL bit in the CHCFG_n register is loaded to the current register set.

3. DMA transfer
   According to the set value, a DMA transfer is performed. For details on the transfer, see from section 15.4.2. Transfer Mode to section 15.4.11. Aborting a Transfer.
4. DMA transfer completion mask
   According to the value set for the DEM bit in the CHCFG_n register, the DMA transfer completion interrupt is masked. When DEM = 1, the DMA transfer completion interrupt is masked. Immediately after the DMA transfer completion interrupt conditions are satisfied, the DEM bit is cleared to 0 automatically.
5. TEND mask
   According to the values set for the TCM bit in the CHCFG_n register, the TEND is masked. When TCM = 1, TEND is masked. Immediately, the DMA transfer completes and the TCM bit is cleared to 0 automatically.
6. Switching register sets automatically
   According to the value set for the RSW bit in the CHCFG_n register, the register set is switched to the other next register set.
7. Successive execution
   According to the value set for the REN bit in the CHCFG_n register, DMA transfers are performed successively. When REN = 0, the EN bit and the TACT bit in the CHSTAT_n register are cleared to 0, and DMAC stops operation. When REN = 1, the DMA transfer continues. Immediately after the conditions for performing DMA transfer again are satisfied with the REN bit, the REN bit is cleared to 0 automatically.

(3) Setting the Register Mode

Select the register set to execute.

Table 15.6 Register mode settings

DMS (CHCFG_n)	RSEL (CHCFG_n)	Description
0	0	Executes Next0 Register Set
	1	Executes Next1 Register Set

DMA transfer completion interrupts can be masked.

Table 15.7 DMA transfer completion interrupt mask settings

DEM (CHCFG_n)	Description
0	When a DMA transfer completes, the DMA transfer completion interrupt is generated
1	Even if a DMA transfer completes, no DMA transfer completion interrupt is generated. When a DMA transfer completes, the DEM bit is cleared to 0 automatically.

TEND output can be masked.

Table 15.8 TEND output mask settings

TCM (CHCFG_n)	Description
0	When a DMA transfer completes, the TEND is asserted
1	Even if a DMA transfer completes, no TEND is asserted. When a DMA transfer completes, the TCM bit is cleared to 0 automatically.

After a DMA transfer, another DMA transfer can be performed successively.

Table 15.9 Settings for automatic execution of register sets (1 of 2)

REN (CHCFG_n)	Operation	Remarks
0	When a DMA transfer of the register set for the RSEL bit completes, the EN bit is cleared to 0, and the DMA operation ends	Set this value if you want to perform a DMA transfer once

Table 15.9 Settings for automatic execution of register sets (2 of 2)

REN (CHCFG_n)	Operation	Remarks
1	After a DMA transfer completes, a DMA transfer is performed for the register set that was selected to be performed successively. When the successive transfer is performed, the REN bit is cleared to 0.	Set this register if you want to perform register sets successively

When a DMA transfer completes, the register set can be switched to the next register set.

Table 15.10 Settings for switching register sets automatically

RSW (CHCFG_n)	Operation	Remarks
0	When a DMA transfer completes, register sets are not switched	Set this value when you want to use one register set only
1	When a DMA transfer completes, the RSEL bit is reversed automatically, and the other register set is selected	Set this value if you want to switch register sets

(4) Register Mode Setting Examples

Using the Next0 register set only.

Table 15.11 Register mode setting example 1

DMS (CHCFG_n)	RSEL (CHCFG_n)	DEM (CHCFG_n)	TCM (CHCFG_n)	RSW (CHCFG_n)	REN (CHCFG_n)
0 (Register mode)	0 (Next0)	0 (Do not mask)	0 (Do not mask)	0 (Do not switch)	0 (Continuous execution not in progress)

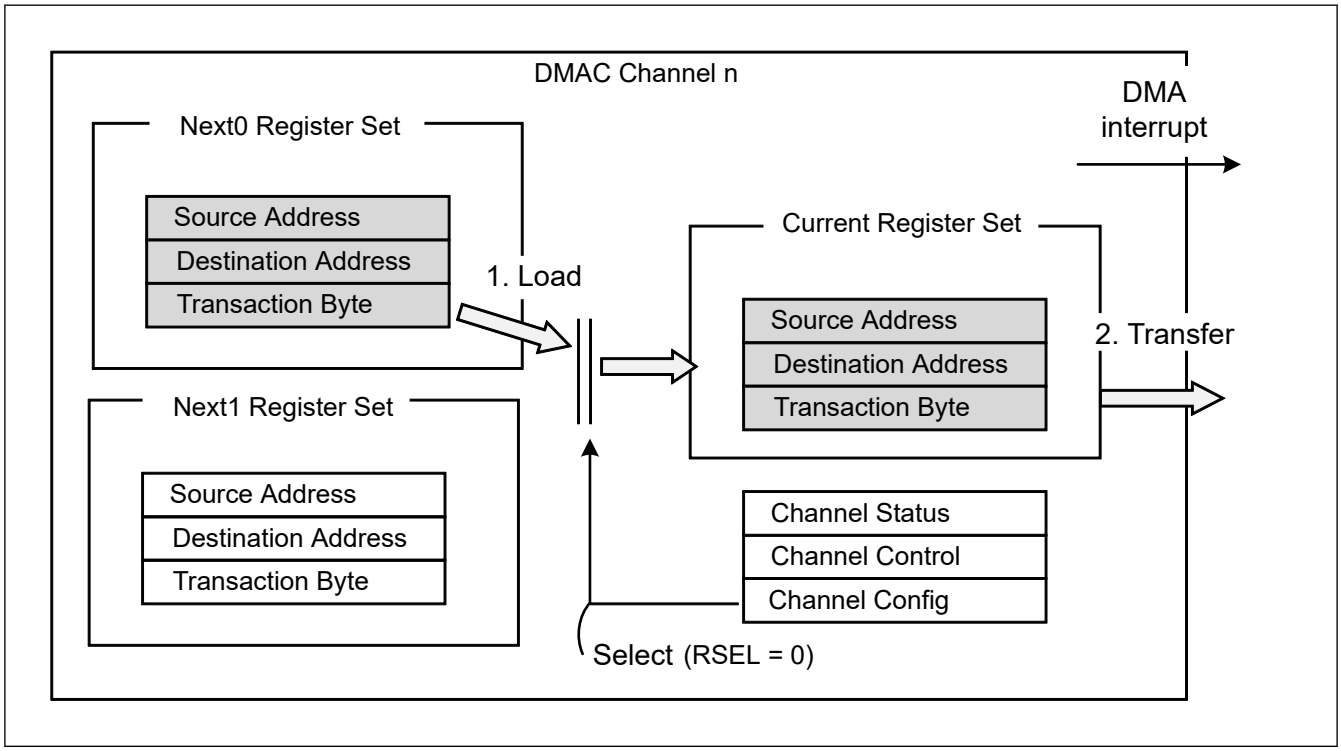


Figure 15.3 Register mode setting example 1

1. Writing 1 to the SETEN bit in the CHCTRL_n register sets 1 to the EN bit in the CHSTAT_n register, and loads Next0 Register Set to the current register set.
2. According to the values set for the current register set and the channel register set, a DMA transfer is performed.
3. Because the DEM bit in the CHCFG_n register is 0, a DMA transfer completion interrupt is generated when a DMA transfer completes.

4. Because the TCM bit in the CHCFG_n register is 0, the TEND is asserted when DMA transfer completes.
5. Because the REN bit in the CHCFG_n register is 0, the EN bit in the CHSTAT_n register is cleared to 0, and the operation ends.

Using two register sets successively.

Table 15.12 Register mode setting example 2

DMS (CHCFG_n)	RSEL (CHCFG_n)	DEM (CHCFG_n)	TCM (CHCFG_n)	RSW (CHCFG_n)	REN (CHCFG_n)
0 (Register mode)	0 (Next0)	1 (Mask)	0 (Do not mask)	1 (Switch)	1 (Continuous execution not in progress)

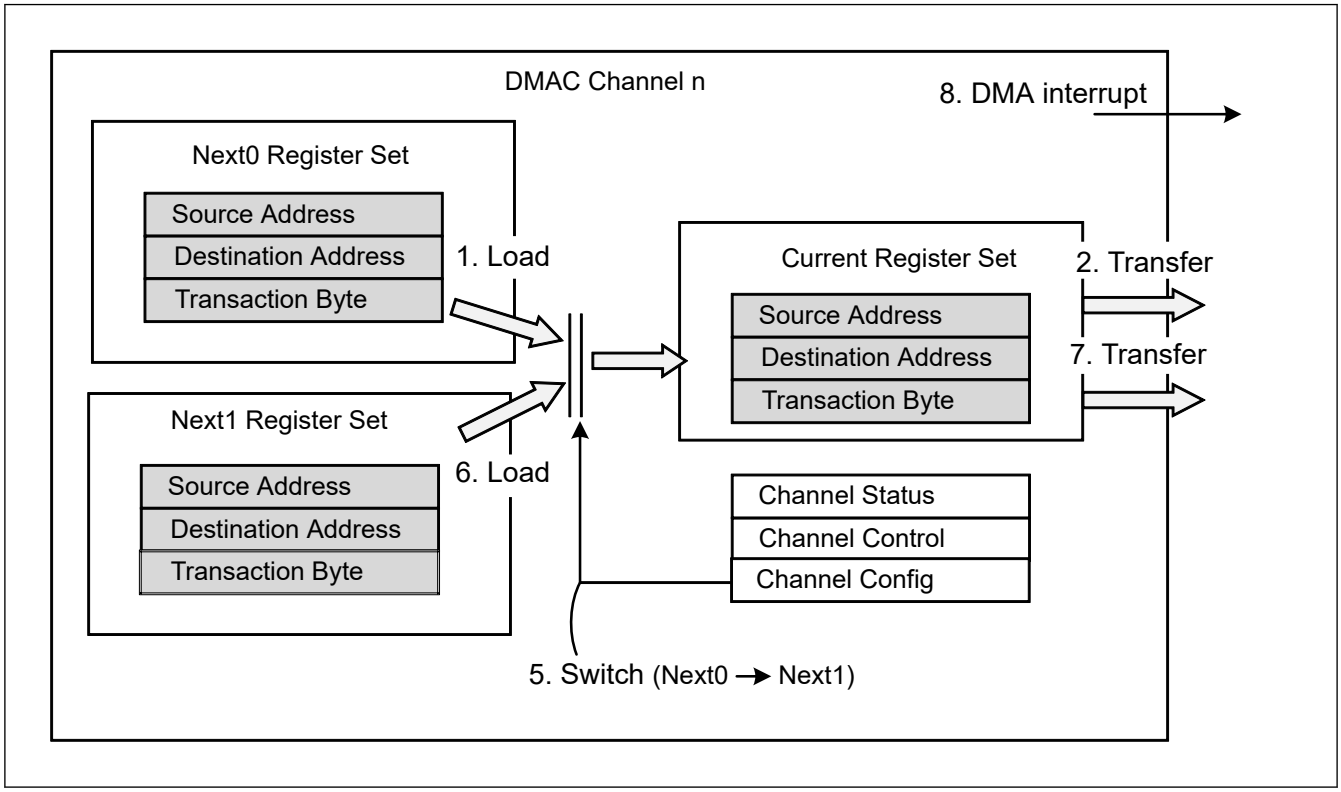


Figure 15.4 Register mode setting example 2

1. Writing 1 to the SETEN bit in the CHCTRL_n register sets 1 to the EN bit in the CHSTAT_n register, and loads Next0 Register Set to the current register set.
2. According to the values set for the current register set and the channel register set, a DMA transfer is performed.
3. Because the DEM bit in the CHCFG_n register is 1, when the DMA transfer completes, no DMA transfer completion interrupt is generated. In addition, the DEM bit is cleared to 0 automatically.
4. Because the REN bit in the CHCFG_n register is 1, DMA transfers are performed successively. In addition, the REN bit is cleared to 0.
5. Because the RSW bit in the CHCFG_n register is 1, the register set to be executed next is switched (RSEL = 0 → 1).
6. Loads Next1 Register Set to Current Register Set.
7. According to the values for Current Register Set and Channel Register Set, DMA transfers are performed.
8. Because the DEM bit in the CHCFG_n register is 0, a DMA transfer completion interrupt is generated when a DMA transfer completes.
9. Because the TCM bit in the CHCFG_n register is 0, the TEND is asserted when DMA transfer completes.
10. Because the REN bit in the CHCFG_n register is 0, the EN bit in the CHSTAT_n register is cleared to 0 automatically.

15.4.1.2 Link Mode

In link mode, a DMA transfer is performed by reading a descriptor in the memory area outside the DMAC as the setting value. Within DMAC, each channel has the next link address (NXLA_n) register and the current link address (CRLA_n) register. Each of them is used to set the address of the descriptor to be executed next and to display the descriptor address of the current DMA transfer respectively.

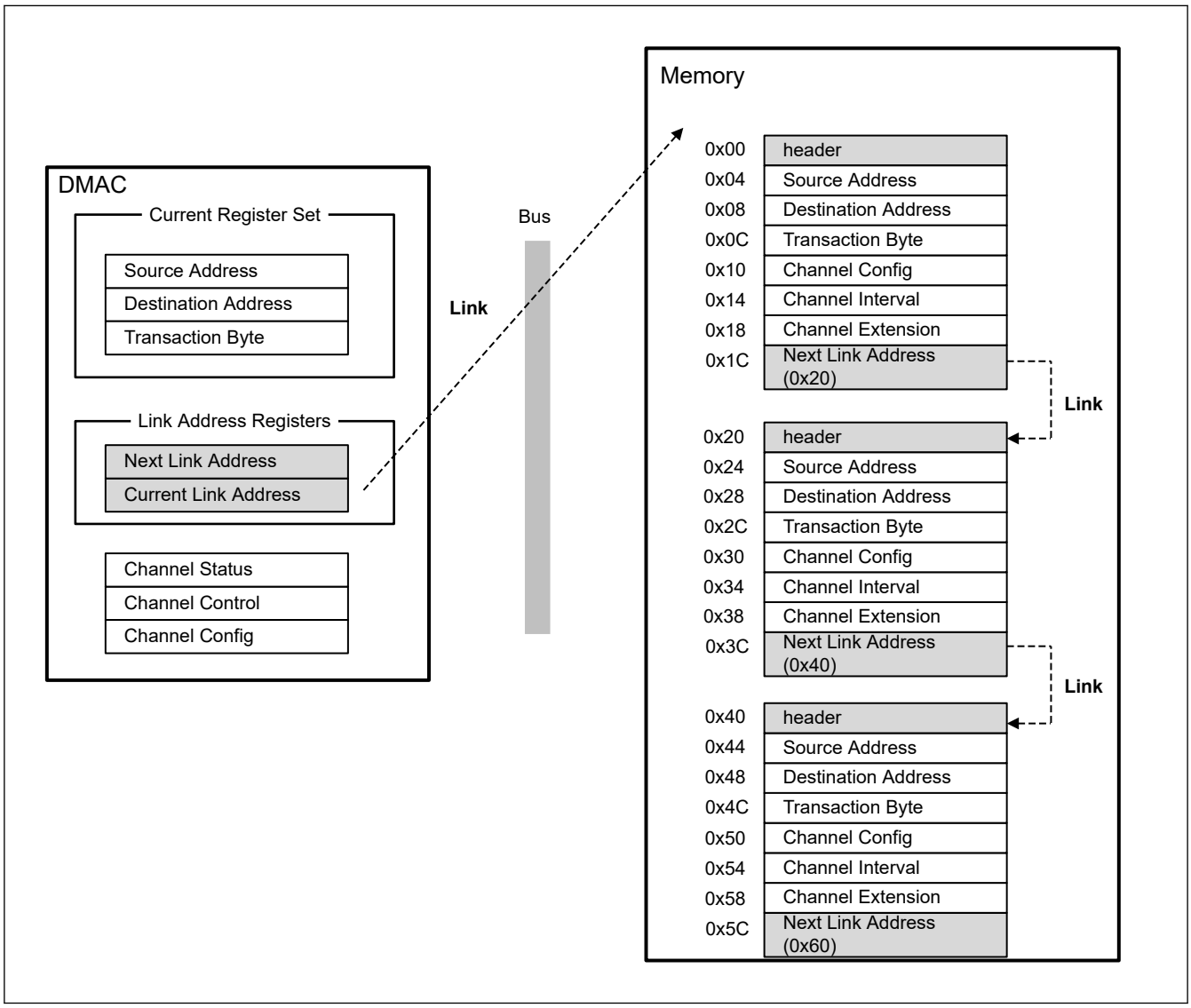


Figure 15.5 Link mode overview

(1) Operation Flows in Link Mode

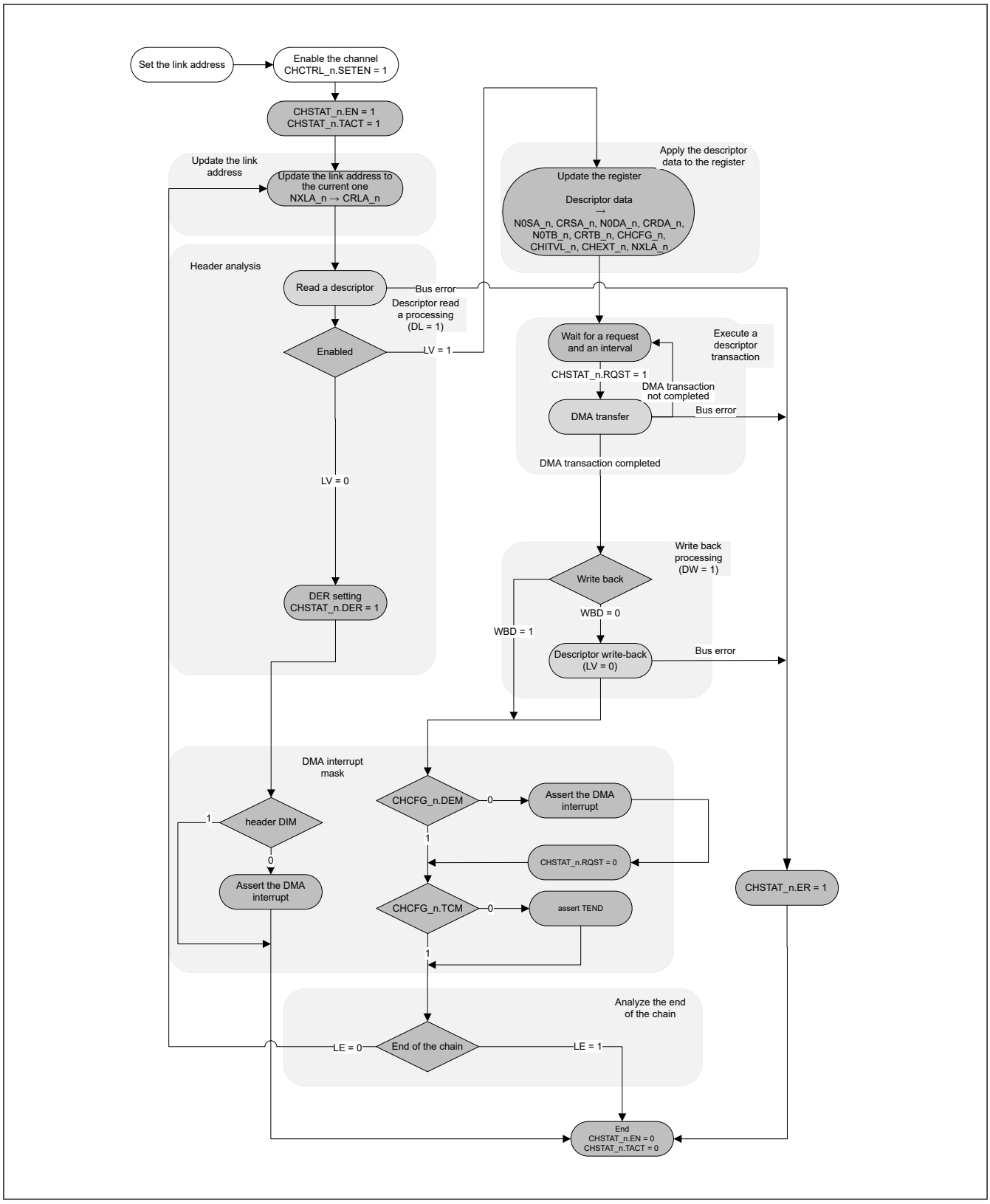


Figure 15.6 Link mode operation flow

(2) Description of Link Mode Operation Flow

1. Channel setting

Set the start address of the link destination in the NXLA_n register.

2. Updating the link address
   Writing 1 to the SETEN bit in the CHCTRL_n register sets 1 to the EN bit and the TACT bit in the CHSTAT_n register, and loads the link address set for the NXLA_n register to the CRLA_n register.
3. Descriptor read and header judgment
   DMAC starts reading a descriptor and checks the header contents. When LV = 0, the descriptor is discarded and the DER bit in the CHSTAT_n register = 1 indicates the end state (the EN bit in the CHSTAT_n register = 0). At that time, if the DIM bit in the header register is 0, a DMA transfer completion interrupt is generated.
4. Descriptor settings
   When LV = 1, the read descriptor data is loaded to the current register set and the channel register set. In addition, the next link destination is loaded to the NXLA_n register.
5. DMA transfer
   According to the set value, a DMA transfer is performed. For details on the transfer, see from section 15.4.2. Transfer Mode to section 15.4.11. Aborting a Transfer.
6. Writing back of header
   When WBD of header = 0, DMAC writes LV = 0 back to the header area.
7. DMA interrupt mask
   When the DEM bit in the CHCFG_n register is 0, a DMA transfer completion interrupt is generated. When DMA transfer is performed using multi descriptors, DMA request should be issued again and RQST is set to 1 to continue the DMA transfer because RQST is set to 0 when DEM = 0.
8. TEND mask
   When the TCM bit in the CHCGF_n register is 0, the TEND is generated.
9. Link end judgment
   When LE of header = 1, the EN bit and the TACT bit in the CHSTAT_n register are cleared to 0, and DMAC stops operation. When LE = 0, the current register set is updated, and the next descriptor read is started.

(3) Register Settings

When using link mode, set the DMS bit in the CHCFG_n register to 1.

Table 15.13 Link mode settings

DMS (CHCFG_n)	Description
1	Operates in link mode. This bit cannot be changed by using a descriptor.

As registers that indicate the link destination, the next link address (NXLA_n) register and the current link address (CRLA_n) register are available.

To start link mode, set the link destination in the NXLA_n register.

The NXLA_n register is updated to the next link after a descriptor is read. In addition, the CRLA_n register indicates the link address of the currently executed descriptor.

Table 15.14 Link address register set

Register	Description
NXLA_n	Sets and displays the next link destination. Before starting link mode, set the address of the link destination for this register.
CRLA_n	Displays the currently executed link destination. This register is read-only.

(4) Descriptor Settings

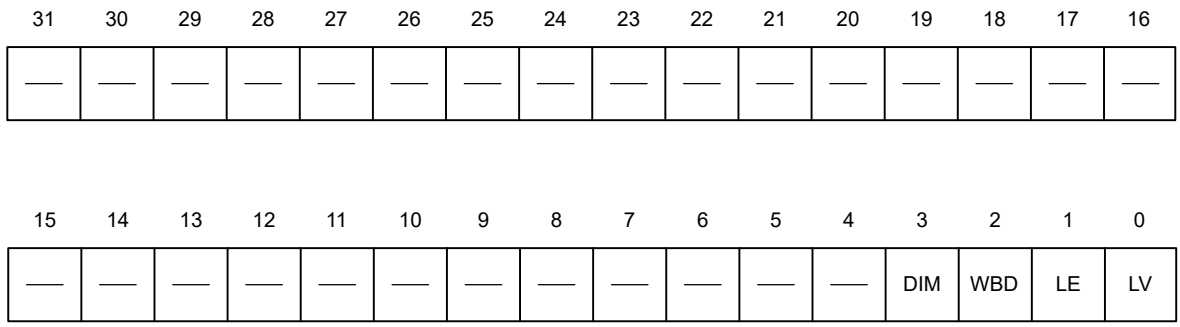
A descriptor should be deployed in the link address as shown in Table 15.15. DMAC interprets data obtained through descriptor read in order.

Table 15.15 Descriptor placement

Address (Link Address + N)*1							
+0x1C	+0x18	+0x14	+0x10	+0x0C	+0x08	+0x04	+0x00
Next link address	Extension	Interval	Config*2	Transaction byte	Destination address	Source address	Header

Note 1. Link address should be aligned to 32-bit boundary.
Note 2. Register mode setting is prohibited.

As shown in Figure 15.7, header indicates descriptor states. This area is read by DMAC before starting a DMA transfer in link mode. In addition, after the DMA transfer, the transfer status is written back by DMAC.



When reading header, DMA performs a write back for this 4 bytes area after DMA transfer and when the descriptor information is completed. Write back data is a header read data except LV, and LV = 0.

Figure 15.7 Header area

Table 15.16 Header area

Bit position	Bit name	Description
0	LV	Link Valid Indicates that this descriptor is enabled. When WBD = 0, DMAC writes 0 after the DMA transfer written in the descriptor. Set 1 when setting header. 0: The descriptor is disabled 1: The descriptor is enabled
1	LE	Link End Indicates that the link ends during DMA transfer of this descriptor. To indicate the end of the link, set this bit to 1. 0: The link continues 1: The link ends
2	WBD	Write Back Disable Masks write back execution of the LV bit. When this bit is 1, DMAC does not perform write-back operation. 0: The LV bit is written back to 0 1: The LV bit is not written back
3	DIM	Descriptor Interrupt Mask When LV = 0 at loading of header, specifies whether DMA transfer completion interrupt mask or not. 0: DMA transfer completion interrupt is asserted 1: DMA transfer completion interrupt is masked
31:4	—	A reserved area. Set 0.

Data in descriptors except header has the same specifications as an on-chip register. However, the DMS bit in the CHCFG_n register cannot be changed by a descriptor. For details on the on-chip register specifications, see section 15.3. Register Descriptions.

For descriptor setting examples, see section 15.6.4. Setting Example 4 (Link Mode).

Figure 15.8 provides an overview of the descriptor area and the DMA transfer area to which DMAC accesses.

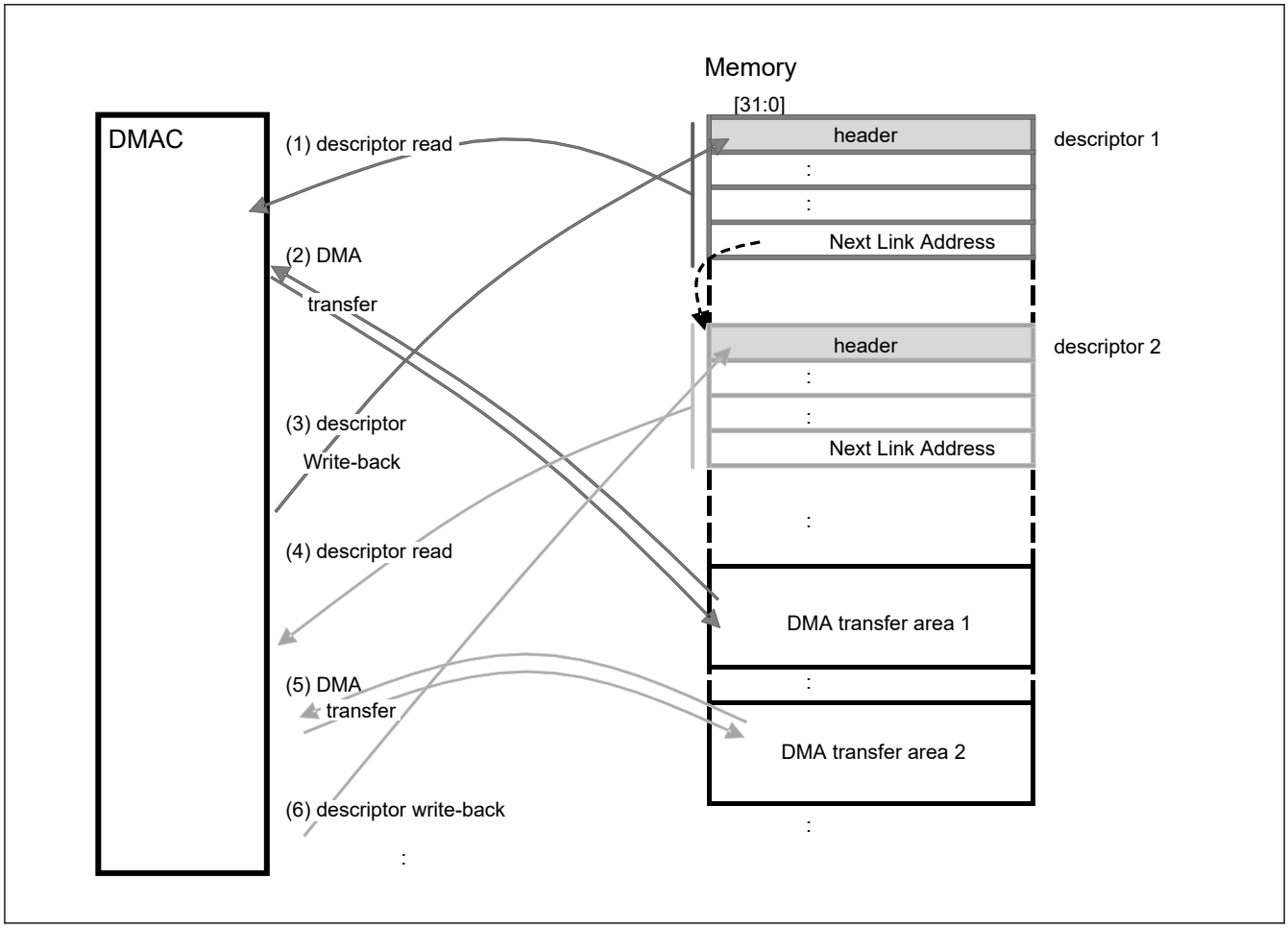


Figure 15.8 Header area

1. Descriptor read
   Loads the value set for the on-chip NXLA_n register to the CRLA_n register, and then reads the descriptor from the memory space (descriptor 1) indicated by the CRLA_n register.
2. DMA transfer
   When the LV bit in header is 1, performs a DMA transfer according to the descriptor information.
3. Descriptor write-back
   After the DMA transfer for the set number of bytes, if the WBD bit in header is 0, the LV bit writes 0 and other fields writes the value read in step 1 as data back to header [31:1] of descriptor 1 in bytes.
4. Descriptor read
   If the LE bit in header of the descriptor which was read previously (step 1) is 0, reads the next descriptor from the address (descriptor 2) indicated by the next link address in the descriptor.
5. DMA transfer
   When the LV bit in header is 1, performs a DMA transfer according to the descriptor information.
6. Descriptor write-back
   After the DMA transfer for the set number of bytes, if the WBD bit in header is 0, the LV bit writes 0 and other fields writes the value read in step 4 as data back to header [31:1] of descriptor 2 in bytes.

Hereafter, repeats steps (4) through (6).

When LE of header is 1 and WBD = 0, DMA transfer proceeds in accord with the descriptor settings, and 0 is written back to the LV bit of the header, after which operation ends.

When LE of header is 1 and WBD = 1, performs a DMA transfer with the setting, and ends the operation (no write-back is performed).

When LV of header is 0 the operation stops (no DMA transfer is performed).

(5) Link Configuration Examples

In link mode, descriptors can be configured as shown in Figure 15.9.

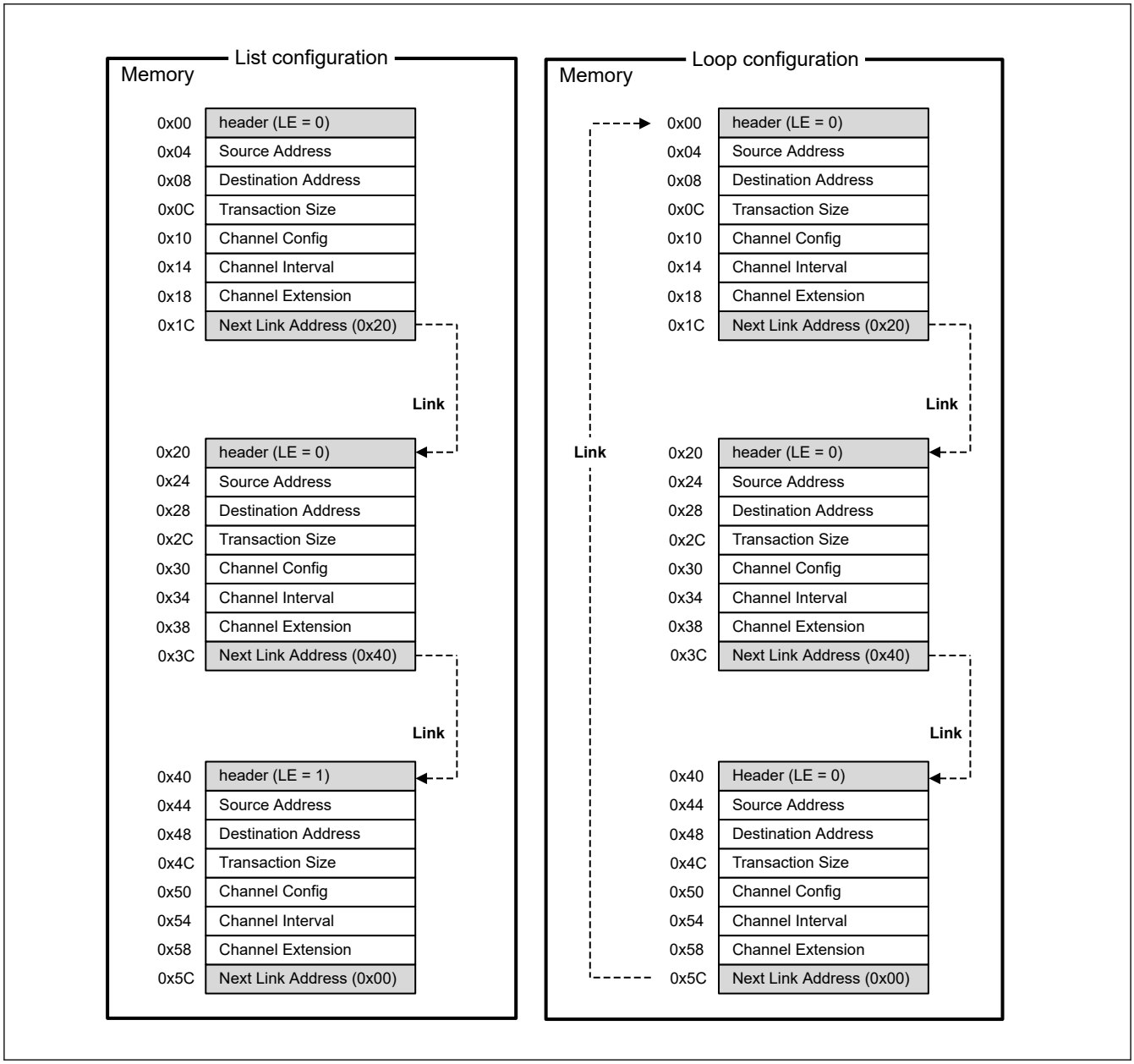


Figure 15.9 Header area

Setting 1 to the LE bit in header of the last descriptor ends the link.

Setting the link destination of the last descriptor to the address of the previous descriptor configures the descriptors in a loop. To end the loop, change the LE bit of header to 1 before DMAC reads a descriptor, or follow the procedure to pause a transfer.

15.4.2 Transfer Mode

Single transfer mode and block transfer mode are supported. When selecting mode, use the TM bit in the CHCFG_n register for each channel.

Table 15.17 Basic transfer settings

Transfer mode	TM (CHCFG_n)	Description
Single transfer	0	For one DMA request, performs a DMA transfer
Block transfer	1	For one DMA request, performs transfers until a DMA transfer completes

15.4.2.1 Single Transfer Mode

When a DMA transfer request is accepted, a DMA transfer is performed on the side (transfer source or transfer destination) indicated by the REQD bit in the CHCFG_n register to activate DACK. Each time a transfer request is accepted, transfer proceeds. This operation is repeated until it reaches the transfer size loaded from the N0TB_n or N1TB_n register to the CRTB_n register. Arbitration between channels is performed for each DMA transfer.

The DACK output timing differs depending on the setting for the REQD bit in the CHCFG_n register or settings for the transfer size (DDS[2:0] and SDS[2:0] in the CHCFG_n register) bit settings. For details, see section 15.4.8. Differences in Operation According to the Transfer Data Size.

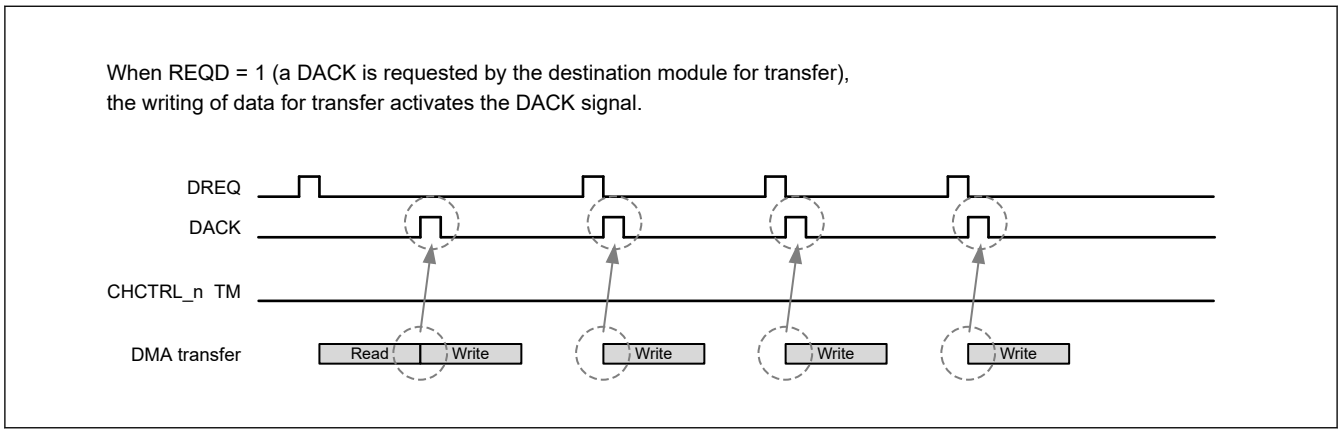


Figure 15.10 Single transfer mode (REQD = 1, SDS > DDS)

15.4.2.2 Block Transfer Mode

Once a DMA transfer request is accepted, the transfer continues until transfer of the number of bytes loaded from the N0TB_n or N1TB_n register to the DMA transfer byte register (CRTB_n register) is completed (After DMA transfer completion, arbitration between channels is performed for each DMA transfer).

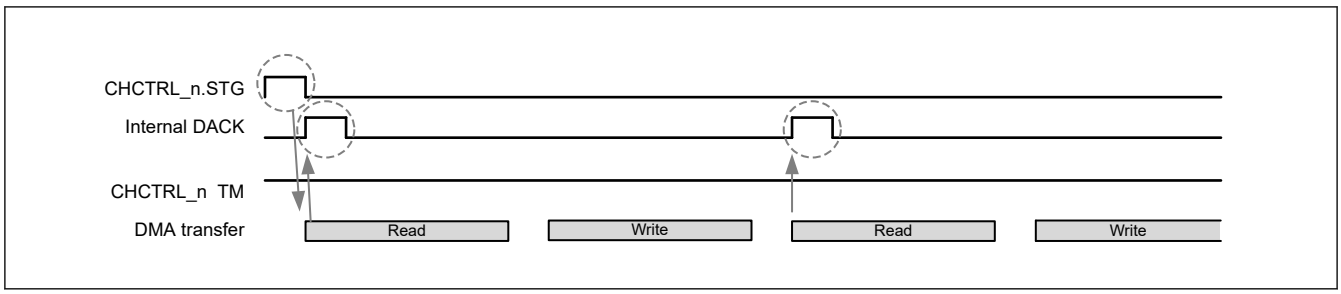


Figure 15.11 Block transfer mode (REQD = 0, SDS < DDS)

15.4.3 DMA Channel Priority Control

As an arbitration method between channels, fixed priority mode and round-robin mode are supported. To select the mode, use the PR bit in the DCTRL register. When the PR bit is 0, fixed priority mode is selected. When the PR bit is set to 1, round-robin mode is selected.

Table 15.18 Priority control settings

Transfer mode	PR (DCTRL)	Description	Applications
Fixed Priority	0	Channels 0 to 7, channels 8 to 15 are fixed priority mode. CH0 (CH8) > CH1 (CH9) > CH2 (CH10) > CH3 (CH11) > CH4 (CH12) > CH5 (CH13) > CH6 (CH14) > CH7 (CH15)	Select this mode if channels have priority
Round-Robin	1	Controls requests in round-robin mode	Select this mode if you want to execute requests equally

15.4.3.1 Fixed Priority Mode

In fixed priority mode, the priority levels are fixed within the group of channels 0 to 7 and within the group of channels 8 to 15. In addition, the priority levels between the group of channels 0 to 7 and the group of channels 8 to 15 are in round-robin mode.

The priority levels immediately after a reset and transfer through DMA channel 0 are shown in Figure 15.12.

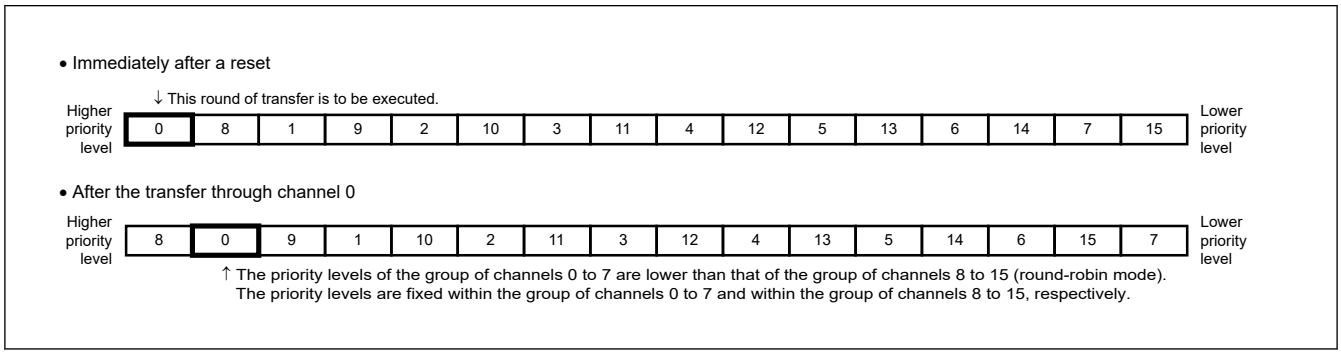
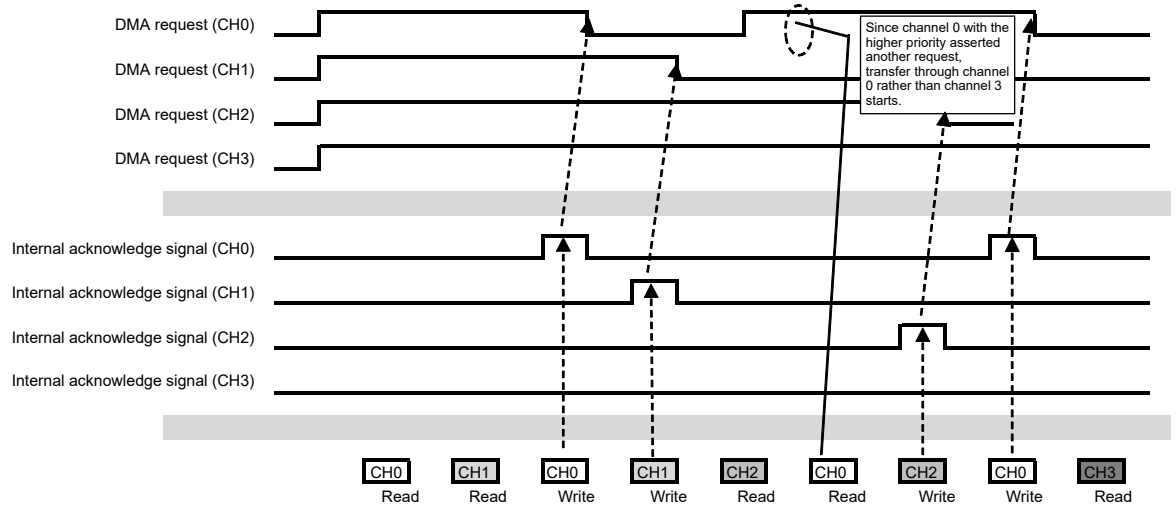


Figure 15.12 Priority levels immediately after a reset and transfer through DMA channel 0

If DMA transfer requests are generated on multiple channels simultaneously, a DMA transfer request with a smaller channel number takes precedence.

Figure 15.13 provides an example of when a DMA transfer is performed in fixed priority mode if another DMA transfer with higher priority is requested.

Example: DMA requests are issued through four channels at the same time.



Note: Channel 0 is handled with the highest priority, but the transfer with the next highest priority is performed because bus arbitration is performed after data has been read from the transfer source of channel 0. Read operation of another channel may interrupt the operation between reading data from and writing data to the same channel.

Figure 15.13 Fixed priority mode (number of channels = 4 and REQD = 1)

15.4.3.2 Round-Robin Mode

In round-robin mode, priority is changed every time a transfer of a channel is accepted so that the lowest priority is given to the channel in which the last transfer is performed.

The priority levels immediately after a reset and transfer through DMA channel 2 are shown in Figure 15.14.

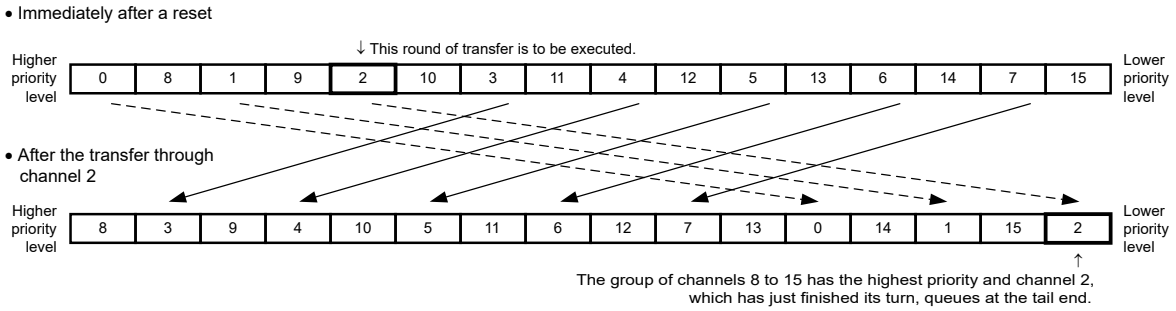
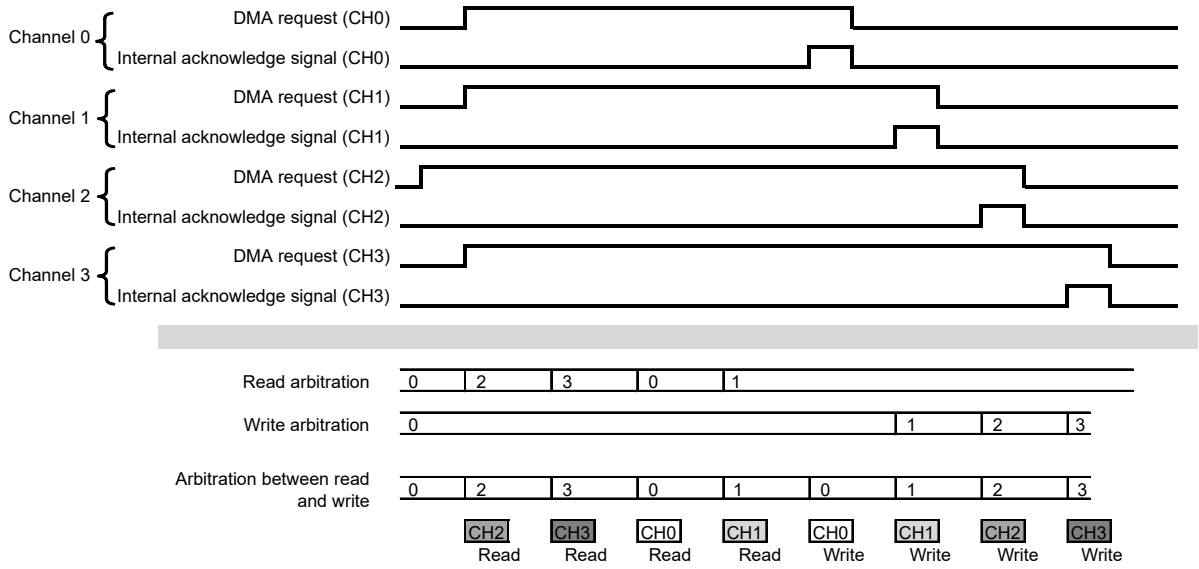


Figure 15.14 Priority levels immediately after a reset and transfer through DMA channel 2

Figure 15.15 provides an example of a DMA transfer in round-robin mode.



Note: Within DMAC, arbitration between read channels and arbitration between write channels are performed. As a result of those arbitrations, further arbitration is performed, and a bus access is issued.

Figure 15.15 Round-robin mode (number of channels = 4 and REQD = 1)

15.4.4 DMA Transfer Request

DMA activation requests have four types including software requests, external requests, on-chip peripheral module requests, and external interrupts.

Select the transfer request source for an external request, on-chip peripheral module request, external interrupt, and software request with the DMACm_RSSELn register (m = 0, 1, or 2, n = 0 to 5).

For details on the DMACm_RSSELn register (m = 0, 1, or 2, n = 0 to 5), see section 15.3.9. DMAC0_RSSELi : DMAC Unit 0 Resource Select Register i (i = 0 to 5) and section 15.3.10. DMAC1_RSSELi : DMAC Unit 1 Resource Select Register i (i = 0 to 5).

15.4.4.1 Specifying Detection Operation of DMA Transfer Requests for Each Source

Detection methods for DMA transfer requests of external requests, on-chip peripheral module requests, external interrupts, and software requests might be specified according to sources.

For each DMA transfer source, set the LVL, LEN, and HEN bits in the CHCFG_n register according to Table 15.19, Table 15.20, and Table 15.21.

For details on edge detection operation, see Figure 15.16 and Figure 15.17. For information about level detection operation, see Figure 15.18 and Figure Figure 15.19.

Table 15.19 Detection operation specification for each source of DMA transfer requests

DMA transfer request source	Detection operation specification of DMA transfer requests	DMA acknowledge signal specification
External request	Detects the rising edge. Detects the falling edge. Detects the high level. Detects the low level.	Depends on specifications of the DMA transfer request source
On-chip peripheral module request	Depends on specifications of the DMA transfer request source. See Table 15.21.	Bus cycle mode
External interrupt	Detects the rising edge. Detects the high level.	Bus cycle mode
Software request	Detects the rising edge	Bus cycle mode

Table 15.20 Method of detecting DMA transfer request signals

Mode	LVL (CHCFG_n)	HIEN (CHCFG_n)	LOEN (CHCFG_n)	Description
Edge detection	0	0	0	Disables detection
			1	Detects the falling edge
		1	0	Detects the rising edge
			1	Detects the falling and rising edges
Level detection	1	0	0	Disables detection
			1	Detects the low level
		1	0	Detects the high level
			1	DMA transfer starts regardless of the request signal level when the SETEN bit in the CHCTRL_n is 1. In this setting, DMA acknowledge must be set to other than level mode. If level mode is set, dead lock will occur.

Table 15.21 DMA transfer request detection operation setting table (1 of 2)

DMA transfer request source	DMA transfer source	Transfer source	Transfer destination	REQ_SELn[8:0]	CHCFG_n						
					TM	AM[2:0]*2	LVL	HIEN	LOEN	REQD*2	SEL[2:0]
External DMA Request	DREQ	Arbitrary	Arbitrary	*4	0/1	001/ 010/ 100	0/1	10/01		0/1	*5
Software Interrupt 0 to 13	INTCPU	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
External Interrupt 0 to 13	IRQ	Arbitrary	Arbitrary	*4	0/1	010	0/1*1	1*1	0*1	0/1	*5
CMT Unit 0 to 2	Compare match	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
CMTW Unit 0, 1	Compare match	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
	Input capture			*4							
	Output compare			*4							
MTU3	Input capture/compare match*3	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
	Overflow/underflow			*4							
GPT unit 0 to 9	Input capture/compare match	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
	Compare match			*4							
	Overflow			*4							
	Underflow			*4							
Ethernet switch	Ethernet switch interrupt	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
	DLR interrupt			*4							
	PRP interrupt			*4							
	Integrated Hub interrupt			*4							
	RX Pattern Matcher interrupt			*4							
	Timer pulse output	Arbitrary	Arbitrary	*4	0/1	010	0	0/1	0/1	0/1	*5
	TDMA timer output			*4							
EtherCAT slave	Sync interrupt	Arbitrary	Arbitrary	*4	0/1	010	0	0/1	0/1	0/1	*5
	EtherCAT interrupt	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
	SOF interrupt			*4							
	EOF interrupt			*4							

Table 15.21 DMA transfer request detection operation setting table (2 of 2)

DMA transfer request source	DMA transfer source	Transfer source	Transfer destination	REQ_SELn[8:0]	CHCFG_n						
					TM	AM[2:0]*2	LVL	HIEN	LOEN	REQD*2	SEL[2:0]
USB	Host interrupt	Arbitrary	Arbitrary	*4	0/1	010	0/1	1	0	0/1	*5
	Function interrupt			*4							
SCI Unit 0 to 4	Receive data full	RDR	Arbitrary	*4	0	010	0	1	0	0	*5
	Transmit data empty	Arbitrary	TDR	*4	0	010	0	1	0	1	*5
SCIE Unit 0 to 11	Receive data full	RDR	Arbitrary	*4	0	010	0	1	0	0	*5
	Transmit data empty	Arbitrary	TDR	*4	0	010	0	1	0	1	*5
IIC Unit 0, 1	Receive data full	ICDRR	Arbitrary	*4	0	010	0	1	0	0	*5
	Transmit data empty	Arbitrary	ICDRT	*4	0	010	0	1	0	1	*5
CANFD	RX FIFO 0 DMA request	CFDRFDF00	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
	RX FIFO 1 DMA request	CFDRFDF01	Arbitrary	*4							
	RX FIFO 2 DMA request	CFDRFDF02	Arbitrary	*4							
	RX FIFO 3 DMA request	CFDRFDF03	Arbitrary	*4							
	RX FIFO 4 DMA request	CFDRFDF04	Arbitrary	*4							
	RX FIFO 5 DMA request	CFDRFDF05	Arbitrary	*4							
	RX FIFO 6 DMA request	CFDRFDF06	Arbitrary	*4							
	RX FIFO 7 DMA request	CFDRFDF07	Arbitrary	*4							
	CH0 first common FIFO DMA request	CFDCFDF00	Arbitrary	*4							
	CH1 first common FIFO DMA request	CFDCFDF03	Arbitrary	*4							
SPI Unit 0 to 2	Receive data full	SPDR	Arbitrary	*4	0	010	0	1	0	0	*5
	Transmit data empty	Arbitrary	SPDR	*4	0	010	0	1	0	1	*5
DSMIF Unit 0 to 9	Current data register update common	DSCDRCHn	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
	Current capture data register A update common	DSCCDRACHn	Arbitrary	*4							
	Current capture data register B update common	DSCCDRBCHn	Arbitrary	*4							
ADC12 Unit 0 to 2	Scan end interrupt	ADDRm	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
	Compare match			*4							
	Compare mismatch			*4							
Mailbox	Mailbox (Host CPU to Cortex-A55/ Cortex-R52) interrupt	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
ENDAT Unit 0 to 15	Interrupt output 1	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
BISS Unit 0 to 15	End of transmission interrupt	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
HDSL Unit 0 to 15	Interrupt output	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5
	Interrupt output (safety related)			*4							
	Fast position value ready interrupt			*4							
	Safe position received interrupt			*4							
AFMT Unit 0 to 15	Communication completion	Arbitrary	Arbitrary	*4	0/1	010	0	1	0	0/1	*5

Note 1. Set the LVL, HIEN, and LOEN bits of the external interrupts (IRQ0 to IRQ13) as follows:

For the setting of the NS_PORTNF_MD register, see section 12.3.2.4. NS_PORTNF_MD : Interrupt Edge Detection Setting Register.

LVL: Set according to the edge/level setting of the NS_PORTNF_MD register.

HIEN: Set to 1 regardless of the detection level of the NS_PORTNF_MD register.

LOEN: Set to 0 regardless of the detection level of the NS_PORTNF_MD register.

Note 2. If the DACK/TEND signal is not used, specify any setting.

Note 3. When an interrupt occurred at the same time as DMA transfer request, DMA transfer completion interrupt does not occur. DMA transfer completion should be confirmed by END bit in CHSTAT_n register.

Note 4. Available trigger source number (event number) is defined in event table. For details, see section xx.xx.xx Event Table.

Note 5. Set the value corresponding to channel as follow:

ch 0 or 8: 0x0

ch 1 or 9: 0x1

ch 2 or 10: 0x2

ch 3 or 11: 0x3

ch 4 or 12: 0x4

ch 5 or 13: 0x5

ch 6 or 14: 0x6

ch 7 or 15: 0x7

Remarks: CHCFG_n register setting values

TM bit	0:	Single transfer
	1:	Block transfer
AM[2:0] bits	001b:	DACK level output
	010b:	DACK bus cycle output
	100b:	Masks DACK output
LVL bit	0:	Detects the edge of a DMA request
	1:	Detects the level of a DMA request
REQD bit	0:	DACK output during read operation
	1:	DACK output during write operation

Note: Instead of an interrupt source from peripheral modules, transfer completion sources of DMAC channels selected by the DMAC source select register are connected to the vector number selected by the DMAC Source Select Register.

15.4.4.2 Edge Detection

Setting the LVL bit in the CHCFG_n register to 0 detects the edge.

Setting the HIEN bit in the CHCFG_n register to 1 detects the rising edge, and setting the LOEN bit in the CHCFG_n register to 1 detects the falling edge.

For DMA requests by the DREQ signal, request the next DMA transfer after the DACK signal is activated.

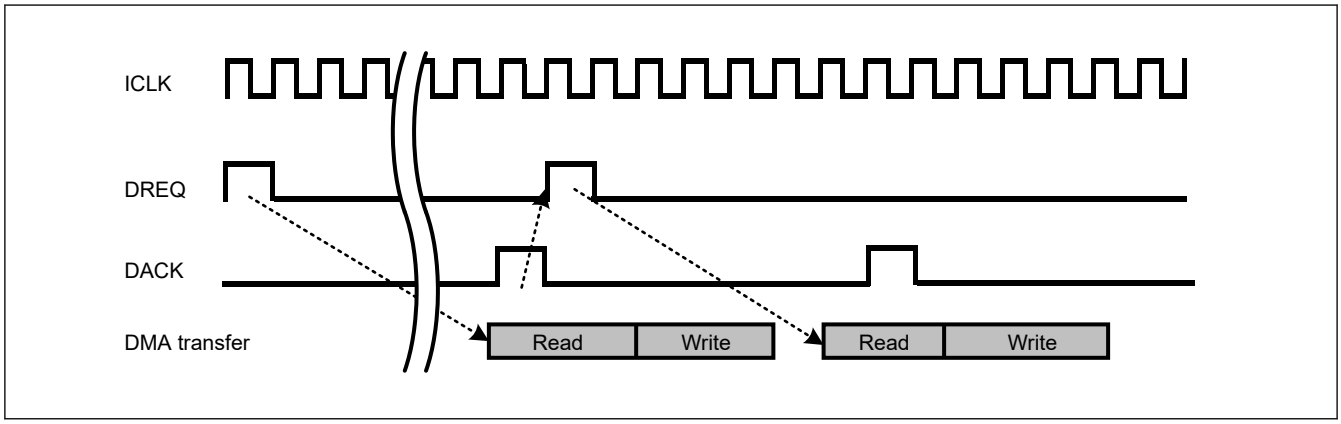


Figure 15.16 Edge detection timing (rising edge (HIEN = 1), DACK output is active during reading (REQD = 0))

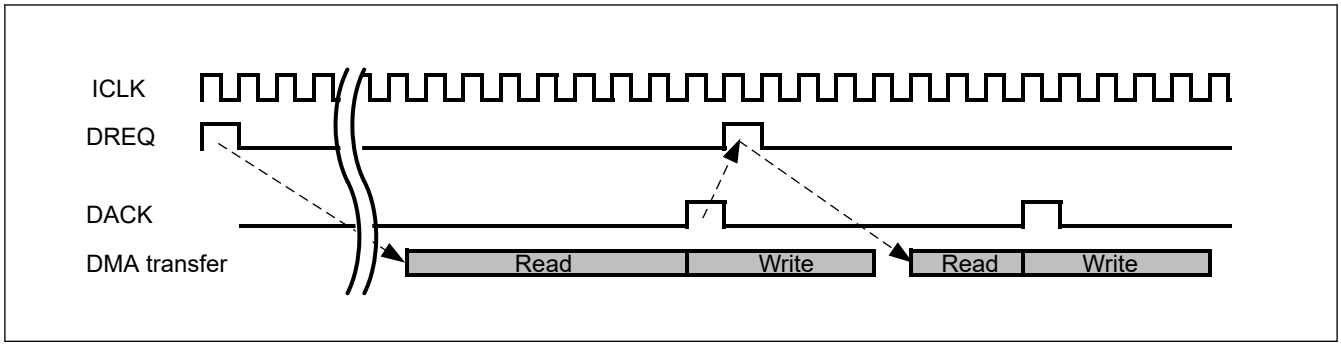


Figure 15.17 Edge detection timing (rising edge (HIEN = 1), DACK output is active during writing (REQD = 1))

15.4.4.3 Level Detection

Setting the LVL bit in the CHCFG_n register to 1 detects the level.

If a DMA transfer request is active (according to HIEN or LOEN settings) for two consecutive clocks (CLK) or more, it is recognized as a DMA request.

If the DACK signal output is set to level mode, the DACK signal output remains at the active level until the DMA transfer request is deactivated.

If you want to request the next DMA transfer, request it after the DACK signal is deactivated.

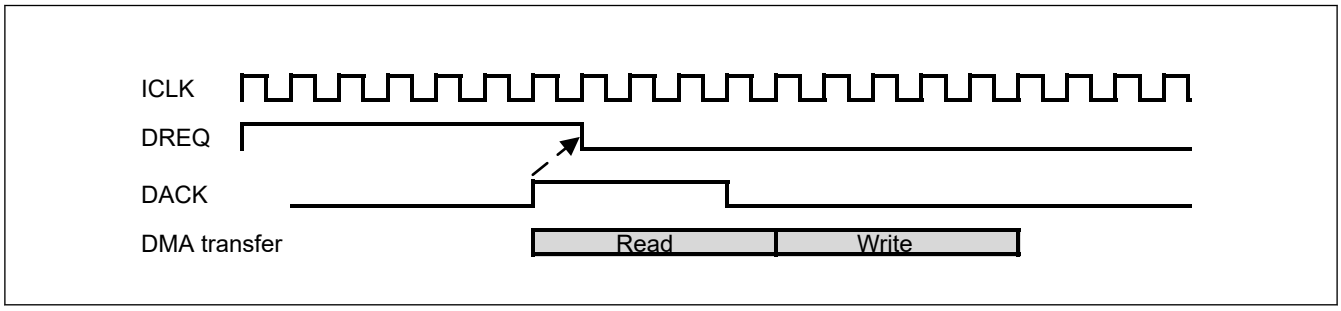


Figure 15.18 Level detection timing (high-level detection (HIEN = 1), DACK output is active during reading (REQD = 0))

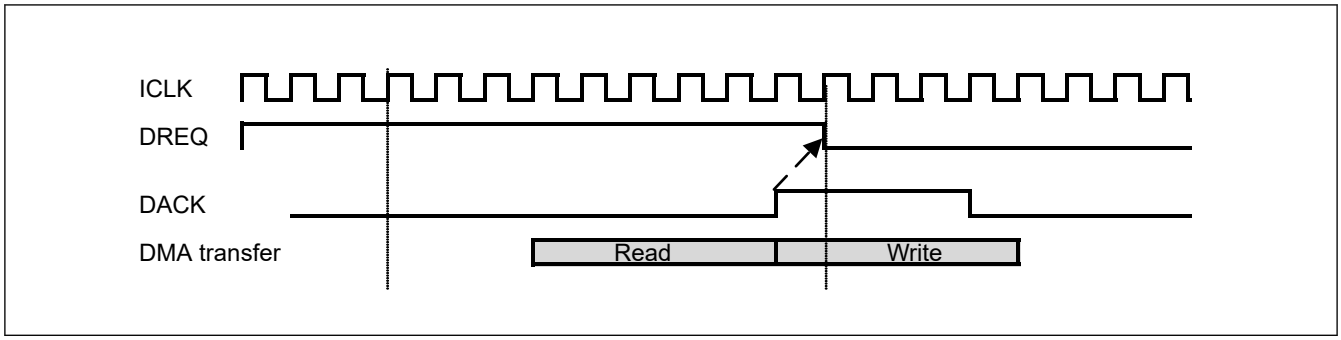


Figure 15.19 Level detection timing (high-level detection (HIEN = 1), DACK output is active during writing (REQD = 1))

15.4.5 DMA Acknowledge Output/DMA Transaction Completion Output Function

The DMA acknowledge signal (DACK) is output as a reception response signal for a DMA transfer request.

The DMA transfer completion signal (TEND) is output when the DACK signal for the last DMA transfer is output.

If the DREQ signal is used for a DMA transfer request, use DACK as the DMA acknowledge signal, and use TEND as the DMA transaction completion output signal.

Set the output mode of the DACK signals with the AM[2:0] bits in the Channel Configuration register (CHCFG_n). Level output and bus cycle output are supported.

The outputs of the DACK and TEND signals on the DACK and TEND pins are active only when the conditions given below are met. To use the DACK and TEND pins, specify appropriate settings for the Port Function Control register (PFC) and the Port Mode Control register (PMC) to use those pins as peripheral functions. For details on pin settings, see section 17, I/O Ports.

The conditions:

- The DREQ pin is set as the source for the DMA transfer request
- The source and destination of the DMA transfer are set as follows:
  - DMA is set to be transferred from an external address space (CSn) to any destination while the CHCFG_n.REQD bit is set to 0 (requested from the source of transfer)
  - DMA is set to be transferred to an external address space (CSn) from any source while the CHCFG_n.REQD bit is set to 1 (requested from the destination of transfer)

The outputs on the pins are not active for the cases other than above. However, the internal signals become active according to the setting of the AM[2:0] bits.

Transfer requested from	Source for DMA transfer	Transferred from	Transferred to	CHCFG_n		External pins used for outputting an active signal	
				AM[2:0]	REQD	DACK Signal	TEND Signal
Request from outside	DREQ	External bus	Anywhere	001/010	0	DACK	TEND
		Anywhere	External bus		1	DACK	TEND
		Internal	Anywhere		0	—	—
		Anywhere	Internal bus		1	—	—
Other than above	Other than above	Anywhere	Anywhere	Any value	Any value	—	—

Note: —: The outputs on the pin are always inactive. Inactive signal level is obtained by inverting the active level set by the bits which specify the acknowledge levels and transfer-end levels.
External bus refers to the CS0 to CS5 spaces and internal bus refers to the address spaces other than the external bus.

15.4.5.1 DMA Acknowledge Signal Output

For the DMA acknowledge signal and DMA transaction completion signal, output mode might be specified according to the source.

For each DMA transfer request source, specify the correct operation with the AM[2:0] bits of the Channel Configuration register (CHCFGn) according to Table 15.19, Table 15.21, and Table 15.22.

For details on level output operation, see Figure 15.20 and Figure 15.21.

For details on bus cycle output operation, see Figure 15.22 and Figure 15.23.

Table 15.22 DACK/TEND pin output settings (1 of 2)

Mode	AM[2] (CHCFG_n)	AM[1:0] (CHCFG_n)	REQD (CHCFG_n)	Applications
Level	0	01b	0 (activated during read operation)	In level mode, enable DACK. The DACK output remains active until the DMA transfer request is deactivated. The TEND signal is activated only when the last DMA transfer is performed.
			1 (activated during write operation)	
Bus cycle	0	10b or 11b	0 (activated during read operation)	The DACK signal remains active during a bus cycle. Use this mode if you want to activate the DACK signal until a bus cycle ends.
			1 (activated during write operation)	

Table 15.22 DACK/TEND pin output settings (2 of 2)

Mode	AM[2] (CHCFG_n)	AM[1:0] (CHCFG_n)	REQD (CHCFG_n)	Applications
Mask	1	Any value	Any value	Set DACK output to inactive (fixed). Use this mode if you do not want to notify the connection destination of DACK output.

15.4.5.2 Level Output

By setting the AM field in the CHCFG_n register to 001b, the mode is switched to level output. DACK output remains active until the DMA transfer request is deactivated.

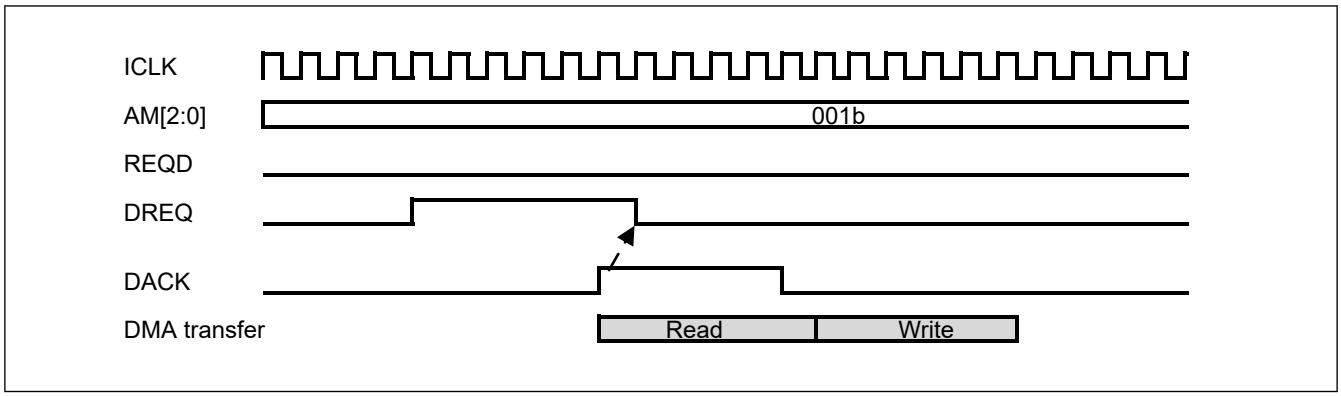


Figure 15.20 DACK output timing (AM[2:0] = 001b, REQD = 0)

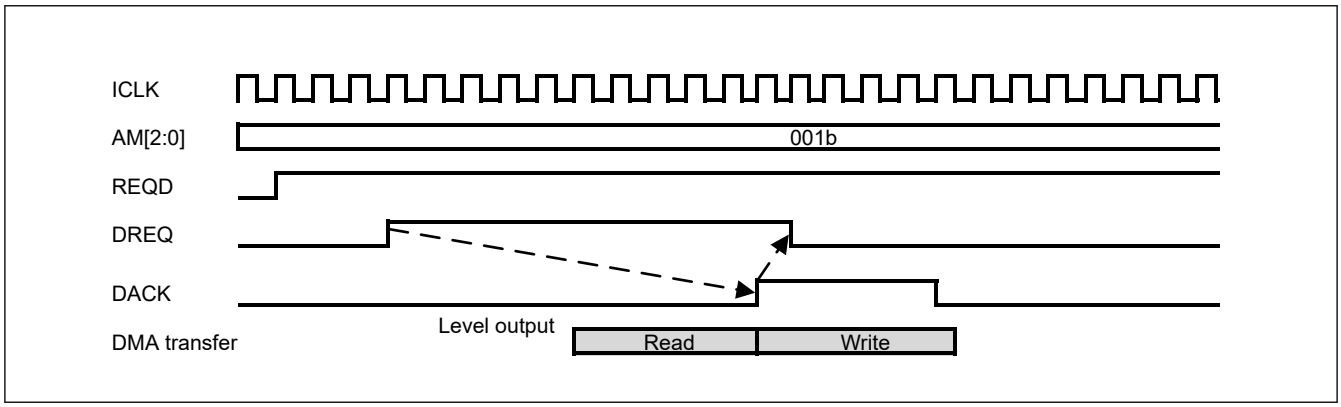


Figure 15.21 DACK output timing (AM[2:0] = 001b, REQD = 1)

15.4.5.3 Bus Cycle Output

By setting the AM field in the CHCFG_n register to 010b, the mode switches to bus cycle output. During a bus cycle, the DACK output remains active.

While the DACK signal is active, the DREQ signal is masked in the CPU. Therefore, even if the DREQ signal is set to level detection, it is not necessary to deactivate the signal on the output side.

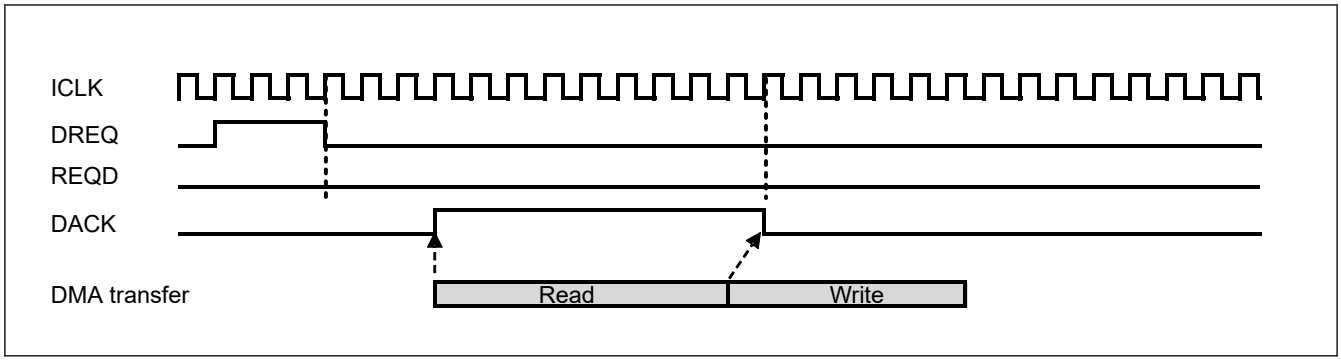


Figure 15.22 Bus cycle output timing (REQD = 0)

- When the REQD bit in the CHCFG_n register = 0 (activated during read operation), the DACK output remains active for the period from when a read request is output on the bus until a cycle elapses after reading the last data.
- If DMA transfer request input is detected by level, DMA transfer request input is deactivated until the next cycle begins after a bus cycle ends.

The following signals trigger the rising and falling of the DACK output:

Rising: Starts a transfer.

Falling: Ends a transfer.

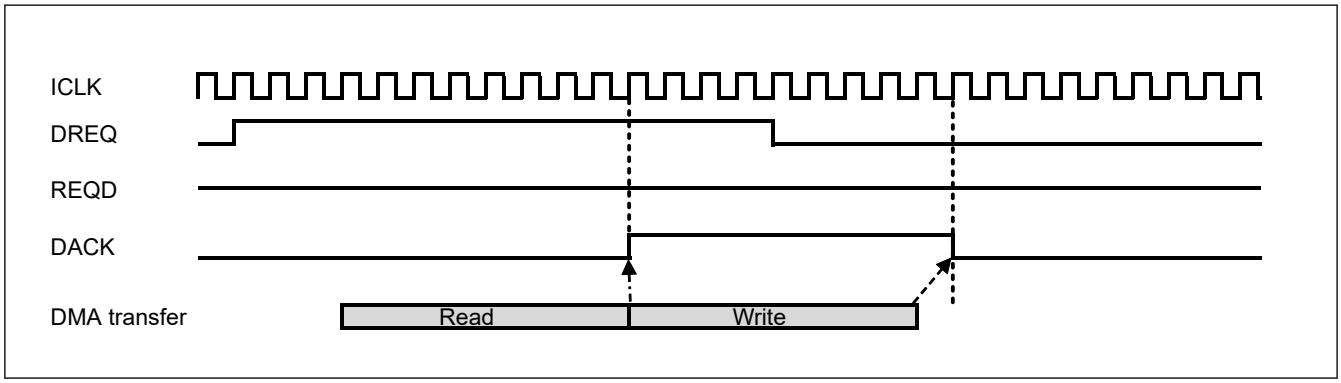


Figure 15.23 Bus cycle output timing (REQD = 1)

- When the REQD bit in the CHCFG_n register = 1 (activated during write operation), the DACK signal remains active for the period from when a write request is output until a clock (ICLK) elapses after responding to the last data.
- If DMA transfer request input is detected by level, DMA transfer request input is deactivated until the next cycle begins after a bus cycle ends.

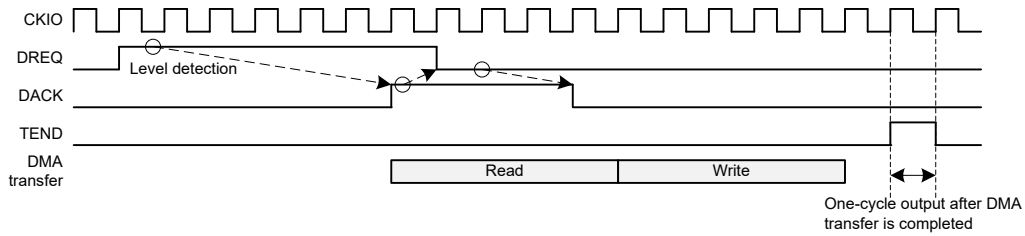
The following signals trigger the rising and falling of the DACK output:

Rising: Starts a transfer.

Falling: Ends a transfer.

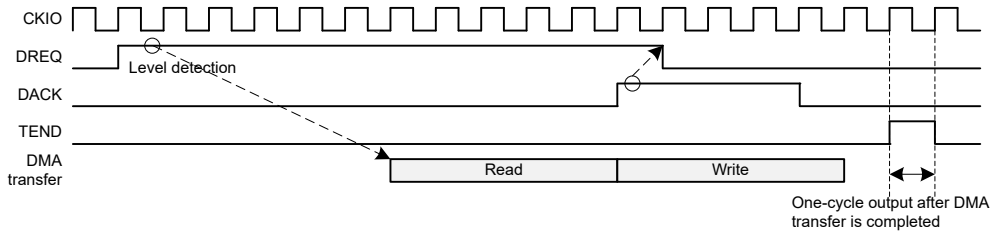
15.4.6 DMA Transaction Completion Signal Output

Figure 15.24 and Figure 15.25 show external DMA interface operation examples. One cycle of the TEND signal is output after the DMA transfer is completed.



Note:
- DMA transfer request: High level detection, request from transfer source (CHCFG_n.REQD = 0)
- DMA acknowledge output: Level mode (CHCFG_n.AM[2:0] = 001b)
- When source and destination are specified to external bus, the TEND signal might be output before write cycle completion.

Figure 15.24 External DMA interface operation example (REQD = 0: read)



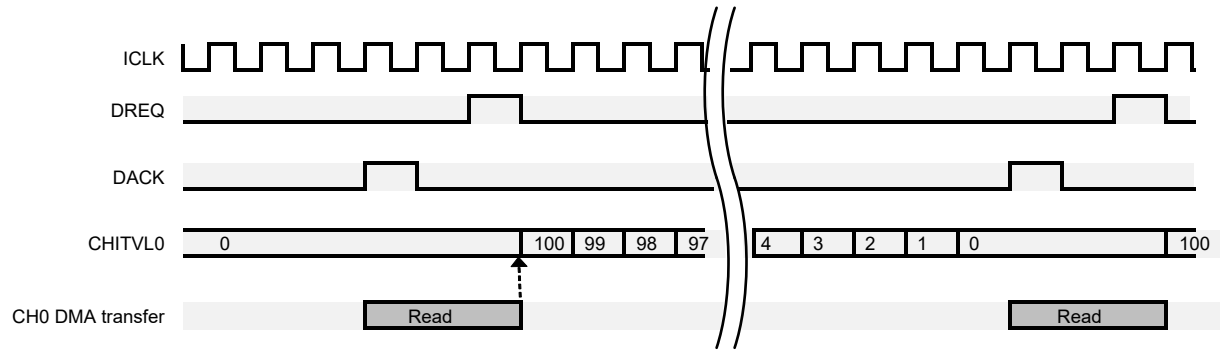
Note:
- DMA transfer request: High level detection, request from transfer source (CHCFG_n.REQD = 1)
- DMA acknowledge output: Level mode (CHCFG_n.AM[2:0] = 001b)

Figure 15.25 External DMA interface operation example (REQD = 1: write)

15.4.7 Interval Count Function

By the setting for the ITVL field in the CHITVL_n register, the execution interval of DMA transfers can be adjusted. This function is used to avoid occupation of a bus by DMAC. Until the count value becomes 0, no DMA transfer for the next DMA request is performed.

Figure 15.26 provides an operation example.



Note: In this example, when REQD = 0 (a transfer request issued by the source module) and SDS < DDS (data size at the destination is greater than that at the source), a preset number of intervals (100) are inserted between several rounds of read access after a transfer request.

Figure 15.26 Interval count (REQD = 0, SDS < DDS)

An interval is inserted after a transfer specified by the REQD bit in the CHCFG_n register. Figure 15.27 shows the relationship between the setting values for the REQD, SDS, and DDS bits in the CHCFG_n register and the cycles to which intervals are applied.

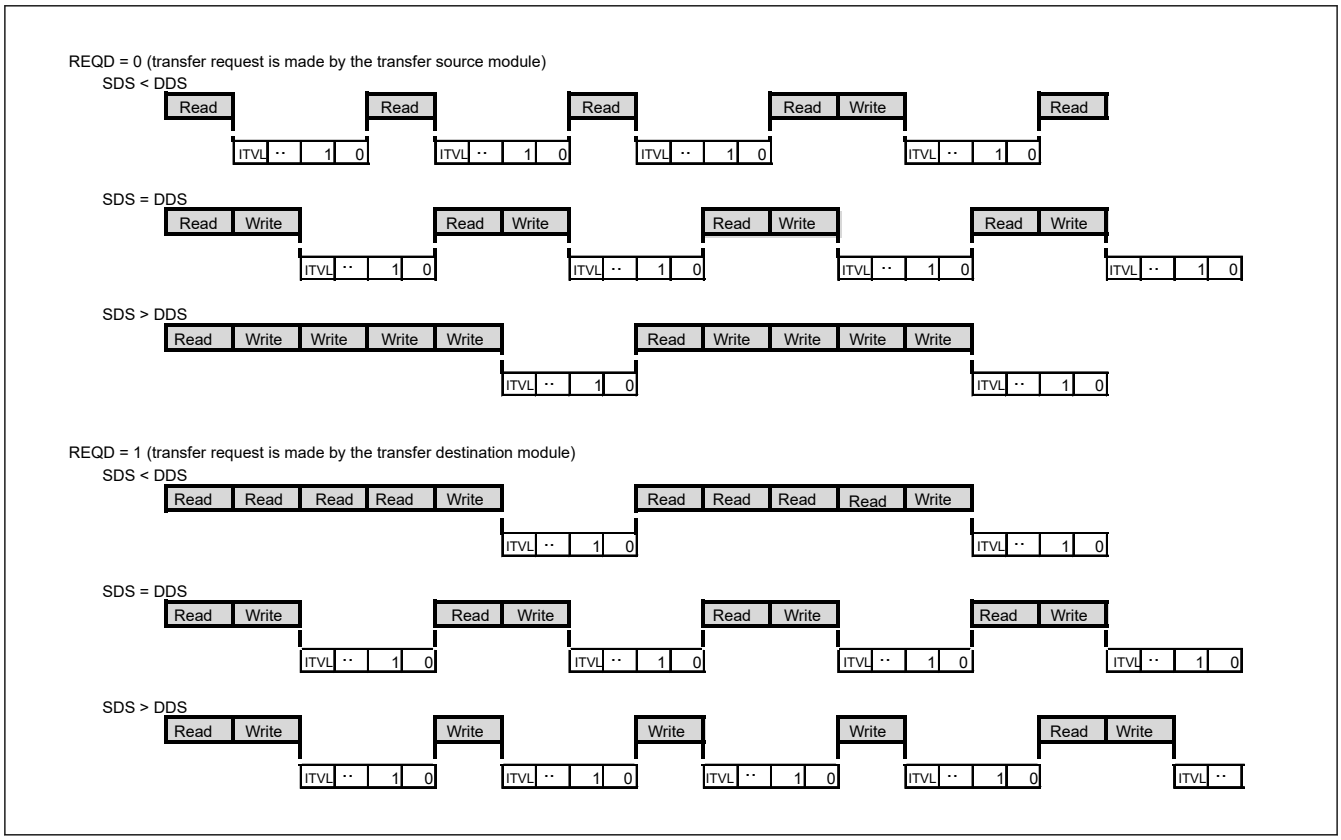


Figure 15.27 DMA transfer settings and interval counts

15.4.8 Differences in Operation According to the Transfer Data Size

15.4.8.1 When the Transfer Data Size on the Transfer Source is Small

When the transfer data size at the destination is large, reading from the source proceeds several times, and this is followed by writing to the destination.

Figure 15.28 shows the timing diagram when the transfer source is 8 bits, and the transfer destination is 32 bits (SDS = 0 and DDS = 2 in the CHCFG_n register) for rising edge detection.

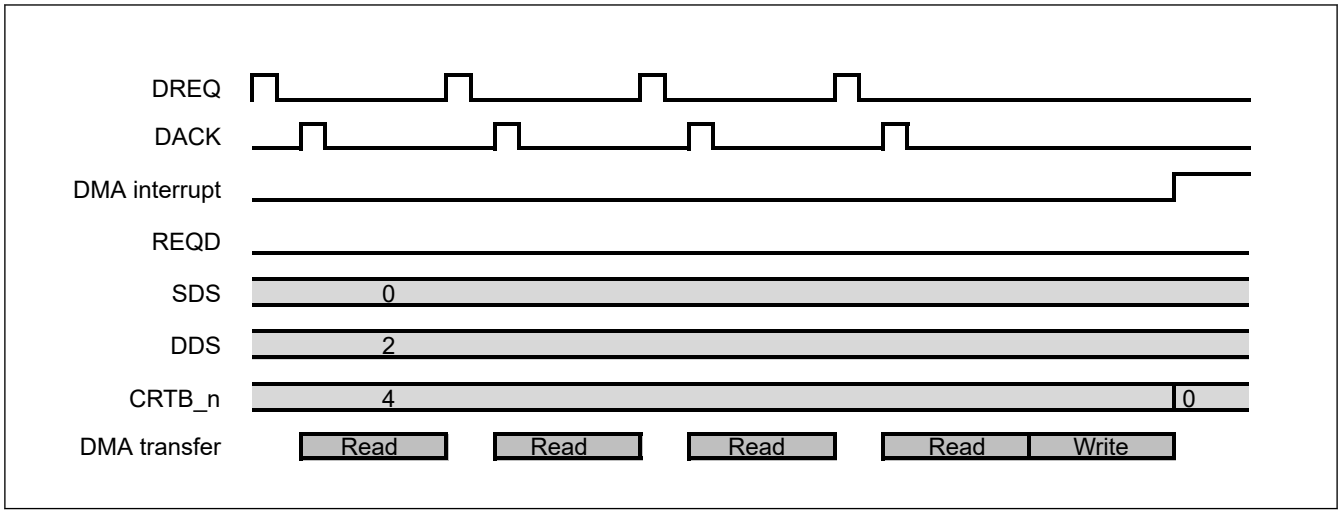


Figure 15.28 When the size of data on the transfer source is small (LVL in CHCFGn = 0, HIEN = 1, REQD = 0, SDS < DDS)

15.4.8.2 When the Transfer Data Size on the Transfer Destination is Small

When the transfer data size on the transfer source is large, after a single read operation, write operation to the transfer destination is performed a few times. Figure 15.29 shows the timing diagram when the transfer source is 64 bits, and the transfer destination is 16 bits (SDS = 3 and DDS = 1 in the CHCFG_n register) for rising edge detection.

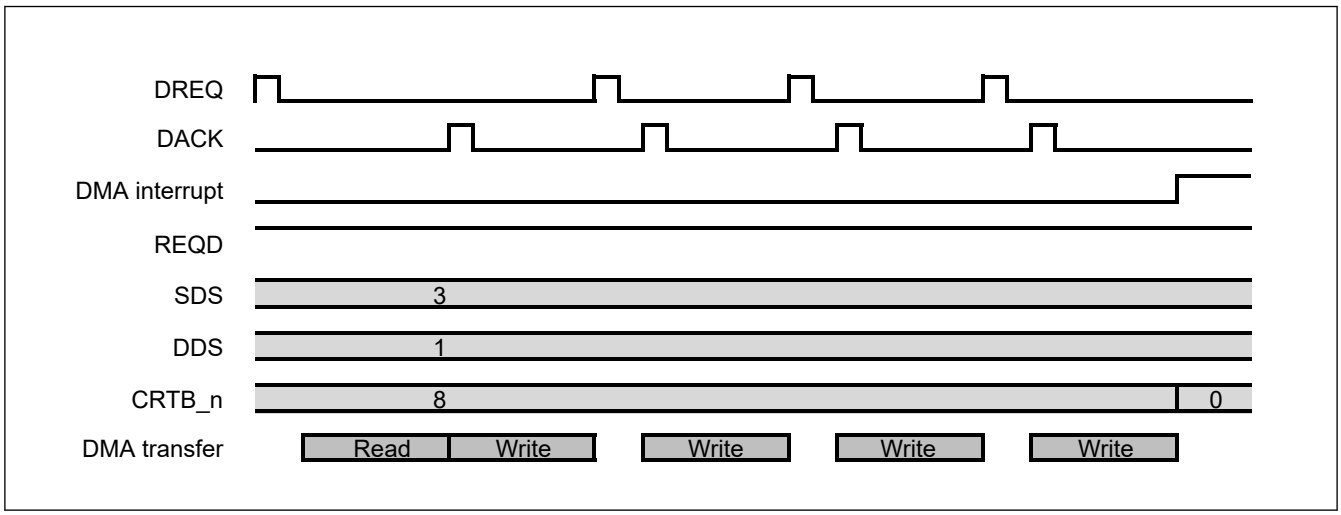


Figure 15.29 When the size of data on the transfer destination is small (LVL = 0, HIEN = 1, REQD = 1, and SDS > DDS in CHCFG_n)

15.4.8.3 When the Size of Transfer Data on the Transfer Source and on the Transfer Destination is the Same

Every time a DMA transfer request is detected, read operation is performed on the transfer source, and write operation is performed on the transfer destination.

Figure 15.30 shows the timing diagram when the transfer source and the transfer destination are 8 bits (SDS = 0 and DDS = 0 in the CHCFG_n register) for rising edge detection.

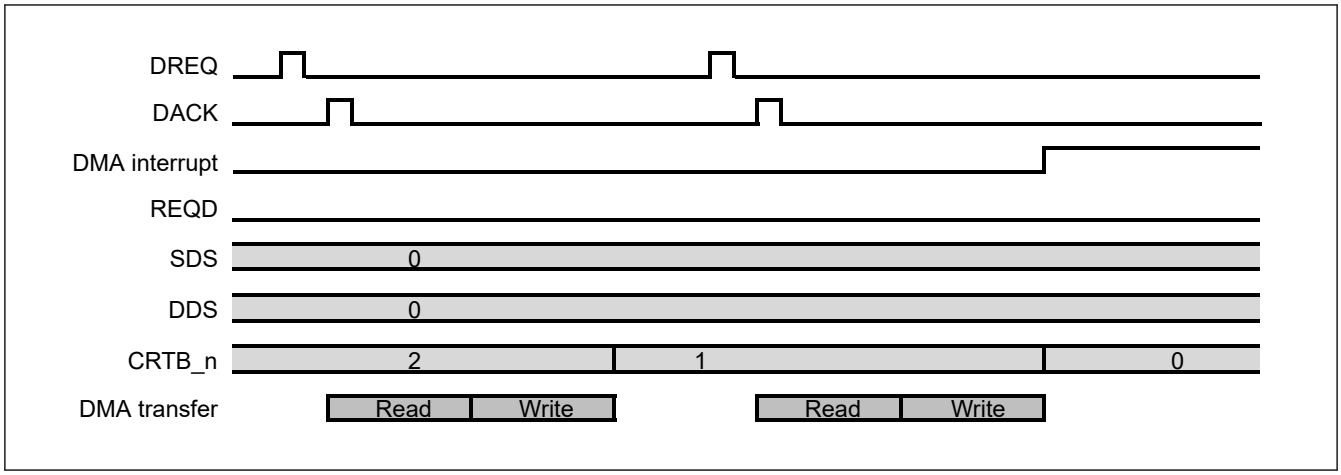


Figure 15.30 When the sizes of data on the transfer source and the transfer destination are the same (LVL = 0, HIEN = 1, REQD = 0, and SDS = DDS in CHCFG_n)

15.4.9 DMA Transfer Status

The CHSTAT_n register indicates the DMA transfer status of each channel.

The TACT bit in the CHSTAT_n register indicates that DMA operation is being performed on channel n. Writing 1 to the SETEN bit in the CHCTRL_n register sets 1. The TACT bit remains 1 while accessing a descriptor, or waiting for a DMA request.

The TACT bit is cleared when the EN bit in the CHSTAT_n register is cleared (for details on clear conditions, see section 15.3.7. CHSTAT_n : Channel Status Register n (n = 0 to 15)), and DMA transfers for the set number of times are finished.

The TACT bit is not cleared even when a DMA transfer finishes, but the EN bit is not cleared (when the REN bit in the CHCFG_n register = 1 in register mode or the next descriptor access is performed in link mode).

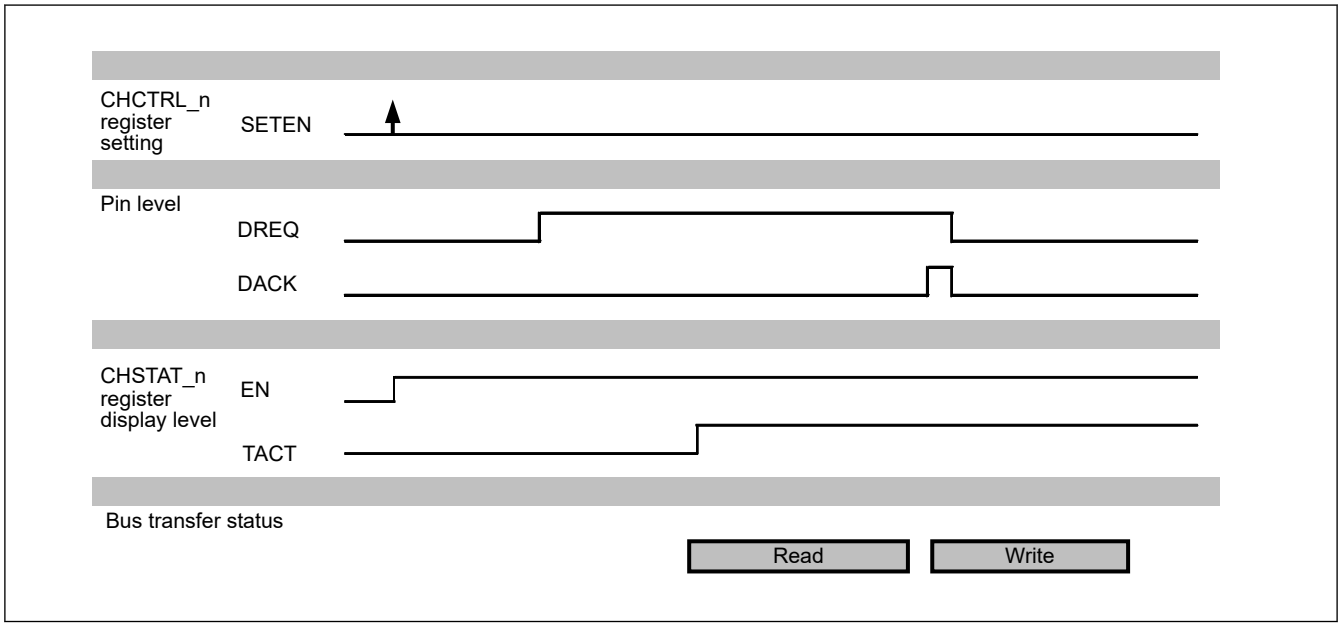


Figure 15.31 DMAC status example 1 (hardware request)

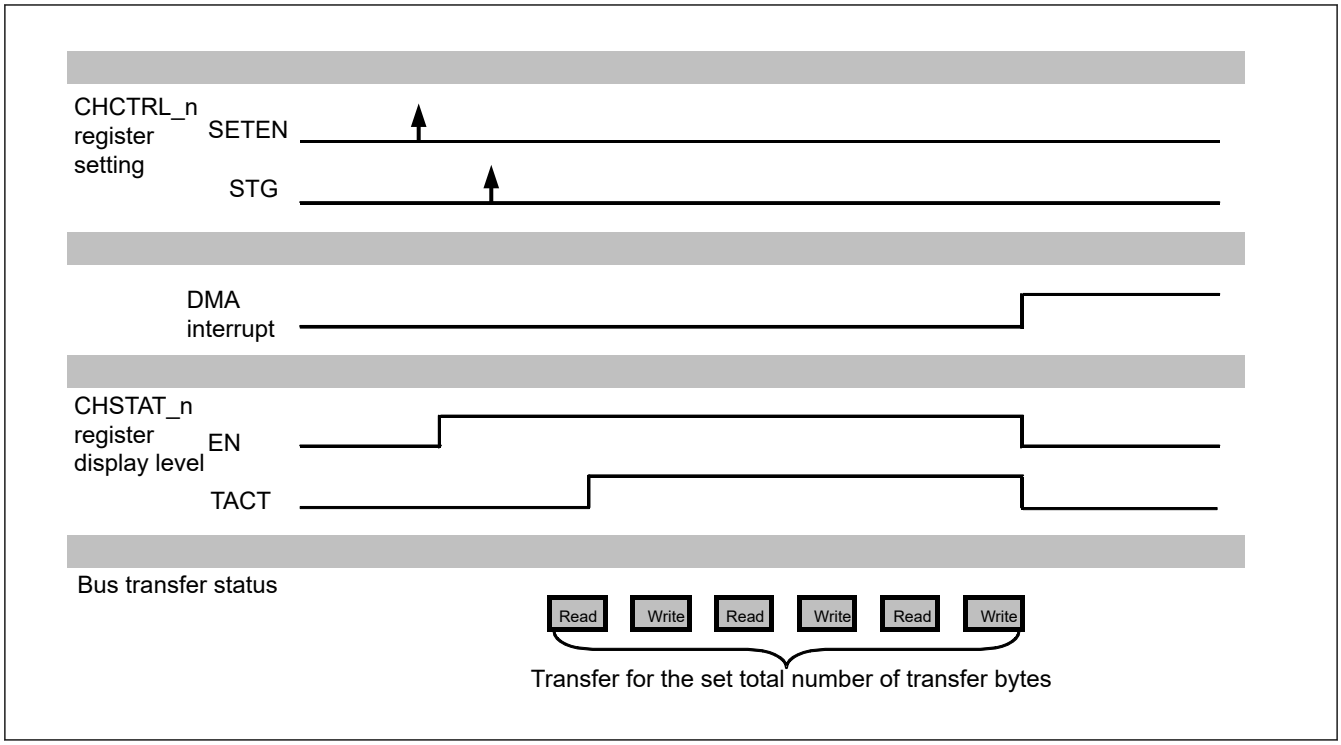


Figure 15.32 DMAC status example 2 (software request)

15.4.10 Suspending a Transfer

You can suspend a DMA transfer at the SETSUS bit in the CHCTRL_n register. At that time, if there is an already running bus cycle, waits for the cycle to end, then suspends the transfer. Writing 1 to the CLRSUS bit in the CHCTRL_n register resumes from the suspended state.

To check if the transfer is suspended, set the SETSUS bit in the CHCTRL_n register, and then make sure that the SUS bit in the CHSTAT_n register or the SUS bit in the DSTAT_SUS register on the applicable channel is set to 1.

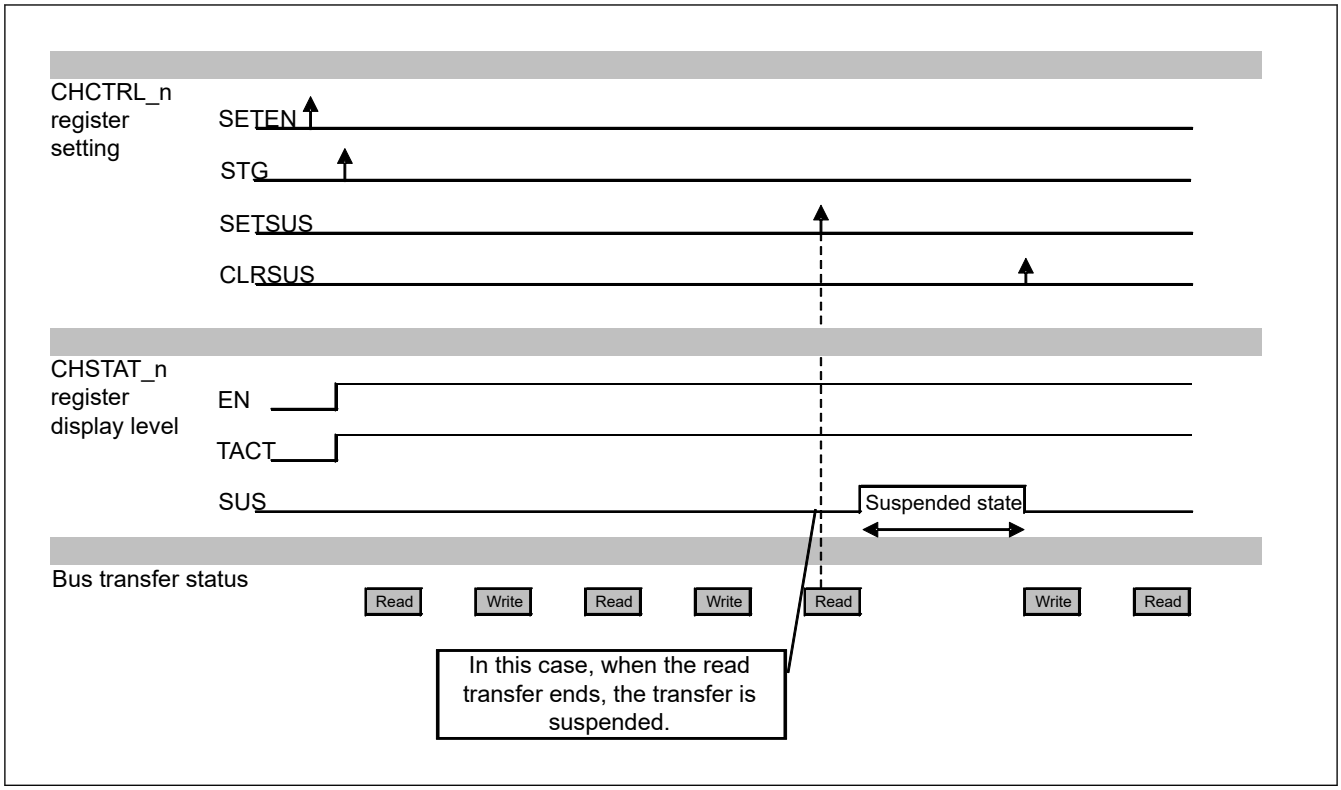


Figure 15.33 DMAC suspended state (software request block transfer)

15.4.11 Aborting a Transfer

During a DMA transfer, if you write 1 to the CLREN bit in the CHCTRL_n register, you can abort the DMA transfer of the channel. As processing after aborting the transfer, you can use the SBE bit in the CHCFG_n register to determine whether to flush data remaining in a buffer when a transfer is suspended. By default, SBE = 0 (do not flush data) is selected.

When this mode (flush data) is activated, if a transfer that is being performed when the CLREN bit in the CHCTRL_n register = 1 is aborted, data remaining in the buffer of DMAC is flushed, and the operation stops.

15.4.11.1 Aborting a Transfer (No Buffer Flush: SBE = 0)

During a DMA transfer, if you write 1 to the CLREN bit in the CHCTRL_n register, you can abort and then stop the DMA transfer. The timing to stop the transfer depends on the value set for the REQD bit. After the transfer stops, write 1 to the SWRST bit in the CHCTRL_n register, and clear the contents within the DMAC before setting the next transfer.

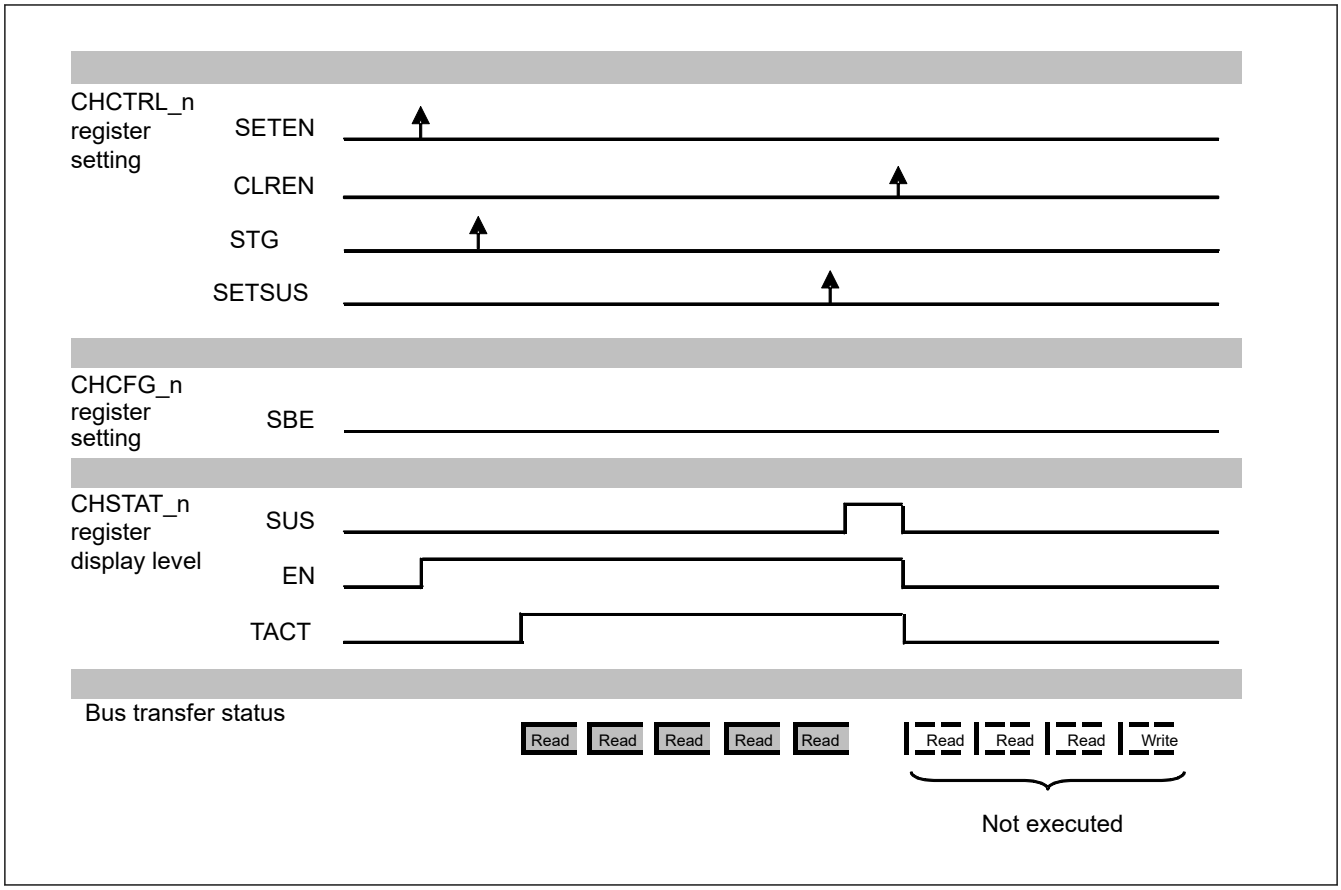


Figure 15.34 Aborting a DMA transfer

- When the TACT bit in the CHSTAT_n register is cleared, you can confirm that the channel stops completely.
- If a DMA transfer is aborted, no DMA transfer completion interrupt is generated.
- When the REQD bit in the CHCFG_n register = 0, the transfer stops when the next read operation completes. If there is data that can be written in the buffer, the transfer stops after the data is written.
- When the REQD bit in the CHCFG_n register = 1, the transfer stops when the next write operation completes.

15.4.11.2 Aborting a Transfer (Buffer Flush: SBE = 1)

During a DMA transfer, if you write 1 to the CLREN bit in the CHCTRL_n register, you can abort the DMA transfer.

When the REQD bit in the CHCFG_n register = 0, already read data is flushed (written), and then the DMA transfer stops. When REQD = 1, flush mode cannot be used.

After the transfer stops, set the SWRST bit in the CHCTRL_n register, and clear the contents within the DMAC before setting the next transfer.

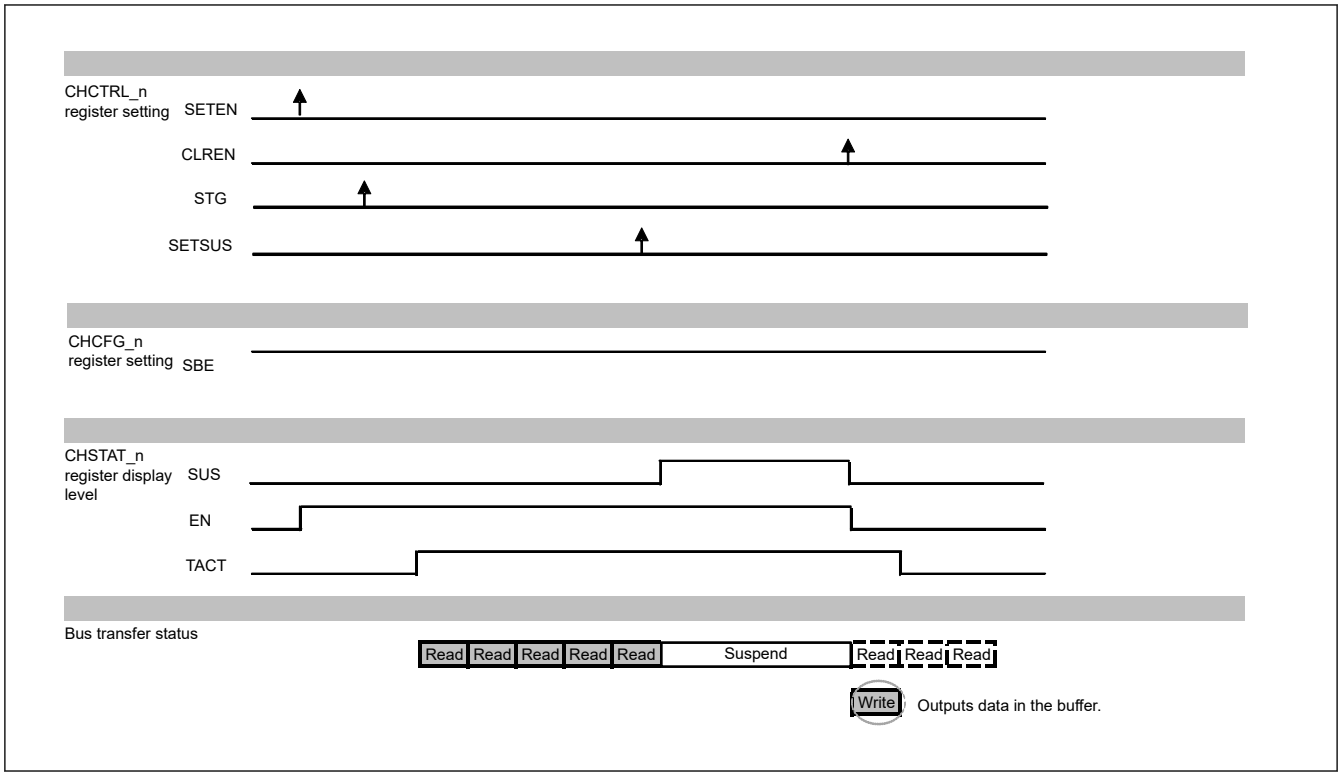


Figure 15.35 Aborting DMA transfer (buffer flush mode)

- Figure 15.35 shows an example when 1 is written to the CLREN bit in the CHCTRL_n register during the fifth read transfer in flush mode (the SBE bit in the CHCFG_n register = 1), and the transfer is aborted. It shows how read data is written, and the DMA transfer stops.
- When the TACT bit in the CHSTAT_n register is cleared to 0, you can confirm that the channel completely stops.

15.4.11.3 Checking if the Channel Stops

When 1 is written to the CLREN bit in the CHCTRL_n register, and the EN bit in the CHSTAT_n register is cleared to 0, if a transfer is already performed on a bus, DMAC cannot stop immediately. To check if DMA completely stops, make sure that the EN bit is cleared to 0, and the TACT bit in the CHSTAT_n register is cleared to 0.

15.4.11.4 Procedure for Aborting a Transfer

The following steps show the procedure for stopping a transfer:

1. Write 1 to the SETSUS bit in the CHCTRL_n register.
2. Perform polling until the SUS bit in the CHCTRL_n register is set to 1. If the EN bit = 0, go to step 6 because DMAC transfer is suspended.
3. Write 1 to the CLREN bit in the CHCTRL_n register.
4. When the SBE bit in the CHCFG_n register = 0, the transfer stops according to value for the REQD bit in the CHCFG_n register. At that time, if SBE = 1, flush mode is used.
5. By reading the CHSTAT_n register, make sure the TACT bit is cleared to 0. When TACT = 0, DMA stops completely. When TACT = 1, perform polling until the TACT bit is cleared to 0.
6. After a transfer is aborted, to perform the next DMA transfer, you must set the SWRST (software reset) bit in the CHCTRL_n register immediately before the next transfer starts.

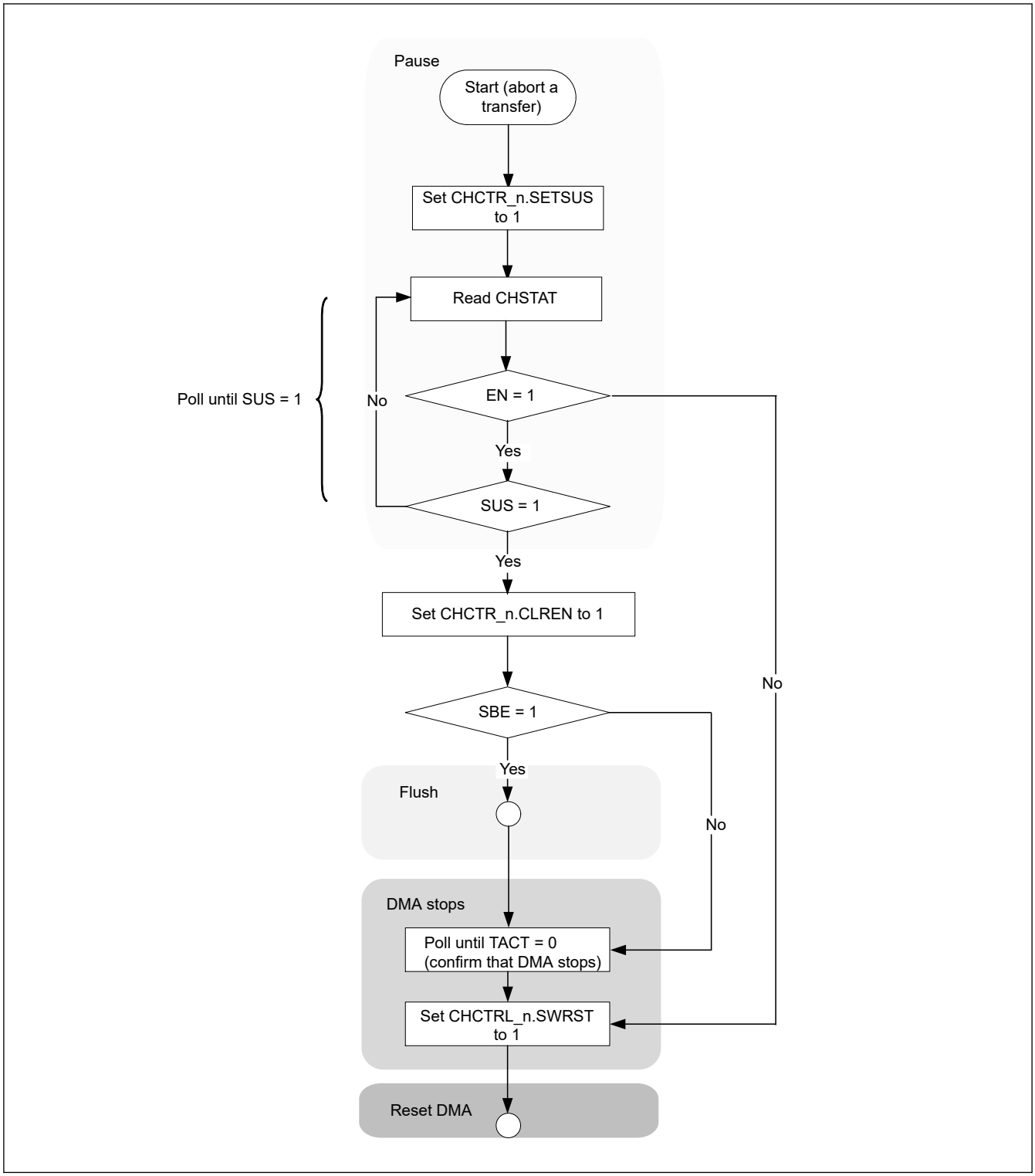


Figure 15.36 Operation flow for aborting a transfer

15.5 Interrupts

15.5.1 Interrupt Sources

DMAC has two types of interrupt sources, DMA transfer completion interrupts and DMA error interrupts for each channel. Table 15.23 shows the relationship among interrupt sources, enable bits, and status flags.

Table 15.23 Interrupt sources of DMAC

Interrupt source		Interrupt Enable bit	Interrupt Status flag	Output condition
DMA transfer completion interrupt	DMA transfer completion	CHCFG_n.DEM	CHSTAT_n.END	When a transfer for the total number of transfer bytes loaded to the CRTB_n register completes (after a write back if write back is performed in link mode)
	Descriptor invalid	DIM in header register		In link mode, when the header of the LV of the read descriptor = 0
DMA error interrupt		— (mask disabled)	CHSTAT_n.ER	When a bus error occurs during a DMA transfer and descriptor access

15.5.2 DMA Transfer Completion Interrupts

A DMA transfer completion interrupt is an interrupt request signal indicating that the DMA transfer completes. Each bit of the DMA transfer completion interrupt corresponds to each channel.

When the transfer of the total number of bytes loaded to the CRTB_n register is completed, the END bit in the CHSTAT_n register is set to 1. At that time, if the DEM bit in the CHCFG_n register = 0, a DMA transfer completion interrupt is generated (n = 0 to 15). To perform write back in link mode, an interrupt is generated after the write back.

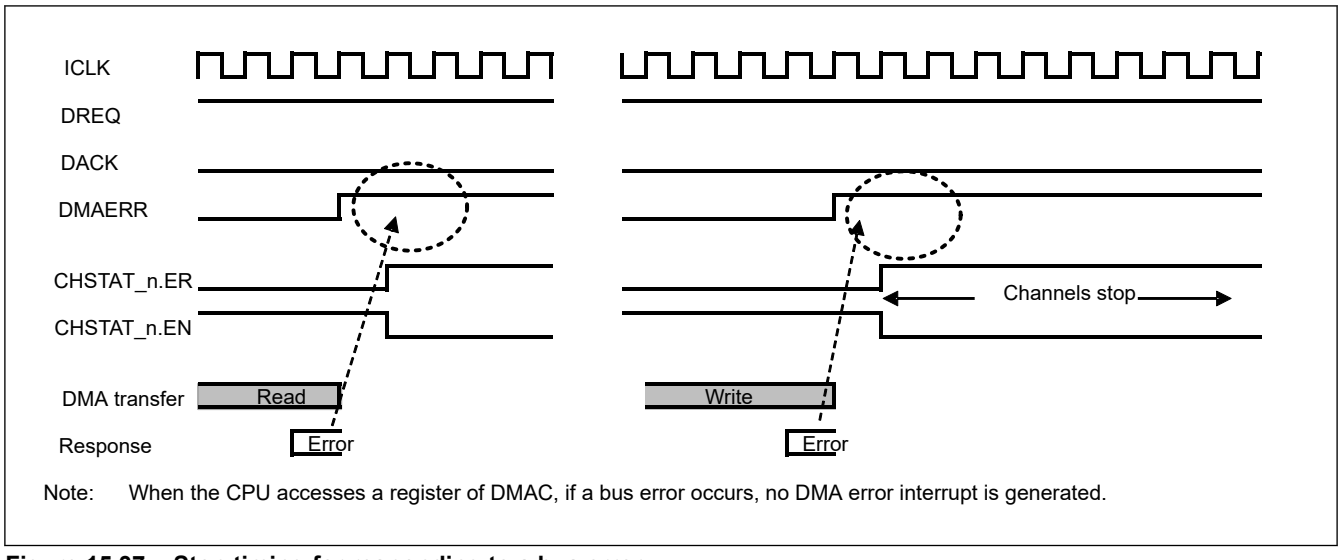
In addition, when link mode and header of the read descriptor is LV = 0, the DER bit in the CHSTAT_n register is set to 1. At that time, if the DIM bit in the header register = 0, a DMA transfer completion interrupt is generated.

15.5.3 DMA Error Interrupt

If a bus error occurs during a DMA transfer or descriptor access, this module determines that an error occurred, and stops the transfer. When a bus error occurs, the EN bit in the CHSTAT_n register for channel n where a transfer is performed is cleared to 0, and the ER bit is set to 1 (n = 0 to 15). In addition, a DMA error interrupt is generated. DMA error interrupts cannot be masked.

Data for a series of error transfers cannot be guaranteed. You must perform the transfer from the beginning by using the following procedure:

1. Set the SWRST bit in the CHCTRL_n register to 1.
2. Set each register again.



Note: When the CPU accesses a register of DMAC, if a bus error occurs, no DMA error interrupt is generated.

Figure 15.37 Stop timing for responding to a bus error

15.6 DMA Setting Examples

This section provides examples of DMA transfers. Table 15.24 lists the transfer conditions for the setting examples described in this section.

Table 15.24 List of transfer conditions for the DMA transfer setting examples

Setting example	DMA mode	Transfer mode	Transfer request
Setting example 1	Register mode	Single transfer mode	Hardware
Setting example 2	Register mode	Block transfer mode	Software
Setting example 3	Register mode (continuous execution)	Block transfer mode	Software
Setting example 4	Link mode	Block transfer mode	Software

15.6.1 Setting Example 1 (Register Hardware Request)

This section provides setting examples of DMA transfers that use hardware requests in register mode.

Table 15.25 DMA transfer setting example 1

Parameter		Description
Channel to use		DMAC0 channel 3
Priority control		Fixed priority
DMA mode		Register mode
Transfer mode		Single transfer mode
Register set to use		Next0 register set
Transfer source	Start address	0x4000_0000
	Address direction	Increment
	Data size	32 bits
Transfer destination	Start address	0x1000_0000
	Address direction	Increment
	Data size	32 bits
Number of DMA transfer bytes		64 bytes
DMA transfer request		Detects the rising edge by hardware (DREQ)
Selection of the side which makes a DMA transfer request		Requested from the transfer-side module
DACK/TEND signal		Outputs by level during read operation
DMA transfer completion interrupt mask		None

Setting Example 1

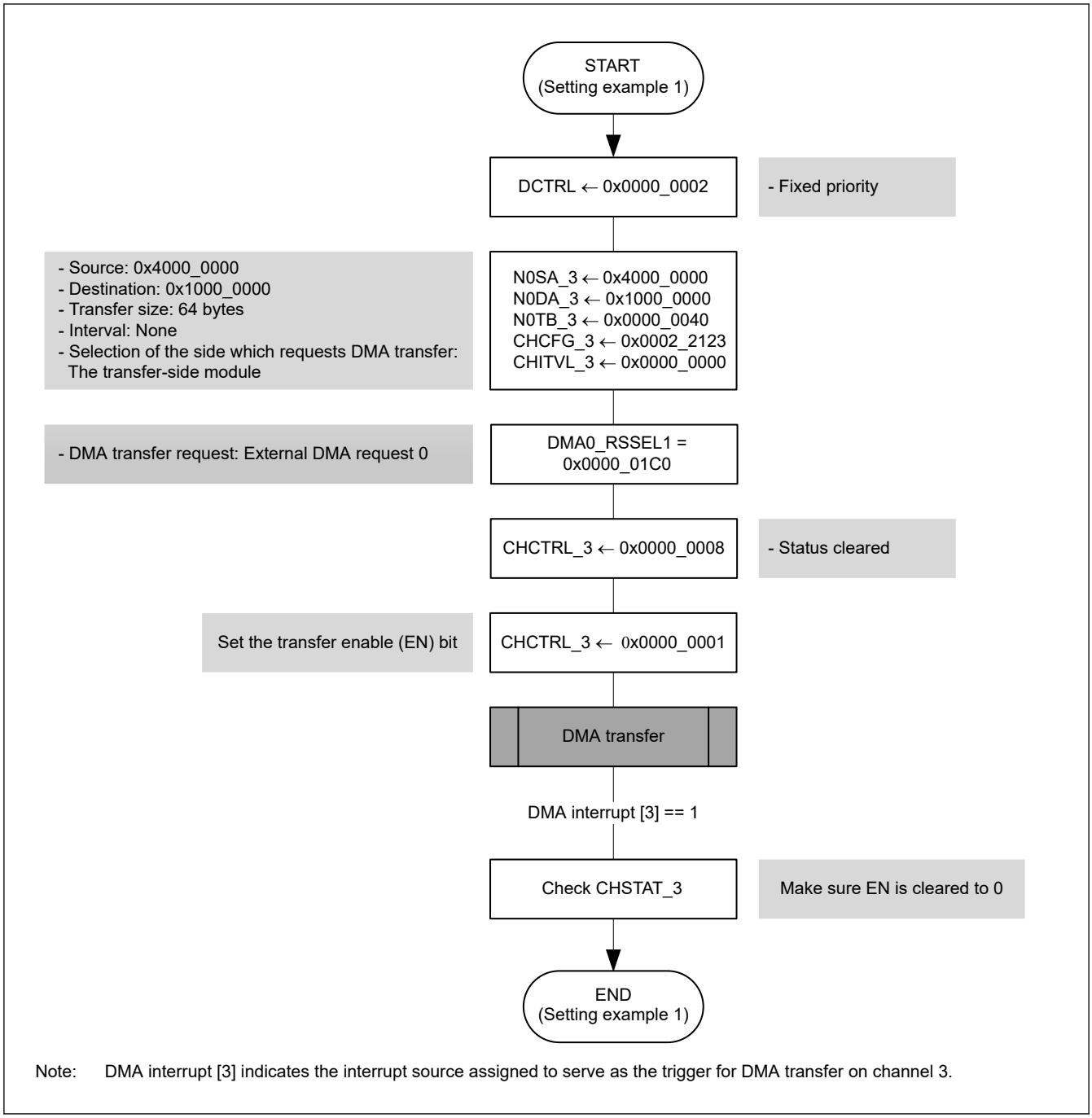
N0SA = 0x4000_0000 (transfer source address)

N0DA = 0x1000_0000 (transfer destination address)

N0TB = 0x0000_0040 (number of transfer bytes)

CHCFG = 0x0002_2123 (configuration)

CHITVL = 0x0000_0000 (interval)



Note: DMA interrupt [3] indicates the interrupt source assigned to serve as the trigger for DMA transfer on channel 3.

Figure 15.38 Setting example 1

15.6.2 Setting Example 2 (Register Mode Software Request)

This section provides a setting example of DMA transfers that use software requests in register mode.

Table 15.26 DMA transfer setting example 2 (1 of 2)

Parameter	Description
Channel to use	DMAC0 channel 2
Priority control	Round-robin
DMA mode	Register mode
Transfer mode	Block transfer mode
Register set to use	Next1 register set

Table 15.26 DMA transfer setting example 2 (2 of 2)

Parameter		Description
Transfer source	Start address	0x1000_0000
	Address direction	Increment
	Data size	8 bits
Transfer destination	Start address	0x1000_1000
	Address direction	Increment
	Data size	256 bits
Number of DMA transfer bytes		128 bytes
DMA transfer request		Software request
DACK/TEND signal		Mask
DMA transfer completion interrupt mask		None

Setting Example 2

DCTRL = 0x0000_0003 (DMA setting)

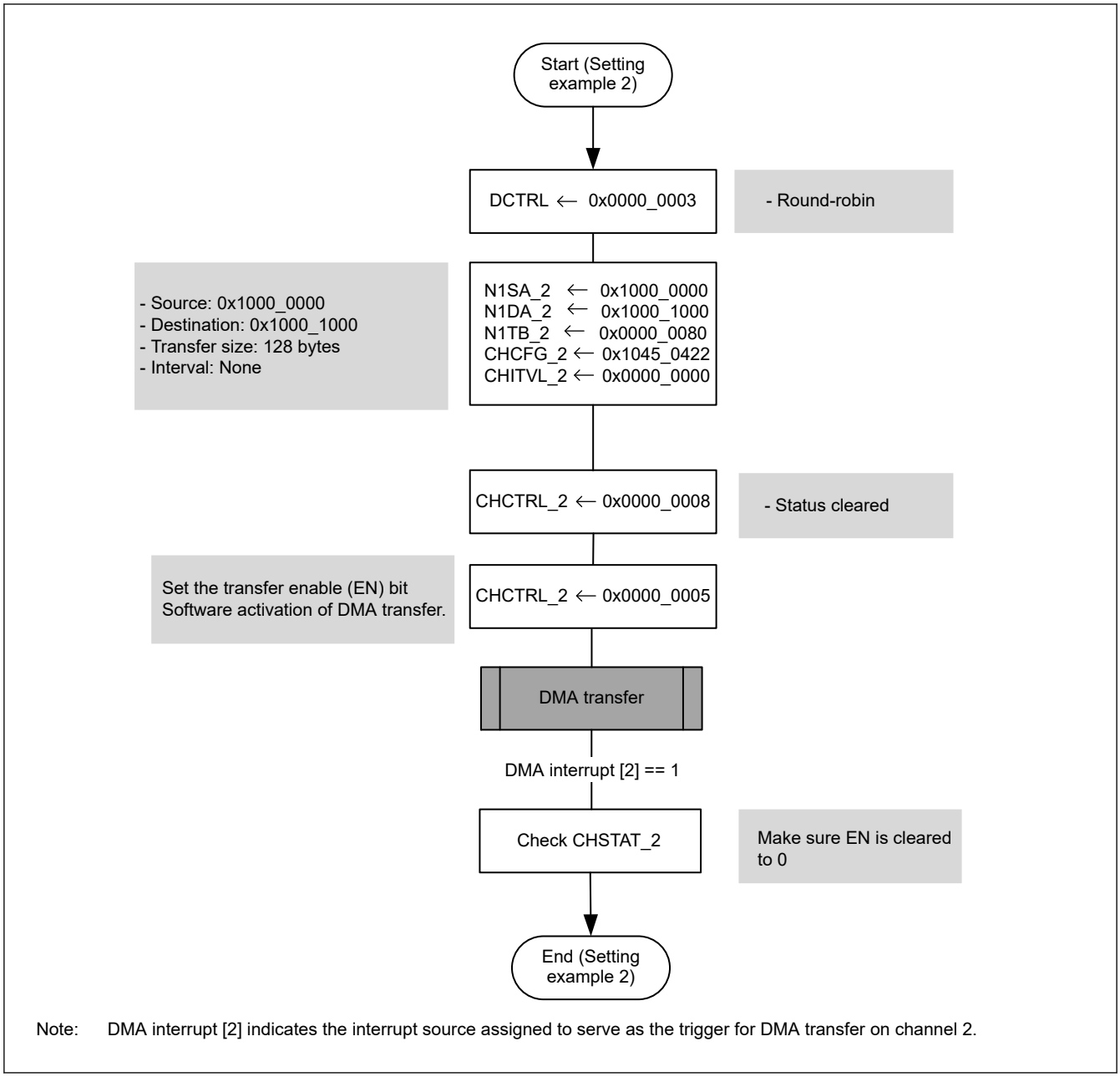
N1SA = 0x1000_0000 (transfer source address)

N1DA = 0x1000_1000 (transfer destination address)

N1TB = 0x0000_0080 (number of transfer bytes)

CHCFG = 0x1045_0422 (configuration)

CHITVL = 0x0000_0000 (interval)



Note: DMA interrupt [2] indicates the interrupt source assigned to serve as the trigger for DMA transfer on channel 2.

Figure 15.39 Setting example 2

15.6.3 Setting Example 3 (Register Mode Continuous Execution)

This section provides a setting example of DMA transfers that use Next0/1 Register Sets in series in register mode.

Table 15.27 DMA transfer setting example 3 (1 of 2)

Parameter	Description
Channel to use	DMAC0 channel 1
Priority control	Round-robin
DMA mode	Register mode
Transfer mode	Block transfer mode
Register set to use	Next0 register set → Next1 register set in series

Table 15.27 DMA transfer setting example 3 (2 of 2)

Parameter		Description
Transfer source (Next0)	Start address	0x1000_4000
	Address direction	Fixed
	Data size	32 bits
	Number of DMA transfer bytes	512 bytes
Transfer destination (Next0)	Start address	0x1000_5000
	Address direction	Fixed
	Data size	512 bits
	Number of DMA transfer bytes	512 bytes
Transfer source (Next1)	Start address	0x1000_6000
	Address direction	Fixed
	Data size	32 bits
	Number of DMA transfer bytes	2048 bytes
Transfer destination (Next1)	Start address	0x1000_7000
	Address direction	Fixed
	Data size	512 bits
	Number of DMA transfer bytes	2048 bytes
DMA transfer request		Software request
DACK/TEND signal		Mask
DMA transfer completion interrupt mask		Masks when Next0 completes

Setting Example 3

DCTRL = 0x0000_0003 (DMA setting)

N0SA = 0x1000_4000 (transfer source address)

N0DA = 0x1000_5000 (transfer destination address)

N0TB = 0x0000_0200 (number of transfer bytes)

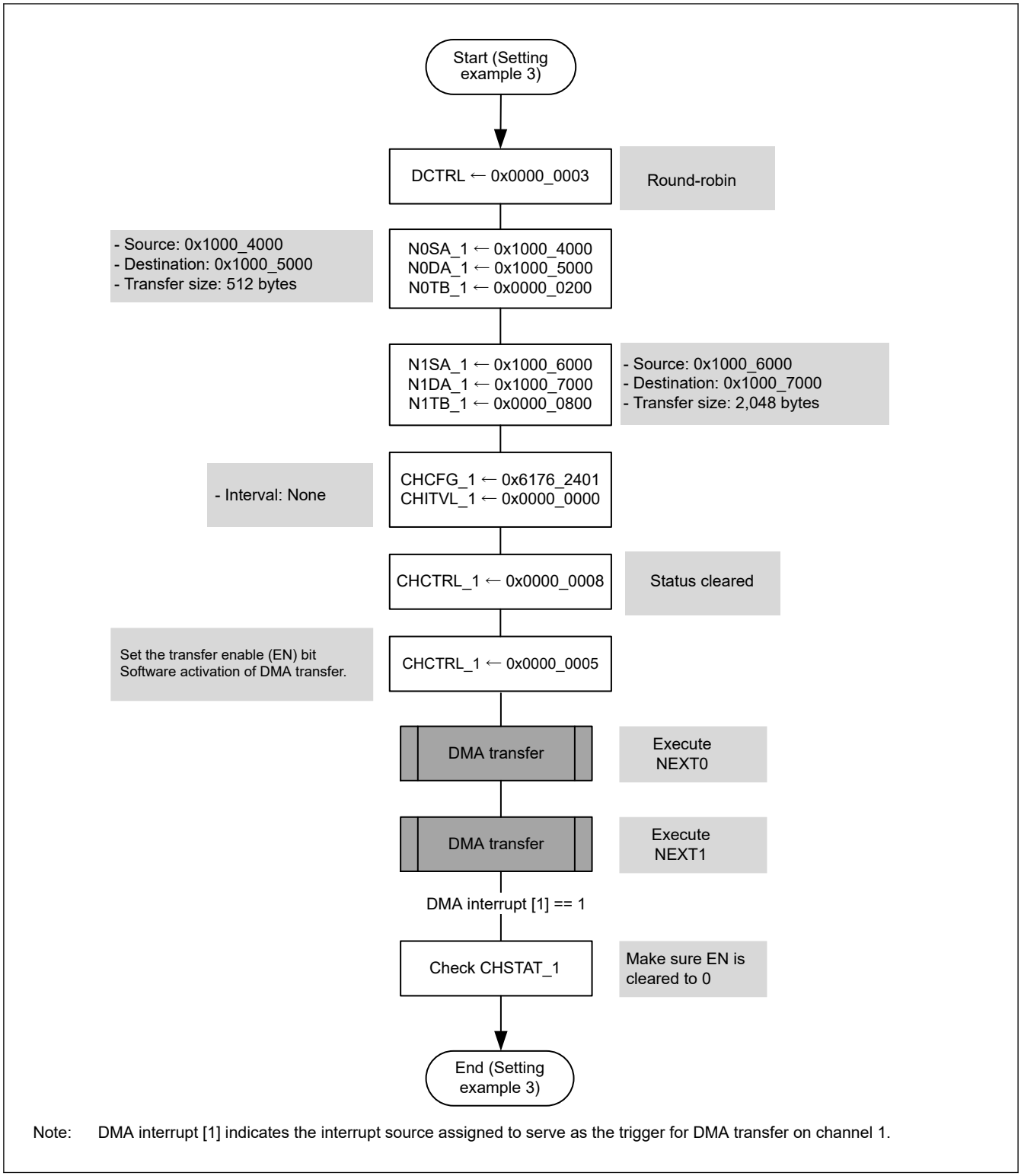
N1SA = 0x1000_6000 (transfer source address)

N1DA = 0x1000_7000 (transfer destination address)

N1TB = 0x0000_0800 (number of transfer bytes)

CHCFG = 0x6176_2401 (configuration)

CHITVL = 0x0000_0000 (interval)



Note: DMA interrupt [1] indicates the interrupt source assigned to serve as the trigger for DMA transfer on channel 1.

Figure 15.40 Setting example 3

15.6.4 Setting Example 4 (Link Mode)

This section provides a setting example when a DMA transfer is performed in link mode.

Table 15.28 DMA transfer setting example 4 (1 of 2)

Parameter	Description
Channel to use	DMAC0 channel 0

Table 15.28 DMA transfer setting example 4 (2 of 2)

Parameter	Description
Priority control	Round-robin
DMA mode	Link mode
Transfer mode	Block transfer mode
Descriptor start address	0x1080_0000

Table 15.29 DMA transfer setting example 4 (descriptor 1)

Parameter		Description
Descriptor start address		0x1080_0000
Next descriptor start address		0x1080_1000
Transfer mode		Block transfer mode
Transfer source	Start address	0x1000_0000
	Address direction	Increment
	Data size	32 bits
Transfer destination	Start address	0x1000_1000
	Address direction	Increment
	Data size	32 bits
Number of DMA transfer bytes		2048 bytes
DMA transfer request		Software request
DACK/TEND signal		Mask
DMA transfer completion interrupt output mask		Masks when a DMA transfer on descriptor 1 completes
Descriptor header	DMA interrupt when LV =1	Issued (DIM = 0)
	LV write back	Done (WBD = 0)
	Next link address	Available (LE = 0)
	Descriptor valid	Valid (LV = 1)

Table 15.30 DMA transfer setting example 4 (descriptor 2) (1 of 2)

Parameter		Description
Descriptor start address		0x1080_1000
Next descriptor start address		0x1080_2000
Transfer mode		Block transfer mode
Transfer source	Start address	0x1000_2000
	Address direction	Increment
	Data size	256 bits
Transfer destination	Start address	0x1000_3000
	Address direction	Increment
	Data size	256 bits
Number of DMA transfer bytes		1024 bytes
DMA transfer request		Software request
DACK/TEND signal		Mask
DMA transfer completion interrupt output mask		Masks when a DMA transfer on descriptor 2 completes

Table 15.30 DMA transfer setting example 4 (descriptor 2) (2 of 2)

Parameter		Description
Descriptor header	DMA interrupt when LV =1	Issued (DIM = 0)
	LV write back	Done (WBD = 0)
	Next link address	Available (LE = 0)
	Descriptor valid	Valid (LV = 1)

Table 15.31 DMA transfer setting example 4 (descriptor 3)

Parameter		Description
Descriptor start address		0x1080_2000
Next descriptor start address		—
Transfer mode		Block transfer mode
Transfer source	Start address	0x1000_4000
	Address direction	Increment
	Data size	512 bits
Transfer destination	Start address	0x1000_6000
	Address direction	Increment
	Data size	512 bits
Number of DMA transfer bytes		4096 bytes
DMA transfer request		Software request
DACK/TEND signal		Mask
DMA transfer completion interrupt output mask		Do not mask
Descriptor header	DMA interrupt when LV =1	Issued (DIM = 0)
	LV write back	Done (WBD = 0)
	Next link address	Not available (LE = 1)
	Descriptor valid	Valid (LV = 1)

Setting Example 4

DCTRL = 0x0000_0003 (DMA setting)

NXLA = 0x1080_0000 (descriptor start address)

CHCFG = 0x8000_0000 (configuration)

Table 15.32 Descriptor settings

Parameter	Descriptor 1	Descriptor 2	Descriptor 3
Header	0x0000_0001	0x0000_0001	0x0000_0003
SA (Source Address)	0x1000_0000	0x1000_2000	0x1000_4000
DA (Destination Address)	0x1000_1000	0x1000_3000	0x1000_6000
TB (Transaction Bytes)	0x0000_0800	0x0000_0400	0x0000_1000
CFG (Configuration)	0x8142_2220	0x8145_5220	0x8046_6220
ITVL (Interval)	0x0000_0000	0x0000_0000	0x0000_0000
EXT (Extension)	0x0000_0000	0x0000_0000	0x0000_0000
NXLA (Next Link Address)	0x1080_1000	0x1080_2000	0x0000_0000

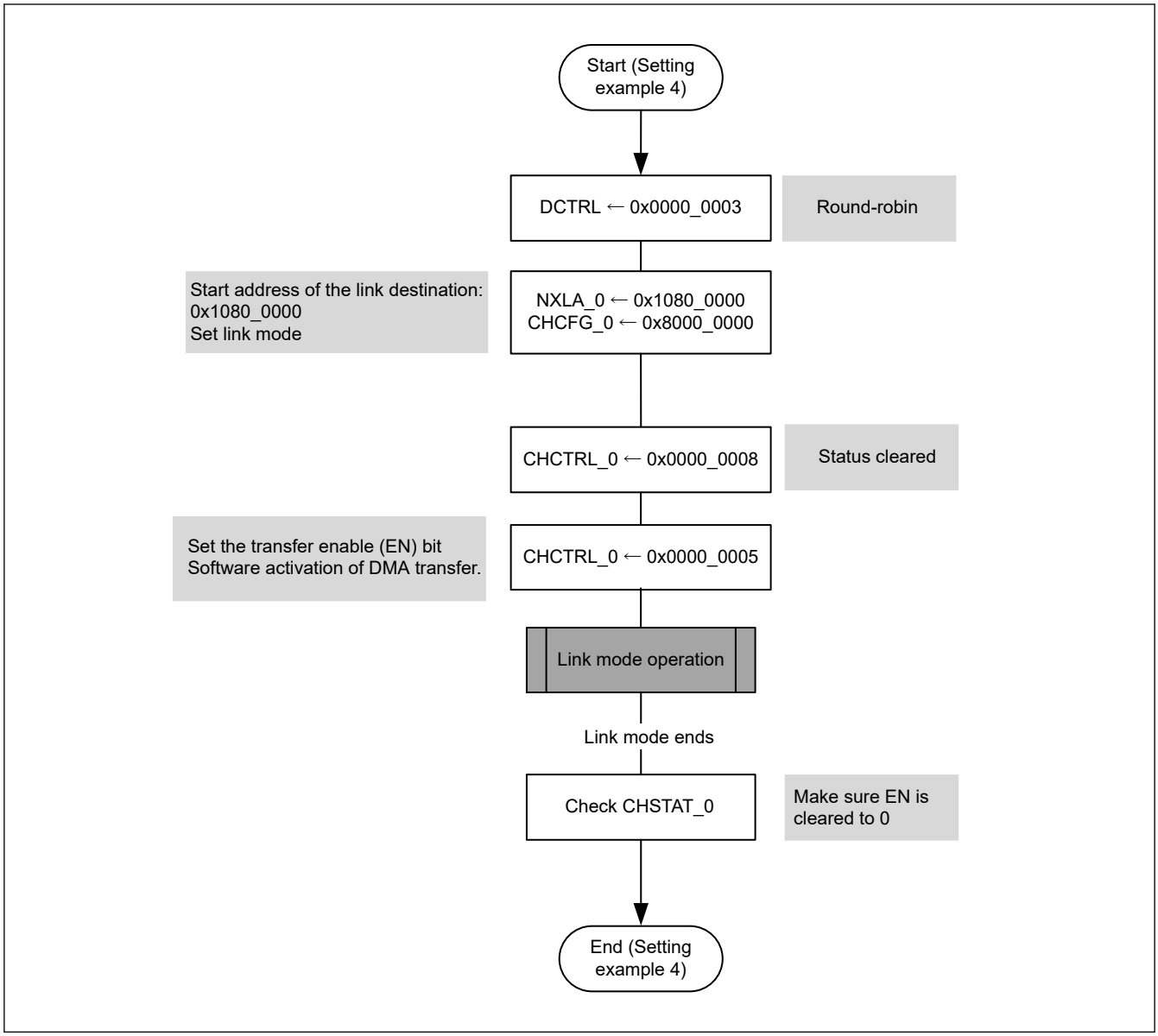


Figure 15.41 Setting example 4

15.6.5 Next Register Continuous Execution Settings

Figure 15.42 shows the flow when using two Next register sets to continue DMA transfers in register mode.

While performing a DMA transfer of a Next register, set the other Next register, and perform the DMA transfers in series.

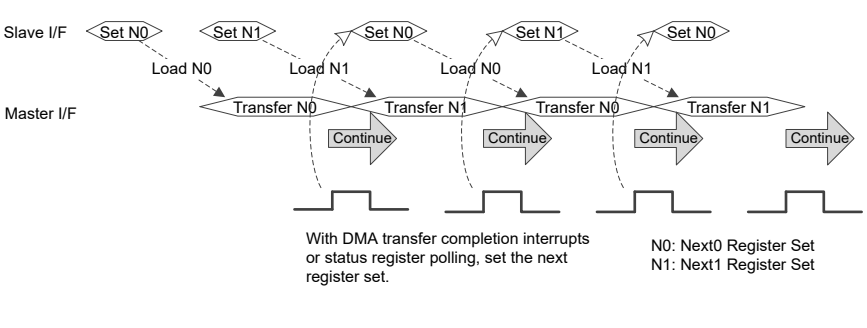


Figure 15.42 Flow for Next register continuous execution setting

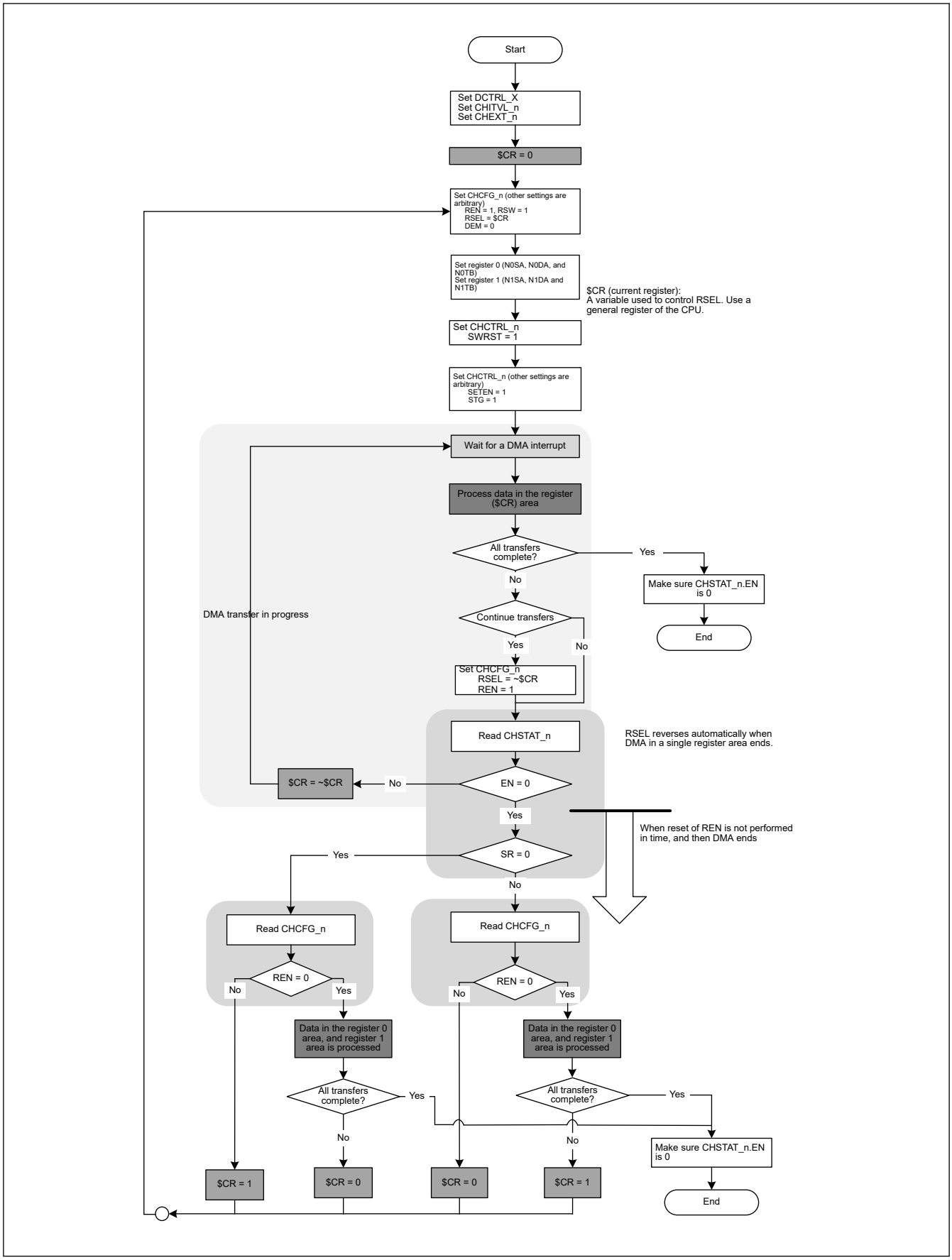


Figure 15.43 Setting example 5

(1) Supplementary information

Save the register sets 0 (N0SA_n, N0DA_n, and N0TB_n registers) and 1 (N1SA_n, N1DA_n, and N1TB_n registers) in a general register (call this register value $CR for convenience).

Every time a DMA transfer of a register set completes (a DMA transfer completion interrupt is generated), the REN bit in the CHCFG_n register is cleared to 0 automatically. To perform transfers in series, write 1 to the REN bit in the CHCFG_n register.

In this mode, two Next registers are executed in series. However, if the REN bit is not set before a DMA transfer completes (before the next DMA transfer completion interrupt is generated), the continuous execution stops. In this case, by reading the SR and EN bits in the CHSTAT_n register, and the REN bit in the CHCFG_n register, you can check how far the transfer has performed. To resume the transfer, perform the procedure described in Figure 15.43.

15.7 Usage Notes

This section provides notes on the DMA Controller module.

- When a transfer of which source and destination are the same, or partially shared area is performed, consistency of data cannot be guaranteed. Therefore, do not perform a transfer of which source and destination address areas overlap.
- For hardware activation, if REQD = 1 (the transfer destination issues a hardware request), SBE = 1 (flush mode) then the forced ejection function cannot be used. The transfer is deactivated by hardware.

15.7.1 When the DACK Signal is Divided and Output

When 4 bytes or more of data is transferred to an 8-bit or 16-bit external device, or 2 bytes or more of data is transferred to an 8-bit external device, a DMA transfer unit is divided into multiple bus cycles. The TEND signal outputs after the last DMA access completion.

When a DMA transfer is divided into multiple bus cycles and the CS signal is deactivated between bus cycles, DACK output is divided in the same way as the CS signal.

Figure 15.44 provides an example of DACK/TEND.

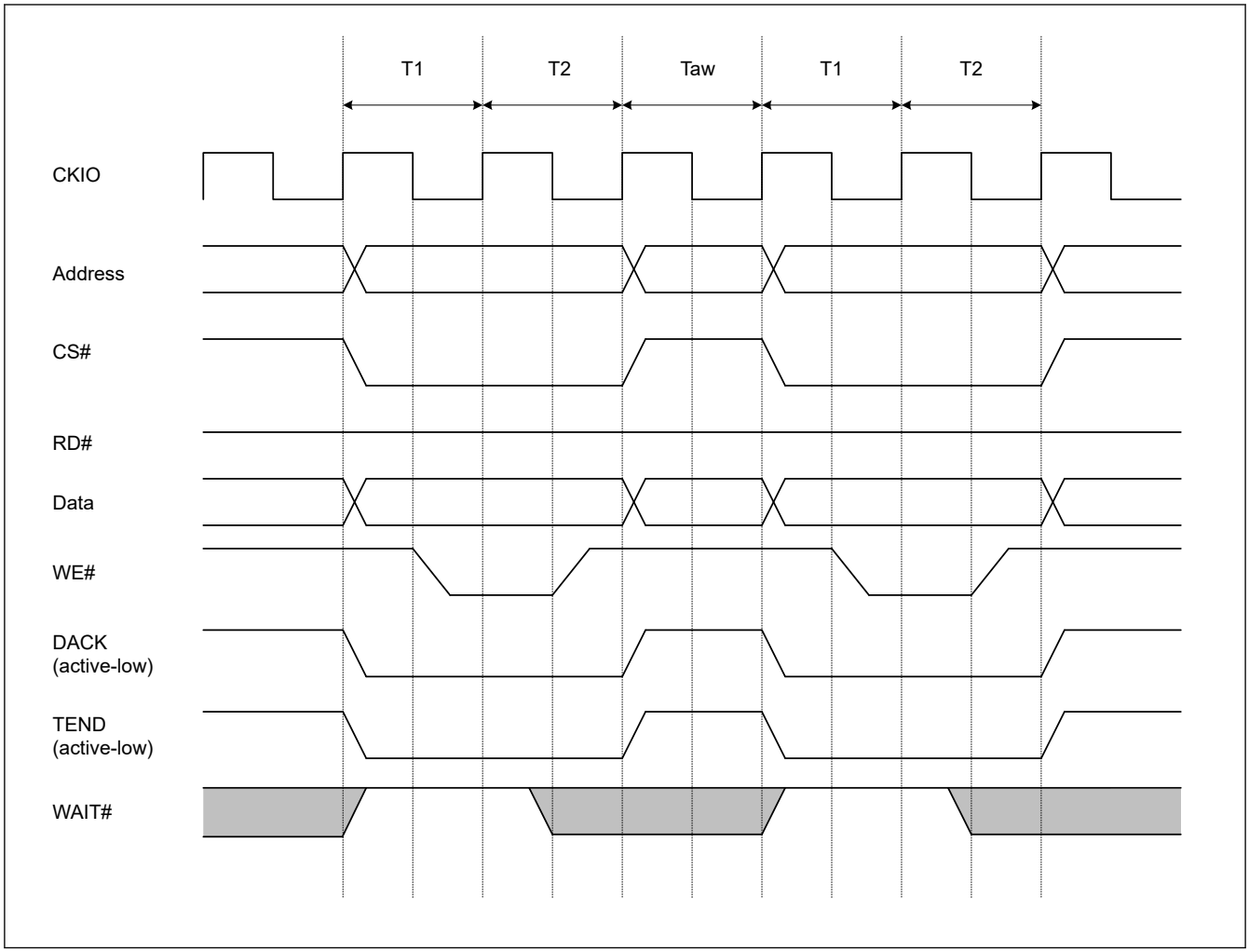


Figure 15.44 DACK divided output example

16. Event Link Controller (ELC)

16.1 Overview

The event link controller (ELC) connects (links) events generated by various peripheral modules to other modules. Event link allows direct cooperation among modules without CPU intervention.

Table 16.1 lists the specifications of the ELC and Figure 16.1 shows a block diagram.

Table 16.1 ELC specifications

Parameter	Specifications
Event link function	• 648 types of event signals can be directly connected to modules. • The operation of timer modules can be selected when an event is input to the timer module. • Event link operation is possible for Port 14 and Port 30. Single port*1: An event link can be set for a single bit specified in a port. Port group*1: An event link can be set for a group of single bits specified within eight I/O ports.

Note 1. The single port and port group specified as the input generate an event according to the change in the connected signal value.

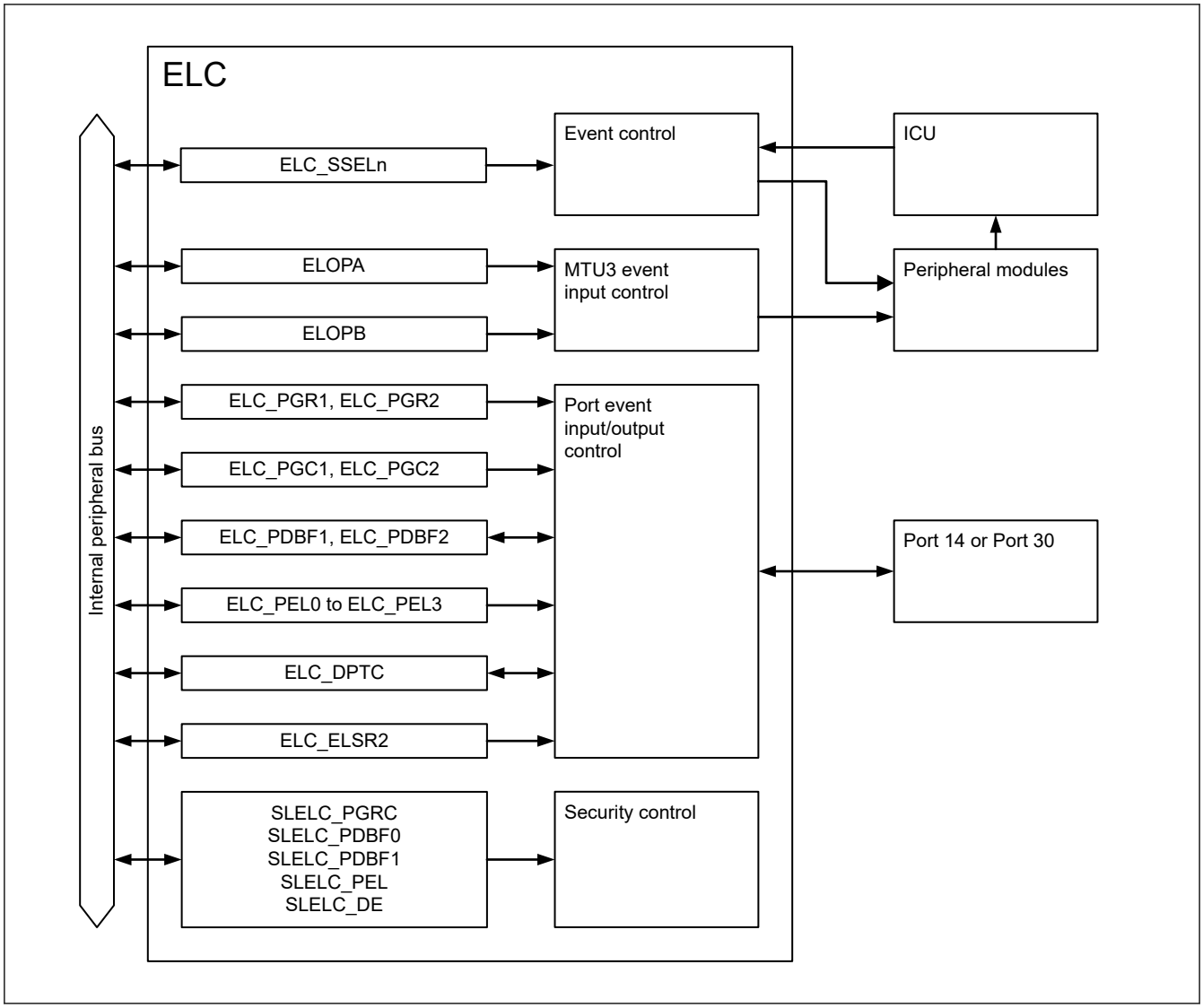


Figure 16.1 Block diagram of event link controller

16.2 Register Map

Table 16.2 ELC register map

Address	Register Symbol	Register Name	Write Protection
0x802A_0698 + 0x04 × n	ELC_SSELn	ELC Event Source Select Register n (n = 0 to 64)	—
0x8029_0000	ELOPA	Event Link Option Setting Register A	—
0x8029_0004	ELOPB	Event Link Option Setting Register B	—
0x802C_0E00 + 0x01 × (n - 1)	ELC_PGRn	ELC Port Group Setting Register n (n = 1, 2)	PRCRN.PRCN2
0x802C_0E02 + 0x01 × (n - 1)	ELC_PGCn	ELC Port Group Control Register n (n = 1, 2)	PRCRN.PRCN2
0x802C_0E04 + 0x04 × (n - 1)	ELC_PDBFn	ELC Port Buffer Register n (n = 1, 2)	PRCRN.PRCN2
0x802C_0E0C+ 0x01 × n	ELC_PELn	ELC Port Setting Register n (n = 0 to 3)	PRCRN.PRCN2
0x802C_0E10	ELC_DPTC	ELC Edge Detection Control Register	PRCRN.PRCN2
0x802C_0E11	ELC_ELSR2	ELC Port Event Control Register	PRCRN.PRCN2
0x802C_1E00	SLELC_PGRC	ELC_PGR/C Security Level Register	PRCRN.PRCN2
0x802C_1E04 + 0x04 × (n -1)	SLELC_PDBFn	ELC_PDBFn Security Level Register (n = 1, 2)	PRCRN.PRCN2
0x802C_1E0C	SLELC_PEL	ELC_PELn Security Level Register	PRCRN.PRCN2
0x802C_1E10	SLELC_DE	ELC_DPTC Security Level Register	PRCRN.PRCN2

Table 16.3 ELC related system control register

Target	Module Reset-Control Register	Module Stop Control Register	Slave Access Control Register
ELC_SSELn (n = 0 to 64)	—	—	SLVACCCTL8.ICU_SL
ELOPA, ELOPB	—	—	SLVACCCTL7.LLPP_SL*1
ELC_PGRn (n =0, 1)	—	—	SLELC_PGRC.EL_PGRn_SL
ELC_PGC n (n =0, 1)	—	—	SLELC_PGRC.EL_PGCn_SL
ELC_PDBF n (n =0, 1)	—	—	SLELC_PDBFn.EL_PDBF_SL
ELC_PEL n (n =0 to 3)	—	—	SLELC_PEL.ELC_PELn_SL
ELC_DPTC	—	—	SLELC_DE.ELC_DPTC_SL
ELC_ELSR2	—	—	SLELC_DE.ELC_ELSR2_SL
SLELC_PGRC SLELC_PDBFn (n =0, 1) SLELC_PEL SLELC_DE	—	—	SLPSRNS.SL

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

16.3 Register Descriptions

This section describes details of the register functions.

16.3.1 ELC_SSELn : ELC Event Source Select Register n (n = 0 to 64)

Base address: ELC = 0x802A_0698

Offset address: 0x00 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:			ELC_SEL2[9:0]										ELC_SEL1[9:0]										ELC_SEL0[9:0]									
Value after reset:	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
9:0	ELC_SEL0[9:0]	Set the number for ELC event source to be linked to the ELC destination. Event source is selectable from event table listed in section section 12.4.2. Event Table. If ELC function is not used, set 0x3FF (Disable ELC function). Setting of the number not allowed in event table is prohibited. See Table 16.4 for correspondence between register/bit assignment and ELC destination.	R/W
19:10	ELC_SEL1[9:0]		R/W
29:20	ELC_SEL2[9:0]		R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

The ELC_SSELn register specifies an event source to be linked to for each destination peripheral module. Event source number set in the ELC_SSELn register are listed in section section 12.4.2. Event Table. Table 16.4 lists the correspondence between the ELC_SSELn register/bits and the destination peripheral modules.

Table 16.4 Correspondence between the ELC_SSELn register and ELC destination (1 of 3)

Register name	Bit name	ELC destination	Register name	Bit name	ELC destination
ELC_SSEL0	ELC_SEL0	MTU0	ELC_SSEL33	ELC_SEL0	DSMIF3 CAP_TRG1
	ELC_SEL1	MTU3		ELC_SEL1	DSMIF3 CAP_TRG2
	ELC_SEL2	MTU4		ELC_SEL2	DSMIF3 CAP_TRG3
ELC_SSEL1	ELC_SEL0	GPT00 event A	ELC_SSEL34	ELC_SEL0	DSMIF3 CAP_TRG4
	ELC_SEL1	GPT00 event B		ELC_SEL1	DSMIF3 CAP_TRG5
	ELC_SEL2	GPT00 event C		ELC_SEL2	DSMIF3 CDCNT_INT_TRG0
ELC_SSEL2	ELC_SEL0	GPT00 event D	ELC_SSEL35	ELC_SEL0	DSMIF3 CDCNT_INT_TRG1
	ELC_SEL1	GPT00 event E		ELC_SEL1	DSMIF3 CDCNT_INT_TRG2
	ELC_SEL2	GPT00 event F		ELC_SEL2	DSMIF4 CAP_TRG0
ELC_SSEL3	ELC_SEL0	GPT01 event A	ELC_SSEL36	ELC_SEL0	DSMIF4 CAP_TRG1
	ELC_SEL1	GPT01 event B		ELC_SEL1	DSMIF4 CAP_TRG2
	ELC_SEL2	GPT01 event C		ELC_SEL2	DSMIF4 CAP_TRG3
ELC_SSEL4	ELC_SEL0	GPT01 event D	ELC_SSEL37	ELC_SEL0	DSMIF4 CAP_TRG4
	ELC_SEL1	GPT01 event E		ELC_SEL1	DSMIF4 CAP_TRG5
	ELC_SEL2	GPT01 event F		ELC_SEL2	DSMIF4 CDCNT_INT_TRG0
ELC_SSEL5	ELC_SEL0	GPT02 event A	ELC_SSEL38	ELC_SEL0	DSMIF4 CDCNT_INT_TRG1
	ELC_SEL1	GPT02 event B		ELC_SEL1	DSMIF4 CDCNT_INT_TRG2
	ELC_SEL2	GPT02 event C		ELC_SEL2	DSMIF5 CAP_TRG0
ELC_SSEL6	ELC_SEL0	GPT02 event D	ELC_SSEL39	ELC_SEL0	DSMIF5 CAP_TRG1
	ELC_SEL1	GPT02 event E		ELC_SEL1	DSMIF5 CAP_TRG2
	ELC_SEL2	GPT02 event F		ELC_SEL2	DSMIF5 CAP_TRG3
ELC_SSEL7	ELC_SEL0	GPT03 event A	ELC_SSEL40	ELC_SEL0	DSMIF5 CAP_TRG4
	ELC_SEL1	GPT03 event B		ELC_SEL1	DSMIF5 CAP_TRG5
	ELC_SEL2	GPT03 event C		ELC_SEL2	DSMIF5 CDCNT_INT_TRG0
ELC_SSEL8	ELC_SEL0	GPT03 event D	ELC_SSEL41	ELC_SEL0	DSMIF5 CDCNT_INT_TRG1
	ELC_SEL1	GPT03 event E		ELC_SEL1	DSMIF5 CDCNT_INT_TRG2
	ELC_SEL2	GPT03 event F		ELC_SEL2	DSMIF6 CAP_TRG0
ELC_SSEL9	ELC_SEL0	GPT04 event A	ELC_SSEL42	ELC_SEL0	DSMIF6 CAP_TRG1
	ELC_SEL1	GPT04 event B		ELC_SEL1	DSMIF6 CAP_TRG2
	ELC_SEL2	GPT04 event C		ELC_SEL2	DSMIF6 CAP_TRG3
ELC_SSEL10	ELC_SEL0	GPT04 event D	ELC_SSEL43	ELC_SEL0	DSMIF6 CAP_TRG4
	ELC_SEL1	GPT04 event E		ELC_SEL1	DSMIF6 CAP_TRG5
	ELC_SEL2	GPT04 event F		ELC_SEL2	DSMIF6 CDCNT_INT_TRG0

Table 16.4 Correspondence between the ELC_SSELn register and ELC destination (2 of 3)

Register name	Bit name	ELC destination	Register name	Bit name	ELC destination
ELC_SSEL11	ELC_SEL0	GPT05 event A	ELC_SSEL44	ELC_SEL0	DSMIF6 CDCNT_INT_TRG1
	ELC_SEL1	GPT05 event B		ELC_SEL1	DSMIF6 CDCNT_INT_TRG2
	ELC_SEL2	GPT05 event C		ELC_SEL2	DSMIF7 CAP_TRG0
ELC_SSEL12	ELC_SEL0	GPT05 event D	ELC_SSEL45	ELC_SEL0	DSMIF7 CAP_TRG1
	ELC_SEL1	GPT05 event E		ELC_SEL1	DSMIF7 CAP_TRG2
	ELC_SEL2	GPT05 event F		ELC_SEL2	DSMIF7 CAP_TRG3
ELC_SSEL13	ELC_SEL0	GPT06 event A	ELC_SSEL46	ELC_SEL0	DSMIF7 CAP_TRG4
	ELC_SEL1	GPT06 event B		ELC_SEL1	DSMIF7 CAP_TRG5
	ELC_SEL2	GPT06 event C		ELC_SEL2	DSMIF7 CDCNT_INT_TRG0
ELC_SSEL14	ELC_SEL0	GPT06 event D	ELC_SSEL47	ELC_SEL0	DSMIF7 CDCNT_INT_TRG1
	ELC_SEL1	GPT06 event E		ELC_SEL1	DSMIF7 CDCNT_INT_TRG2
	ELC_SEL2	GPT06 event F		ELC_SEL2	DSMIF8 CAP_TRG0
ELC_SSEL15	ELC_SEL0	GPT07 event A	ELC_SSEL48	ELC_SEL0	DSMIF8 CAP_TRG1
	ELC_SEL1	GPT07 event B		ELC_SEL1	DSMIF8 CAP_TRG2
	ELC_SEL2	GPT07 event C		ELC_SEL2	DSMIF8 CAP_TRG3
ELC_SSEL16	ELC_SEL0	GPT07 event D	ELC_SSEL49	ELC_SEL0	DSMIF8 CAP_TRG4
	ELC_SEL1	GPT07 event E		ELC_SEL1	DSMIF8 CAP_TRG5
	ELC_SEL2	GPT07 event F		ELC_SEL2	DSMIF8 CDCNT_INT_TRG0
ELC_SSEL17	ELC_SEL0	GPT08 event A	ELC_SSEL50	ELC_SEL0	DSMIF8 CDCNT_INT_TRG1
	ELC_SEL1	GPT08 event B		ELC_SEL1	DSMIF8 CDCNT_INT_TRG2
	ELC_SEL2	GPT08 event C		ELC_SEL2	DSMIF9 CAP_TRG0
ELC_SSEL18	ELC_SEL0	GPT08 event D	ELC_SSEL51	ELC_SEL0	DSMIF9 CAP_TRG1
	ELC_SEL1	GPT08 event E		ELC_SEL1	DSMIF9 CAP_TRG2
	ELC_SEL2	GPT08 event F		ELC_SEL2	DSMIF9 CAP_TRG3
ELC_SSEL19	ELC_SEL0	GPT09 event A	ELC_SSEL52	ELC_SEL0	DSMIF9 CAP_TRG4
	ELC_SEL1	GPT09 event B		ELC_SEL1	DSMIF9 CAP_TRG5
	ELC_SEL2	GPT09 event C		ELC_SEL2	DSMIF9 CDCNT_INT_TRG0
ELC_SSEL20	ELC_SEL0	GPT09 event D	ELC_SSEL53	ELC_SEL0	DSMIF9 CDCNT_INT_TRG1
	ELC_SEL1	GPT09 event E		ELC_SEL1	DSMIF9 CDCNT_INT_TRG2
	ELC_SEL2	GPT09 event F		ELC_SEL2	Encoder I/F SS trigger 0
ELC_SSEL21	ELC_SEL0	GPT09 event G	ELC_SSEL54	ELC_SEL0	Encoder I/F SS trigger 1
	ELC_SEL1	GPT09 event H		ELC_SEL1	Encoder I/F SS trigger 2
	ELC_SEL2	ADC0 A		ELC_SEL2	Encoder I/F SS trigger 3
ELC_SSEL22	ELC_SEL0	ADC0 B	ELC_SSEL55	ELC_SEL0	Encoder I/F SS trigger 4
	ELC_SEL1	ADC1 A		ELC_SEL1	Encoder I/F SS trigger 5
	ELC_SEL2	ADC1 B		ELC_SEL2	Encoder I/F SS trigger 6
ELC_SSEL23	ELC_SEL0	ADC2 A	ELC_SSEL56	ELC_SEL0	Encoder I/F SS trigger 7
	ELC_SEL1	ADC2 B		ELC_SEL1	Encoder I/F SS trigger 8
	ELC_SEL2	DSMIF0 CAP_TRG0		ELC_SEL2	Encoder I/F SS trigger 9
ELC_SSEL24	ELC_SEL0	DSMIF0 CAP_TRG1	ELC_SSEL57	ELC_SEL0	Encoder I/F SS trigger 10
	ELC_SEL1	DSMIF0 CAP_TRG2		ELC_SEL1	Encoder I/F SS trigger 11
	ELC_SEL2	DSMIF0 CAP_TRG3		ELC_SEL2	Encoder I/F SS trigger 12
ELC_SSEL25	ELC_SEL0	DSMIF0 CAP_TRG4	ELC_SSEL58	ELC_SEL0	Encoder I/F SS trigger 13
	ELC_SEL1	DSMIF0 CAP_TRG5		ELC_SEL1	Encoder I/F SS trigger 14
	ELC_SEL2	DSMIF0 CDCNT_INT_TRG0		ELC_SEL2	Encoder I/F SS trigger 15

Table 16.4 Correspondence between the ELC_SSELn register and ELC destination (3 of 3)

Register name	Bit name	ELC destination	Register name	Bit name	ELC destination
ELC_SSEL26	ELC_SEL0	DSMIF0 CDCNT_INT_TRG1	ELC_SSEL59	ELC_SEL0	ESC LATCH0
	ELC_SEL1	DSMIF0 CDCNT_INT_TRG2		ELC_SEL1	ESC LATCH1
	ELC_SEL2	DSMIF1 CAP_TRG0		ELC_SEL2	GMAC0 PTP timer capture 0
ELC_SSEL27	ELC_SEL0	DSMIF1 CAP_TRG1	ELC_SSEL60	ELC_SEL0	GMAC0 PTP timer capture 1
	ELC_SEL1	DSMIF1 CAP_TRG2		ELC_SEL1	GMAC1 PTP timer capture 0
	ELC_SEL2	DSMIF1 CAP_TRG3		ELC_SEL2	GMAC1 PTP timer capture 1
ELC_SSEL28	ELC_SEL0	DSMIF1 CAP_TRG4	ELC_SSEL61	ELC_SEL0	GMAC2 PTP timer capture 0
	ELC_SEL1	DSMIF1 CAP_TRG5		ELC_SEL1	GMAC2 PTP timer capture 1
	ELC_SEL2	DSMIF1 CDCNT_INT_TRG0		ELC_SEL2	Output port group 0
ELC_SSEL29	ELC_SEL0	DSMIF1 CDCNT_INT_TRG1	ELC_SSEL62	ELC_SEL0	Output port group 1
	ELC_SEL1	DSMIF1 CDCNT_INT_TRG2		ELC_SEL1	Input port group 0
	ELC_SEL2	DSMIF2 CAP_TRG0		ELC_SEL2	Input port group 1
ELC_SSEL30	ELC_SEL0	DSMIF2 CAP_TRG1	ELC_SSEL63	ELC_SEL0	Single port 0
	ELC_SEL1	DSMIF2 CAP_TRG2		ELC_SEL1	Single port 1
	ELC_SEL2	DSMIF2 CAP_TRG3		ELC_SEL2	Single port 2
ELC_SSEL31	ELC_SEL0	DSMIF2 CAP_TRG4	ELC_SSEL64	ELC_SEL0	Single port 3
	ELC_SEL1	DSMIF2 CAP_TRG5		ELC_SEL1	ENCOUT trigger
	ELC_SEL2	DSMIF2 CDCNT_INT_TRG0		ELC_SEL2	Reserved
ELC_SSEL32	ELC_SEL0	DSMIF2 CDCNT_INT_TRG1	—		
	ELC_SEL1	DSMIF2 CDCNT_INT_TRG2			
	ELC_SEL2	DSMIF3 CAP_TRG0			

16.3.2 ELOPA : Event Link Option Setting Register A

Base address: ELO = 0x8029_0000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	MTU3MD[1:0]		—	—	—	—	MTU0MD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	MTU0MD[1:0]	MTU0 Operation Select 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture (The MTU0.TCNT value is captured into MTU0.TGRA.) 1 1: Event is disabled.	R/W
5:2	—	These bits are read as 1. The write value should be 1.	R/W
7:6	MTU3MD[1:0]	MTU3 Operation Select 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture (The MTU3.TCNT value is captured into MTU3.TGRA.) 1 1: Event is disabled.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The ELOPA register determines the operation of MTU0 and MTU3 when an event is input. The event setting should be disabled when the ELC function is not used.

16.3.3 ELOPB : Event Link Option Setting Register B

Base address: ELO = 0x8029_0000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MTU4MD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	MTU4MD[1:0]	MTU4 Operation Select 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture (The MTU4.TCNT value is captured into MTU4.TGRA.) 1 1: Event is disabled.	R/W
7:2	—	These bits are read as 1. The write value should be 1.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The ELOPB register determines the operation of MTU4 when an event is input. The event setting should be disabled when the ELC function is not used.

16.3.4 ELC_PGRn : ELC Port Group Setting Register n (n = 1, 2)

Base address: PORT_NSR = 0x802C_0000

Offset address: 0xE00 + 0x01 × (n − 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PG7 to PG0	Port Group Setting 0: The port bit is not specified as a member of the same group. 1: The port bit is specified as a member of the same group.	R/W

The ELC_PGRn register specifies a group for I/O port bits. This register specifies each port bit in the same eight I/O ports as the member of a group. One to eight port bits can be specified as the members of the same group as required.

Table 16.5 Registers related to port groups and corresponding port numbers

Port number	Port Group Setting Register (ELC_PGRn)	Port Group Control Register (ELC_PGCn)	Port Buffer Register (ELC_PDBFn)
Port 14	ELC_PGR1 register	ELC_PGC1 register	ELC_PDBF1 register
Port 30	ELC_PGR2 register	ELC_PGC2 register	ELC_PDBF2 register

16.3.5 ELC_PGCn : ELC Port Group Control Register n (n = 1, 2)

Base address: PORT_NSR = 0x802C_0000

Offset address: 0xE02 + 0x01 × (n − 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PGCO[2:0]			—	PGCOVE	PGCI[1:0]	
Value after reset:	1	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
1:0	PGCI[1:0]	Event Output Edge Select 0 0: Event is generated on detection of the rising edge of the external input signal. 0 1: Event is generated on detection of the falling edge of the external input signal. 1 x: Event is generated on detection of both the rising and falling edges of the external input signal.	R/W
2	PGCOVE	PDBF Overwrite 0: Overwriting ELC_PDBFn register is disabled. 1: Overwriting ELC_PDBFn register is enabled.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
6:4	PGCO[2:0]	Port Group Operation Select 0 0 0: 0 is output when the event is input. 0 0 1: 1 is output when the event is input. 0 1 0: The toggled (inverted) value is output when the event is input. The target Port m Mode register should be set to 11b: Output (Output data is input to input buffer) 0 1 1: The buffer value is output when the event is input. 1 x x: Setting prohibited	R/W
7	—	This bit is read as 1. The write value should be 1.	R/W

For the output port group, the ELC_PGCn register specifies the form of outputting the signal externally via the port when the event signal is input. For the input port group, this register enables/disables overwriting of the ELC_PDBFn register and specifies the conditions of event generation (edge of the externally input signal).

16.3.6 ELC_PDBFn : ELC Port Buffer Register n (n = 1, 2)

Base address: PORT_NSR = 0x802C_0000

Offset address: 0xE04 + 0x04 × (n − 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PB7 to PB0	Port Buffer n ● In case Port is output Data is transferred from ELC_PDBFn register to Pm register when an event is input. ● In case Port is input External input data is transferred to ELC_PDBFn register when an event is input. Write access to the bit specified as a member of the input port group is invalid.	R/W

The ELC_PDBFn register is an 8-bit readable/writable register used in combination with the ELC_PGRn register.

16.3.7 ELC_PELn : ELC Port Setting Register n (n = 0 to 3)

Base address: PORT_NSR = 0x802C_0000

Offset address: 0xE0C+ 0x01 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PSM[1:0]		PSP[1:0]		PSB[2:0]		
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	PSB[2:0]	Bit Number Specification A bit number in eight I/O ports is specified.	R/W
4:3	PSP[1:0]	Port Number Specification 0 0: Setting prohibited 0 1: Port 14 (corresponding to ELC_PGR1) 1 0: Port 30 (corresponding to ELC_PGR2) 1 1: Setting prohibited	R/W
6:5	PSM[1:0]	Event Link Specification For the output port, data to be output from the port is specified. For the input port, the edge on which the event is to be output is specified. 0 0: For the output port, 0 is output when the event is input. For the input port, the event is output on detection of the rising edge. 0 1: For the output port, 1 is output when the event is input. For the input port, the event is output on detection of the falling edge. 1 x: For the output port, the toggled (inverted) value is output when the event is input. For the input port, the event is output on detection of both the rising and falling edges. The target Port m Mode register should be set to 11b: Output (Output data is input to input buffer)	R/W
7	—	This bit is read as 1. The write value should be 1.	R/W

The ELC_PELn register specifies the single port to which an event is to be linked, the port operation on the event signal input, and the conditions of event generation. In this LSI, a total of 4 bits in Port 14 or Port 30 can be specified as single ports.

16.3.8 ELC_DPTC : ELC Edge Detection Control Register

Base address: PORT_NSR = 0x802C_0000

Offset address: 0xE10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	PTC3	PTC2	PTC1	PTC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PTC3 to PTC0	Single Input Port n Edge Detection 0: Disable edge detection for Single Input Port n 1: Enable edge detection for Single Input Port n	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

The ELC_DPTC register specifies input edge detection function for single input port linked to the event.

16.3.9 ELC_ELSR2 : ELC Port Event Control Register

Base address: PORT_NSR = 0x802C_0000

Offset address: 0xE11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PES3	PES2	PES1	PES0	PEG2	PEG1	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
3:2	PEG2 to PEG1	ELC Port Buffer Register (ELC_PDBFn) write access control. When set to 1, writing to the ELC_PDBFn register via Internal peripheral bus is disabled, preventing overwriting. 0: When not use event link function of input port group n 1: When use event link function of input port group n	R/W
7:4	PES3 to PES0	Single Port n Event Link Function Enable 0: Disable Single Port event link function 1: Enable Single Port event link function	R/W

The ELC_ELSR2 register specifies input port function linked to ELC. When enabled in this register, corresponding ELC_SSELn should be enabled.

16.3.10 SLELC_PGRC : ELC_PGR/ELC_PGC Security Level Register

Base address: PORT_NSR = 0x802C_0000

Offset address: 0x1E00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ELC_PGC2_SL		—	—	—	—	—	—	ELC_PGC1_SL	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ELC_PGR2_SL		—	—	—	—	—	—	ELC_PGR1_SL	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ELC_PGR1_SL	ELC_PGR1 register Security Level	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	ELC_PGR2_SL	ELC_PGR2 register Security Level	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	ELC_PGC1_SL	ELC_PGC1 register Security Level	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	ELC_PGC2_SL	ELC_PGC2 register Security Level	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The SLELC_PGRC register is used to select the security level of the ELC_PGRn and ELC_PGCn registers. For details of security level, see section section 13.4.4.2. TrustZone Slave (TZS).

16.3.11 SLELC_PDBFn : ELC_PDBFn Security Level Register n (n = 1, 2)

Base address: PORT_NSR = 0x802C_0000

Offset address: 0x1E04 + 0x04 × (n − 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ELC_PDBF_SL	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ELC_PDBF_SL	ELC_PDBFn register Security Level	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The SLELC_PDBFn register is used to select the security level of the ELC_PDBFn register. For details of security level, see section section 13.4.4.2. TrustZone Slave (TZS).

16.3.12 SLELC_PEL : ELC_PEL Security Level Register

Base address: PORT_NSR = 0x802C_0000

Offset address: 0x1E0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ELC_PEL3_SL		—	—	—	—	—	—	ELC_PEL2_SL	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ELC_PEL1_SL		—	—	—	—	—	—	ELC_PEL0_SL	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ELC_PEL0_SL	ELC_PEL0 register Security Level	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	ELC_PEL1_SL	ELC_PEL1 register Security Level	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	ELC_PEL2_SL	ELC_PEL2 register Security Level	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	ELC_PEL3_SL	ELC_PEL3 register Security Level	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The SLELC_PEL register is used to select the security level of the ELC_PELn registers. For details of security level, see section section 13.4.4.2. TrustZone Slave (TZS).

16.3.13 SLELC_DE : ELC_DPTC/ELC_ELSR2 Security Level Register

Base address: PORT_NSR = 0x802C_0000

Offset address: 0x1E10

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ELC_ELSR2_SL[1:0]		—	—	—	—	—	—	ELC_DPTC_SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ELC_DPTC_SL[1:0]	ELC_DPTC register Security Level	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	ELC_ELSR2_SL[1:0]	ELC_ELSR2 register Security Level	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

The SLELC_DE register is used to select the security level of the ELC_DPTC and ELC_DLSR2 registers. For details of security level, see section section 13.4.4.2. TrustZone Slave (TZS).

16.4 Operation

16.4.1 Event Linkage

ELC connects events generated by various source peripheral modules to other destination modules. It allows direct interlinked operation among source and destination modules without using software.

When source events are specified in the ELC_SSELn registers, the corresponding destination peripheral modules can be operated at generation of the specified events. A single destination peripheral module can link only with a single event source. Set the ELC_SSELn register after completing the initialization of the destination peripheral module to operate by an event. Table 16.6 lists the operations of destination peripheral modules when an event signal is input. When ELC is not used in the destination peripheral module, set 0x3FF to the corresponding ELC_SSELn register.

Table 16.6 ELC destination and operation (1 of 6)

No.	ELC destination	Operation
0	MTU0	The following operations can be selected by setting the ELOPA and ELOPB registers: ● Starts counting when an event signal is input ● Restarts counting when an event signal is input ● Performs input-capture operation when an event signal is input
1	MTU3	
2	MTU4	
3	GPT00 event A	The following operations can be selected by setting GPT registers in GPT00: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
4	GPT00 event B	
5	GPT00 event C	
6	GPT00 event D	
7	GPT00 event E	
8	GPT00 event F	
9	GPT01 event A	The following operations can be selected by setting GPT registers in GPT01: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
10	GPT01 event B	
11	GPT01 event C	
12	GPT01 event D	
13	GPT01 event E	
14	GPT01 event F	

Table 16.6 ELC destination and operation (2 of 6)

No.	ELC destination	Operation
15	GPT02 event A	The following operations can be selected by setting GPT registers in GPT02: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
16	GPT02 event B	
17	GPT02 event C	
18	GPT02 event D	
19	GPT02 event E	
20	GPT02 event F	
21	GPT03 event A	The following operations can be selected by setting GPT registers in GPT03: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
22	GPT03 event B	
23	GPT03 event C	
24	GPT03 event D	
25	GPT03 event E	
26	GPT03 event F	
27	GPT04 event A	The following operations can be selected by setting GPT registers in GPT04: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
28	GPT04 event B	
29	GPT04 event C	
30	GPT04 event D	
31	GPT04 event E	
32	GPT04 event F	
33	GPT05 event A	The following operations can be selected by setting GPT registers in GPT05: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
34	GPT05 event B	
35	GPT05 event C	
36	GPT05 event D	
37	GPT05 event E	
38	GPT05 event F	
39	GPT06 event A	The following operations can be selected by setting GPT registers in GPT06: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
40	GPT06 event B	
41	GPT06 event C	
42	GPT06 event D	
43	GPT06 event E	
44	GPT06 event F	
45	GPT07 event A	The following operations can be selected by setting GPT registers in GPT07: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
46	GPT07 event B	
47	GPT07 event C	
48	GPT07 event D	
49	GPT07 event E	
50	GPT07 event F	

Table 16.6 ELC destination and operation (3 of 6)

No.	ELC destination	Operation
51	GPT08 event A	The following operations can be selected by setting GPT registers in GPT08: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
52	GPT08 event B	
53	GPT08 event C	
54	GPT08 event D	
55	GPT08 event E	
56	GPT08 event F	
57	GPT09 event A	The following operations can be selected by setting GPT registers in GPT09: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
58	GPT09 event B	
59	GPT09 event C	
60	GPT09 event D	
61	GPT09 event E	
62	GPT09 event F	
63	GPT09 event G	
64	GPT09 event H	
65	ADC0 A	Starts A/D conversion when an event signal is input
66	ADC0 B	
67	ADC1 A	
68	ADC1 B	
69	ADC2 A	
70	ADC2 B	
71	DSMIF0 CAP_TRG0	Current Data capture trigger for DSMIF0
72	DSMIF0 CAP_TRG1	
73	DSMIF0 CAP_TRG2	
74	DSMIF0 CAP_TRG3	
75	DSMIF0 CAP_TRG4	
76	DSMIF0 CAP_TRG5	
77	DSMIF0 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger for DSMIF0
78	DSMIF0 CDCNT_INT_TRG1	
79	DSMIF0 CDCNT_INT_TRG2	
80	DSMIF1 CAP_TRG0	Current Data capture trigger for DSMIF1
81	DSMIF1 CAP_TRG1	
82	DSMIF1 CAP_TRG2	
83	DSMIF1 CAP_TRG3	
84	DSMIF1 CAP_TRG4	
85	DSMIF1 CAP_TRG5	
86	DSMIF1 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger for DSMIF1
87	DSMIF1 CDCNT_INT_TRG1	
88	DSMIF1 CDCNT_INT_TRG2	

Table 16.6 ELC destination and operation (4 of 6)

No.	ELC destination	Operation
89	DSMIF2 CAP_TRG0	Current Data capture trigger for DSMIF2
90	DSMIF2 CAP_TRG1	
91	DSMIF2 CAP_TRG2	
92	DSMIF2 CAP_TRG3	
93	DSMIF2 CAP_TRG4	
94	DSMIF2 CAP_TRG5	
95	DSMIF2 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger for DSMIF2
96	DSMIF2 CDCNT_INT_TRG1	
97	DSMIF2 CDCNT_INT_TRG2	
98	DSMIF3 CAP_TRG0	Current Data capture trigger for DSMIF3
99	DSMIF3 CAP_TRG1	
100	DSMIF3 CAP_TRG2	
101	DSMIF3 CAP_TRG3	
102	DSMIF3 CAP_TRG4	
103	DSMIF3 CAP_TRG5	
104	DSMIF3 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger for DSMIF3
105	DSMIF3 CDCNT_INT_TRG1	
106	DSMIF3 CDCNT_INT_TRG2	
107	DSMIF4 CAP_TRG0	Current Data capture trigger for DSMIF4
108	DSMIF4 CAP_TRG1	
109	DSMIF4 CAP_TRG2	
110	DSMIF4 CAP_TRG3	
111	DSMIF4 CAP_TRG4	
112	DSMIF4 CAP_TRG5	
113	DSMIF4 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger for DSMIF4
114	DSMIF4 CDCNT_INT_TRG1	
115	DSMIF4 CDCNT_INT_TRG2	
116	DSMIF5 CAP_TRG0	Current Data capture trigger for DSMIF5
117	DSMIF5 CAP_TRG1	
118	DSMIF5 CAP_TRG2	
119	DSMIF5 CAP_TRG3	
120	DSMIF5 CAP_TRG4	
121	DSMIF5 CAP_TRG5	
122	DSMIF5 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger for DSMIF5
123	DSMIF5 CDCNT_INT_TRG1	
124	DSMIF5 CDCNT_INT_TRG2	

Table 16.6 ELC destination and operation (5 of 6)

No.	ELC destination	Operation
125	DSMIF6 CAP_TRG0	Current Data capture trigger for DSMIF6
126	DSMIF6 CAP_TRG1	
127	DSMIF6 CAP_TRG2	
128	DSMIF6 CAP_TRG3	
129	DSMIF6 CAP_TRG4	
130	DSMIF6 CAP_TRG5	
131	DSMIF6 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger for DSMIF6
132	DSMIF6 CDCNT_INT_TRG1	
133	DSMIF6 CDCNT_INT_TRG2	
134	DSMIF7 CAP_TRG0	Current Data capture trigger for DSMIF7
135	DSMIF7 CAP_TRG1	
136	DSMIF7 CAP_TRG2	
137	DSMIF7 CAP_TRG3	
138	DSMIF7 CAP_TRG4	
139	DSMIF7 CAP_TRG5	
140	DSMIF7 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger for DSMIF7
141	DSMIF7 CDCNT_INT_TRG1	
142	DSMIF7 CDCNT_INT_TRG2	
143	DSMIF8 CAP_TRG0	Current Data capture trigger for DSMIF8
144	DSMIF8 CAP_TRG1	
145	DSMIF8 CAP_TRG2	
146	DSMIF8 CAP_TRG3	
147	DSMIF8 CAP_TRG4	
148	DSMIF8 CAP_TRG5	
149	DSMIF8 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger for DSMIF8
150	DSMIF8 CDCNT_INT_TRG1	
151	DSMIF8 CDCNT_INT_TRG2	
152	DSMIF9 CAP_TRG0	Current Data capture trigger for DSMIF9
153	DSMIF9 CAP_TRG1	
154	DSMIF9 CAP_TRG2	
155	DSMIF9 CAP_TRG3	
156	DSMIF9 CAP_TRG4	
157	DSMIF9 CAP_TRG5	
158	DSMIF9 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger for DSMIF9
159	DSMIF9 CDCNT_INT_TRG1	
160	DSMIF9 CDCNT_INT_TRG2	

Table 16.6 ELC destination and operation (6 of 6)

No.	ELC destination	Operation
161	Encoder I/F SS trigger 0	Encoder interface subsystem trigger
162	Encoder I/F SS trigger 1	
163	Encoder I/F SS trigger 2	
164	Encoder I/F SS trigger 3	
165	Encoder I/F SS trigger 4	
166	Encoder I/F SS trigger 5	
167	Encoder I/F SS trigger 6	
168	Encoder I/F SS trigger 7	
169	Encoder I/F SS trigger 8	
170	Encoder I/F SS trigger 9	
171	Encoder I/F SS trigger 10	
172	Encoder I/F SS trigger 11	
173	Encoder I/F SS trigger 12	
174	Encoder I/F SS trigger 13	
175	Encoder I/F SS trigger 14	
176	Encoder I/F SS trigger 15	
177	ESC LATCH0	Capture current value of EtherCAT Slave Controller DC
178	ESC LATCH1	
179	GMAC0 PTP timer capture 0	Capture current value of PTP timer
180	GMAC0 PTP timer capture 1	
181	GMAC1 PTP timer capture 0	
182	GMAC1 PTP timer capture 1	
183	GMAC2 PTP timer capture 0	
184	GMAC2 PTP timer capture 1	
185	Output port group 0 (Port 14)	• Change the Pm value to the value specified in the ELC_PGCn register • Transfer ELC_PDBFn value to the Pm register
186	Output port group 1 (Port 30)	
187	Input port group 0 (Port 14)	Transfers the signal value of the external pin to the ELC_PDBFn register
188	Input port group 1 (Port 30)	
189	Single port 0	• In case single port is output Changes the Pm value to the value specified in the ELC_PELn register • In case single port is input Generates an event when edge detection on the single input port
190	Single port 1	
191	Single port 2	
192	Single port 3	
193	ENCOUT trigger	ENCOUT Trigger

16.4.2 Operation of MTU When Event is Input

The operations are performed depending on the ELOPA and ELOPB registers when an event is input.

16.4.2.1 Count Start Operation

When an event is input, the timer starts counting, which sets the count start bit[*1] in each timer control register to 1. An event that is input while the count start bit is 1 is invalid.

Note 1. Refer to the register descriptions on starting the timer in the relevant Peripheral Timer Module section.

16.4.2.2 Count Restart Operation

When an event is input, the timer counter is initialized. Since the count start bit*1 in each timer control register is retained, counting is restarted when an event signal is input while the count start bit is 1.

Note 1. Refer to the register descriptions on starting the timer in the relevant Peripheral Timer Module section.

16.4.2.3 Input Capture Operation

When an event is input, the timer performs input-capture operation.

16.4.3 Operation of GPT When Event is Input

Six event signals for each GPT00 to GPT08 and eight event signals for GPT09 are connected to every channel of each GPT unit as GPT event source A to F for GPT00 to GPT08 and A to H for GPT09. To set GPT operation when inputting an event signal, enable any of the event sources A to F for GPT00 to GPT08 and A to H for GPT09 by a corresponding bit in registers listed in Table 16.7.

Table 16.7 Operations when event is input and corresponding source select registers

Operations on event	Register symbol	Register name
Count start	GTSSR	General PWM Timer Start Source Select Register
Count stop	GTPSR	General PWM Timer Stop Source Select Register
Counter clear	GTCSR	General PWM Timer Clear Source Select Register
Up-count	GTUPSR	General PWM Timer Count-Up Source Select Register
Down-count	GTDNSR	General PWM Timer Count-Down Source Select Register
Input capture A	GTICASR	General PWM Timer Input Capture Source Select Register A
Input capture B	GTICBSR	General PWM Timer Input Capture Source Select Register B

16.4.4 Operation of A/D Converter When Event is Input

The A/D converter starts A/D conversion when the ADCSR.ADST bit*1 is set to 1.

Note 1. For details, see section 41, 12-Bit A/D converter (ADC12).

16.4.5 I/O Port Operation on Event Input and Event Generation

The I/O port operation to be performed on event input to the port and the operation causing the port to generate an event can be set.

16.4.5.1 Single Ports and Port Groups

There are two event link modes:

- Event link to single ports
- Event link to port groups

In the former mode, events can be connected to eight I/O ports. In the latter mode, events can be connected to port groups consisting of any two or more bits in the same eight I/O ports.

A single port can be set by specifying any bit in the I/O port*1 to which an event can be connected using the ELC_PELn register. A port group can be set by specifying any one or more bits in the I/O port*1 to which an event can be connected using the ELC_PGRn register. One input port group and one output port group can be set in the same I/O port.

If the I/O port bit is specified as both a single port and a member of a port group, both functions are enabled when the relevant port is input, whereas only the port group function is enabled when the relevant port is output.

Set the PMm register to select the direction of the I/O ports.

Note 1. Port 14 and Port 30

16.4.5.2 Single Input Port Operation on Event Generation

A single input port which is specified by the PMm register generates an event when the signal value of the external pin connected to the relevant port changes. The event generation condition is specified using the ELC_PELn register. An example of event linking operation by a single input port is shown as [1] in Figure 16.2.

16.4.5.3 Single Output Ports Operation on Event Input

When an event is input to a single output port which is specified by the PMm register, the signal of the external pin connected to the relevant port changes according to the settings of the ELC_PELn register. This changes the signal value of the external pin connected to the relevant port. An example of event linking operation by a single output port is shown as [2] in Figure 16.2.

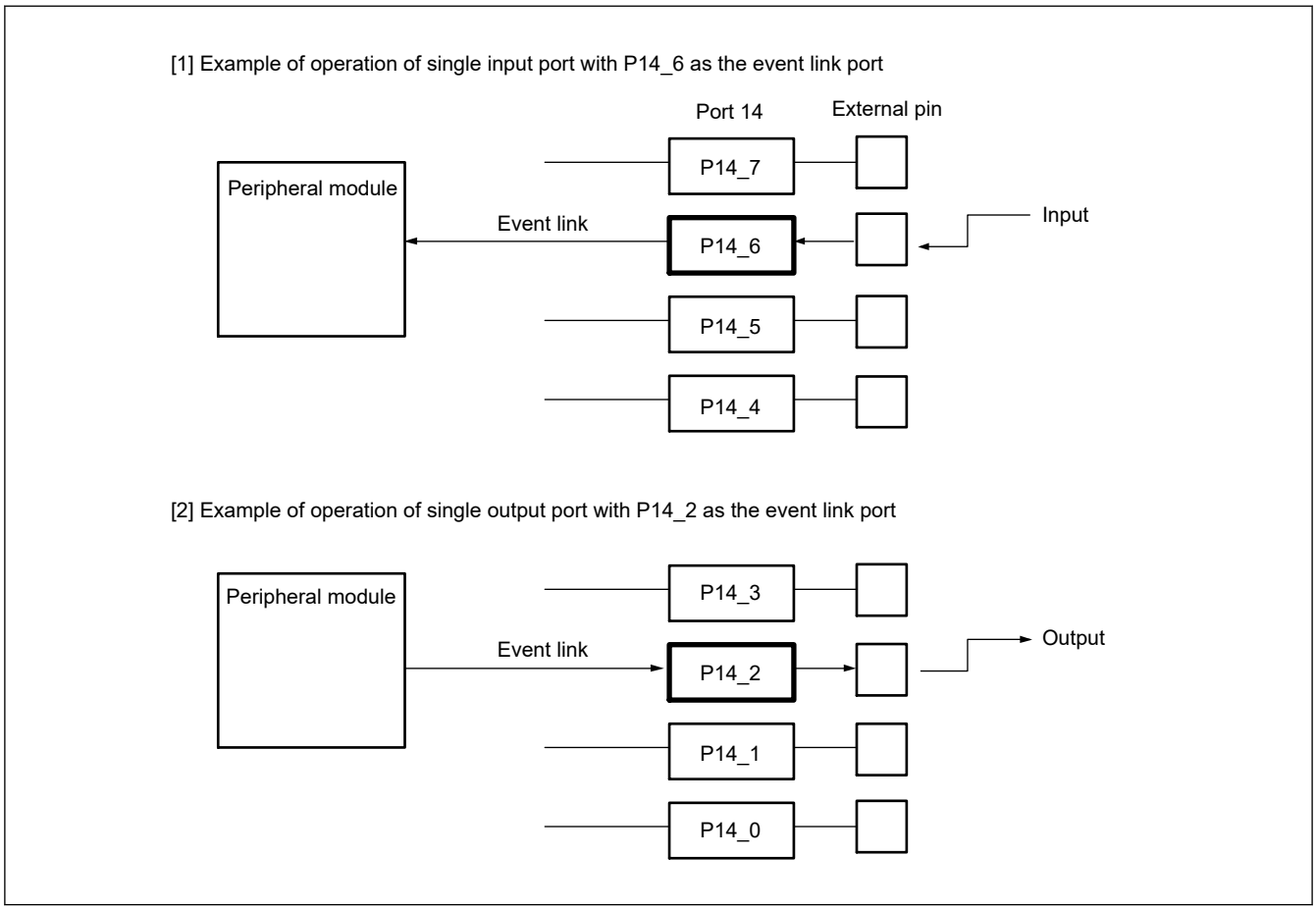


Figure 16.2 Event linkage related to single ports (port 14)

16.4.5.4 Input Port Group Operation on Event Generation

An input port group which is specified by the PMm register generates an event when the signal value of any one of the external pins connected to the relevant port group changes. The event generation condition is specified using the PGCIn bit of the ELC_PGCn register.

16.4.5.5 Input Port Group Operation on Event Input

When an event is input to an input port group, the signal value of the external pin on event input is transferred to the ELC_PDBFn register. In this case, only the values of the bits specified as members of the input port group are transferred. If another event is input to the input port group in this state, operations are performed depending on the ELC_PGCn.PGCOVE bit setting as described below. An example of operation is shown in Figure 16.3.

- ELC_PGCn.PGCOVE = 0 (overwriting is disabled)

If the value that was transferred to the ELC_PDBFn register on the previous event input has already been read by the CPU, the signal value of the external pin is transferred to the ELC_PDBFn register. If not read, the signal value of the external pin is not transferred and the input event is invalid.

- ELC_PGCn.PGCOVE = 1 (overwriting is enabled)
  When another event is input to an input port group, the signal value of the external pin is transferred to the ELC_PDBFn register.

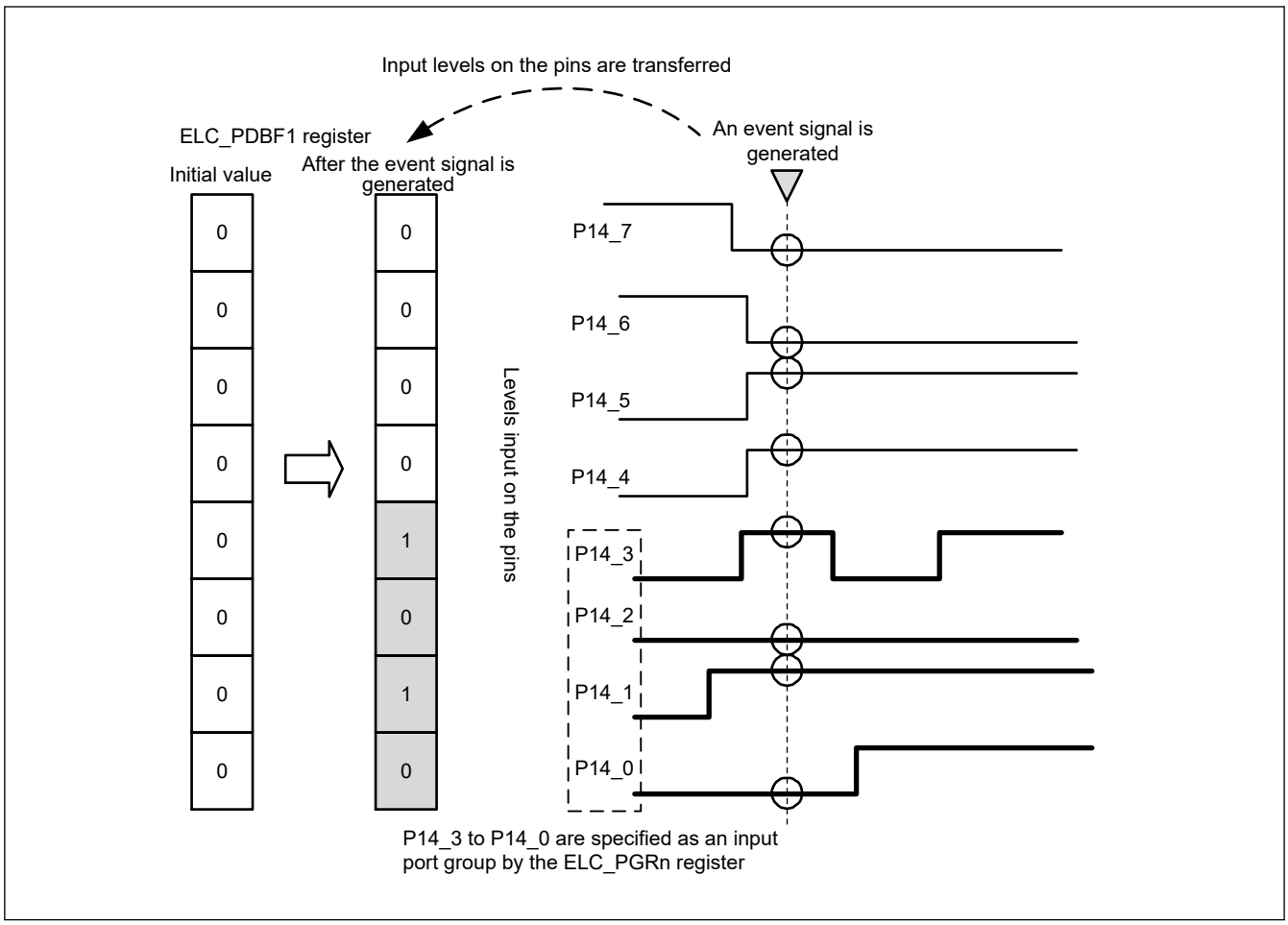


Figure 16.3 Input port group operation on event input (port 14)

16.4.5.6 Output Port Group Operation on Event Input

If an event is input to an output port group while the ELC_PGCn.PGCOn bit is 000b, 001b, or 010b, the Pm register value is changed to the value which was specified in the ELC_PGCn register.

If an event is input to an output port group while the ELC_PGCn.PGCOn bit is 011b, the ELC_PDBFn value is transferred to the Pm register of the port which was specified in the ELC_PGRn register. Example of operation of the output port group on an event input (when ELC_PGCn.PGCOn = 011b) is shown in Figure 16.4.

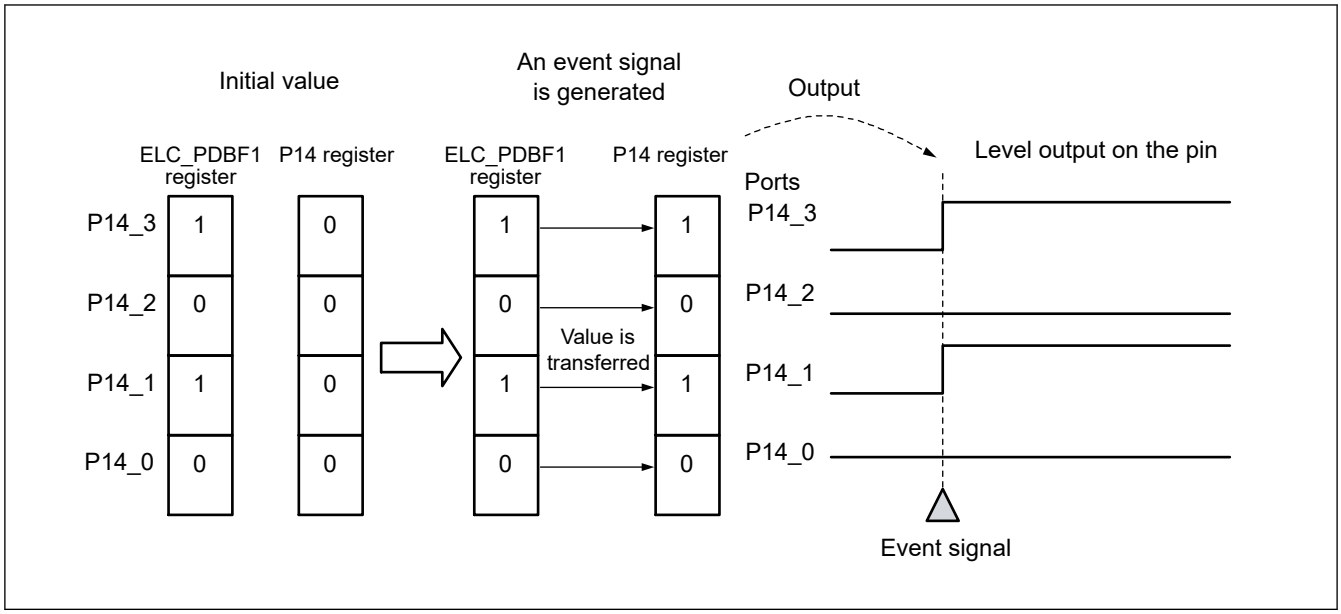


Figure 16.4 Event linkage operation of output port group (port 14)

16.4.5.7 Restrictions on Writing to Pm and ELC_PDBFn Registers by a CPU

For event linkage through the I/O ports, following restrictions apply when a CPU writes the Pm and ELC_PDBFn registers.

- If port bits are specified as members of the input port group, write access to the relevant bits in the ELC_PDBFn register is invalid.
- If port bits are specified as members of the output port group, write access to the relevant bits in the Pm register is invalid.
- If a port bit is specified as a single output port and the event linkage is set (by the ELC_SSELn register) for the port, write access to the relevant bit in the Pm register is invalid.

16.4.6 Example of Procedure for Linking Events

The following procedure describes the steps for linking events.

1. Set the operation of the module to which an event is to be linked.
2. If events are linked to ports, set the registers associated with the ports as follows:
   - I/O port setting
     Pm: Set the initial values of the output ports.
     PMm: Set the I/O direction of the ports.
   - ELC setting
     ELC_PGRn: If ports are used as a port group, set the ports (in bit units) to be grouped.
     ELC_PGCn: Set the operation of the port group.
     ELC_PELn: If ports are used as single ports, set the ports, the operation of the ports when an event is input, and the condition when an event is generated.
     ELC_DPTC: If pots are used as single input port edge detection, set the ports.
     ELC_ELSR2: If ports are used as input port groups, set the PEGn bits. If ports are used as single ports, set the PESn bits.
3. For the ELC_SSELn register associated with the module to which an event signal is to be linked, set the number of the event signal.
4. If events are to be linked to timer modules, set the ELOPA and ELOPB registers corresponding to the timers as required.
5. Set the operation of the module from which an event is output, and activate the module. This allows the event output from the module to start the module to which an event is linked as preset.
6. To stop event linkage of independent modules, set 0x3FF to ELC_SELx bits in the ELC_SSEL register corresponding to the modules.

Note: Setting the ELC_PDBFn register
- Output port groups
  Set the ELC_PGCn register before setting the ELC_PDBFn register. The value of the ELC_PGCn register can be modified if this precedes an event trigger which causes the value of the Pm register to change.
- Input port groups
  Setting the ELC_PDBFn is not required. However, since the value after a reset is 0x00, if the ELC_PDBFn register is used to confirm changes from H to L due to an event input, set the ELC_PDBFn bit for the pin for which you want to confirm this to be 1.

16.5 Usage Notes

16.5.1 Setting Clocks

To link events, it is necessary for the ELC and the related modules to be enabled. The modules cannot operate if the related modules are in the module-stop state or if the low power consumption mode causes the modules to stop (all-module stop mode).

17. I/O Ports

17.1 Overview

The I/O port pins operate as general I/O port pins, I/O pins for peripheral modules, or interrupt input pins. All pins are set to non-use immediately after a reset (Hi-Z input protection) excluding P08_1 to P08_5 which are shared with JTAG pins and reset output pin. The pin functions are switched by register settings. The I/O ports and peripheral modules for each pin are specified in the associated registers.

The configuration of the I/O ports differs depending on the package.

Table 17.1 shows the specifications of I/O ports.

Table 17.1 Specifications of I/O ports for 729/576-pin FCBGA (1 of 2)

	Package		Package	
Port	729-pin FCBGA	Number of pins	576-pin FCBGA	Number of pins
Port 00	P00_0 to P00_7	8	P00_0 to P00_4	5
Port 01	P01_0 to P01_7	8	P01_0 to P01_7	8
Port 02	P02_0 to P02_7	8	P02_0 to P02_7	8
Port 03	P03_0 to P03_7	8	P03_0 to P03_4	5
Port 04	P04_0 to P04_7	8	—	0
Port 05	P05_0 to P05_7	8	P05_1 to P05_7	7
Port 06	P06_0 to P06_7	8	P06_0 to P06_7	8
Port 07	P07_0 to P07_7	8	—	0
Port 08	P08_0 to P08_7	8	P08_1 to P08_6	6
Port 09	P09_0 to P09_7	8	P09_4 to P09_7	4
Port 10	P10_0 to P10_7	8	P10_0 to P10_7	8
Port 11	P11_0 to P11_7	8	P11_0	1
Port 12	P12_0 to P12_7	8	P12_0 to P12_7	8
Port 13	P13_0 to P13_7	8	P13_0 to P13_7	8
Port 14	P14_0 to P14_7	8	P14_0 to P14_7	8
Port 15	P15_0 to P15_7	8	—	0
Port 16	P16_0 to P16_7	8	P16_5 to P16_7	3
Port 17	P17_0 to P17_7	8	P17_0 to P17_7	8
Port 18	P18_0 to P18_7	8	P18_0 to P18_7	8
Port 19	P19_0 to P19_7	8	—	0
Port 20	P20_0 to P20_7	8	P20_0 to P20_7	8
Port 21	P21_0 to P21_7	8	P21_0 to P21_7	8
Port 22	P22_0 to P22_7	8	P22_0 to P22_7	8
Port 23	P23_0 to P23_7	8	P23_0	1
Port 24	P24_0 to P24_7	8	P24_5 to P24_7	3
Port 25	P25_0 to P25_7	8	P25_0 to P25_7	8
Port 26	P26_0 to P26_7	8	P26_0 to P26_7	8
Port 27	P27_0 to P27_7	8	P27_0 to P27_6	7
Port 28	P28_0 to P28_7	8	—	0
Port 29	P29_0 to P29_7	8	P29_1 to P29_7	7
Port 30	P30_0 to P30_7	8	P30_0 to P30_7	8
Port 31	P31_0 to P31_7	8	P31_0 to P31_6	7

Table 17.1 Specifications of I/O ports for 729/576-pin FCBGA (2 of 2)

Port	Package		Package	
	729-pin FCBGA	Number of pins	576-pin FCBGA	Number of pins
Port 32	P32_0 to P32_7	8	—	0
Port 33	P33_0 to P33_7	8	P33_2 to P33_7	6
Port 34	P34_0 to P34_7	8	P34_0 to P34_6	7
Port 35	P35_0 to P35_6	7	—	0

17.2 I/O Port Configuration

Figure 17.1 shows a connection diagram for the I/O port registers.

This figure shows the logic for reference, not the actual circuit.

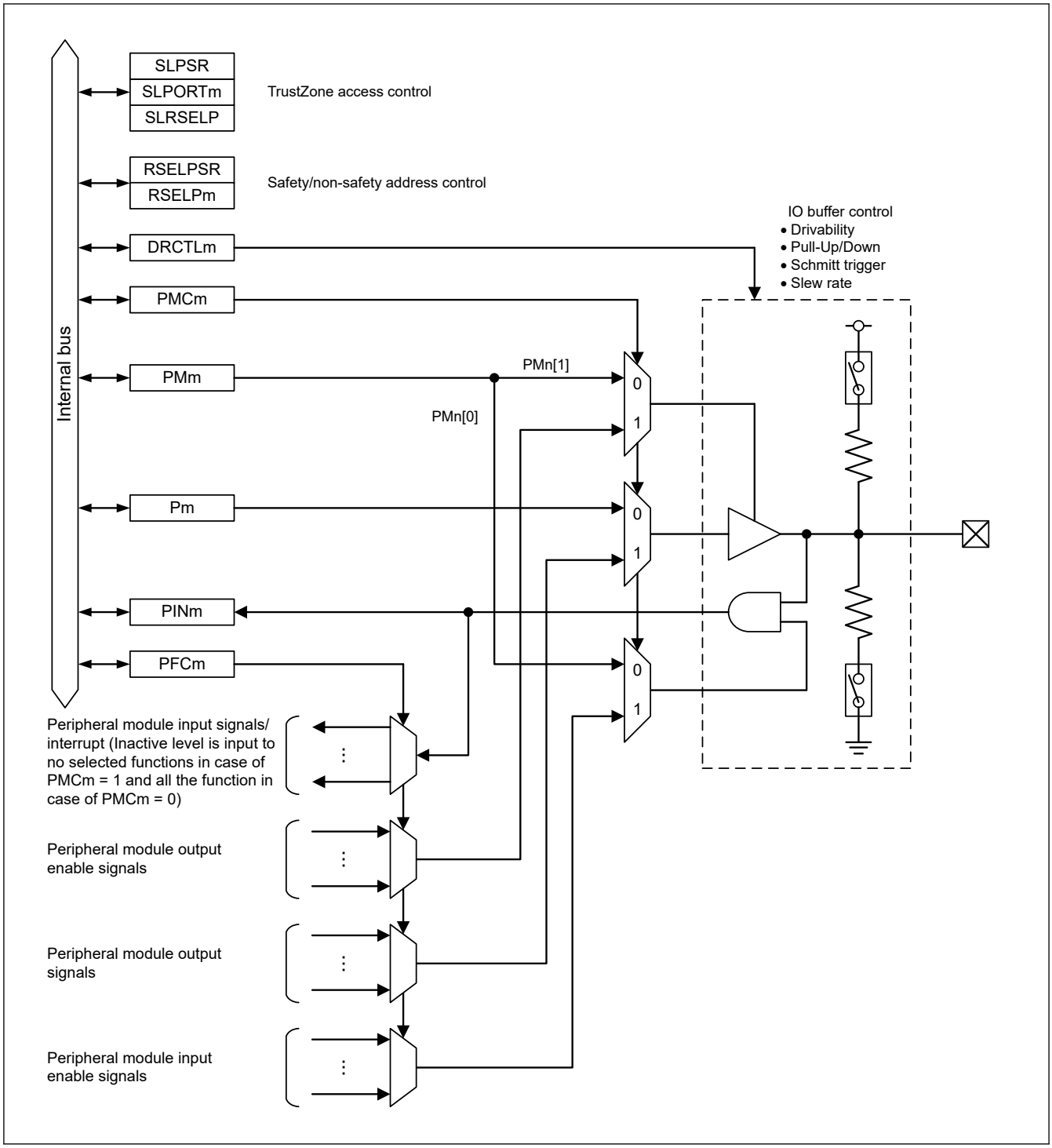


Figure 17.1 Basic circuit configuration of port

The pin assignments for each module/function are shown below. Please see section 17.5. Multiplexed Pin Configurations for detail of pin multiplex mapping.

Table 17.2 List of multiplexed pin configurations (1 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Clock	—	CKIO	P08_6	✓	✓
		ETH0_REFCLK	P21_7	✓	✓
		ETH1_REFCLK	P26_4	✓	✓
		ETH2_REFCLK	P31_0	✓	✓
		ETH3_REFCLK	P34_6	✓	✓
		RMII0_REFCLK	P21_7	✓	✓
		RMII1_REFCLK	P26_4	✓	✓
		RMII2_REFCLK	P31_0	✓	✓
		RMII3_REFCLK	P34_6	✓	✓
Operating mode	—	MD0	P24_6	✓	✓
		MD1	P24_7	✓	✓
		MD2	P25_0	✓	✓
		MDD	P06_6	✓	✓
		MDV	P20_1	✓	✓
		MDW0	P25_1	✓	✓
		MDW1	P25_2	✓	✓
System Control	—	RSTOUT#	P08_5	✓	✓
Debugging interface	—	TCK	P08_3	✓	✓
		TDI	P08_2	✓	✓
		TDO	P08_4	✓	✓
		TMS	P08_1	✓	✓
Bus State Controller	—	A0	P08_7	✓	—
			P14_0	✓	✓
		A1	P10_4	✓	✓
		A2	P10_5	✓	✓
		A3	P10_6	✓	✓
		A4	P10_7	✓	✓
		A5	P11_0	✓	✓
		A6	P17_4	✓	✓
		A7	P17_5	✓	✓
		A8	P18_0	✓	✓
		A9	P18_1	✓	✓
		A10	P18_2	✓	✓
		A11	P18_3	✓	✓
		A12	P18_4	✓	✓
		A13	P18_5	✓	✓
		A14	P18_6	✓	✓
		A15	P18_7	✓	✓
		A16	P31_6	✓	✓
			P33_2	✓	✓

Table 17.2 List of multiplexed pin configurations (2 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Bus State Controller	—	A17	P23_0	✓	✓
			P33_3	✓	✓
		A18	P22_7	✓	✓
			P33_4	✓	✓
		A19	P22_6	✓	✓
			P33_5	✓	✓
		A20	P22_5	✓	✓
			P33_6	✓	✓
		A21	P22_4	✓	✓
			P33_7	✓	✓
		A22	P22_3	✓	✓
			P34_0	✓	✓
		A23	P22_2	✓	✓
			P34_1	✓	✓
		A24	P34_2	✓	✓
		A25	P34_3	✓	✓
		BS#	P14_2	✓	✓
		CS0#	P10_2	✓	✓
		CS2#	P26_6	✓	✓
			P34_4	✓	✓
		CS3#	P26_7	✓	✓
			P34_5	✓	✓
		CS5#	P27_0	✓	✓
			P34_6	✓	✓
		D0	P00_0	✓	✓
		D1	P00_1	✓	✓
		D2	P00_2	✓	✓
		D3	P00_3	✓	✓
		D4	P00_4	✓	✓
		D5	P02_5	✓	✓
		D6	P02_6	✓	✓
		D7	P02_7	✓	✓
		D8	P03_0	✓	✓
		D9	P03_1	✓	✓
		D10	P03_2	✓	✓
		D11	P03_3	✓	✓
		D12	P03_4	✓	✓
		D13	P09_4	✓	✓
		D14	P09_5	✓	✓
		D15	P09_6	✓	✓

Table 17.2 List of multiplexed pin configurations (3 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Bus State Controller	—	D16	P12_0	✓	✓
		D17	P12_1	✓	✓
		D18	P12_2	✓	✓
		D19	P12_3	✓	✓
		D20	P12_4	✓	✓
		D21	P12_5	✓	✓
		D22	P12_6	✓	✓
		D23	P12_7	✓	✓
		D24	P13_0	✓	✓
		D25	P13_1	✓	✓
		D26	P13_2	✓	✓
		D27	P13_3	✓	✓
		D28	P13_4	✓	✓
		D29	P13_5	✓	✓
		D30	P13_6	✓	✓
		D31	P13_7	✓	✓
		RD#	P10_3	✓	✓
		RD/WR#	P14_1	✓	✓
		WAIT#	P10_1	✓	✓
		WE0#	P09_7	✓	✓
		WE1#	P10_0	✓	✓
		WE2#	P17_6	✓	✓
		WE3#/AH#	P17_7	✓	✓
DMA Controller	—	DACK	P14_4	✓	✓
			P17_5	✓	✓
			P31_5	✓	✓
			P35_0	✓	—
		DREQ	P14_3	✓	✓
			P17_4	✓	✓
			P31_4	✓	✓
			P34_7	✓	—
		TEND	P14_5	✓	✓
			P18_0	✓	✓
			P31_6	✓	✓
			P35_1	✓	—
Interrupt	—	SEI	P00_0	✓	✓
			P04_5	✓	—
			P08_6	✓	✓
			P18_2	✓	✓
			P26_6	✓	✓

Table 17.2 List of multiplexed pin configurations (4 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Interrupt	—	IRQ0	P00_1	✓	✓
			P04_6	✓	—
			P08_7	✓	—
			P10_6	✓	✓
			P15_5	✓	—
			P18_3	✓	✓
			P26_7	✓	✓
		IRQ1	P00_2	✓	✓
			P04_7	✓	—
			P10_2	✓	✓
			P12_4	✓	✓
			P15_6	✓	—
			P18_4	✓	✓
			P27_0	✓	✓
		IRQ2	P00_3	✓	✓
			P05_0	✓	—
			P10_3	✓	✓
			P12_7	✓	✓
			P16_0	✓	—
			P18_5	✓	✓
			P27_1	✓	✓
		IRQ3	P00_4	✓	✓
			P05_1	✓	✓
			P10_4	✓	✓
			P13_2	✓	✓
			P18_6	✓	✓
			P27_2	✓	✓
		IRQ4	P00_6	✓	—
			P05_2	✓	✓
			P10_0	✓	✓
			P11_1	✓	—
			P13_5	✓	✓
			P18_7	✓	✓
			P27_7	✓	—
		IRQ5	P00_7	✓	—
			P05_3	✓	✓
			P11_2	✓	—
			P14_0	✓	✓
			P22_3	✓	✓
			P28_0	✓	—

Table 17.2 List of multiplexed pin configurations (5 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Interrupt	—	IRQ6	P01_0	✓	✓
			P05_4	✓	✓
			P11_3	✓	—
			P14_3	✓	✓
			P22_4	✓	✓
			P28_1	✓	—
		IRQ7	P02_0	✓	✓
			P10_1	✓	✓
			P11_4	✓	—
			P18_0	✓	✓
			P22_5	✓	✓
			P28_2	✓	—
		IRQ8	P02_1	✓	✓
			P06_2	✓	✓
			P08_5	✓	✓
			P14_6	✓	✓
			P22_6	✓	✓
			P29_4	✓	✓
		IRQ9	P02_2	✓	✓
			P06_3	✓	✓
			P10_7	✓	✓
			P14_7	✓	✓
			P22_7	✓	✓
			P29_5	✓	✓
		IRQ10	P02_3	✓	✓
			P06_4	✓	✓
			P16_3	✓	—
			P23_0	✓	✓
			P30_2	✓	✓
		IRQ11	P02_4	✓	✓
			P06_5	✓	✓
			P16_4	✓	—
			P22_0	✓	✓
			P24_0	✓	—
			P30_3	✓	✓
		IRQ12	P03_2	✓	✓
			P06_7	✓	✓
			P17_0	✓	✓
			P24_1	✓	—
			P26_5	✓	✓
			P33_3	✓	✓

Table 17.2 List of multiplexed pin configurations (6 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Interrupt	—	IRQ13	P03_3	✓	✓
			P07_0	✓	—
			P11_0	✓	✓
			P17_1	✓	✓
			P24_2	✓	—
			P31_1	✓	✓
			P33_4	✓	✓
		IRQ14	P03_4	✓	✓
			P07_1	✓	—
			P13_7	✓	✓
			P17_2	✓	✓
			P24_3	✓	—
			P30_7	✓	✓
			P33_5	✓	✓
			P34_7	✓	—
		IRQ15	P03_5	✓	—
			P07_2	✓	—
			P17_3	✓	✓
			P18_1	✓	✓
			P24_4	✓	—
			P33_6	✓	✓
Multi-Function Timer Pulse Unit	MTU	MTCLKA	P00_6	✓	—
			P03_3	✓	✓
		MTCLKB	P00_7	✓	—
			P03_4	✓	✓
		MTCLKC	P02_0	✓	✓
			P04_5	✓	—
			P10_2	✓	✓
		MTCLKD	P02_1	✓	✓
			P04_6	✓	—
			P10_3	✓	✓
	MTU0	MTIOC0A	P01_6	✓	✓
			P04_7	✓	—
			P10_5	✓	✓
		MTIOC0B	P01_7	✓	✓
			P05_0	✓	—
			P10_6	✓	✓
		MTIOC0C	P02_0	✓	✓
			P04_5	✓	—
		MTIOC0D	P02_1	✓	✓
			P04_6	✓	—

Table 17.2 List of multiplexed pin configurations (7 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Multi-Function Timer Pulse Unit	MTU1	MTIOC1A	P01_0	✓	✓
			P02_2	✓	✓
			P03_2	✓	✓
			P03_6	✓	—
			P10_4	✓	✓
			P27_5	✓	✓
		MTIOC1B	P00_7	✓	—
			P02_3	✓	✓
			P03_1	✓	✓
			P03_7	✓	—
			P10_5	✓	✓
			P27_6	✓	✓
	MTU2	MTIOC2A	P10_2	✓	✓
			P27_3	✓	✓
		MTIOC2B	P10_3	✓	✓
			P27_4	✓	✓
	MTU3	MTIOC3A	P01_0	✓	✓
			P03_5	✓	—
		MTIOC3B	P00_0	✓	✓
			P02_5	✓	✓
		MTIOC3C	P01_1	✓	✓
			P03_6	✓	—
		MTIOC3D	P00_1	✓	✓
			P02_6	✓	✓
	MTU4	MTIOC4A	P00_2	✓	✓
			P02_7	✓	✓
		MTIOC4B	P00_4	✓	✓
			P03_1	✓	✓
		MTIOC4C	P00_3	✓	✓
			P03_0	✓	✓
		MTIOC4D	P00_5	✓	—
			P03_2	✓	✓
	MTU5	MTIC5U	P01_3	✓	✓
			P02_7	✓	✓
			P10_7	✓	✓
		MTIC5V	P01_4	✓	✓
			P03_0	✓	✓
			P12_0	✓	✓
		MTIC5W	P01_5	✓	✓
			P03_5	✓	—
			P12_1	✓	✓

Table 17.2 List of multiplexed pin configurations (8 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Multi-Function Timer Pulse Unit	MTU6	MTIOC6A	P02_2	✓	✓
			P04_7	✓	—
		MTIOC6B	P01_2	✓	✓
			P03_7	✓	—
			P09_4	✓	✓
		MTIOC6C	P02_3	✓	✓
			P05_0	✓	—
		MTIOC6D	P01_3	✓	✓
			P04_0	✓	—
			P09_5	✓	✓
	MTU7	MTIOC7A	P01_4	✓	✓
			P04_1	✓	—
			P09_6	✓	✓
		MTIOC7B	P01_6	✓	✓
			P04_3	✓	—
			P10_0	✓	✓
		MTIOC7C	P01_5	✓	✓
			P04_2	✓	—
			P09_7	✓	✓
		MTIOC7D	P01_7	✓	✓
			P04_4	✓	—
			P10_1	✓	✓
	MTU8	MTIOC8A	P01_1	✓	✓
			P02_5	✓	✓
		MTIOC8B	P01_2	✓	✓
			P02_6	✓	✓
		MTIOC8C	P00_5	✓	—
			P03_3	✓	✓
		MTIOC8D	P00_6	✓	—
			P03_4	✓	✓
Port Output Enable 3	POE3	POE0#	P02_4	✓	✓
			P14_3	✓	✓
			P31_2	✓	✓
		POE4#	P06_7	✓	✓
			P14_4	✓	✓
			P31_3	✓	✓
		POE8#	P07_1	✓	—
			P14_5	✓	✓
			P31_4	✓	✓

Table 17.2 List of multiplexed pin configurations (9 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Port Output Enable 3	POE3	POE10#	P07_2	✓	—
			P14_6	✓	✓
			P31_5	✓	✓
		POE11#	P07_3	✓	—
			P14_7	✓	✓
			P31_6	✓	✓
General PWM Timer	GPT00	GTADSM00_0	P17_4	✓	✓
			P33_2	✓	✓
		GTADSM00_1	P17_5	✓	✓
			P33_3	✓	✓
		GTIOC00_0A	P00_0	✓	✓
		GTIOC00_0B	P00_1	✓	✓
		GTIOC00_1A	P00_2	✓	✓
		GTIOC00_1B	P00_3	✓	✓
		GTIOC00_2A	P00_4	✓	✓
		GTIOC00_2B	P00_5	✓	—
			P01_0	✓	✓
		GTIOC00_3A	P00_6	✓	—
			P10_7	✓	✓
		GTIOC00_3B	P00_7	✓	—
			P11_0	✓	✓
		GTIOC00_4A	P01_0	✓	✓
		GTIOC00_4B	P01_1	✓	✓
	GPT01	GTADSM01_0	P17_6	✓	✓
			P33_4	✓	✓
		GTADSM01_1	P17_7	✓	✓
			P33_5	✓	✓
		GTIOC01_0A	P01_2	✓	✓
			P32_2	✓	—
		GTIOC01_0B	P01_3	✓	✓
			P32_3	✓	—
		GTIOC01_1A	P01_4	✓	✓
			P32_4	✓	—
		GTIOC01_1B	P01_5	✓	✓
			P32_5	✓	—
		GTIOC01_2A	P01_6	✓	✓
			P32_6	✓	—
		GTIOC01_2B	P01_7	✓	✓
			P32_7	✓	—
		GTIOC01_3A	P02_0	✓	✓
			P12_5	✓	✓

Table 17.2 List of multiplexed pin configurations (10 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
General PWM Timer	GPT01	GTIOC01_3B	P02_1	✓	✓
			P12_6	✓	✓
		GTIOC01_4A	P02_2	✓	✓
		GTIOC01_4B	P02_3	✓	✓
	GPT02	GTADSM02_0	P18_0	✓	✓
			P33_6	✓	✓
		GTADSM02_1	P18_1	✓	✓
			P33_7	✓	✓
		GTIOC02_0A	P02_5	✓	✓
			P27_1	✓	✓
		GTIOC02_0B	P02_6	✓	✓
			P27_2	✓	✓
		GTIOC02_1A	P02_7	✓	✓
			P27_3	✓	✓
		GTIOC02_1B	P03_0	✓	✓
			P27_4	✓	✓
		GTIOC02_2A	P03_1	✓	✓
			P27_5	✓	✓
		GTIOC02_2B	P03_2	✓	✓
			P27_6	✓	✓
		GTIOC02_3A	P03_3	✓	✓
			P13_0	✓	✓
		GTIOC02_3B	P03_4	✓	✓
			P13_1	✓	✓
		GTIOC02_4A	P03_5	✓	—
		GTIOC02_4B	P03_6	✓	—
	GPT03	GTADSM03_0	P18_2	✓	✓
			P34_0	✓	✓
		GTADSM03_1	P18_3	✓	✓
			P34_1	✓	✓
		GTIOC03_0A	P03_7	✓	—
			P16_5	✓	✓
			P34_0	✓	✓
		GTIOC03_0B	P04_0	✓	—
			P16_6	✓	✓
			P34_1	✓	✓
		GTIOC03_1A	P04_1	✓	—
			P16_7	✓	✓
			P34_2	✓	✓

Table 17.2 List of multiplexed pin configurations (11 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
General PWM Timer	GPT03	GTIOC03_1B	P04_2	✓	—
			P17_0	✓	✓
			P34_3	✓	✓
		GTIOC03_2A	P04_3	✓	—
			P17_1	✓	✓
			P34_4	✓	✓
		GTIOC03_2B	P04_4	✓	—
			P17_2	✓	✓
			P34_5	✓	✓
		GTIOC03_3A	P04_5	✓	—
			P13_3	✓	✓
		GTIOC03_3B	P04_6	✓	—
			P13_4	✓	✓
		GTIOC03_4A	P04_7	✓	—
		GTIOC03_4B	P05_0	✓	—
	GPT04	GTADSM04_0	P18_4	✓	✓
			P34_2	✓	✓
		GTADSM04_1	P18_5	✓	✓
			P34_3	✓	✓
		GTIOC04_0A	P01_2	✓	✓
			P09_4	✓	✓
		GTIOC04_0B	P01_3	✓	✓
			P09_5	✓	✓
		GTIOC04_1A	P01_4	✓	✓
			P09_6	✓	✓
		GTIOC04_1B	P01_5	✓	✓
			P09_7	✓	✓
		GTIOC04_2A	P01_6	✓	✓
			P10_0	✓	✓
		GTIOC04_2B	P01_7	✓	✓
			P10_1	✓	✓
		GTIOC04_3A	P10_2	✓	✓
			P13_6	✓	✓
		GTIOC04_3B	P10_3	✓	✓
			P13_7	✓	✓
		GTIOC04_4A	P10_4	✓	✓
		GTIOC04_4B	P10_5	✓	✓
	GPT05	GTADSM05_0	P18_6	✓	✓
			P34_4	✓	✓
		GTADSM05_1	P18_7	✓	✓
			P34_5	✓	✓

Table 17.2 List of multiplexed pin configurations (12 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
General PWM Timer	GPT05	GTIOC05_0A	P10_6	✓	✓
			P11_6	✓	—
		GTIOC05_0B	P10_7	✓	✓
			P11_7	✓	—
		GTIOC05_1A	P12_0	✓	✓
		GTIOC05_1B	P12_1	✓	✓
		GTIOC05_2A	P12_2	✓	✓
		GTIOC05_2B	P12_3	✓	✓
		GTIOC05_3A	P12_4	✓	✓
			P14_1	✓	✓
		GTIOC05_3B	P12_5	✓	✓
			P14_2	✓	✓
		GTIOC05_4A	P12_6	✓	✓
		GTIOC05_4B	P12_7	✓	✓
	GPT06	GTADSM06_0	P02_5	✓	✓
			P34_7	✓	—
		GTADSM06_1	P02_6	✓	✓
			P35_0	✓	—
		GTIOC06_0A	P14_1	✓	✓
			P22_7	✓	✓
		GTIOC06_0B	P14_2	✓	✓
			P23_0	✓	✓
		GTIOC06_1A	P14_3	✓	✓
			P23_1	✓	—
		GTIOC06_1B	P14_4	✓	✓
			P23_2	✓	—
		GTIOC06_2A	P14_5	✓	✓
			P23_3	✓	—
		GTIOC06_2B	P14_6	✓	✓
			P23_4	✓	—
		GTIOC06_3A	P13_5	✓	✓
			P14_4	✓	✓
		GTIOC06_3B	P13_6	✓	✓
			P14_5	✓	✓
		GTIOC06_4A	P13_7	✓	✓
		GTIOC06_4B	P14_0	✓	✓
	GPT07	GTADSM07_0	P02_7	✓	✓
			P35_1	✓	—
		GTADSM07_1	P03_0	✓	✓
			P35_2	✓	—
		GTIOC07_0A	P19_0	✓	—

Table 17.2 List of multiplexed pin configurations (13 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
General PWM Timer	GPT07	GTIOC07_0B	P19_1	✓	—
		GTIOC07_1A	P19_2	✓	—
		GTIOC07_1B	P19_3	✓	—
		GTIOC07_2A	P19_4	✓	—
		GTIOC07_2B	P19_5	✓	—
		GTIOC07_3A	P18_1	✓	✓
			P18_4	✓	✓
		GTIOC07_3B	P18_2	✓	✓
			P18_5	✓	✓
		GTIOC07_4A	P18_6	✓	✓
		GTIOC07_4B	P18_7	✓	✓
	GPT08	GTADSM08_0	P03_1	✓	✓
			P35_3	✓	—
		GTADSM08_1	P03_2	✓	✓
			P35_4	✓	—
		GTIOC08_0A	P27_7	✓	—
		GTIOC08_0B	P28_0	✓	—
		GTIOC08_1A	P28_1	✓	—
		GTIOC08_1B	P28_2	✓	—
		GTIOC08_2A	P28_3	✓	—
		GTIOC08_2B	P28_4	✓	—
		GTIOC08_3A	P08_6	✓	✓
			P27_3	✓	✓
		GTIOC08_3B	P08_7	✓	—
			P27_4	✓	✓
		GTIOC08_4A	P27_5	✓	✓
		GTIOC08_4B	P27_6	✓	✓
	GPT09	GTADSM09_0	P03_3	✓	✓
			P35_5	✓	—
		GTADSM09_1	P03_4	✓	✓
			P35_6	✓	—
		GTIOC09_0A	P14_1	✓	✓
			P29_1	✓	✓
		GTIOC09_0B	P14_2	✓	✓
			P29_2	✓	✓
		GTIOC09_1A	P14_3	✓	✓
			P29_3	✓	✓
		GTIOC09_1B	P14_4	✓	✓
			P29_4	✓	✓
		GTIOC09_2A	P14_5	✓	✓
			P29_5	✓	✓

Table 17.2 List of multiplexed pin configurations (14 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
General PWM Timer	GPT09	GTIOC09_2B	P14_6	✓	✓
			P29_6	✓	✓
		GTIOC09_3A	P14_7	✓	✓
			P29_7	✓	✓
		GTIOC09_3B	P15_0	✓	—
			P30_0	✓	✓
		GTIOC09_4A	P15_1	✓	—
			P30_1	✓	✓
		GTIOC09_4B	P15_2	✓	—
			P30_2	✓	✓
		GTIOC09_5A	P15_3	✓	—
			P30_3	✓	✓
		GTIOC09_5B	P15_4	✓	—
			P30_4	✓	✓
		GTIOC09_6A	P15_5	✓	—
			P30_5	✓	✓
		GTIOC09_6B	P15_6	✓	—
			P30_6	✓	✓
	GPT10	GTIOC10_0A	P09_4	✓	✓
			P32_2	✓	—
		GTIOC10_0B	P09_5	✓	✓
			P32_3	✓	—
		GTIOC10_1A	P09_6	✓	✓
			P32_4	✓	—
		GTIOC10_1B	P09_7	✓	✓
			P32_5	✓	—
		GTIOC10_2A	P10_0	✓	✓
			P32_6	✓	—
		GTIOC10_2B	P10_1	✓	✓
			P32_7	✓	—
		GTIOC10_3A	P10_2	✓	✓
			P33_0	✓	—
		GTIOC10_3B	P10_3	✓	✓
			P33_1	✓	—

Table 17.2 List of multiplexed pin configurations (15 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Port Output Enable for GPT	—	GTETRGA	P06_3	✓	✓
			P17_3	✓	✓
			P22_1	✓	✓
		GTETRGB	P06_4	✓	✓
			P17_4	✓	✓
			P22_2	✓	✓
		GTETRGC	P06_5	✓	✓
			P17_5	✓	✓
			P22_3	✓	✓
		GTETRGD	P06_7	✓	✓
			P17_6	✓	✓
			P22_4	✓	✓
		GTETRGSA	P08_5	✓	✓
			P16_3	✓	—
			P22_5	✓	✓
			P31_0	✓	✓
		GTETRGSB	P08_6	✓	✓
			P16_4	✓	—
			P22_6	✓	✓
			P31_1	✓	✓
Compare Match Timer W	CMTW0	CMTW0_TIC0	P02_5	✓	✓
			P03_7	✓	—
			P12_0	✓	✓
			P14_4	✓	✓
		CMTW0_TIC1	P02_7	✓	✓
			P04_1	✓	—
			P12_2	✓	✓
			P14_6	✓	✓
		CMTW0_TOC0	P02_6	✓	✓
			P04_0	✓	—
			P12_1	✓	✓
			P14_5	✓	✓
		CMTW0_TOC1	P03_0	✓	✓
			P04_2	✓	—
			P12_3	✓	✓
			P14_7	✓	✓

Table 17.2 List of multiplexed pin configurations (16 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Compare Match Timer W	CMTW1	CMTW1_TIC0	P03_1	✓	✓
			P04_3	✓	—
			P12_4	✓	✓
			P17_4	✓	✓
		CMTW1_TIC1	P03_3	✓	✓
			P04_5	✓	—
			P12_6	✓	✓
			P17_6	✓	✓
		CMTW1_TOC0	P03_2	✓	✓
			P04_4	✓	—
			P12_5	✓	✓
			P17_5	✓	✓
		CMTW1_TOC1	P03_4	✓	✓
			P04_6	✓	—
			P12_7	✓	✓
			P17_7	✓	✓
Real Time Clock	RTC	RTCAT1HZ	P03_4	✓	✓
			P08_0	✓	—
			P14_1	✓	✓
			P18_3	✓	✓
Ethernet Subsystem	ETH0	ETH0_COL	P22_4	✓	✓
		ETH0_CRS	P22_3	✓	✓
		ETH0_RXCLK	P20_6	✓	✓
		ETH0_RXD0	P20_7	✓	✓
		ETH0_RXD1	P21_0	✓	✓
		ETH0_RXD2	P21_1	✓	✓
		ETH0_RXD3	P21_2	✓	✓
		ETH0_RXDV	P21_3	✓	✓
		ETH0_RXER	P22_2	✓	✓
		ETH0_TXCLK	P20_0	✓	✓
		ETH0_TXD0	P20_1	✓	✓
		ETH0_TXD1	P20_2	✓	✓
		ETH0_TXD2	P20_3	✓	✓
		ETH0_TXD3	P20_4	✓	✓
		ETH0_TXEN	P20_5	✓	✓
		ETH0_TXER	P22_1	✓	✓

Table 17.2 List of multiplexed pin configurations (17 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Ethernet Subsystem	ETH1	ETH1_COL	P18_3	✓	✓
			P23_0	✓	✓
			P27_1	✓	✓
		ETH1_CRS	P18_2	✓	✓
			P22_7	✓	✓
			P27_0	✓	✓
		ETH1_RXCLK	P25_3	✓	✓
		ETH1_RXD0	P25_4	✓	✓
		ETH1_RXD1	P25_5	✓	✓
		ETH1_RXD2	P25_6	✓	✓
		ETH1_RXD3	P25_7	✓	✓
		ETH1_RXDV	P26_0	✓	✓
		ETH1_RXER	P26_7	✓	✓
			P34_6	✓	✓
		ETH1_TXCLK	P24_5	✓	✓
		ETH1_TXD0	P24_6	✓	✓
		ETH1_TXD1	P24_7	✓	✓
		ETH1_TXD2	P25_0	✓	✓
		ETH1_TXD3	P25_1	✓	✓
		ETH1_TXEN	P25_2	✓	✓
		ETH1_TXER	P26_6	✓	✓
	ETH2	ETH2_COL	P31_5	✓	✓
		ETH2_CRS	P31_4	✓	✓
		ETH2_RXCLK	P29_7	✓	✓
		ETH2_RXD0	P30_0	✓	✓
		ETH2_RXD1	P30_1	✓	✓
		ETH2_RXD2	P30_2	✓	✓
		ETH2_RXD3	P30_3	✓	✓
		ETH2_RXDV	P30_4	✓	✓
		ETH2_RXER	P31_1	✓	✓
			P31_3	✓	✓
		ETH2_TXCLK	P29_1	✓	✓
		ETH2_TXD0	P29_2	✓	✓
		ETH2_TXD1	P29_3	✓	✓
		ETH2_TXD2	P29_4	✓	✓
		ETH2_TXD3	P29_5	✓	✓
		ETH2_TXEN	P29_6	✓	✓
		ETH2_TXER	P31_2	✓	✓

Table 17.2 List of multiplexed pin configurations (18 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Ethernet Subsystem	ETH3	ETH3_COL	P00_3	✓	✓
			P02_3	✓	✓
			P35_2	✓	—
		ETH3_CRS	P00_2	✓	✓
			P02_2	✓	✓
			P35_1	✓	—
		ETH3_RXCLK	P34_0	✓	✓
		ETH3_RXD0	P34_1	✓	✓
		ETH3_RXD1	P34_2	✓	✓
		ETH3_RXD2	P34_3	✓	✓
		ETH3_RXD3	P34_4	✓	✓
		ETH3_RXDV	P34_5	✓	✓
		ETH3_RXER	P00_1	✓	✓
			P02_1	✓	✓
			P35_0	✓	—
		ETH3_TXCLK	P33_2	✓	✓
		ETH3_TXD0	P33_3	✓	✓
		ETH3_TXD1	P33_4	✓	✓
		ETH3_TXD2	P33_5	✓	✓
		ETH3_TXD3	P33_6	✓	✓
		ETH3_TXEN	P33_7	✓	✓
		ETH3_TXER	P00_0	✓	✓
			P02_0	✓	✓
			P34_7	✓	—
Ethernet MAC	GMAC0	GMAC0_MDC	P21_4	✓	✓
		GMAC0_MDIO	P21_5	✓	✓
		GMAC0_PTPTRG0	P22_5	✓	✓
		GMAC0_PTPTRG1	P22_6	✓	✓
	GMAC1	GMAC1_MDC	P06_7	✓	✓
			P18_2	✓	✓
			P26_1	✓	✓
		GMAC1_MDIO	P07_0	✓	—
			P18_3	✓	✓
			P26_2	✓	✓
		GMAC1_PTPTRG0	P27_2	✓	✓
		GMAC1_PTPTRG1	P27_3	✓	✓
	GMAC2	GMAC2_MDC	P30_5	✓	✓
		GMAC2_MDIO	P30_6	✓	✓
		GMAC2_PTPTRG0	P31_6	✓	✓
		GMAC2_PTPTRG1	P31_7	✓	—

Table 17.2 List of multiplexed pin configurations (19 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Ethernet Switch	—	ETHSW_LPI0	P23_6	✓	—
		ETHSW_LPI1	P23_7	✓	—
		ETHSW_LPI2	P24_0	✓	—
		ETHSW_MDC	P21_4	✓	✓
			P26_1	✓	✓
			P30_5	✓	✓
		ETHSW_MDIO	P21_5	✓	✓
			P26_2	✓	✓
			P30_6	✓	✓
		ETHSW_PHYLINK0	P21_6	✓	✓
		ETHSW_PHYLINK1	P26_3	✓	✓
		ETHSW_PHYLINK2	P30_7	✓	✓
		ETHSW_PTPOUT0	P17_6	✓	✓
		ETHSW_PTPOUT1	P17_7	✓	✓
		ETHSW_PTPOUT2	P14_0	✓	✓
			P31_4	✓	✓
		ETHSW_PTPOUT3	P18_7	✓	✓
			P31_5	✓	✓
		ETHSW_TDMAOUT0	P27_7	✓	—
			P31_6	✓	✓
		ETHSW_TDMAOUT1	P28_0	✓	—
			P31_3	✓	✓
		ETHSW_TDMAOUT2	P22_7	✓	✓
			P28_1	✓	—
		ETHSW_TDMAOUT3	P23_0	✓	✓
			P28_2	✓	—
EtherCAT slave controller	—	ESC_I2CCLK	P14_6	✓	✓
			P23_3	✓	—
			P24_3	✓	—
			P34_5	✓	✓
		ESC_I2CDATA	P14_7	✓	✓
			P23_4	✓	—
			P24_4	✓	—
			P34_6	✓	✓
		ESC_IRQ	P14_4	✓	✓
			P23_1	✓	—
		ESC_LATCH0	P22_5	✓	✓
		ESC_LATCH1	P22_6	✓	✓
		ESC_LEDERR	P18_1	✓	✓
			P27_2	✓	✓
			P31_3	✓	✓

Table 17.2 List of multiplexed pin configurations (20 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
EtherCAT slave controller	—	ESC_LEDRUN	P11_1	✓	—
			P18_0	✓	✓
			P31_6	✓	✓
		ESC_LEDSTER	P18_4	✓	✓
			P26_7	✓	✓
		ESC_LINKACT0	P16_3	✓	—
			P22_7	✓	✓
		ESC_LINKACT1	P16_4	✓	—
			P23_0	✓	✓
		ESC_LINKACT2	P14_3	✓	✓
			P23_5	✓	—
		ESC_MDC	P21_4	✓	✓
			P26_1	✓	✓
			P30_5	✓	✓
		ESC_MDIO	P21_5	✓	✓
			P26_2	✓	✓
			P30_6	✓	✓
		ESC_PHYLINK0	P21_6	✓	✓
		ESC_PHYLINK1	P26_3	✓	✓
		ESC_PHYLINK2	P30_7	✓	✓
		ESC_RESETOUT#	P11_0	✓	✓
			P14_5	✓	✓
			P23_2	✓	—
			P26_6	✓	✓
			P34_7	✓	—
		ESC_SYNC0	P14_0	✓	✓
			P17_6	✓	✓
			P31_4	✓	✓
		ESC_SYNC1	P17_7	✓	✓
			P18_7	✓	✓
			P31_5	✓	✓
USB 2.0 host/function module	—	USB_EXICEN	P00_2	✓	✓
			P00_7	✓	—
			P02_4	✓	✓
		USB_OVRCUR	P00_1	✓	✓
			P00_6	✓	—
			P02_3	✓	✓
		USB_VBUSEN	P00_0	✓	✓
			P00_5	✓	—
			P02_2	✓	✓

Table 17.2 List of multiplexed pin configurations (21 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Serial Communications Interface	SCI0	CTS0#	P10_5	✓	✓
			P18_1	✓	✓
		DE0	P10_6	✓	✓
			P17_4	✓	✓
		RXD0/SCL0/MISO0	P10_2	✓	✓
			P17_6	✓	✓
			P27_4	✓	✓
		SCK0	P10_1	✓	✓
			P17_5	✓	✓
			P27_3	✓	✓
		SS0#/CTS0#/RTS0#	P10_4	✓	✓
			P18_0	✓	✓
		TXD0/SDA0/MOSI0	P10_3	✓	✓
			P17_7	✓	✓
			P27_5	✓	✓
	SCI1	CTS1#	P11_3	✓	—
			P18_6	✓	✓
		DE1	P11_4	✓	—
			P18_7	✓	✓
		RXD1/SCL1/MISO1	P11_0	✓	✓
			P18_3	✓	✓
			P33_3	✓	✓
		SCK1	P10_7	✓	✓
			P18_2	✓	✓
			P33_2	✓	✓
		SS1#/CTS1#/RTS1#	P11_2	✓	—
			P18_5	✓	✓
		TXD1/SDA1/MOSI1	P11_1	✓	—
			P18_4	✓	✓
			P33_4	✓	✓
	SCI2	CTS2#	P12_7	✓	✓
		DE2	P13_0	✓	✓
		RXD2/SCL2/MISO2	P12_4	✓	✓
			P33_5	✓	✓
		SCK2	P12_3	✓	✓
		SS2#/CTS2#/RTS2#	P12_6	✓	✓
		TXD2/SDA2/MOSI2	P12_5	✓	✓
			P33_6	✓	✓

Table 17.2 List of multiplexed pin configurations (22 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Serial Communications Interface	SCI3	CTS3#	P13_7	✓	✓
		DE3	P14_0	✓	✓
		RXD3/SCL3/MISO3	P13_4	✓	✓
			P34_4	✓	✓
		SCK3	P13_3	✓	✓
		SS3#/CTS3#/RTS3#	P13_6	✓	✓
		TXD3/SDA3/MOSI3	P13_5	✓	✓
			P34_5	✓	✓
	SCI4	CTS4#	P14_5	✓	✓
		DE4	P14_6	✓	✓
		RXD4/SCL4/MISO4	P14_2	✓	✓
			P35_5	✓	—
		SCK4	P14_1	✓	✓
		SS4#/CTS4#/RTS4#	P14_4	✓	✓
		TXD4/SDA4/MOSI4	P14_3	✓	✓
			P35_6	✓	—
	SCI5	CTS5#	P16_0	✓	—
			P22_5	✓	✓
		DE5	P16_1	✓	—
			P22_6	✓	✓
		RXD5/SCL5/MISO5	P16_3	✓	—
			P22_2	✓	✓
		SCK5	P16_2	✓	—
			P22_3	✓	✓
		SS5#/CTS5#/RTS5#	P15_7	✓	—
			P22_4	✓	✓
		TXD5/SDA5/MOSI5	P16_4	✓	—
			P22_1	✓	✓
Serial Communications Interface for Encoder	SCIE00	DEE00	P00_7	✓	—
			P14_3	✓	✓
			P28_6	✓	—
		RXDE00	P01_1	✓	✓
			P14_5	✓	✓
			P29_0	✓	—
		SCKE00	P00_6	✓	—
			P14_2	✓	✓
			P28_5	✓	—
		TXDE00	P01_0	✓	✓
			P14_4	✓	✓
			P28_7	✓	—

Table 17.2 List of multiplexed pin configurations (23 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Serial Communications Interface for Encoder	SCIE01	DEE01	P02_1	✓	✓
			P26_6	✓	✓
			P31_7	✓	—
			P33_3	✓	✓
		RXDE01	P02_3	✓	✓
			P27_0	✓	✓
			P32_1	✓	—
			P33_5	✓	✓
		SCKE01	P02_0	✓	✓
			P26_5	✓	✓
			P31_6	✓	✓
			P33_2	✓	✓
		TXDE01	P02_2	✓	✓
			P26_7	✓	✓
			P32_0	✓	—
			P33_4	✓	✓
	SCIE02	DEE02	P03_4	✓	✓
		RXDE02	P03_2	✓	✓
			P03_6	✓	—
		SCKE02	P03_3	✓	✓
		TXDE02	P03_1	✓	✓
			P03_5	✓	—
	SCIE03	DEE03	P04_6	✓	—
			P13_1	✓	✓
		RXDE03	P05_0	✓	—
			P13_3	✓	✓
		SCKE03	P04_5	✓	—
			P13_0	✓	✓
		TXDE03	P04_7	✓	—
			P13_2	✓	✓
	SCIE04	DEE04	P00_7	✓	—
			P07_2	✓	—
			P10_3	✓	✓
		RXDE04	P01_1	✓	✓
			P07_4	✓	—
			P10_5	✓	✓
		SCKE04	P00_6	✓	—
			P07_1	✓	—
			P10_2	✓	✓

Table 17.2 List of multiplexed pin configurations (24 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Serial Communications Interface for Encoder	SCIE04	TXDE04	P01_0	✓	✓
			P07_3	✓	—
			P10_4	✓	✓
	SCIE05	DEE05	P07_6	✓	—
			P12_5	✓	✓
		RXDE05	P08_0	✓	—
			P12_7	✓	✓
		SCKE05	P07_5	✓	—
			P12_4	✓	✓
		TXDE05	P07_7	✓	—
			P12_6	✓	✓
	SCIE06	DEE06	P15_1	✓	—
			P33_7	✓	✓
		RXDE06	P15_3	✓	—
			P34_1	✓	✓
		SCKE06	P15_0	✓	—
			P33_6	✓	✓
		TXDE06	P15_2	✓	—
			P34_0	✓	✓
	SCIE07	DEE07	P15_5	✓	—
			P34_3	✓	✓
		RXDE07	P15_7	✓	—
			P34_5	✓	✓
		SCKE07	P15_4	✓	—
			P34_2	✓	✓
		TXDE07	P15_6	✓	—
			P16_0	✓	—
			P34_4	✓	✓
	SCIE08	DEE08	P07_2	✓	—
			P13_1	✓	✓
			P28_6	✓	—
		RXDE08	P07_4	✓	—
			P13_3	✓	✓
			P29_0	✓	—
		SCKE08	P07_1	✓	—
			P13_0	✓	✓
			P28_5	✓	—
		TXDE08	P07_3	✓	—
			P13_2	✓	✓
			P28_7	✓	—

Table 17.2 List of multiplexed pin configurations (25 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Serial Communications Interface for Encoder	SCIE09	DEE09	P13_5	✓	✓
			P18_5	✓	✓
			P29_2	✓	✓
			P31_3	✓	✓
		RXDE09	P13_7	✓	✓
			P18_7	✓	✓
			P29_4	✓	✓
			P31_5	✓	✓
		SCKE09	P13_4	✓	✓
			P18_4	✓	✓
			P29_1	✓	✓
			P31_2	✓	✓
		TXDE09	P13_6	✓	✓
			P18_6	✓	✓
			P29_3	✓	✓
			P31_4	✓	✓
	SCIE10	DEE10	P18_5	✓	✓
			P27_4	✓	✓
			P29_6	✓	✓
			P32_3	✓	—
		RXDE10	P18_7	✓	✓
			P27_6	✓	✓
			P30_0	✓	✓
			P32_5	✓	—
		SCKE10	P18_4	✓	✓
			P27_3	✓	✓
			P29_5	✓	✓
			P32_2	✓	—
		TXDE10	P18_6	✓	✓
			P27_5	✓	✓
			P29_7	✓	✓
			P32_4	✓	—
	SCIE11	DEE11	P30_2	✓	✓
			P31_7	✓	—
			P32_7	✓	—
		RXDE11	P30_4	✓	✓
			P32_1	✓	—
			P33_1	✓	—
		SCKE11	P30_1	✓	✓
			P31_6	✓	✓
			P32_6	✓	—

Table 17.2 List of multiplexed pin configurations (26 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Serial Communications Interface for Encoder	SCIE11	TXDE11	P30_3	✓	✓
			P32_0	✓	—
			P33_0	✓	—
I2C Bus Interface	IIC0	IIC_SCL0	P00_6	✓	—
			P02_3	✓	✓
			P02_5	✓	✓
			P04_5	✓	—
			P06_2	✓	✓
			P07_1	✓	—
			P07_7	✓	—
			P14_6	✓	✓
			P23_3	✓	—
		IIC_SDA0	P00_7	✓	—
			P02_4	✓	✓
			P04_6	✓	—
			P06_3	✓	✓
			P07_2	✓	—
			P08_0	✓	—
			P14_7	✓	✓
			P23_4	✓	—
	IIC1	IIC_SCL1	P01_0	✓	✓
			P03_3	✓	✓
			P04_7	✓	—
			P06_4	✓	✓
			P07_3	✓	—
			P08_5	✓	✓
			P08_7	✓	—
			P24_3	✓	—
			P34_5	✓	✓
		IIC_SDA1	P02_0	✓	✓
			P03_4	✓	✓
			P05_0	✓	—
			P06_5	✓	✓
			P07_4	✓	—
			P08_6	✓	✓
			P24_4	✓	—
			P34_6	✓	✓

Table 17.2 List of multiplexed pin configurations (27 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
I2C Bus Interface	IIC2	IIC_SCL2	P02_1	✓	✓
			P02_3	✓	✓
			P03_5	✓	—
			P05_2	✓	✓
			P06_7	✓	✓
			P07_5	✓	—
			P08_7	✓	—
		IIC_SDA2	P02_2	✓	✓
			P03_6	✓	—
			P05_4	✓	✓
			P07_0	✓	—
			P07_6	✓	—
			P09_0	✓	—
CAN-FD Interface	CAN0	CANRX0	P17_4	✓	✓
			P18_2	✓	✓
			P20_3	✓	✓
			P22_2	✓	✓
			P24_3	✓	—
			P26_6	✓	✓
			P28_5	✓	—
		CANRXDP0	P18_0	✓	✓
			P21_1	✓	✓
			P22_3	✓	✓
			P25_0	✓	✓
			P26_7	✓	✓
			P28_7	✓	—
		CANTX0	P17_5	✓	✓
			P18_3	✓	✓
			P20_4	✓	✓
			P22_1	✓	✓
			P24_4	✓	—
			P26_5	✓	✓
			P28_6	✓	—
		CANTXDP0	P18_1	✓	✓
			P21_2	✓	✓
			P22_4	✓	✓
			P25_1	✓	✓
			P27_0	✓	✓
			P29_0	✓	—

Table 17.2 List of multiplexed pin configurations (28 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
CAN-FD Interface	CAN1	CANRX1	P12_0	✓	✓
			P18_4	✓	✓
			P21_4	✓	✓
			P22_5	✓	✓
			P25_6	✓	✓
			P27_1	✓	✓
		CANRXDP1	P12_2	✓	✓
			P18_6	✓	✓
			P21_6	✓	✓
			P22_7	✓	✓
			P26_1	✓	✓
			P27_3	✓	✓
		CANTX1	P12_1	✓	✓
			P18_5	✓	✓
			P21_5	✓	✓
			P22_6	✓	✓
			P25_7	✓	✓
			P27_2	✓	✓
		CANTXDP1	P12_3	✓	✓
			P18_7	✓	✓
			P21_7	✓	✓
			P23_0	✓	✓
			P26_2	✓	✓
			P27_4	✓	✓

Table 17.2 List of multiplexed pin configurations (29 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Serial Peripheral Interface	SPI0	SPI_MISO0	P27_4	✓	✓
			P31_6	✓	✓
			P33_5	✓	✓
		SPI_MOSI0	P27_3	✓	✓
			P31_5	✓	✓
			P33_4	✓	✓
		SPI_RSPCK0	P27_2	✓	✓
			P31_4	✓	✓
			P33_3	✓	✓
		SPI_SSL00	P27_5	✓	✓
			P31_7	✓	—
			P33_6	✓	✓
		SPI_SSL01	P26_7	✓	✓
			P32_0	✓	—
		SPI_SSL02	P27_0	✓	✓
			P32_1	✓	—
		SPI_SSL03	P27_1	✓	✓
			P32_2	✓	—
	SPI1	SPI_MISO1	P28_1	✓	—
			P32_5	✓	—
			P33_4	✓	✓
			P35_1	✓	—
		SPI_MOSI1	P28_0	✓	—
			P32_4	✓	—
			P33_3	✓	✓
			P35_3	✓	—
		SPI_RSPCK1	P27_7	✓	—
			P32_3	✓	—
			P33_2	✓	✓
			P35_5	✓	—
		SPI_SSL10	P28_2	✓	—
			P32_6	✓	—
			P33_5	✓	✓
			P35_2	✓	—
		SPI_SSL11	P28_3	✓	—
			P32_7	✓	—
			P33_6	✓	✓
			P35_4	✓	—
		SPI_SSL12	P28_4	✓	—
			P33_0	✓	—
			P35_6	✓	—

Table 17.2 List of multiplexed pin configurations (30 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Serial Peripheral Interface	SPI1	SPI_SSL13	P28_5	✓	—
			P33_1	✓	—
	SPI2	SPI_MISO2	P30_3	✓	✓
			P34_1	✓	✓
		SPI_MOSI2	P30_2	✓	✓
			P34_0	✓	✓
		SPI_RSPCK2	P33_7	✓	✓
		SPI_SSL20	P29_4	✓	✓
			P34_2	✓	✓
		SPI_SSL21	P29_5	✓	✓
			P34_3	✓	✓
		SPI_SSL22	P29_6	✓	✓
			P34_4	✓	✓
		SPI_SSL23	P13_6	✓	✓
			P29_7	✓	✓
			P34_5	✓	✓
	SPI3	SPI_MISO3	P13_2	✓	✓
			P30_7	✓	✓
			P35_0	✓	—
		SPI_MOSI3	P13_1	✓	✓
			P30_6	✓	✓
			P34_7	✓	—
		SPI_RSPCK3	P13_0	✓	✓
			P30_5	✓	✓
			P34_6	✓	✓
		SPI_SSL30	P13_3	✓	✓
			P31_0	✓	✓
			P31_4	✓	✓
			P33_2	✓	✓
			P35_1	✓	—
		SPI_SSL31	P13_4	✓	✓
			P31_1	✓	✓
			P31_5	✓	✓
			P35_2	✓	—
		SPI_SSL32	P13_5	✓	✓
			P31_2	✓	✓
			P35_3	✓	—
		SPI_SSL33	P31_3	✓	✓
			P35_4	✓	—

Table 17.2 List of multiplexed pin configurations (31 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
expanded Serial Peripheral Interface	xSPI0	XSPI0_CKN	P05_2	✓	✓
		XSPI0_CKP	P05_1	✓	✓
		XSPI0_CS0#	P05_3	✓	✓
		XSPI0_CS1#	P05_4	✓	✓
		XSPI0_DS	P05_5	✓	✓
		XSPI0_ECS0#	P07_5	✓	—
		XSPI0_ECS1#	P07_6	✓	—
		XSPI0_INT0#	P07_3	✓	—
		XSPI0_INT1#	P07_4	✓	—
		XSPI0_IO0	P05_6	✓	✓
		XSPI0_IO1	P05_7	✓	✓
		XSPI0_IO2	P06_0	✓	✓
		XSPI0_IO3	P06_1	✓	✓
		XSPI0_IO4	P06_2	✓	✓
		XSPI0_IO5	P06_3	✓	✓
		XSPI0_IO6	P06_4	✓	✓
		XSPI0_IO7	P06_5	✓	✓
		XSPI0_RESET0#	P06_6	✓	✓
		XSPI0_RESET1#	P07_0	✓	—
		XSPI0_RSTO0#	P07_1	✓	—
		XSPI0_RSTO1#	P07_2	✓	—
		XSPI0_WP0#	P07_7	✓	—
		XSPI0_WP1#	P08_0	✓	—
	xSPI1	XSPI1_CKP	P01_0	✓	✓
		XSPI1_CS0#	P01_1	✓	✓
		XSPI1_CS1#	P01_2	✓	✓
		XSPI1_DS	P01_3	✓	✓
		XSPI1_IO0	P01_4	✓	✓
		XSPI1_IO1	P01_5	✓	✓
		XSPI1_IO2	P01_6	✓	✓
		XSPI1_IO3	P01_7	✓	✓
		XSPI1_IO4	P02_0	✓	✓
		XSPI1_IO5	P02_1	✓	✓
		XSPI1_IO6	P02_2	✓	✓
		XSPI1_IO7	P02_3	✓	✓

Table 17.2 List of multiplexed pin configurations (32 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Delta-sigma Interface	DSMIF0	MCLK00	P02_5	✓	✓
			P07_1	✓	—
			P10_2	✓	✓
			P13_0	✓	✓
		MCLK01	P02_7	✓	✓
			P07_3	✓	—
			P10_4	✓	✓
			P13_2	✓	✓
		MCLK02	P07_5	✓	—
			P08_5	✓	✓
			P12_4	✓	✓
		MDAT00	P02_6	✓	✓
			P07_2	✓	—
			P10_3	✓	✓
			P13_1	✓	✓
		MDAT01	P03_0	✓	✓
			P07_4	✓	—
			P10_5	✓	✓
			P13_3	✓	✓
		MDAT02	P07_6	✓	—
			P08_6	✓	✓
			P12_5	✓	✓

Table 17.2 List of multiplexed pin configurations (33 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Delta-sigma Interface	DSMIF1	MCLK10	P07_7	✓	—
			P10_2	✓	✓
			P12_6	✓	✓
			P18_2	✓	✓
		MCLK11	P08_6	✓	✓
			P10_4	✓	✓
			P18_4	✓	✓
		MCLK12	P09_0	✓	—
			P18_6	✓	✓
		MDAT10	P08_0	✓	—
			P10_3	✓	✓
			P12_7	✓	✓
			P18_3	✓	✓
		MDAT11	P08_7	✓	—
			P10_5	✓	✓
			P18_5	✓	✓
		MDAT12	P09_1	✓	—
			P18_7	✓	✓
	DSMIF2	MCLK20	P01_1	✓	✓
			P09_2	✓	—
		MCLK21	P02_0	✓	✓
			P10_6	✓	✓
		MCLK22	P02_2	✓	✓
			P11_0	✓	✓
		MDAT20	P02_4	✓	✓
			P09_3	✓	—
		MDAT21	P02_1	✓	✓
			P10_7	✓	✓
		MDAT22	P02_3	✓	✓
			P11_1	✓	—
	DSMIF3	MCLK30	P11_2	✓	—
			P30_7	✓	✓
		MCLK31	P11_4	✓	—
			P31_3	✓	✓
		MCLK32	P14_7	✓	✓
			P31_5	✓	✓
		MDAT30	P11_3	✓	—
			P31_2	✓	✓
		MDAT31	P11_5	✓	—
			P31_4	✓	✓

Table 17.2 List of multiplexed pin configurations (34 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Delta-sigma Interface	DSMIF3	MDAT32	P15_0	✓	—
			P31_6	✓	✓
	DSMIF4	MCLK40	P13_4	✓	✓
			P15_1	✓	—
		MCLK41	P13_6	✓	✓
			P15_3	✓	—
		MCLK42	P14_0	✓	✓
			P15_5	✓	—
		MDAT40	P13_5	✓	✓
			P15_2	✓	—
		MDAT41	P13_7	✓	✓
			P15_4	✓	—
		MDAT42	P14_1	✓	✓
			P15_6	✓	—
	DSMIF5	MCLK50	P15_7	✓	—
			P33_2	✓	✓
		MCLK51	P16_1	✓	—
			P33_4	✓	✓
		MCLK52	P19_6	✓	—
			P33_6	✓	✓
		MDAT50	P16_0	✓	—
			P33_3	✓	✓
		MDAT51	P16_2	✓	—
			P33_5	✓	✓
		MDAT52	P19_7	✓	—
			P33_7	✓	✓
	DSMIF6	MCLK60	P23_5	✓	—
		MCLK61	P23_7	✓	—
		MCLK62	P24_1	✓	—
		MDAT60	P23_6	✓	—
		MDAT61	P24_0	✓	—
		MDAT62	P24_2	✓	—
	DSMIF7	MCLK70	P09_4	✓	✓
			P24_3	✓	—
		MCLK71	P09_6	✓	✓
			P28_5	✓	—
		MCLK72	P10_0	✓	✓
			P28_7	✓	—
		MDAT70	P09_5	✓	✓
			P24_4	✓	—

Table 17.2 List of multiplexed pin configurations (35 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Delta-sigma Interface	DSMIF7	MDAT71	P09_7	✓	✓
			P28_6	✓	—
		MDAT72	P10_1	✓	✓
			P29_0	✓	—
	DSMIF8	MCLK80	P31_2	✓	✓
		MCLK81	P31_4	✓	✓
		MCLK82	P33_0	✓	—
		MDAT80	P31_3	✓	✓
		MDAT81	P31_5	✓	✓
		MDAT82	P33_1	✓	—
	DSMIF9	MCLK90	P35_1	✓	—
		MCLK91	P35_3	✓	—
		MCLK92	P35_5	✓	—
		MDAT90	P35_2	✓	—
		MDAT91	P35_4	✓	—
		MDAT92	P35_6	✓	—
12-Bit A/D Converter	ADC120	ADTRG0#	P00_2	✓	✓
			P04_4	✓	—
			P34_4	✓	✓
			P35_3	✓	—
	ADC121	ADTRG1#	P00_3	✓	✓
			P04_5	✓	—
			P34_5	✓	✓
			P35_4	✓	—
	ADC122	ADTRG2#	P00_4	✓	✓
			P04_6	✓	—
			P34_6	✓	✓
			P35_2	✓	—
Serial host interface	—	HSPI_CK	P27_6	✓	✓
		HSPI_CS#	P27_1	✓	✓
		HSPI_INT#	P27_0	✓	✓
		HSPI_IO0	P27_2	✓	✓
		HSPI_IO1	P27_3	✓	✓
		HSPI_IO2	P27_4	✓	✓
		HSPI_IO3	P27_5	✓	✓
		HSPI_IO4	P31_2	✓	✓
		HSPI_IO5	P31_3	✓	✓
		HSPI_IO6	P31_4	✓	✓
		HSPI_IO7	P31_5	✓	✓

Table 17.2 List of multiplexed pin configurations (36 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Mailbox	—	MBX_HINT#	P02_4	✓	✓
			P04_6	✓	—
			P08_0	✓	—
			P14_4	✓	✓
Encoder interface subsystem	ENCIF00	ENCIFCK00	P00_6	✓	—
			P14_2	✓	✓
			P28_5	✓	—
		ENCIFDI00	P01_1	✓	✓
			P14_5	✓	✓
			P29_0	✓	—
		ENCIFDO00	P01_0	✓	✓
			P14_4	✓	✓
			P28_7	✓	—
		ENCIFOE00	P00_7	✓	—
			P14_3	✓	✓
			P28_6	✓	—
	ENCIF01	ENCIFCK01	P02_0	✓	✓
			P26_5	✓	✓
			P31_6	✓	✓
			P33_2	✓	✓
		ENCIFDI01	P02_3	✓	✓
			P27_0	✓	✓
			P32_1	✓	—
			P33_5	✓	✓
		ENCIFDO01	P02_2	✓	✓
			P26_7	✓	✓
			P32_0	✓	—
			P33_4	✓	✓
		ENCIFOE01	P02_1	✓	✓
			P26_6	✓	✓
			P31_7	✓	—
			P33_3	✓	✓
	ENCIF02	ENCIFCK02	P03_3	✓	✓
		ENCIFDI02	P03_2	✓	✓
			P03_6	✓	—
		ENCIFDO02	P03_1	✓	✓
			P03_5	✓	—
		ENCIFOE02	P03_4	✓	✓
	ENCIF03	ENCIFCK03	P04_5	✓	—
			P13_0	✓	✓

Table 17.2 List of multiplexed pin configurations (37 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Encoder interface subsystem	ENCIF03	ENCIFDI03	P05_0	✓	—
			P13_3	✓	✓
		ENCIFDO03	P04_7	✓	—
			P13_2	✓	✓
		ENCIFOE03	P04_6	✓	—
			P13_1	✓	✓
	ENCIF04	ENCIFCK04	P00_6	✓	—
			P07_1	✓	—
			P10_2	✓	✓
		ENCIFDI04	P01_1	✓	✓
			P07_4	✓	—
			P10_5	✓	✓
		ENCIFDO04	P01_0	✓	✓
			P07_3	✓	—
			P10_4	✓	✓
		ENCIFOE04	P00_7	✓	—
			P07_2	✓	—
			P10_3	✓	✓
	ENCIF05	ENCIFCK05	P07_5	✓	—
			P12_4	✓	✓
		ENCIFDI05	P08_0	✓	—
			P12_7	✓	✓
		ENCIFDO05	P07_7	✓	—
			P12_6	✓	✓
		ENCIFOE05	P07_6	✓	—
			P12_5	✓	✓
	ENCIF06	ENCIFCK06	P15_0	✓	—
			P33_6	✓	✓
		ENCIFDI06	P15_3	✓	—
			P34_1	✓	✓
		ENCIFDO06	P15_2	✓	—
			P34_0	✓	✓
		ENCIFOE06	P15_1	✓	—
			P33_7	✓	✓
	ENCIF07	ENCIFCK07	P15_4	✓	—
			P34_2	✓	✓
		ENCIFDI07	P15_7	✓	—
			P34_5	✓	✓
		ENCIFDO07	P15_6	✓	—
			P34_4	✓	✓

Table 17.2 List of multiplexed pin configurations (38 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Encoder interface subsystem	ENCIF07	ENCIFOE07	P15_5	✓	—
			P34_3	✓	✓
	ENCIF08	ENCIFCK08	P28_5	✓	—
		ENCIFDI08	P29_0	✓	—
		ENCIFDO08	P28_7	✓	—
		ENCIFOE08	P28_6	✓	—
	ENCIF09	ENCIFCK09	P29_1	✓	✓
			P31_2	✓	✓
		ENCIFDI09	P29_4	✓	✓
			P31_5	✓	✓
		ENCIFDO09	P29_3	✓	✓
			P31_4	✓	✓
		ENCIFOE09	P29_2	✓	✓
			P31_3	✓	✓
	ENCIF10	ENCIFCK10	P29_5	✓	✓
			P32_2	✓	—
		ENCIFDI10	P30_0	✓	✓
			P32_5	✓	—
		ENCIFDO10	P29_7	✓	✓
			P32_4	✓	—
		ENCIFOE10	P29_6	✓	✓
			P32_3	✓	—
	ENCIF11	ENCIFCK11	P30_1	✓	✓
			P32_6	✓	—
		ENCIFDI11	P30_4	✓	✓
			P33_1	✓	—
		ENCIFDO11	P30_3	✓	✓
			P33_0	✓	—
		ENCIFOE11	P30_2	✓	✓
			P32_7	✓	—
	ENCIF12	ENCIFCK12	P07_1	✓	—
			P13_0	✓	✓
		ENCIFDI12	P07_4	✓	—
			P13_3	✓	✓
		ENCIFDO12	P07_3	✓	—
			P13_2	✓	✓
		ENCIFOE12	P07_2	✓	—
			P13_1	✓	✓
	ENCIF13	ENCIFCK13	P13_4	✓	✓
			P18_4	✓	✓

Table 17.2 List of multiplexed pin configurations (39 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
Encoder interface subsystem	ENCIF13	ENCIFDI13	P13_7	✓	✓
			P18_7	✓	✓
		ENCIFDO13	P13_6	✓	✓
			P18_6	✓	✓
		ENCIFOE13	P13_5	✓	✓
			P18_5	✓	✓
	ENCIF14	ENCIFCK14	P18_4	✓	✓
			P27_3	✓	✓
		ENCIFDI14	P18_7	✓	✓
			P27_6	✓	✓
		ENCIFDO14	P18_6	✓	✓
			P27_5	✓	✓
		ENCIFOE14	P18_5	✓	✓
			P27_4	✓	✓
	ENCIF15	ENCIFCK15	P31_6	✓	✓
		ENCIFDI15	P32_1	✓	—
		ENCIFDO15	P32_0	✓	—
		ENCIFOE15	P31_7	✓	—
EnDat 2.2	ENDAT00	DUEI00	P00_0	✓	✓
			P11_5	✓	—
			P23_1	✓	—
		SI00#	P00_2	✓	✓
			P11_7	✓	—
			P23_3	✓	—
		TST_OUT00	P00_1	✓	✓
			P11_6	✓	—
			P23_2	✓	—
	ENDAT01	DUEI01	P00_3	✓	✓
			P12_0	✓	✓
			P23_4	✓	—
		SI01#	P00_5	✓	—
			P12_2	✓	✓
			P23_6	✓	—
		TST_OUT01	P00_4	✓	✓
			P12_1	✓	✓
			P23_5	✓	—
	ENDAT02	DUEI02	P01_2	✓	✓
			P14_1	✓	✓
			P23_7	✓	—

Table 17.2 List of multiplexed pin configurations (40 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
EnDat 2.2	ENDAT02	SI02#	P01_4	✓	✓
			P14_7	✓	✓
			P24_1	✓	—
		TST_OUT02	P01_3	✓	✓
			P14_6	✓	✓
			P24_0	✓	—
	ENDAT03	DUEI03	P01_5	✓	✓
			P16_0	✓	—
			P25_3	✓	✓
		SI03#	P01_7	✓	✓
			P16_2	✓	—
			P25_5	✓	✓
		TST_OUT03	P01_6	✓	✓
			P16_1	✓	—
			P25_4	✓	✓
	ENDAT04	DUEI04	P03_7	✓	—
			P16_3	✓	—
			P25_6	✓	✓
		SI04#	P04_1	✓	—
			P16_5	✓	✓
			P26_0	✓	✓
		TST_OUT04	P04_0	✓	—
			P16_4	✓	—
			P25_7	✓	✓
	ENDAT05	DUEI05	P04_2	✓	—
			P16_6	✓	✓
			P27_7	✓	—
		SI05#	P04_4	✓	—
			P17_0	✓	✓
			P28_1	✓	—
		TST_OUT05	P04_3	✓	—
			P16_7	✓	✓
			P28_0	✓	—
	ENDAT06	DUEI06	P05_1	✓	✓
			P17_1	✓	✓
			P28_2	✓	—
		SI06#	P05_3	✓	✓
			P17_3	✓	✓
			P28_4	✓	—

Table 17.2 List of multiplexed pin configurations (41 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
EnDat 2.2	ENDAT06	TST_OUT06	P05_2	✓	✓
			P17_2	✓	✓
			P28_3	✓	—
	ENDAT07	DUEI07	P05_4	✓	✓
			P17_4	✓	✓
			P30_5	✓	✓
		SI07#	P05_6	✓	✓
			P17_6	✓	✓
			P30_7	✓	✓
		TST_OUT07	P05_5	✓	✓
			P17_5	✓	✓
			P30_6	✓	✓
	ENDAT08	DUEI08	P05_7	✓	✓
			P17_7	✓	✓
			P34_6	✓	✓
		SI08#	P06_1	✓	✓
			P18_1	✓	✓
			P35_0	✓	—
		TST_OUT08	P06_0	✓	✓
			P18_0	✓	✓
			P34_7	✓	—
	ENDAT09	DUEI09	P06_2	✓	✓
			P19_0	✓	—
			P35_1	✓	—
		SI09#	P06_4	✓	✓
			P19_2	✓	—
			P35_3	✓	—
		TST_OUT09	P06_3	✓	✓
			P19_1	✓	—
			P35_2	✓	—
	ENDAT10	DUEI10	P08_1	✓	✓
			P19_3	✓	—
			P35_4	✓	—
		SI10#	P08_3	✓	✓
			P19_5	✓	—
			P35_6	✓	—
		TST_OUT10	P08_2	✓	✓
			P19_4	✓	—
			P35_5	✓	—
	ENDAT11	DUEI11	P08_6	✓	✓
			P20_5	✓	✓

Table 17.2 List of multiplexed pin configurations (42 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
EnDat 2.2	ENDAT11	SI11#	P09_0	✓	—
			P20_7	✓	✓
		TST_OUT11	P08_7	✓	—
			P20_6	✓	✓
	ENDAT12	DUEI12	P09_1	✓	—
			P21_0	✓	✓
		SI12#	P09_3	✓	—
			P21_2	✓	✓
		TST_OUT12	P09_2	✓	—
			P21_1	✓	✓
	ENDAT13	DUEI13	P09_4	✓	✓
			P21_3	✓	✓
		SI13#	P09_6	✓	✓
			P21_5	✓	✓
		TST_OUT13	P09_5	✓	✓
			P21_4	✓	✓
	ENDAT14	DUEI14	P09_7	✓	✓
			P22_3	✓	✓
		SI14#	P10_1	✓	✓
			P22_5	✓	✓
		TST_OUT14	P10_0	✓	✓
			P22_4	✓	✓
	ENDAT15	DUEI15	P11_1	✓	—
			P22_6	✓	✓
		SI15#	P11_3	✓	—
			P23_0	✓	✓
		TST_OUT15	P11_2	✓	—
			P22_7	✓	✓
HIPERFACE DSL	HDSL00	HDSL00_CLK1	P00_2	✓	✓
			P20_7	✓	✓
		HDSL00_CLK2	P00_6	✓	—
			P21_3	✓	✓
		HDSL00_LINK	P00_0	✓	✓
			P20_5	✓	✓
		HDSL00_MISO1	P00_4	✓	✓
			P21_1	✓	✓
		HDSL00_MISO2	P01_0	✓	✓
			P21_5	✓	✓
		HDSL00_MOSI1	P00_5	✓	—
			P21_2	✓	✓

Table 17.2 List of multiplexed pin configurations (43 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
HIPERFACE DSL	HDSL00	HDSL00_MOSI2	P01_1	✓	✓
			P21_6	✓	✓
		HDSL00_SEL1	P00_3	✓	✓
			P21_0	✓	✓
		HDSL00_SEL2	P00_7	✓	—
			P21_4	✓	✓
		HDSL00_SMPL	P00_1	✓	✓
			P20_6	✓	✓
	HDSL01	HDSL01_CLK1	P01_4	✓	✓
			P22_1	✓	✓
		HDSL01_CLK2	P02_0	✓	✓
			P22_5	✓	✓
		HDSL01_LINK	P01_2	✓	✓
			P21_7	✓	✓
		HDSL01_MISO1	P01_6	✓	✓
			P22_3	✓	✓
		HDSL01_MISO2	P02_2	✓	✓
			P22_7	✓	✓
		HDSL01_MOSI1	P01_7	✓	✓
			P22_4	✓	✓
		HDSL01_MOSI2	P02_3	✓	✓
			P23_0	✓	✓
		HDSL01_SEL1	P01_5	✓	✓
			P22_2	✓	✓
		HDSL01_SEL2	P02_1	✓	✓
			P22_6	✓	✓
		HDSL01_SMPL	P01_3	✓	✓
			P22_0	✓	✓
	HDSL02	HDSL02_CLK1	P02_6	✓	✓
			P23_3	✓	—
		HDSL02_CLK2	P03_2	✓	✓
			P23_7	✓	—
		HDSL02_LINK	P02_4	✓	✓
			P23_1	✓	—
		HDSL02_MISO1	P03_0	✓	✓
			P23_5	✓	—
		HDSL02_MISO2	P03_4	✓	✓
			P24_1	✓	—
		HDSL02_MOSI1	P03_1	✓	✓
			P23_6	✓	—

Table 17.2 List of multiplexed pin configurations (44 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
HIPERFACE DSL	HDSL02	HDSL02_MOSI2	P03_5	✓	—
			P24_2	✓	—
		HDSL02_SEL1	P02_7	✓	✓
			P23_4	✓	—
		HDSL02_SEL2	P03_3	✓	✓
			P24_0	✓	—
		HDSL02_SMPL	P02_5	✓	✓
			P23_2	✓	—
	HDSL03	HDSL03_CLK1	P04_0	✓	—
			P24_5	✓	✓
		HDSL03_CLK2	P04_4	✓	—
			P25_6	✓	✓
		HDSL03_LINK	P03_6	✓	—
			P24_3	✓	—
		HDSL03_MISO1	P04_2	✓	—
			P25_4	✓	✓
		HDSL03_MISO2	P04_6	✓	—
			P26_0	✓	✓
		HDSL03_MOSI1	P04_3	✓	—
			P25_5	✓	✓
		HDSL03_MOSI2	P04_7	✓	—
			P26_1	✓	✓
		HDSL03_SEL1	P04_1	✓	—
			P25_3	✓	✓
		HDSL03_SEL2	P04_5	✓	—
			P25_7	✓	✓
		HDSL03_SMPL	P03_7	✓	—
			P24_4	✓	—
	HDSL04	HDSL04_CLK1	P05_2	✓	✓
			P26_5	✓	✓
		HDSL04_CLK2	P05_6	✓	✓
			P27_1	✓	✓
		HDSL04_LINK	P05_0	✓	—
			P26_2	✓	✓
		HDSL04_MISO1	P05_4	✓	✓
			P26_7	✓	✓
		HDSL04_MISO2	P06_0	✓	✓
			P27_3	✓	✓
		HDSL04_MOSI1	P05_5	✓	✓
			P27_0	✓	✓

Table 17.2 List of multiplexed pin configurations (45 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
HIPERFACE DSL	HDSL04	HDSL04_MOSI2	P06_1	✓	✓
			P27_4	✓	✓
		HDSL04_SEL1	P05_3	✓	✓
			P26_6	✓	✓
		HDSL04_SEL2	P05_7	✓	✓
			P27_2	✓	✓
		HDSL04_SMPL	P05_1	✓	✓
			P26_3	✓	✓
	HDSL05	HDSL05_CLK1	P06_4	✓	✓
			P27_7	✓	—
		HDSL05_CLK2	P07_1	✓	—
			P28_3	✓	—
		HDSL05_LINK	P06_2	✓	✓
			P27_5	✓	✓
		HDSL05_MISO1	P06_7	✓	✓
			P28_1	✓	—
		HDSL05_MISO2	P07_3	✓	—
			P28_5	✓	—
		HDSL05_MOSI1	P07_0	✓	—
			P28_2	✓	—
		HDSL05_MOSI2	P07_4	✓	—
			P28_6	✓	—
		HDSL05_SEL1	P06_5	✓	✓
			P28_0	✓	—
		HDSL05_SEL2	P07_2	✓	—
			P28_4	✓	—
		HDSL05_SMPL	P06_3	✓	✓
			P27_6	✓	✓
	HDSL06	HDSL06_CLK1	P07_7	✓	—
			P29_1	✓	✓
		HDSL06_CLK2	P08_3	✓	✓
			P29_5	✓	✓
		HDSL06_LINK	P07_5	✓	—
			P28_7	✓	—
		HDSL06_MISO1	P08_1	✓	✓
			P29_3	✓	✓
		HDSL06_MISO2	P08_5	✓	✓
			P29_7	✓	✓
		HDSL06_MOSI1	P08_2	✓	✓
			P29_4	✓	✓

Table 17.2 List of multiplexed pin configurations (46 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
HIPERFACE DSL	HDSL06	HDSL06_MOSI2	P08_6	✓	✓
			P30_0	✓	✓
		HDSL06_SEL1	P08_0	✓	—
			P29_2	✓	✓
		HDSL06_SEL2	P08_4	✓	✓
			P29_6	✓	✓
		HDSL06_SMPL	P07_6	✓	—
			P29_0	✓	—
	HDSL07	HDSL07_CLK1	P09_1	✓	—
			P30_3	✓	✓
		HDSL07_CLK2	P09_5	✓	✓
			P30_7	✓	✓
		HDSL07_LINK	P08_7	✓	—
			P30_1	✓	✓
		HDSL07_MISO1	P09_3	✓	—
			P30_5	✓	✓
		HDSL07_MISO2	P09_7	✓	✓
			P31_1	✓	✓
		HDSL07_MOSI1	P09_4	✓	✓
			P30_6	✓	✓
		HDSL07_MOSI2	P10_0	✓	✓
			P31_2	✓	✓
		HDSL07_SEL1	P09_2	✓	—
			P30_4	✓	✓
		HDSL07_SEL2	P09_6	✓	✓
			P31_0	✓	✓
		HDSL07_SMPL	P09_0	✓	—
			P30_2	✓	✓
	HDSL08	HDSL08_CLK1	P10_3	✓	✓
			P31_5	✓	✓
		HDSL08_CLK2	P10_7	✓	✓
			P32_1	✓	—
		HDSL08_LINK	P10_1	✓	✓
			P31_3	✓	✓
		HDSL08_MISO1	P10_5	✓	✓
			P31_7	✓	—
		HDSL08_MISO2	P11_1	✓	—
			P32_3	✓	—
		HDSL08_MOSI1	P10_6	✓	✓
			P32_0	✓	—

Table 17.2 List of multiplexed pin configurations (47 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
HIPERFACE DSL	HDSL08	HDSL08_MOSI2	P11_2	✓	—
			P32_4	✓	—
		HDSL08_SEL1	P10_4	✓	✓
			P31_6	✓	✓
		HDSL08_SEL2	P11_0	✓	✓
			P32_2	✓	—
		HDSL08_SMPL	P10_2	✓	✓
			P31_4	✓	✓
	HDSL09	HDSL09_CLK1	P11_5	✓	—
			P32_7	✓	—
		HDSL09_CLK2	P12_1	✓	✓
			P33_3	✓	✓
		HDSL09_LINK	P11_3	✓	—
			P32_5	✓	—
		HDSL09_MISO1	P11_7	✓	—
			P33_1	✓	—
		HDSL09_MISO2	P12_3	✓	✓
			P33_5	✓	✓
		HDSL09_MOSI1	P12_0	✓	✓
			P33_2	✓	✓
		HDSL09_MOSI2	P12_4	✓	✓
			P33_6	✓	✓
		HDSL09_SEL1	P11_6	✓	—
			P33_0	✓	—
		HDSL09_SEL2	P12_2	✓	✓
			P33_4	✓	✓
		HDSL09_SMPL	P11_4	✓	—
			P32_6	✓	—
	HDSL10	HDSL10_CLK1	P12_7	✓	✓
			P34_1	✓	✓
		HDSL10_CLK2	P13_3	✓	✓
			P34_5	✓	✓
		HDSL10_LINK	P12_5	✓	✓
			P33_7	✓	✓
		HDSL10_MISO1	P13_1	✓	✓
			P34_3	✓	✓
		HDSL10_MISO2	P13_5	✓	✓
			P34_7	✓	—
		HDSL10_MOSI1	P13_2	✓	✓
			P34_4	✓	✓

Table 17.2 List of multiplexed pin configurations (48 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
HIPERFACE DSL	HDSL10	HDSL10_MOSI2	P13_6	✓	✓
			P35_0	✓	—
		HDSL10_SEL1	P13_0	✓	✓
			P34_2	✓	✓
		HDSL10_SEL2	P13_4	✓	✓
			P34_6	✓	✓
		HDSL10_SMPL	P12_6	✓	✓
			P34_0	✓	✓
	HDSL11	HDSL11_CLK1	P14_1	✓	✓
			P35_3	✓	—
		HDSL11_CLK2	P14_5	✓	✓
		HDSL11_LINK	P13_7	✓	✓
			P35_1	✓	—
		HDSL11_MISO1	P14_3	✓	✓
			P35_5	✓	—
		HDSL11_MISO2	P14_7	✓	✓
		HDSL11_MOSI1	P14_4	✓	✓
			P35_6	✓	—
		HDSL11_MOSI2	P15_0	✓	—
		HDSL11_SEL1	P14_2	✓	✓
			P35_4	✓	—
		HDSL11_SEL2	P14_6	✓	✓
		HDSL11_SMPL	P14_0	✓	✓
			P35_2	✓	—
	HDSL12	HDSL12_CLK1	P15_3	✓	—
		HDSL12_CLK2	P15_7	✓	—
		HDSL12_LINK	P15_1	✓	—
		HDSL12_MISO1	P15_5	✓	—
		HDSL12_MISO2	P16_1	✓	—
		HDSL12_MOSI1	P15_6	✓	—
		HDSL12_MOSI2	P16_2	✓	—
		HDSL12_SEL1	P15_4	✓	—
		HDSL12_SEL2	P16_0	✓	—
		HDSL12_SMPL	P15_2	✓	—
	HDSL13	HDSL13_CLK1	P16_5	✓	✓
		HDSL13_CLK2	P17_1	✓	✓
		HDSL13_LINK	P16_3	✓	—
		HDSL13_MISO1	P16_7	✓	✓
		HDSL13_MISO2	P17_3	✓	✓
		HDSL13_MOSI1	P17_0	✓	✓
		HDSL13_MOSI2	P17_4	✓	✓

Table 17.2 List of multiplexed pin configurations (49 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
HIPERFACE DSL	HDSL13	HDSL13_SEL1	P16_6	✓	✓
		HDSL13_SEL2	P17_2	✓	✓
		HDSL13_SMPL	P16_4	✓	—
	HDSL14	HDSL14_CLK1	P17_7	✓	✓
		HDSL14_CLK2	P18_3	✓	✓
		HDSL14_LINK	P17_5	✓	✓
		HDSL14_MISO1	P18_1	✓	✓
		HDSL14_MISO2	P18_5	✓	✓
		HDSL14_MOSI1	P18_2	✓	✓
		HDSL14_MOSI2	P18_6	✓	✓
		HDSL14_SEL1	P18_0	✓	✓
		HDSL14_SEL2	P18_4	✓	✓
		HDSL14_SMPL	P17_6	✓	✓
	HDSL15	HDSL15_CLK1	P19_1	✓	—
		HDSL15_CLK2	P19_5	✓	—
		HDSL15_LINK	P18_7	✓	✓
		HDSL15_MISO1	P19_3	✓	—
		HDSL15_MISO2	P19_7	✓	—
		HDSL15_MOSI1	P19_4	✓	—
		HDSL15_MOSI2	P20_0	✓	✓
		HDSL15_SEL1	P19_2	✓	—
		HDSL15_SEL2	P19_6	✓	—
		HDSL15_SMPL	P19_0	✓	—
ENCOUT	—	POUTA	P02_5	✓	✓
			P10_6	✓	✓
			P31_2	✓	✓
		POUTB	P02_6	✓	✓
			P10_7	✓	✓
			P31_4	✓	✓
		POUTZ	P02_7	✓	✓
			P11_0	✓	✓
			P31_5	✓	✓
LCD Controller	—	DISP_CLK	P09_4	✓	✓
		DISP_DATAB0	P17_7	✓	✓
		DISP_DATAB1	P18_0	✓	✓
		DISP_DATAB2	P18_1	✓	✓
		DISP_DATAB3	P18_2	✓	✓
		DISP_DATAB4	P18_3	✓	✓
		DISP_DATAB5	P18_4	✓	✓
		DISP_DATAB6	P18_5	✓	✓
		DISP_DATAB7	P18_6	✓	✓

Table 17.2 List of multiplexed pin configurations (50 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
LCD Controller	—	DISP_DATAG0	P11_0	✓	✓
		DISP_DATAG1	P14_1	✓	✓
		DISP_DATAG2	P14_2	✓	✓
		DISP_DATAG3	P14_3	✓	✓
		DISP_DATAG4	P14_4	✓	✓
		DISP_DATAG5	P14_5	✓	✓
		DISP_DATAG6	P14_6	✓	✓
		DISP_DATAG7	P17_6	✓	✓
		DISP_DATAR0	P10_0	✓	✓
		DISP_DATAR1	P10_1	✓	✓
		DISP_DATAR2	P10_2	✓	✓
		DISP_DATAR3	P10_3	✓	✓
		DISP_DATAR4	P10_4	✓	✓
		DISP_DATAR5	P10_5	✓	✓
		DISP_DATAR6	P10_6	✓	✓
		DISP_DATAR7	P10_7	✓	✓
		DISP_DE	P09_7	✓	✓
		DISP_HSYNC	P09_5	✓	✓
		DISP_VSYNC	P09_6	✓	✓
SD/eMMC Host Interface	SDHI0	SD0_CD	P14_1	✓	✓
			P22_5	✓	✓
		SD0_CLK	P12_0	✓	✓
		SD0_CMD	P12_1	✓	✓
		SD0_DATA0	P12_2	✓	✓
		SD0_DATA1	P12_3	✓	✓
		SD0_DATA2	P12_4	✓	✓
		SD0_DATA3	P12_5	✓	✓
		SD0_DATA4	P12_6	✓	✓
		SD0_DATA5	P12_7	✓	✓
		SD0_DATA6	P13_0	✓	✓
		SD0_DATA7	P13_1	✓	✓
		SD0_IOVS	P02_6	✓	✓
			P14_7	✓	✓
		SD0_PWEN	P02_5	✓	✓
			P14_6	✓	✓
		SD0_RST#	P13_2	✓	✓
		SD0_WP	P14_2	✓	✓
			P22_6	✓	✓
	SDHI1	SD1_CD	P14_3	✓	✓
			P17_4	✓	✓
		SD1_CLK	P16_5	✓	✓

Table 17.2 List of multiplexed pin configurations (51 of 51)

Module/Function	Unit/Channel	Pin function	Allocation port	Package	
				729-pin	576-pin
SD/eMMC Host Interface	SDHI1	SD1_CMD	P16_6	✓	✓
		SD1_DATA0	P16_7	✓	✓
		SD1_DATA1	P17_0	✓	✓
		SD1_DATA2	P17_1	✓	✓
		SD1_DATA3	P17_2	✓	✓
		SD1_IOVS	P08_6	✓	✓
			P17_7	✓	✓
			P18_1	✓	✓
			P18_3	✓	✓
			P30_7	✓	✓
			P34_4	✓	✓
		SD1_PWEN	P08_5	✓	✓
			P17_6	✓	✓
			P18_0	✓	✓
			P18_2	✓	✓
			P34_3	✓	✓
		SD1_WP	P14_4	✓	✓
			P17_5	✓	✓
PCI Express Gen3	PCIE0	PCIE_RSTOUT0B	P33_3	✓	✓
	PCIE1	PCIE_RSTOUT1B	P33_4	✓	✓

Note: ✓: Available
—: Not available

17.3 Register Map

Table 17.3 I/O Port register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x812C_0000 + 0x001 × m (m = 00 to 12) 0x802B_0000 + 0x001 × m (m = 00 to 12) 0x802C_0000 + 0x001 × m (m = 13 to 35)	Pm	Port m Register (m = 00 to 35)	—
0x812C_0200 + 0x002 × m (m = 00 to 12) 0x802B_0200 + 0x002 × m (m = 00 to 12) 0x802C_0200 + 0x002 × m (m = 13 to 35)	PMm	Port m Mode Register (m = 00 to 35)	—
0x812C_0400 + 0x001 × m (m = 00 to 12) 0x802B_0400 + 0x001 × m (m = 00 to 12) 0x802C_0400 + 0x001 × m (m = 13 to 35)	PMCm	Port m Mode Control Register (m = 00 to 35)	PRCRS.PRCS2 (m = 00 to 12) PRCRN.PRCN2 (m = 13 to 35)
0x812C_0600 + 0x008 × m (m = 00 to 12) 0x802B_0600 + 0x008 × m (m = 00 to 12) 0x802C_0600 + 0x008 × m (m = 13 to 35)	PFCm	Port m Function Control Register (m = 00 to 35)	PRCRS.PRCS2 (m = 00 to 12) PRCRN.PRCN2 (m = 13 to 35)
0x812C_0800 + 0x001 × m (m = 00 to 12) 0x802B_0800 + 0x001 × m (m = 00 to 12) 0x802C_0800 + 0x001 × m (m = 13 to 35)	PINm	Port m Input Register (m = 00 to 35)	—
0x812C_0A00 + 0x008 × m (m = 00 to 12) 0x802B_0A00 + 0x008 × m (m = 00 to 12) 0x802C_0A00 + 0x008 × m (m = 13 to 35)	DRCTLm	I/O Buffer m Function Switching Register (m = 00 to 35)	PRCRS.PRCS2 (m = 00 to 12) PRCRN.PRCN2 (m = 13 to 35)
0x812C_0C00 + 0x001 × m (m = 00 to 12) 0x802B_0C00 + 0x001 × m (m = 00 to 12)	RSELPm	Port m Region Select Register (m = 00 to 12)	PRCRS.PRCS2

Table 17.3 I/O Port register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x812C_1000 + 0x001 × m (m = 00 to 12) 0x802B_1000 + 0x001 × m (m = 00 to 12) 0x802C_1000 + 0x001 × m (m = 13 to 35)	SLPORTm	Port m Access Control Register (m = 00 to 35)	PRCRS.PRCS2 (m = 00 to 12) PRCRN.PRCN2 (m = 13 to 35)
0x812C_1C00 0x802B_1C00	SLRSELP	RSELP Access Control Register	PRCRS.PRCS2
0x812C_1F00 0x802B_1F00	SLPSR	Port Security Register Access Control Register	PRCRS.PRCS2
0x802C_1F00	SLPSRNS	Port NS Security Register Access Control Register	PRCRN.PRCN2
0x812C_1F04 0x802B_1F04	RSELPSR	Port Security Register Access Region Select Register	PRCRS.PRCS2

Table 17.4 I/O Port related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
Pm, PMm, PMCm, PFCm, PINm, DRCTLm	—	—	SLPORTm.SL
RSELPm	—	—	SLRSELP.SL
SLPORTm (m = 00 to 12), SLRSELP, RSELPSR	—	—	SLPSR.SL
SLPORTm (m = 13 to 35)	—	—	SLPSRNS.SL
SLPSR, SLPSRNS	—	—	Fixed to 3

17.4 Register Descriptions

17.4.1 Pm : Port m Register (m = 00 to 35)

Base address: PORT_SRS = 0x812C_0000 (m = 00 to 12)
PORT_SRN = 0x802B_0000 (m = 00 to 12)
PORT_NSR = 0x802C_0000 (m = 13 to 35)

Offset address: 0x000 + 0x001 × m

Bit position:	7	6	5	4	3	2	1	0
Bit field:	POUT_7	POUT_6	POUT_5	POUT_4	POUT_3	POUT_2	POUT_1	POUT_0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W[*1]
7:0	POUT_0 to POUT_7	Pm_n Output Data Store (n: bit position) 0: Low output 1: High output	R/W

Note: Available pins depend on port group. For details, see Table 17.1.
Note 1. If access address is not region selected by RSELPm register, the bit is read only. (m = 00 to 12)

The Pm register holds the data to be output from the pins used for general I/O.

17.4.2 PMm : Port m Mode Register (m = 00 to 35)

Base address: PORT_SRS = 0x812C_0000 (m = 00 to 12)
PORT_SRN = 0x802B_0000 (m = 00 to 12)
PORT_NSR = 0x802C_0000 (m = 13 to 35)

Offset address: 0x200 + 0x002 × m

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PM7[1:0]		PM6[1:0]		PM5[1:0]		PM4[1:0]		PM3[1:0]		PM2[1:0]		PM1[1:0]		PM0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W*1
1:0	PM0[1:0]	Pm_0 I/O Select 0 0: Hi-Z 0 1: Input 1 0: Output (Output data is not input to input buffer) 1 1: Output (Output data is input to input buffer)	R/W
3:2	PM1[1:0]	Pm_1 I/O Select This function is the same as PM0[1:0].	R/W
5:4	PM2[1:0]	Pm_2 I/O Select This function is the same as PM0[1:0].	R/W
7:6	PM3[1:0]	Pm_3 I/O Select This function is the same as PM0[1:0].	R/W
9:8	PM4[1:0]	Pm_4 I/O Select This function is the same as PM0[1:0].	R/W
11:10	PM5[1:0]	Pm_5 I/O Select This function is the same as PM0[1:0].	R/W
13:12	PM6[1:0]	Pm_6 I/O Select This function is the same as PM0[1:0].	R/W
15:14	PM7[1:0]	Pm_7 I/O Select This function is the same as PM0[1:0].	R/W

Note: Available pins depend on port group. For details, see Table 17.1.
Note 1. If access address is not region selected by RSELPm register, the bit is read only. (m = 00 to 12)

The PMm register is used to select I/O function for the pins used for general I/O.

17.4.3 PMCm : Port m Mode Control Register (m = 00 to 35)

Base address: PORT_SRS = 0x812C_0000 (m = 00 to 12)
PORT_SRN = 0x802B_0000 (m = 00 to 12)
PORT_NSR = 0x802C_0000 (m = 13 to 35)

Offset address: 0x400 + 0x001 × m

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0
Value after reset:	0	0	0*2	0*2	0*2	0*2	0*2	0

Bit	Symbol	Function	R/W*1
7:0	PMC0 to PMC7	Pm_n Pin Mode Control (n: bit position) 0: Uses the pin as a general I/O pin. 1: Uses the pin as an I/O port for peripheral functions.	R/W

Note: Available pins depend on port group. For details, see Table 17.1.
Note 1. If access address is not region selected by RSELPm register, the bit is read only. (m = 00 to 12)
Note 2. Value after reset is 1 for PMC1 to PMC5 bits in PMC08 register since JTAG (TMS, TDI, TDO and TCK) and RSTOUT# pins are selected at initial state.

The PMCm register is used to select the function of the pins of the port.

17.4.4 PFCm : Port m Function Control Register (m = 00 to 35)

Base address: PORT_SRS = 0x812C_0000 (m = 00 to 12)
PORT_SRN = 0x802B_0000 (m = 00 to 12)
PORT_NSR = 0x802C_0000 (m = 13 to 35)

Offset address: 0x600 + 0x008 × m

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	—	—	PFC7[5:0]						—	—	PFC6[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	—	—	PFC5[5:0]						—	—	PFC4[5:0]					
Value after reset:	0	0	0	0	0	0	0	0*2	0	0	0	0	0	0	0	0*2

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	PFC3[5:0]						—	—	PFC2[5:0]					
Value after reset:	0	0	0	0	0	0	0	0*2	0	0	0	0	0	0	0	0*2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	PFC1[5:0]						—	—	PFC0[5:0]					
Value after reset:	0	0	0	0	0	0	0	0*2	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W*1
5:0	PFC0[5:0]	Pm_0 Pin function Select Select one function from maximum 64 shared functions. For details of pin multiplex mapping, see section 17.5. Multiplexed Pin Configurations.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	PFC1[5:0]	Pm_1 Pin function Select This function is the same as PFC0[5:0].	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
21:16	PFC2[5:0]	Pm_2 Pin function Select This function is the same as PFC0[5:0].	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
29:24	PFC3[5:0]	Pm_3 Pin function Select This function is the same as PFC0[5:0].	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W
37:32	PFC4[5:0]	Pm_4 Pin function Select This function is the same as PFC0[5:0].	R/W
39:38	—	These bits are read as 0. The write value should be 0.	R/W
45:40	PFC5[5:0]	Pm_5 Pin function Select This function is the same as PFC0[5:0].	R/W
47:46	—	These bits are read as 0. The write value should be 0.	R/W
53:48	PFC6[5:0]	Pm_6 Pin function Select This function is the same as PFC0[5:0].	R/W
55:54	—	These bits are read as 0. The write value should be 0.	R/W
61:56	PFC7[5:0]	Pm_7 Pin function Select This function is the same as PFC0[5:0].	R/W
63:62	—	These bits are read as 0. The write value should be 0.	R/W

Note: Available pins depend on port group. For details, see Table 17.1.
Note 1. If access address is not region selected by RSELPm register, the bit is read only. (m = 00 to 12)

Note 2. Value after reset is 1 for PFC1[0] to PMC5[0] bits in PFC08 register since JTAG (TMS, TDI, TDO and TCK) and RSTOUT# pins are selected at initial state.

The PFCm register is used to select peripheral function for the pins.

17.4.5 PINm : Port m Input Register (m = 00 to 35)

Base address: PORT_SRS = 0x812C_0000 (m = 00 to 12)
PORT_SRN = 0x802B_0000 (m = 00 to 12)
PORT_NSR = 0x802C_0000 (m = 13 to 35)

Offset address: 0x800 + 0x001 × m

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PIN0 to PIN7	Pm_n Pin Input (n: bit position) 0: Low input 1: High input	R

Note: Available pins depend on port group. For details, see Table 17.1.

The PINm register reflects the states of the individual input port pins.

17.4.6 DRCTLm : I/O Buffer m Function Switching Register (m = 00 to 35)

Base address: PORT_SRS = 0x812C_0000 (m = 00 to 12)
PORT_SRN = 0x802B_0000 (m = 00 to 12)
PORT_NSR = 0x802C_0000 (m = 13 to 35)

Offset address: 0xA00 + 0x008 × m

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	—	—	SR7	SMT7	PUD7[1:0]		DRV7[1:0]		—	—	SR6	SMT6	PUD6[1:0]		DRV6[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	—	—	SR5	SMT5	PUD5[1:0]		DRV5[1:0]		—	—	SR4	SMT4	PUD4[1:0]		DRV4[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	SR3	SMT3	PUD3[1:0]		DRV3[1:0]		—	—	SR2	SMT2	PUD2[1:0]		DRV2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	SR1	SMT1	PUD1[1:0]		DRV1[1:0]		—	—	SR0	SMT0	PUD0[1:0]		DRV0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W*1
1:0	DRV0[1:0]	Pm_0 Driving Ability Control 0 0: Low 0 1: Middle 1 0: High 1 1: Ultra High	R/W

Bit	Symbol	Function	R/W*1
3:2	PUD0[1:0]	Pm_0 Pull-Up/Down Control 0 0: No pull-up/down 0 1: Pull-up 1 0: Pull-down 1 1: Setting prohibited	R/W
4	SMT0	Pm_0 Schmitt Trigger Control 0: Disable Schmitt Trigger input 1: Enable Schmitt Trigger input	R/W
5	SR0	Pm_0 Slew Rate Control 0: Slow 1: Fast	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DRV1[1:0]	Pm_1 Driving Ability Control This function is the same as DRV0[1:0].	R/W
11:10	PUD1[1:0]	Pm_1 Pull-Up/Down Control This function is the same as PUD0[1:0].	R/W
12	SMT1	Pm_1 Schmitt Trigger Control This function is the same as SMT0.	R/W
13	SR1	Pm_1 Slew Rate Control This function is the same as SR0.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
17:16	DRV2[1:0]	Pm_2 Driving Ability Control This function is same as DRV0[1:0].	R/W
19:18	PUD2[1:0]	Pm_2 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W
20	SMT2	Pm_2 Schmitt Trigger Control This function is the same as SMT0.	R/W
21	SR2	Pm_2 Slew Rate Control This function is the same as SR0.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	DRV3[1:0]	Pm_3 Driving Ability Control This function is same as DRV0[1:0].	R/W
27:26	PUD3[1:0]	Pm_3 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W
28	SMT3	Pm_3 Schmitt Trigger Control This function is the same as SMT0.	R/W
29	SR3	Pm_3 Slew Rate Control This function is the same as SR0.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W
33:32	DRV4[1:0]	Pm_4 Driving Ability Control This function is same as DRV0[1:0].	R/W
35:34	PUD4[1:0]	Pm_4 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W
36	SMT4	Pm_4 Schmitt Trigger Control This function is the same as SMT0.	R/W
37	SR4	Pm_4 Slew Rate Control This function is the same as SR0.	R/W
39:38	—	These bits are read as 0. The write value should be 0.	R/W
41:40	DRV5[1:0]	Pm_5 Driving Ability Control This function is same as DRV0[1:0].	R/W
43:42	PUD5[1:0]	Pm_5 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W

Bit	Symbol	Function	R/W*1
44	SMT5	Pm_5 Schmitt Trigger Control This function is the same as SMT0.	R/W
45	SR5	Pm_5 Slew Rate Control This function is the same as SR0.	R/W
47:46	—	These bits are read as 0. The write value should be 0.	R/W
49:48	DRV6[1:0]	Pm_6 Driving Ability Control This function is same as DRV0[1:0].	R/W
51:50	PUD6[1:0]	Pm_6 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W
52	SMT6	Pm_6 Schmitt Trigger Control This function is the same as SMT0.	R/W
53	SR6	Pm_6 Slew Rate Control This function is the same as SR0.	R/W
55:54	—	These bits are read as 0. The write value should be 0.	R/W
57:56	DRV7[1:0]	Pm_7 Driving Ability Control This function is same as DRV0[1:0].	R/W
59:58	PUD7[1:0]	Pm_7 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W
60	SMT7	Pm_7 Schmitt Trigger Control This function is the same as SMT0.	R/W
61	SR7	Pm_7 Slew Rate Control This function is the same as SR0.	R/W
63:62	—	These bits are read as 0. The write value should be 0.	R/W

Note: Available pins depend on port group. For details, see Table 17.1.
Note 1. If access address is not region selected by RSELPm register, the bit is read only. (m = 00 to 12)

The DRCTLm register is used to control function of I/O buffers. This register is valid even if the pin is used as either general I/O pin or an I/O port for peripheral functions in PMCm register.

17.4.7 RSELPm : Port m Region Select Register (m = 00 to 12)

Base address: PORT_SRS = 0x812C_0000 (m = 00 to 12)
PORT_SRN = 0x802B_0000 (m = 00 to 12)

Offset address: 0xC00 + 0x001 × m

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RS0 to RS7	Pm_n pin I/O port registers Region Select (n = bit position) 0: Safety region (Address is 0x812C_0xxx) 1: Non safety region (Address is 0x802B_0xxx)	R/W*1

Note: Available pins depend on port group. For details, see Table 17.1.
Note 1. Access from 0x802B_0C00 + 0x001 × m is read only.

The RSELPm register is used to specify which is the attribution of the I/O port, Safety or Non safety. I/O ports registers except Non-safety dedicated I/O ports (m = 13 to 35) registers are defined two address, Safety region and Non safety region. When Safety region is selected in this register, access to Non safety region in I/O ports registers is limited to read-only though access to Safety region is allowed R/W. On the other hand, when Non safety region is selected in this register, access to Safety region in I/O ports registers is limited to read-only though access to Non safety region is allowed R/W. (Except PINm register due to read-only originally.)

17.4.8 SLPORTm : Port m Access Control Register (m = 00 to 35)

Base address: PORT_SRS = 0x812C_0000 (m = 00 to 12)
PORT_SRN = 0x802B_0000 (m = 00 to 12)
PORT_NSR = 0x802C_0000 (m = 13 to 35)

Offset address: 0x1000 + 0x001 × m

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SL[1:0]	Pm group I/O ports registers Security Level 0 0: Security Level 0 0 1: Security Level 1 1 0: Security Level 2 1 1: Security Level 3	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

The SLPORTm register is used to specify the security level for Pm group I/O port registers (Pm, PMm, PMCm, PFCm, PINm, and DRCTLm). I/O port registers for all the pins in the same port group are set to the same security level which is specified in this register.

17.4.9 SLRSELP : RSELP Access Control Register

Base address: PORT_SRS = 0x812C_0000
PORT_SRN = 0x802B_0000

Offset address: 0x1C00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SL[1:0]	RSELPm (m = 00 to 12) register Security Level 0 0: Security Level 0 0 1: Security Level 1 1 0: Security Level 2 1 1: Security Level 3	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

The SLRSELP register is used to specify the security level for RSELPm registers (m = 00 to 12). All RSELPm registers (m = 00 to 12) are set to the same security level which is specified in this register.

17.4.10 SLPSR : Port Security Register Access Control Register

Base address: PORT_SRS = 0x812C_0000
PORT_SRN = 0x802B_0000

Offset address: 0x1F00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
1:0	SL[1:0]	Port security control register Security Level 0 0: Security Level 0 0 1: Security Level 1 1 0: Security Level 2 1 1: Security Level 3	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The SLPSR register is used to specify the security level for Port security control registers (SLPORTm (m = 00 to 12), SLRSELP, and RSELPSR). All the target registers are set to the same security level which is specified in this register.

17.4.11 SLPSRNS : Port NS Security Register Access Control Register

Base address: PORT_NSR = 0x802C_0000

Offset address: 0x1F00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
1:0	SL[1:0]	Port security control register and ELC security control register Security Level 0 0: Security Level 0 0 1: Security Level 1 1 0: Security Level 2 1 1: Security Level 3	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The SLPSRNS register is used to specify the security level for Port security control registers (SLPORTm (m = 13 to 35)) and ELC security control registers (SLELC_PGRC, SLELC_PDBFn, SLELC_PEL and SLELC_DE). All the target registers are set to the same security level which is specified in this register.

17.4.12 RSELPSR : Port Security Register Access Region Select Register

Base address: PORT_SRS = 0x812C_0000
PORT_SRN = 0x802B_0000

Offset address: 0x1F04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RS	Port security control register Region Select 0: Safety region (Address is 0x812C_1xxx) 1: Non-safety region (Address is 0x802B_1xxx)	R/W*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access from 0x802B_1F04 is read only.

The RSELPSR register is used to specify which is the attribution of the Port security control registers (SLPORTm, SLRSELP, and SLPSR), Safety or Non-safety. Port security control registers are defined two address, Safety region and Non-safety region. When Safety region is selected in this register, access to Non-safety region in Port security control registers is limited to read-only though access to Safety region is allowed R/W. On the other hand, when Non-safety region is selected in this register, access to Safety region in Port security control registers is limited to read-only though access to Non-safety region is allowed R/W.

17.5 Multiplexed Pin Configurations

Select the function number with PFCn bit of Port m Function Control Register (PFCm: m = 00 to 35).

One of three types of IO buffers, Type A, B, or C is used each port. Depending on the IO type, recommended DRCTLm register setting may be different each peripheral function. For details, see Table 58.11.

Each port belongs to one voltage domain. Voltage domain VCC1833_n (n = 0 to 7) are selectable from 1.8 V or 3.3 V according to the selected peripheral function. When using Ethernet with RGMII, 1.8 V xSPI, or 1.8 V eMMC, supply 1.8 V to the corresponding voltage domain (VCC1833_n). When using SD, corresponding voltage domain (VCC1833_n) can be changed between 1.8 V and 3.3 V during operation. In other cases, supply 3.3 V to the corresponding voltage domain (VCC1833_n). Note that 1.8 V operation is not supported for other peripheral function than Ethernet with RGMII, xSPI, and SDHI.

17.5.1 Port 00 Function Assignment

Table 17.5 Port 00 function assignment (1 of 2)

Function (PFC)	Port	P00_0	P00_1	P00_2	P00_3	P00_4	P00_5	P00_6	P00_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	SEI	IRQ0	IRQ1	IRQ2	IRQ3	—	IRQ4	IRQ5
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	D0	D1	D2	D3	D4	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	MTIOC4B	MTIOC4D	MTCLKA	MTCLKB
	0x07	—	—	—	—	—	MTIOC8C	MTIOC8D	MTIOC1B
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC00_0A	GTIOC00_0B	GTIOC00_1A	GTIOC00_1B	GTIOC00_2A	GTIOC00_2B	GTIOC00_3A	GTIOC00_3B
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	ETH3_TXER	ETH3_RXER	ETH3_CRS	ETH3_COL	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	USB_VBUSEN	USB_OVRCUR	USB_EXICEN	—	—	USB_VBUSEN	USB_OVRCUR	USB_EXICEN
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	—	—	—	—	SCKE00	DEE00
	0x16	—	—	—	—	—	—	SCKE04	DEE04
IIC	0x17	—	—	—	—	—	—	IIC_SCL0	IIC_SDA0
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—

Table 17.5 Port 00 function assignment (2 of 2)

Function (PFC)	Port	P00_0	P00_1	P00_2	P00_3	P00_4	P00_5	P00_6	P00_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	ADTRG0#	ADTRG1#	ADTRG2#	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	ENCIFCK00	ENCIFOE00
	0x23	—	—	—	—	—	—	ENCIFCK04	ENCIFOE04
ENDAT	0x24	DUEI00	TST_OUT00	SI00#	DUEI01	TST_OUT01	SI01#	—	—
HDSL	0x25	HDSL00_LINK	HDSL00_SMPL	HDSL00_CLK1	HDSL00_SEL1	HDSL00_MISO1	HDSL00_MOSI1	HDSL00_CLK2	HDSL00_SEL2
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.2 Port 01 Function Assignment

Table 17.6 Port 01 function assignment (1 of 2)

Function (PFC)	Port	P01_0	P01_1	P01_2	P01_3	P01_4	P01_5	P01_6	P01_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5
Interrupt	0x00	IRQ6	—	—	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	MTIOC3A	MTIOC3C	MTIOC6B	MTIOC6D	MTIOC7A	MTIOC7C	MTIOC7B	MTIOC7D
	0x07	MTIOC1A	MTIOC8A	MTIOC8B	MTIC5U	MTIC5V	MTIC5W	MTIOC0A	MTIOC0B
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC00_4A	GTIOC00_4B	GTIOC01_0A	GTIOC01_0B	GTIOC01_1A	GTIOC01_1B	GTIOC01_2A	GTIOC01_2B
	0x0A	GTIOC00_2B	—	GTIOC04_0A	GTIOC04_0B	GTIOC04_1A	GTIOC04_1B	GTIOC04_2A	GTIOC04_2B
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—

Table 17.6 Port 01 function assignment (2 of 2)

Function (PFC)	Port	P01_0	P01_1	P01_2	P01_3	P01_4	P01_5	P01_6	P01_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	TXDE00	RXDE00	—	—	—	—	—	—
	0x16	TXDE04	RXDE04	—	—	—	—	—	—
IIC	0x17	IIC_SCL1	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	XSPI1_CKP	XSPI1_CS0#	XSPI1_CS1#	XSPI1_DS	XSPI1_IO0	XSPI1_IO1	XSPI1_IO2	XSPI1_IO3
DSMIF	0x1D	—	MCLK20	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	ENCIFDO00	ENCIFDI00	—	—	—	—	—	—
	0x23	ENCIFDO04	ENCIFDI04	—	—	—	—	—	—
ENDAT	0x24	—	—	DUEI02	TST_OUT02	SI02#	DUEI03	TST_OUT03	SI03#
HDSL	0x25	HDSL00_MISO2	HDSL00_MOSI2	HDSL01_LINK	HDSL01_SMPL	HDSL01_CLK1	HDSL01_SEL1	HDSL01_MISO1	HDSL01_MOSI1
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.3 Port 02 Function Assignment

Table 17.7 Port 02 function assignment (1 of 2)

Function (PFC)	Port	P02_0	P02_1	P02_2	P02_3	P02_4	P02_5	P02_6	P02_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type A	Type A	Type A
	Voltage domain	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD33	VDD33	VDD33
Interrupt	0x00	IRQ7	IRQ8	IRQ9	IRQ10	IRQ11	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	D5	D6	D7
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	MTCLKC	MTCLKD	MTIOC6A	MTIOC6C	—	MTIOC3B	MTIOC3D	MTIOC4A
	0x07	MTIOC0C	MTIOC0D	MTIOC1A	MTIOC1B	—	MTIOC8A	MTIOC8B	MTIC5U
POE3	0x08	—	—	—	—	POE0#	—	—	—

Table 17.7 Port 02 function assignment (2 of 2)

Function (PFC)	Port	P02_0	P02_1	P02_2	P02_3	P02_4	P02_5	P02_6	P02_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type A	Type A	Type A
	Voltage domain	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD1833_5	VDD33	VDD33	VDD33
GPT	0x09	GTIOC01_3A	GTIOC01_3B	GTIOC01_4A	GTIOC01_4B	—	GTIOC02_0A	GTIOC02_0B	GTIOC02_1A
	0x0A	—	—	—	—	—	GTADSM06_0	GTADSM06_1	GTADSM07_0
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	CMTW0_TIC0	CMTW0_TOC0	CMTW0_TIC1
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	ETH3_TXER	ETH3_RXER	ETH3_CRS	ETH3_COL	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	USB_VBUSEN	USB_OVRCUR	USB_EXICEN	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	SCKE01	DEE01	TXDE01	RXDE01	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	IIC_SDA1	IIC_SCL2	IIC_SDA2	IIC_SCL0	IIC_SDA0	IIC_SCL0	—	—
	0x18	—	—	—	IIC_SCL2	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	XSPI1_IO4	XSPI1_IO5	XSPI1_IO6	XSPI1_IO7	—	—	—	—
DSMIF	0x1D	MCLK21	MDAT21	MCLK22	MDAT22	MDAT20	MCLK00	MDAT00	MCLK01
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	MBX_HINT#	—	—	—
ENCIF	0x22	ENCIFCK01	ENCIFOE01	ENCIFDO01	ENCIFDI01	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	—	—	—	—	—	—	—
HDSL	0x25	HDSL01_CLK2	HDSL01_SEL2	HDSL01_MISO2	HDSL01_MOSI2	HDSL02_LINK	HDSL02_SMPL	HDSL02_CLK1	HDSL02_SEL1
ENCOUT	0x26	—	—	—	—	—	POUTA	POUTB	POUTZ
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	SD0_PWEN	SD0_IOVS	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.4 Port 03 Function Assignment

Table 17.8 Port 03 function assignment (1 of 2)

Function (PFC)	Port	P03_0	P03_1	P03_2	P03_3	P03_4	P03_5	P03_6	P03_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	—	—	IRQ12	IRQ13	IRQ14	IRQ15	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	D8	D9	D10	D11	D12	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	MTIOC4C	MTIOC4B	MTIOC4D	MTCLKA	MTCLKB	MTIOC3A	MTIOC3C	MTIOC6B
	0x07	MTIC5V	MTIOC1B	MTIOC1A	MTIOC8C	MTIOC8D	MTIC5W	MTIOC1A	MTIOC1B
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC02_1B	GTIOC02_2A	GTIOC02_2B	GTIOC02_3A	GTIOC02_3B	GTIOC02_4A	GTIOC02_4B	GTIOC03_0A
	0x0A	GTADSM07_1	GTADSM08_0	GTADSM08_1	GTADSM09_0	GTADSM09_1	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	CMTW0_TOC1	CMTW1_TIC0	CMTW1_TOC0	CMTW1_TIC1	CMTW1_TOC1	—	—	CMTW0_TIC0
RTC	0x0E	—	—	—	—	RTCAT1HZ	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	TXDE02	RXDE02	SCKE02	DEE02	TXDE02	RXDE02	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	IIC_SCL1	IIC_SDA1	IIC_SCL2	IIC_SDA2	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	MDAT01	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	ENCIFDO02	ENCIFDI02	ENCIFCK02	ENCIFOE02	ENCIFDO02	ENCIFDI02	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	—	—	—	—	—	—	DUEI04
HDSL	0x25	HDSL02_MISO1	HDSL02_MOSI1	HDSL02_CLK2	HDSL02_SEL2	HDSL02_MISO2	HDSL02_MOSI2	HDSL03_LINK	HDSL03_SMPL
ENCOUT	0x26	—	—	—	—	—	—	—	—

Table 17.8 Port 03 function assignment (2 of 2)

Function (PFC)	Port	P03_0	P03_1	P03_2	P03_3	P03_4	P03_5	P03_6	P03_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.5 Port 04 Function Assignment

Table 17.9 Port 04 function assignment (1 of 2)

Function (PFC)	Port	P04_0	P04_1	P04_2	P04_3	P04_4	P04_5	P04_6	P04_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	—	—	—	—	—	SEI	IRQ0	IRQ1
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	MTIOC6D	MTIOC7A	MTIOC7C	MTIOC7B	MTIOC7D	MTCLKC	MTCLKD	MTIOC6A
	0x07	—	—	—	—	—	MTIOC0C	MTIOC0D	MTIOC0A
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC03_0B	GTIOC03_1A	GTIOC03_1B	GTIOC03_2A	GTIOC03_2B	GTIOC03_3A	GTIOC03_3B	GTIOC03_4A
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	CMTW0_TOC0	CMTW0_TIC1	CMTW0_TOC1	CMTW1_TIC0	CMTW1_TOC0	CMTW1_TIC1	CMTW1_TOC1	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	—	—	—	SCKE03	DEE03	TXDE03
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	IIC_SCL0	IIC_SDA0	IIC_SCL1
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—

Table 17.9 Port 04 function assignment (2 of 2)

Function (PFC)	Port	P04_0	P04_1	P04_2	P04_3	P04_4	P04_5	P04_6	P04_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	ADTRG0#	ADTRG1#	ADTRG2#	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	MBX_HINT#	—
ENCIF	0x22	—	—	—	—	—	ENCIFCK03	ENCIFOE03	ENCIFDO03
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	TST_OUT04	SI04#	DUEI05	TST_OUT05	SI05#	—	—	—
HDSL	0x25	HDSL03_CLK1	HDSL03_SEL1	HDSL03_MISO1	HDSL03_MOSI1	HDSL03_CLK2	HDSL03_SEL2	HDSL03_MISO2	HDSL03_MOSI2
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.6 Port 05 Function Assignment

Table 17.10 Port 05 function assignment (1 of 2)

Function (PFC)	Port	P05_0	P05_1	P05_2	P05_3	P05_4	P05_5	P05_6	P05_7
	IO Type	Type A	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD33	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4
Interrupt	0x00	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	MTIOC6C	—	—	—	—	—	—	—
	0x07	MTIOC0B	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC03_4B	—	—	—	—	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—

Table 17.10 Port 05 function assignment (2 of 2)

Function (PFC)	Port	P05_0	P05_1	P05_2	P05_3	P05_4	P05_5	P05_6	P05_7
	IO Type	Type A	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD33	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	RXDE03	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	IIC_SDA1	—	IIC_SCL2	—	IIC_SDA2	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	XSPI0_CKP	XSPI0_CKN	XSPI0_CS0#	XSPI0_CS1#	XSPI0_DS	XSPI0_IO0	XSPI0_IO1
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	ENCIFDI03	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	DUEI06	TST_OUT06	SI06#	DUEI07	TST_OUT07	SI07#	DUEI08
HDSL	0x25	HDSL04_LINK	HDSL04_SMPL	HDSL04_CLK1	HDSL04_SEL1	HDSL04_MISO1	HDSL04_MOSI1	HDSL04_CLK2	HDSL04_SEL2
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.7 Port 06 Function Assignment

Table 17.11 Port 06 function assignment (1 of 2)

Function (PFC)	Port	P06_0	P06_1	P06_2	P06_3	P06_4	P06_5	P06_6	P06_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4
Interrupt	0x00	—	—	IRQ8	IRQ9	IRQ10	IRQ11	—	IRQ12
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	POE4#

Table 17.11 Port 06 function assignment (2 of 2)

Function (PFC)	Port	P06_0	P06_1	P06_2	P06_3	P06_4	P06_5	P06_6	P06_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4
GPT	0x09	—	—	—	—	—	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	GTETRGA	GTETRGB	GTETRGC	—	GTETRGD
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	GMAC1_MDC
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	IIC_SCL0	IIC_SDA0	IIC_SCL1	IIC_SDA1	—	IIC_SCL2
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	XSPI0_IO2	XSPI0_IO3	XSPI0_IO4	XSPI0_IO5	XSPI0_IO6	XSPI0_IO7	XSPI0_RESET0#	—
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	TST_OUT08	SI08#	DUEI09	TST_OUT09	SI09#	—	—	—
HDSL	0x25	HDSL04_MISO2	HDSL04_MOSI2	HDSL05_LINK	HDSL05_SMPL	HDSL05_CLK1	HDSL05_SEL1	—	HDSL05_MISO1
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.8 Port 07 Function Assignment

Table 17.12 Port 07 function assignment (1 of 2)

Function (PFC)	Port	P07_0	P07_1	P07_2	P07_3	P07_4	P07_5	P07_6	P07_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4
Interrupt	0x00	IRQ13	IRQ14	IRQ15	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	POE8#	POE10#	POE11#	—	—	—	—
GPT	0x09	—	—	—	—	—	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	GMAC1_MDIO	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	SCKE04	DEE04	TXDE04	RXDE04	SCKE05	DEE05	TXDE05
	0x16	—	SCKE08	DEE08	TXDE08	RXDE08	—	—	—
IIC	0x17	IIC_SDA2	IIC_SCL0	IIC_SDA0	IIC_SCL1	IIC_SDA1	IIC_SCL2	IIC_SDA2	IIC_SCL0
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	XSPI0_RESET1#	XSPI0_RSTO0#	XSPI0_RSTO1#	XSPI0_INT0#	XSPI0_INT1#	XSPI0_ECS0#	XSPI0_ECS1#	XSPI0_WP0#
DSMIF	0x1D	—	MCLK00	MDAT00	MCLK01	MDAT01	MCLK02	MDAT02	MCLK10
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	ENCIFCK04	ENCIFOE04	ENCIFDO04	ENCIFDI04	ENCIFCK05	ENCIFOE05	ENCIFDO05
	0x23	—	ENCIFCK12	ENCIFOE12	ENCIFDO12	ENCIFDI12	—	—	—
ENDAT	0x24	—	—	—	—	—	—	—	—
HDSL	0x25	HDSL05_MOSI1	HDSL05_CLK2	HDSL05_SEL2	HDSL05_MISO2	HDSL05_MOSI2	HDSL06_LINK	HDSL06_SMPL	HDSL06_CLK1
ENCOUT	0x26	—	—	—	—	—	—	—	—

Table 17.12 Port 07 function assignment (2 of 2)

Function (PFC)	Port	P07_0	P07_1	P07_2	P07_3	P07_4	P07_5	P07_6	P07_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4	VDD1833_4
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.9 Port 08 Function Assignment

Table 17.13 Port 08 function assignment (1 of 2)

Function (PFC)	Port	P08_0	P08_1	P08_2	P08_3	P08_4	P08_5	P08_6	P08_7
	IO Type	Type B	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD1833_4	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	—	—	—	—	—	IRQ8	SEI	IRQ0
SYS/DBG	0x01	—	TMS	TDI	TCK	TDO	RSTOUT#	—	—
Clock	0x02	—	—	—	—	—	—	CKIO	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	A0
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	—	—	—	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	GTIOC08_3A	GTIOC08_3B
POEG	0x0C	—	—	—	—	—	GTETRGSA	GTETRGSB	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	RTCAT1HZ	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	RXDE05	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	IIC_SDA0	—	—	—	—	IIC_SCL1	IIC_SDA1	IIC_SCL2
	0x18	—	—	—	—	—	—	—	IIC_SCL1
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	XSPI0_WP1#	—	—	—	—	—	—	—
DSMIF	0x1D	MDAT10	—	—	—	—	MCLK02	MDAT02	—
	0x1E	—	—	—	—	—	—	MCLK11	MDAT11

Table 17.13 Port 08 function assignment (2 of 2)

Function (PFC)	Port	P08_0	P08_1	P08_2	P08_3	P08_4	P08_5	P08_6	P08_7
	IO Type	Type B	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD1833_4	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	MBX_HINT#	—	—	—	—	—	—	—
ENCIF	0x22	ENCIFDI05	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	DUEI10	TST_OUT10	SI10#	—	—	DUEI11	TST_OUT11
HDSL	0x25	HDSL06_SEL1	HDSL06_MISO1	HDSL06_MOSI1	HDSL06_CLK2	HDSL06_SEL2	HDSL06_MISO2	HDSL06_MOSI2	HDSL07_LINK
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	SD1_PWEN	SD1_IOVS	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.10 Port 09 Function Assignment

Table 17.14 Port 09 function assignment (1 of 2)

Function (PFC)	Port	P09_0	P09_1	P09_2	P09_3	P09_4	P09_5	P09_6	P09_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	—	—	—	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	D13	D14	D15	WE0#
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	MTIOC6B	MTIOC6D	MTIOC7A	MTIOC7C
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	—	—	GTIOC04_0A	GTIOC04_0B	GTIOC04_1A	GTIOC04_1B
	0x0A	—	—	—	—	GTIOC10_0A	GTIOC10_0B	GTIOC10_1A	GTIOC10_1B
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—

Table 17.14 Port 09 function assignment (2 of 2)

Function (PFC)	Port	P09_0	P09_1	P09_2	P09_3	P09_4	P09_5	P09_6	P09_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	IIC_SDA2	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	MCLK12	MDAT12	MCLK20	MDAT20	MCLK70	MDAT70	MCLK71	MDAT71
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	SI11#	DUEI12	TST_OUT12	SI12#	DUEI13	TST_OUT13	SI13#	DUEI14
HDSL	0x25	HDSL07_SMPL	HDSL07_CLK1	HDSL07_SEL1	HDSL07_MISO1	HDSL07_MOSI1	HDSL07_CLK2	HDSL07_SEL2	HDSL07_MISO2
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	DISP_CLK	DISP_HSYNC	DISP_VSYNC	DISP_DE
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.11 Port 10 Function Assignment

Table 17.15 Port 10 function assignment (1 of 2)

Function (PFC)	Port	P10_0	P10_1	P10_2	P10_3	P10_4	P10_5	P10_6	P10_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	IRQ4	IRQ7	IRQ1	IRQ2	IRQ3	—	IRQ0	IRQ9
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	WE1#	WAIT#	CS0#	RD#	A1	A2	A3	A4
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	MTIOC7B	MTIOC7D	MTCLKC	MTCLKD	MTIOC1A	MTIOC1B	—	—
	0x07	—	—	MTIOC2A	MTIOC2B	—	MTIOC0A	MTIOC0B	MTIC5U
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC04_2A	GTIOC04_2B	GTIOC04_3A	GTIOC04_3B	GTIOC04_4A	GTIOC04_4B	GTIOC05_0A	GTIOC05_0B
	0x0A	GTIOC10_2A	GTIOC10_2B	GTIOC10_3A	GTIOC10_3B	—	—	—	—
	0x0B	—	—	—	—	—	—	—	GTIOC00_3A

Table 17.15 Port 10 function assignment (2 of 2)

Function (PFC)	Port	P10_0	P10_1	P10_2	P10_3	P10_4	P10_5	P10_6	P10_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	SCK0	RXD0/SCL0/ MISO0	TXD0/SDA0/ MOSI0	SS0#/CTS0#/ RTS0#	CTS0#	DE0	SCK1
SCIE	0x15	—	—	SCKE04	DEE04	TXDE04	RXDE04	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	MCLK72	MDAT72	MCLK10	MDAT10	MCLK11	MDAT11	MCLK21	MDAT21
	0x1E	—	—	MCLK00	MDAT00	MCLK01	MDAT01	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	ENCIFCK04	ENCIFOE04	ENCIFDO04	ENCIFDI04	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	TST_OUT14	SI14#	—	—	—	—	—	—
HDSL	0x25	HDSL07_MOSI2	HDSL08_LINK	HDSL08_SMPL	HDSL08_CLK1	HDSL08_SEL1	HDSL08_MISO1	HDSL08_MOSI1	HDSL08_CLK2
ENCOUT	0x26	—	—	—	—	—	—	POUTA	POUTB
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	DISP_DATAR0	DISP_DATAR1	DISP_DATAR2	DISP_DATAR3	DISP_DATAR4	DISP_DATAR5	DISP_DATAR6	DISP_DATAR7
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.12 Port 11 Function Assignment

Table 17.16 Port 11 function assignment (1 of 3)

Function (PFC)	Port	P11_0	P11_1	P11_2	P11_3	P11_4	P11_5	P11_6	P11_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	IRQ13	IRQ4	IRQ5	IRQ6	IRQ7	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—

Table 17.16 Port 11 function assignment (2 of 3)

Function (PFC)	Port	P11_0	P11_1	P11_2	P11_3	P11_4	P11_5	P11_6	P11_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	A5	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	—	—	—	—	GTIOC05_0A	GTIOC05_0B
	0x0A	—	—	—	—	—	—	—	—
	0x0B	GTIOC00_3B	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	ESC_RESETOUT#	ESC_LEDRUN	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	RXD1/SCL1/MISO1	TXD1/SDA1/MOSI1	SS1#/CTS1#/RTS1#	CTS1#	DE1	—	—	—
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	MCLK22	MDAT22	MCLK30	MDAT30	MCLK31	MDAT31	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	DUEI15	TST_OUT15	SI15#	—	DUEI00	TST_OUT00	SI00#
HDSL	0x25	HDSL08_SEL2	HDSL08_MISO2	HDSL08_MOSI2	HDSL09_LINK	HDSL09_SMPL	HDSL09_CLK1	HDSL09_SEL1	HDSL09_MISO1
ENCOUT	0x26	POUTZ	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	DISP_DATAG0	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—

Table 17.16 Port 11 function assignment (3 of 3)

Function (PFC)	Port	P11_0	P11_1	P11_2	P11_3	P11_4	P11_5	P11_6	P11_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.13 Port 12 Function Assignment

Table 17.17 Port 12 function assignment (1 of 2)

Function (PFC)	Port	P12_0	P12_1	P12_2	P12_3	P12_4	P12_5	P12_6	P12_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6
Interrupt	0x00	—	—	—	—	IRQ1	—	—	IRQ2
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	D16	D17	D18	D19	D20	D21	D22	D23
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	MTIC5V	MTIC5W	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC05_1A	GTIOC05_1B	GTIOC05_2A	GTIOC05_2B	GTIOC05_3A	GTIOC05_3B	GTIOC05_4A	GTIOC05_4B
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	GTIOC01_3A	GTIOC01_3B	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	CMTW0_TIC0	CMTW0_TOC0	CMTW0_TIC1	CMTW0_TOC1	CMTW1_TIC0	CMTW1_TOC0	CMTW1_TIC1	CMTW1_TOC1
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	SCK2	RXD2/SCL2/MISO2	TXD2/SDA2/MOSI2	SS2#/CTS2#/RTS2#	CTS2#
SCIE	0x15	—	—	—	—	SCKE05	DEE05	TXDE05	RXDE05
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	CANRX1	CANTX1	CANRXDP1	CANTXDP1	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	MCLK02	MDAT02	MCLK10	MDAT10
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—

Table 17.17 Port 12 function assignment (2 of 2)

Function (PFC)	Port	P12_0	P12_1	P12_2	P12_3	P12_4	P12_5	P12_6	P12_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	ENCIFCK05	ENCIFOE05	ENCIFDO05	ENCIFDI05
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	DUEI01	TST_OUT01	SI01#	—	—	—	—	—
HDSL	0x25	HDSL09_MOSI1	HDSL09_CLK2	HDSL09_SEL2	HDSL09_MISO2	HDSL09_MOSI2	HDSL10_LINK	HDSL10_SMPL	HDSL10_CLK1
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	SD0_CLK	SD0_CMD	SD0_DATA0	SD0_DATA1	SD0_DATA2	SD0_DATA3	SD0_DATA4	SD0_DATA5
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.14 Port 13 Function Assignment

Table 17.18 Port 13 function assignment (1 of 2)

Function (PFC)	Port	P13_0	P13_1	P13_2	P13_3	P13_4	P13_5	P13_6	P13_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6
Interrupt	0x00	—	—	IRQ3	—	—	IRQ4	—	IRQ14
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	D24	D25	D26	D27	D28	D29	D30	D31
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	—	—	—	GTIOC06_3A	GTIOC06_3B	GTIOC06_4A
	0x0A	—	—	—	—	—	—	—	—
	0x0B	GTIOC02_3A	GTIOC02_3B	—	GTIOC03_3A	GTIOC03_3B	—	GTIOC04_3A	GTIOC04_3B
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	DE2	—	—	SCK3	RXD3/SCL3/MISO3	TXD3/SDA3/MOSI3	SS3#/CTS3#/RTS3#	CTS3#

Table 17.18 Port 13 function assignment (2 of 2)

Function (PFC)	Port	P13_0	P13_1	P13_2	P13_3	P13_4	P13_5	P13_6	P13_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6	VDD1833_6
SCIE	0x15	SCKE08	DEE08	TXDE08	RXDE08	SCKE09	DEE09	TXDE09	RXDE09
	0x16	SCKE03	DEE03	TXDE03	RXDE03	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	SPI_RSPCK3	SPI_MOSI3	SPI_MISO3	SPI_SSL30	SPI_SSL31	SPI_SSL32	SPI_SSL23	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	MCLK00	MDAT00	MCLK01	MDAT01	MCLK40	MDAT40	MCLK41	MDAT41
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	ENCIFCK12	ENCIFOE12	ENCIFDO12	ENCIFDI12	ENCIFCK13	ENCIFOE13	ENCIFDO13	ENCIFDI13
	0x23	ENCIFCK03	ENCIFOE03	ENCIFDO03	ENCIFDI03	—	—	—	—
ENDAT	0x24	—	—	—	—	—	—	—	—
HDSL	0x25	HDSL10_SEL1	HDSL10_MISO1	HDSL10_MOSI1	HDSL10_CLK2	HDSL10_SEL2	HDSL10_MISO2	HDSL10_MOSI2	HDSL11_LINK
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	SD0_DATA6	SD0_DATA7	SD0_RST#	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.15 Port 14 Function Assignment

Table 17.19 Port 14 function assignment (1 of 2)

Function (PFC)	Port	P14_0	P14_1	P14_2	P14_3	P14_4	P14_5	P14_6	P14_7
	IO Type	Type B	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD1833_6	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	IRQ5	—	—	IRQ6	—	—	IRQ8	IRQ9
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	A0	RD/WR#	BS#	—	—	—	—	—
DMAC	0x05	—	—	—	DREQ	DACK	TEND	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	POE0#	POE4#	POE8#	POE10#	POE11#
GPT	0x09	GTIOC06_4B	GTIOC06_0A	GTIOC06_0B	GTIOC06_1A	GTIOC06_1B	GTIOC06_2A	GTIOC06_2B	—
	0x0A	—	GTIOC09_0A	GTIOC09_0B	GTIOC09_1A	GTIOC09_1B	GTIOC09_2A	GTIOC09_2B	GTIOC09_3A
	0x0B	—	GTIOC05_3A	GTIOC05_3B	—	GTIOC06_3A	GTIOC06_3B	—	—

Table 17.19 Port 14 function assignment (2 of 2)

Function (PFC)	Port	P14_0	P14_1	P14_2	P14_3	P14_4	P14_5	P14_6	P14_7
	IO Type	Type B	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD1833_6	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	CMTW0_TIC0	CMTW0_TOC0	CMTW0_TIC1	CMTW0_TOC1
RTC	0x0E	—	RTCAT1HZ	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	ETHSW_PTPOUT2	—	—	—	—	—	—	—
ESC	0x12	ESC_SYNC0	—	—	ESC_LINKACT2	ESC_IRQ	ESC_RESETOUT#	ESC_I2CCLK	ESC_I2CDATA
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	DE3	SCK4	RXD4/SCL4/MISO4	TXD4/SDA4/MOSI4	SS4#/CTS4#/RTS4#	CTS4#	DE4	—
SCIE	0x15	—	—	SCKE00	DEE00	TXDE00	RXDE00	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	IIC_SCL0	IIC_SDA0
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	MCLK42	MDAT42	—	—	—	—	—	MCLK32
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	MBX_HINT#	—	—	—
ENCIF	0x22	—	—	ENCIFCK00	ENCIFOE00	ENCIFDO00	ENCIFDI00	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	DUEI02	—	—	—	—	TST_OUT02	SI02#
HDSL	0x25	HDSL11_SMPL	HDSL11_CLK1	HDSL11_SEL1	HDSL11_MISO1	HDSL11_MOSI1	HDSL11_CLK2	HDSL11_SEL2	HDSL11_MISO2
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	DISP_DATAG1	DISP_DATAG2	DISP_DATAG3	DISP_DATAG4	DISP_DATAG5	DISP_DATAG6	—
SDHI	0x29	—	SD0_CD	SD0_WP	SD1_CD	SD1_WP	—	SD0_PWEN	SD0_IOVS
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.16 Port 15 Function Assignment

Table 17.20 Port 15 function assignment (1 of 2)

Function (PFC)	Port	P15_0	P15_1	P15_2	P15_3	P15_4	P15_5	P15_6	P15_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	—	—	—	—	—	IRQ0	IRQ1	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	—	—	—	—	—	—
	0x0A	GTIOC09_3B	GTIOC09_4A	GTIOC09_4B	GTIOC09_5A	GTIOC09_5B	GTIOC09_6A	GTIOC09_6B	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	SS5#/CTS5#/ RTS5#
SCIE	0x15	SCKE06	DEE06	TXDE06	RXDE06	SCKE07	DEE07	TXDE07	RXDE07
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	MDAT32	MCLK40	MDAT40	MCLK41	MDAT41	MCLK42	MDAT42	MCLK50
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	ENCIFCK06	ENCIFOE06	ENCIFDO06	ENCIFDI06	ENCIFCK07	ENCIFOE07	ENCIFDO07	ENCIFDI07
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	—	—	—	—	—	—	—
HDSL	0x25	HDSL11_MOSI2	HDSL12_LINK	HDSL12_SMPL	HDSL12_CLK1	HDSL12_SEL1	HDSL12_MISO1	HDSL12_MOSI1	HDSL12_CLK2
ENCOUT	0x26	—	—	—	—	—	—	—	—

Table 17.20 Port 15 function assignment (2 of 2)

Function (PFC)	Port	P15_0	P15_1	P15_2	P15_3	P15_4	P15_5	P15_6	P15_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.17 Port 16 Function Assignment

Table 17.21 Port 16 function assignment (1 of 2)

Function (PFC)	Port	P16_0	P16_1	P16_2	P16_3	P16_4	P16_5	P16_6	P16_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type B	Type B	Type B
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD1833_7	VDD1833_7	VDD1833_7
Interrupt	0x00	IRQ2	—	—	IRQ10	IRQ11	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	—	—	—	GTIOC03_0A	GTIOC03_0B	GTIOC03_1A
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	GTETRGSA	GTETRGSB	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	ESC_LINKACT0	ESC_LINKACT1	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	CTS5#	DE5	SCK5	RXD5/SCL5/MISO5	TXD5/SDA5/MOSI5	—	—	—
SCIE	0x15	TXDE07	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—

Table 17.21 Port 16 function assignment (2 of 2)

Function (PFC)	Port	P16_0	P16_1	P16_2	P16_3	P16_4	P16_5	P16_6	P16_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type B	Type B	Type B
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD1833_7	VDD1833_7	VDD1833_7
DSMIF	0x1D	MDAT50	MCLK51	MDAT51	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	DUEI03	TST_OUT03	SI03#	DUEI04	TST_OUT04	SI04#	DUEI05	TST_OUT05
HDSL	0x25	HDSL12_SEL2	HDSL12_MISO2	HDSL12_MOSI2	HDSL13_LINK	HDSL13_SMPL	HDSL13_CLK1	HDSL13_SEL1	HDSL13_MISO1
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	SD1_CLK	SD1_CMD	SD1_DATA0
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.18 Port 17 Function Assignment

Table 17.22 Port 17 function assignment (1 of 2)

Function (PFC)	Port	P17_0	P17_1	P17_2	P17_3	P17_4	P17_5	P17_6	P17_7
	IO Type	Type B	Type B	Type B	Type B	Type A	Type A	Type A	Type A
	Voltage domain	VDD1833_7	VDD1833_7	VDD1833_7	VDD1833_7	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	IRQ12	IRQ13	IRQ14	IRQ15	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	A6	A7	WE2#	WE3#/AH#
DMAC	0x05	—	—	—	—	DREQ	DACK	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC03_1B	GTIOC03_2A	GTIOC03_2B	—	—	—	—	—
	0x0A	—	—	—	—	GTADSM00_0	GTADSM00_1	GTADSM01_0	GTADSM01_1
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	GTETRGA	GTETRGB	GTETRGC	GTETRGD	—
CMTW	0x0D	—	—	—	—	CMTW1_TIC0	CMTW1_TOC0	CMTW1_TIC1	CMTW1_TOC1
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	ETHSW_PTPOUT0	ETHSW_PTPOUT1
ESC	0x12	—	—	—	—	—	—	ESC_SYNC0	ESC_SYNC1

Table 17.22 Port 17 function assignment (2 of 2)

Function (PFC)	Port	P17_0	P17_1	P17_2	P17_3	P17_4	P17_5	P17_6	P17_7
	IO Type	Type B	Type B	Type B	Type B	Type A	Type A	Type A	Type A
	Voltage domain	VDD1833_7	VDD1833_7	VDD1833_7	VDD1833_7	VDD33	VDD33	VDD33	VDD33
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	DE0	SCK0	RXD0/SCL0/ MISO0	TXD0/SDA0/ MOSI0
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	CANRX0	CANTX0	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	SI05#	DUEI06	TST_OUT06	SI06#	DUEI07	TST_OUT07	SI07#	DUEI08
HDSL	0x25	HDSL13_MOSI1	HDSL13_CLK2	HDSL13_SEL2	HDSL13_MISO2	HDSL13_MOSI2	HDSL14_LINK	HDSL14_SMPL	HDSL14_CLK1
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	DISP_DATAG7	DISP_DATAB0
SDHI	0x29	SD1_DATA1	SD1_DATA2	SD1_DATA3	—	SD1_CD	SD1_WP	SD1_PWEN	SD1_IOVS
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.19 Port 18 Function Assignment

Table 17.23 Port 18 function assignment (1 of 2)

Function (PFC)	Port	P18_0	P18_1	P18_2	P18_3	P18_4	P18_5	P18_6	P18_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	IRQ7	IRQ15	SEI	IRQ0	IRQ1	IRQ2	IRQ3	IRQ4
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	A8	A9	A10	A11	A12	A13	A14	A15
DMAC	0x05	TEND	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—

Table 17.23 Port 18 function assignment (2 of 2)

Function (PFC)	Port	P18_0	P18_1	P18_2	P18_3	P18_4	P18_5	P18_6	P18_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
GPT	0x09	—	—	—	—	GTIOC07_3A	GTIOC07_3B	GTIOC07_4A	GTIOC07_4B
	0x0A	GTADSM02_0	GTADSM02_1	GTADSM03_0	GTADSM03_1	GTADSM04_0	GTADSM04_1	GTADSM05_0	GTADSM05_1
	0x0B	—	GTIOC07_3A	GTIOC07_3B	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	RTCAT1HZ	—	—	—	—
ETHSS	0x0F	—	—	ETH1_CRS	ETH1_COL	—	—	—	—
GMAC	0x10	—	—	GMAC1_MDC	GMAC1_MDIO	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	ETHSW_PTPOUT3
ESC	0x12	ESC_LEDRUN	ESC_LEDERR	—	—	ESC_LEDSTER	—	—	ESC_SYNC1
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	SS0#/CTS0#/RTS0#	CTS0#	SCK1	RXD1/SCL1/MISO1	TXD1/SDA1/MOSI1	SS1#/CTS1#/RTS1#	CTS1#	DE1
SCIE	0x15	—	—	—	—	SCKE09	DEE09	TXDE09	RXDE09
	0x16	—	—	—	—	SCKE10	DEE10	TXDE10	RXDE10
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	CANRXDP0	CANTXDP0	CANRX0	CANTX0	CANRX1	CANTX1	CANRXDP1	CANTXDP1
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	MCLK10	MDAT10	MCLK11	MDAT11	MCLK12	MDAT12
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	ENCIFCK13	ENCIFOE13	ENCIFDO13	ENCIFDI13
	0x23	—	—	—	—	ENCIFCK14	ENCIFOE14	ENCIFDO14	ENCIFDI14
ENDAT	0x24	TST_OUT08	SI08#	—	—	—	—	—	—
HDSL	0x25	HDSL14_SEL1	HDSL14_MISO1	HDSL14_MOSI1	HDSL14_CLK2	HDSL14_SEL2	HDSL14_MISO2	HDSL14_MOSI2	HDSL15_LINK
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	DISP_DATAB1	DISP_DATAB2	DISP_DATAB3	DISP_DATAB4	DISP_DATAB5	DISP_DATAB6	DISP_DATAB7	—
SDHI	0x29	SD1_PWEN	SD1_IOVS	SD1_PWEN	SD1_IOVS	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.20 Port 19 Function Assignment

Table 17.24 Port 19 function assignment (1 of 2)

Function (PFC)	Port	P19_0	P19_1	P19_2	P19_3	P19_4	P19_5	P19_6	P19_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	—	—	—	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC07_0A	GTIOC07_0B	GTIOC07_1A	GTIOC07_1B	GTIOC07_2A	GTIOC07_2B	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	—	MCLK52	MDAT52
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	DUEI09	TST_OUT09	SI09#	DUEI10	TST_OUT10	SI10#	—	—
HDSL	0x25	HDSL15_SMPL	HDSL15_CLK1	HDSL15_SEL1	HDSL15_MISO1	HDSL15_MOSI1	HDSL15_CLK2	HDSL15_SEL2	HDSL15_MISO2
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—

Table 17.24 Port 19 function assignment (2 of 2)

Function (PFC)	Port	P19_0	P19_1	P19_2	P19_3	P19_4	P19_5	P19_6	P19_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.21 Port 20 Function Assignment

Table 17.25 Port 20 function assignment (1 of 2)

Function (PFC)	Port	P20_0	P20_1	P20_2	P20_3	P20_4	P20_5	P20_6	P20_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0
Interrupt	0x00	—	—	—	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	—	—	—	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	ETH0_TXCLK	ETH0_TXD0	ETH0_TXD1	ETH0_TXD2	ETH0_TXD3	ETH0_TXEN	ETH0_RXCLK	ETH0_RXD0
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	CANRX0	CANTX0	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—

Table 17.25 Port 20 function assignment (2 of 2)

Function (PFC)	Port	P20_0	P20_1	P20_2	P20_3	P20_4	P20_5	P20_6	P20_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	—	—	—	—	DUEI11	TST_OUT11	SI11#
HDSL	0x25	HDSL15_MOSI2	—	—	—	—	HDSL00_LINK	HDSL00_SMPL	HDSL00_CLK1
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.22 Port 21 Function Assignment

Table 17.26 Port 21 function assignment (1 of 2)

Function (PFC)	Port	P21_0	P21_1	P21_2	P21_3	P21_4	P21_5	P21_6	P21_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0
Interrupt	0x00	—	—	—	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	ETH0_REFCLK
	0x03	—	—	—	—	—	—	—	RMII0_REFCLK
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	—	—	—	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	ETH0_RXD1	ETH0_RXD2	ETH0_RXD3	ETH0_RXDV	—	—	—	—
GMAC	0x10	—	—	—	—	GMAC0_MDC	GMAC0_MDIO	—	—
ETHSW	0x11	—	—	—	—	ETHSW_MDC	ETHSW_MDIO	ETHSW_PHYLINK0	—
ESC	0x12	—	—	—	—	ESC_MDC	ESC_MDIO	ESC_PHYLINK0	—
USB	0x13	—	—	—	—	—	—	—	—

Table 17.26 Port 21 function assignment (2 of 2)

Function (PFC)	Port	P21_0	P21_1	P21_2	P21_3	P21_4	P21_5	P21_6	P21_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0	VDD1833_0
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	CANRXDP0	CANTXDP0	—	CANRX1	CANTX1	CANRXDP1	CANTXDP1
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	DUEI12	TST_OUT12	SI12#	DUEI13	TST_OUT13	SI13#	—	—
HDSL	0x25	HDSL00_SEL1	HDSL00_MISO1	HDSL00_MOSI1	HDSL00_CLK2	HDSL00_SEL2	HDSL00_MISO2	HDSL00_MOSI2	HDSL01_LINK
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.23 Port 22 Function Assignment

Table 17.27 Port 22 function assignment (1 of 2)

Function (PFC)	Port	P22_0	P22_1	P22_2	P22_3	P22_4	P22_5	P22_6	P22_7
	IO Type	Type B	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD1833_0	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	IRQ11	—	—	IRQ5	IRQ6	IRQ7	IRQ8	IRQ9
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	A23	A22	A21	A20	A19	A18
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—

Table 17.27 Port 22 function assignment (2 of 2)

Function (PFC)	Port	P22_0	P22_1	P22_2	P22_3	P22_4	P22_5	P22_6	P22_7
	IO Type	Type B	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD1833_0	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
GPT	0x09	—	—	—	—	—	—	—	GTIOC06_0A
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	GTETRGA	GTETRGB	GTETRGC	GTETRGD	GTETRGSA	GTETRGSB	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	ETH0_TXER	ETH0_RXER	ETH0_CRS	ETH0_COL	—	—	ETH1_CRS
GMAC	0x10	—	—	—	—	—	GMAC0_PTPTRG0	GMAC0_PTPTRG1	—
ETHSW	0x11	—	—	—	—	—	—	—	ETHSW_TDMAOUT2
ESC	0x12	—	—	—	—	—	ESC_LATCH0	ESC_LATCH1	ESC_LINKACT0
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	TXD5/SDA5/MOSI5	RXD5/SCL5/MISO5	SCK5	SS5#/CTS5#/RTS5#	CTS5#	DE5	—
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	CANTX0	CANRX0	CANRXDP0	CANTXDP0	CANRX1	CANTX1	CANRXDP1
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	—	—	DUEI14	TST_OUT14	SI14#	DUEI15	TST_OUT15
HDSL	0x25	HDSL01_SMPL	HDSL01_CLK1	HDSL01_SEL1	HDSL01_MISO1	HDSL01_MOSI1	HDSL01_CLK2	HDSL01_SEL2	HDSL01_MISO2
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	SD0_CD	SD0_WP	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.24 Port 23 Function Assignment

Table 17.28 Port 23 function assignment (1 of 2)

Function (PFC)	Port	P23_0	P23_1	P23_2	P23_3	P23_4	P23_5	P23_6	P23_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	IRQ10	—	—	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	A17	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC06_0B	GTIOC06_1A	GTIOC06_1B	GTIOC06_2A	GTIOC06_2B	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	ETH1_COL	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	ETHSW_TDMAOUT3	—	—	—	—	—	ETHSW_LPI0	ETHSW_LPI1
ESC	0x12	ESC_LINKACT1	ESC_IRQ	ESC_RESETOUT#	ESC_I2CCLK	ESC_I2CDATA	ESC_LINKACT2	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	IIC_SCL0	IIC_SDA0	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	CANTXDP1	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	MCLK60	MDAT60	MCLK61
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	SI15#	DUEI00	TST_OUT00	SI00#	DUEI01	TST_OUT01	SI01#	DUEI02
HDSL	0x25	HDSL01_MOSI2	HDSL02_LINK	HDSL02_SMPL	HDSL02_CLK1	HDSL02_SEL1	HDSL02_MISO1	HDSL02_MOSI1	HDSL02_CLK2
ENCOUT	0x26	—	—	—	—	—	—	—	—

Table 17.28 Port 23 function assignment (2 of 2)

Function (PFC)	Port	P23_0	P23_1	P23_2	P23_3	P23_4	P23_5	P23_6	P23_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.25 Port 24 Function Assignment

Table 17.29 Port 24 function assignment (1 of 2)

Function (PFC)	Port	P24_0	P24_1	P24_2	P24_3	P24_4	P24_5	P24_6	P24_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type B	Type B	Type B
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD1833_1	VDD1833_1	VDD1833_1
Interrupt	0x00	IRQ11	IRQ12	IRQ13	IRQ14	IRQ15	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	—	—	—	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	ETH1_TXCLK	ETH1_TXD0	ETH1_TXD1
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	ETHSW_LPI2	—	—	—	—	—	—	—
ESC	0x12	—	—	—	ESC_I2CCLK	ESC_I2CDATA	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	IIC_SCL1	IIC_SDA1	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	CANRX0	CANTX0	—	—	—
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—

Table 17.29 Port 24 function assignment (2 of 2)

Function (PFC)	Port	P24_0	P24_1	P24_2	P24_3	P24_4	P24_5	P24_6	P24_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type B	Type B	Type B
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD1833_1	VDD1833_1	VDD1833_1
DSMIF	0x1D	MDAT61	MCLK62	MDAT62	MCLK70	MDAT70	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	TST_OUT02	SI02#	—	—	—	—	—	—
HDSL	0x25	HDSL02_SEL2	HDSL02_MISO2	HDSL02_MOSI2	HDSL03_LINK	HDSL03_SMPL	HDSL03_CLK1	—	—
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.26 Port 25 Function Assignment

Table 17.30 Port 25 function assignment (1 of 2)

Function (PFC)	Port	P25_0	P25_1	P25_2	P25_3	P25_4	P25_5	P25_6	P25_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1
Interrupt	0x00	—	—	—	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	—	—	—	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	ETH1_TXD2	ETH1_TXD3	ETH1_TXEN	ETH1_RXCLK	ETH1_RXD0	ETH1_RXD1	ETH1_RXD2	ETH1_RXD3
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—

Table 17.30 Port 25 function assignment (2 of 2)

Function (PFC)	Port	P25_0	P25_1	P25_2	P25_3	P25_4	P25_5	P25_6	P25_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	CANRXDP0	CANTXDP0	—	—	—	—	CANRX1	CANTX1
SPI	0x1A	—	—	—	—	—	—	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	—	—	DUEI03	TST_OUT03	SI03#	DUEI04	TST_OUT04
HDSL	0x25	—	—	—	HDSL03_SEL1	HDSL03_MISO1	HDSL03_MOSI1	HDSL03_CLK2	HDSL03_SEL2
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.27 Port 26 Function Assignment

Table 17.31 Port 26 function assignment (1 of 2)

Function (PFC)	Port	P26_0	P26_1	P26_2	P26_3	P26_4	P26_5	P26_6	P26_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type A	Type A
	Voltage domain	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD33	VDD33
Interrupt	0x00	—	—	—	—	—	IRQ12	SEI	IRQ0
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	ETH1_REFCLK	—	—	—
	0x03	—	—	—	—	RMII1_REFCLK	—	—	—
BSC	0x04	—	—	—	—	—	—	CS2#	CS3#
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—

Table 17.31 Port 26 function assignment (2 of 2)

Function (PFC)	Port	P26_0	P26_1	P26_2	P26_3	P26_4	P26_5	P26_6	P26_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type A	Type A
	Voltage domain	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD1833_1	VDD33	VDD33
GPT	0x09	—	—	—	—	—	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	ETH1_RXDV	—	—	—	—	—	ETH1_TXER	ETH1_RXER
GMAC	0x10	—	GMAC1_MDC	GMAC1_MDIO	—	—	—	—	—
ETHSW	0x11	—	ETHSW_MDC	ETHSW_MDIO	ETHSW_PHYLINK1	—	—	—	—
ESC	0x12	—	ESC_MDC	ESC_MDIO	ESC_PHYLINK1	—	—	ESC_RESETOUT#	ESC_LEDSTER
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	—	—	—	SCKE01	DEE01	TXDE01
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	CANRXDP1	CANTXDP1	—	—	CANTX0	CANRX0	CANRXDP0
SPI	0x1A	—	—	—	—	—	—	—	SPI_SSL01
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	ENCIFCK01	ENCIFOE01	ENCIFDO01
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	SI04#	—	—	—	—	—	—	—
HDSL	0x25	HDSL03_MISO2	HDSL03_MOSI2	HDSL04_LINK	HDSL04_SMPL	—	HDSL04_CLK1	HDSL04_SEL1	HDSL04_MISO1
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.28 Port 27 Function Assignment

Table 17.32 Port 27 function assignment (1 of 2)

Function (PFC)	Port	P27_0	P27_1	P27_2	P27_3	P27_4	P27_5	P27_6	P27_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	IRQ1	IRQ2	IRQ3	—	—	—	—	IRQ4
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	CS5#	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	MTIOC2A	MTIOC2B	MTIOC1A	MTIOC1B	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	—	GTIOC08_3A	GTIOC08_3B	GTIOC08_4A	GTIOC08_4B	GTIOC08_0A
	0x0A	—	GTIOC02_0A	GTIOC02_0B	GTIOC02_1A	GTIOC02_1B	GTIOC02_2A	GTIOC02_2B	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	ETH1_CRS	ETH1_COL	—	—	—	—	—	—
GMAC	0x10	—	—	GMAC1_PTPTRG0	GMAC1_PTPTRG1	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	ETHSW_TDMAOUT0
ESC	0x12	—	—	ESC_LEDERR	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	SCK0	RXD0/SCL0/MISO0	TXD0/SDA0/MOSI0	—	—
SCIE	0x15	RXDE01	—	—	SCKE10	DEE10	TXDE10	RXDE10	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	CANTXDP0	CANRX1	CANTX1	CANRXDP1	CANTXDP1	—	—	—
SPI	0x1A	SPI_SSL02	SPI_SSL03	SPI_RSPCK0	SPI_MOSI0	SPI_MISO0	SPI_SSL00	—	SPI_RSPCK1
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	HSPI_INT#	HSPI_CS#	HSPI_IO0	HSPI_IO1	HSPI_IO2	HSPI_IO3	HSPI_CK	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	ENCIFDI01	—	—	ENCIFCK14	ENCIFOE14	ENCIFDO14	ENCIFDI14	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	—	—	—	—	—	—	DUEI05
HDSL	0x25	HDSL04_MOSI1	HDSL04_CLK2	HDSL04_SEL2	HDSL04_MISO2	HDSL04_MOSI2	HDSL05_LINK	HDSL05_SMPL	HDSL05_CLK1

Table 17.32 Port 27 function assignment (2 of 2)

Function (PFC)	Port	P27_0	P27_1	P27_2	P27_3	P27_4	P27_5	P27_6	P27_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.29 Port 28 Function Assignment

Table 17.33 Port 28 function assignment (1 of 2)

Function (PFC)	Port	P28_0	P28_1	P28_2	P28_3	P28_4	P28_5	P28_6	P28_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	IRQ5	IRQ6	IRQ7	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC08_0B	GTIOC08_1A	GTIOC08_1B	GTIOC08_2A	GTIOC08_2B	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	ETHSW_TDMAOUT1	ETHSW_TDMAOUT2	ETHSW_TDMAOUT3	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	—	—	—	SCKE08	DEE08	TXDE08
	0x16	—	—	—	—	—	SCKE00	DEE00	TXDE00
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	CANRX0	CANTX0	CANRXDP0
SPI	0x1A	SPI_MOSI1	SPI_MISO1	SPI_SSL10	SPI_SSL11	SPI_SSL12	SPI_SSL13	—	—
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—

Table 17.33 Port 28 function assignment (2 of 2)

Function (PFC)	Port	P28_0	P28_1	P28_2	P28_3	P28_4	P28_5	P28_6	P28_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
DSMIF	0x1D	—	—	—	—	—	MCLK71	MDAT71	MCLK72
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	ENCIFCK08	ENCIFOE08	ENCIFDO08
	0x23	—	—	—	—	—	ENCIFCK00	ENCIFOE00	ENCIFDO00
ENDAT	0x24	TST_OUT05	SI05#	DUEI06	TST_OUT06	SI06#	—	—	—
HDSL	0x25	HDSL05_SEL1	HDSL05_MISO1	HDSL05_MOSI1	HDSL05_CLK2	HDSL05_SEL2	HDSL05_MISO2	HDSL05_MOSI2	HDSL06_LINK
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.30 Port 29 Function Assignment

Table 17.34 Port 29 function assignment (1 of 2)

Function (PFC)	Port	P29_0	P29_1	P29_2	P29_3	P29_4	P29_5	P29_6	P29_7
	IO Type	Type A	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD33	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2
Interrupt	0x00	—	—	—	—	IRQ8	IRQ9	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	GTIOC09_0A	GTIOC09_0B	GTIOC09_1A	GTIOC09_1B	GTIOC09_2A	GTIOC09_2B	GTIOC09_3A
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	ETH2_TXCLK	ETH2_TXD0	ETH2_TXD1	ETH2_TXD2	ETH2_TXD3	ETH2_TXEN	ETH2_RXCLK
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—

Table 17.34 Port 29 function assignment (2 of 2)

Function (PFC)	Port	P29_0	P29_1	P29_2	P29_3	P29_4	P29_5	P29_6	P29_7
	IO Type	Type A	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD33	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	RXDE08	SCKE09	DEE09	TXDE09	RXDE09	SCKE10	DEE10	TXDE10
	0x16	RXDE00	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	CANTXDP0	—	—	—	—	—	—	—
SPI	0x1A	—	—	—	—	SPI_SSL20	SPI_SSL21	SPI_SSL22	SPI_SSL23
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	MDAT72	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	ENCIFDI08	ENCIFCK09	ENCIFOE09	ENCIFDO09	ENCIFDI09	ENCIFCK10	ENCIFOE10	ENCIFDO10
	0x23	ENCIFDI00	—	—	—	—	—	—	—
ENDAT	0x24	—	—	—	—	—	—	—	—
HDSL	0x25	HDSL06_SMPL	HDSL06_CLK1	HDSL06_SEL1	HDSL06_MISO1	HDSL06_MOSI1	HDSL06_CLK2	HDSL06_SEL2	HDSL06_MISO2
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.31 Port 30 Function Assignment

Table 17.35 Port 30 function assignment (1 of 2)

Function (PFC)	Port	P30_0	P30_1	P30_2	P30_3	P30_4	P30_5	P30_6	P30_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2
Interrupt	0x00	—	—	IRQ10	IRQ11	—	—	—	IRQ14
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—

Table 17.35 Port 30 function assignment (2 of 2)

Function (PFC)	Port	P30_0	P30_1	P30_2	P30_3	P30_4	P30_5	P30_6	P30_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2	VDD1833_2
GPT	0x09	GTIOC09_3B	GTIOC09_4A	GTIOC09_4B	GTIOC09_5A	GTIOC09_5B	GTIOC09_6A	GTIOC09_6B	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	ETH2_RXD0	ETH2_RXD1	ETH2_RXD2	ETH2_RXD3	ETH2_RXDV	—	—	—
GMAC	0x10	—	—	—	—	—	GMAC2_MDC	GMAC2_MDIO	—
ETHSW	0x11	—	—	—	—	—	ETHSW_MDC	ETHSW_MDIO	ETHSW_PHYLINK2
ESC	0x12	—	—	—	—	—	ESC_MDC	ESC_MDIO	ESC_PHYLINK2
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	RXDE10	SCKE11	DEE11	TXDE11	RXDE11	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	—	—	SPI_MOSI2	SPI_MISO2	—	SPI_RSPCK3	SPI_MOSI3	SPI_MISO3
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	MCLK30
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	ENCIFDI10	ENCIFCK11	ENCIFOE11	ENCIFDO11	ENCIFDI11	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	—	—	—	—	DUEI07	TST_OUT07	SI07#
HDSL	0x25	HDSL06_MOSI2	HDSL07_LINK	HDSL07_SMPL	HDSL07_CLK1	HDSL07_SEL1	HDSL07_MISO1	HDSL07_MOSI1	HDSL07_CLK2
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	SD1_IOVS
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.32 Port 31 Function Assignment

Table 17.36 Port 31 function assignment (1 of 2)

Function (PFC)	Port	P31_0	P31_1	P31_2	P31_3	P31_4	P31_5	P31_6	P31_7
	IO Type	Type B	Type B	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD1833_2	VDD1833_2	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	—	IRQ13	—	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	ETH2_REFCLK	—	—	—	—	—	—	—
	0x03	RMII2_REFCLK	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	A16	—
DMAC	0x05	—	—	—	—	DREQ	DACK	TEND	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	POE0#	POE4#	POE8#	POE10#	POE11#	—
GPT	0x09	—	—	—	—	—	—	—	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	GTETRGSA	GTETRGSB	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	ETH2_RXER	ETH2_TXER	ETH2_RXER	ETH2_CRS	ETH2_COL	—	—
GMAC	0x10	—	—	—	—	—	—	GMAC2_PTPTRG0	GMAC2_PTPTRG1
ETHSW	0x11	—	—	—	ETHSW_TDMAOUT1	ETHSW_PTPOUT2	ETHSW_PTPOUT3	ETHSW_TDMAOUT0	—
ESC	0x12	—	—	—	ESC_LEDERR	ESC_SYNC0	ESC_SYNC1	ESC_LEDRUN	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	—	—	SCKE09	DEE09	TXDE09	RXDE09	SCKE11	DEE11
	0x16	—	—	—	—	—	—	SCKE01	DEE01
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	SPI_SSL30	SPI_SSL31	SPI_SSL32	SPI_SSL33	SPI_RSPCK0	SPI_MOSI0	SPI_MISO0	SPI_SSL00
	0x1B	—	—	—	—	SPI_SSL30	SPI_SSL31	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	MCLK80	MDAT80	MCLK81	MDAT81	—	—
	0x1E	—	—	MDAT30	MCLK31	MDAT31	MCLK32	MDAT32	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	HSPI_IO4	HSPI_IO5	HSPI_IO6	HSPI_IO7	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	ENCIFCK09	ENCIFOE09	ENCIFDO09	ENCIFDI09	ENCIFCK15	ENCIFOE15
	0x23	—	—	—	—	—	—	ENCIFCK01	ENCIFOE01
ENDAT	0x24	—	—	—	—	—	—	—	—

Table 17.36 Port 31 function assignment (2 of 2)

Function (PFC)	Port	P31_0	P31_1	P31_2	P31_3	P31_4	P31_5	P31_6	P31_7
	IO Type	Type B	Type B	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD1833_2	VDD1833_2	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
HDSL	0x25	HDSL07_SEL2	HDSL07_MISO2	HDSL07_MOSI2	HDSL08_LINK	HDSL08_SMPL	HDSL08_CLK1	HDSL08_SEL1	HDSL08_MISO1
ENCOUT	0x26	—	—	POUTA	—	POUTB	POUTZ	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.33 Port 32 Function Assignment

Table 17.37 Port 32 function assignment (1 of 2)

Function (PFC)	Port	P32_0	P32_1	P32_2	P32_3	P32_4	P32_5	P32_6	P32_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
Interrupt	0x00	—	—	—	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	—	—	GTIOC10_0A	GTIOC10_0B	GTIOC10_1A	GTIOC10_1B	GTIOC10_2A	GTIOC10_2B
	0x0A	—	—	GTIOC01_0A	GTIOC01_0B	GTIOC01_1A	GTIOC01_1B	GTIOC01_2A	GTIOC01_2B
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	—	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	—	—	—
SCIE	0x15	TXDE11	RXDE11	SCKE10	DEE10	TXDE10	RXDE10	SCKE11	DEE11
	0x16	TXDE01	RXDE01	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	SPI_SSL01	SPI_SSL02	SPI_SSL03	SPI_RSPCK1	SPI_MOSI1	SPI_MISO1	SPI_SSL10	SPI_SSL11
	0x1B	—	—	—	—	—	—	—	—

Table 17.37 Port 32 function assignment (2 of 2)

Function (PFC)	Port	P32_0	P32_1	P32_2	P32_3	P32_4	P32_5	P32_6	P32_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	ENCIFDO15	ENCIFDI15	ENCIFCK10	ENCIFOE10	ENCIFDO10	ENCIFDI10	ENCIFCK11	ENCIFOE11
	0x23	ENCIFDO01	ENCIFDI01	—	—	—	—	—	—
ENDAT	0x24	—	—	—	—	—	—	—	—
HDSL	0x25	HDSL08_MOSI1	HDSL08_CLK2	HDSL08_SEL2	HDSL08_MISO2	HDSL08_MOSI2	HDSL09_LINK	HDSL09_SMPL	HDSL09_CLK1
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.34 Port 33 Function Assignment

Table 17.38 Port 33 function assignment (1 of 2)

Function (PFC)	Port	P33_0	P33_1	P33_2	P33_3	P33_4	P33_5	P33_6	P33_7
	IO Type	Type A	Type A	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD33	VDD33	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3
Interrupt	0x00	—	—	—	IRQ12	IRQ13	IRQ14	IRQ15	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	A16	A17	A18	A19	A20	A21
DMAC	0x05	—	—	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTIOC10_3A	GTIOC10_3B	GTADSM00_0	GTADSM00_1	GTADSM01_0	GTADSM01_1	GTADSM02_0	GTADSM02_1
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	—	—	ETH3_TXCLK	ETH3_TXD0	ETH3_TXD1	ETH3_TXD2	ETH3_TXD3	ETH3_TXEN
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—

Table 17.38 Port 33 function assignment (2 of 2)

Function (PFC)	Port	P33_0	P33_1	P33_2	P33_3	P33_4	P33_5	P33_6	P33_7
	IO Type	Type A	Type A	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VDD33	VDD33	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	SCK1	RXD1/SCL1/ MISO1	TXD1/SDA1/ MOSI1	RXD2/SCL2/ MISO2	TXD2/SDA2/ MOSI2	—
SCIE	0x15	TXDE11	RXDE11	SCKE01	DEE01	TXDE01	RXDE01	SCKE06	DEE06
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	SPI_SSL12	SPI_SSL13	SPI_RSPCK1	SPI_MOSI1	SPI_MISO1	SPI_SSL10	SPI_SSL11	SPI_RSPCK2
	0x1B	—	—	SPI_SSL30	SPI_RSPCK0	SPI_MOSI0	SPI_MISO0	SPI_SSL00	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	MCLK82	MDAT82	MCLK50	MDAT50	MCLK51	MDAT51	MCLK52	MDAT52
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	—	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	ENCIFDO11	ENCIFDI11	ENCIFCK01	ENCIFOE01	ENCIFDO01	ENCIFDI01	ENCIFCK06	ENCIFOE06
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	—	—	—	—	—	—	—
HDSL	0x25	HDSL09_SEL 1	HDSL09_MIS O1	HDSL09_MOS I1	HDSL09_CLK 2	HDSL09_SEL 2	HDSL09_MIS O2	HDSL09_MOS I2	HDSL10_LINK
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	PCIE_RSTOU T0B	PCIE_RSTOU T1B	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.35 Port 34 Function Assignment

Table 17.39 Port 34 function assignment (1 of 2)

Function (PFC)	Port	P34_0	P34_1	P34_2	P34_3	P34_4	P34_5	P34_6	P34_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type A
	Voltage domain	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD33
Interrupt	0x00	—	—	—	—	—	—	—	IRQ14
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	ETH3_REFCL K	—
	0x03	—	—	—	—	—	—	RMII3_REFCL K	—
BSC	0x04	A22	A23	A24	A25	CS2#	CS3#	CS5#	—
DMAC	0x05	—	—	—	—	—	—	—	DREQ
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—

Table 17.39 Port 34 function assignment (2 of 2)

Function (PFC)	Port	P34_0	P34_1	P34_2	P34_3	P34_4	P34_5	P34_6	P34_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type A
	Voltage domain	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD1833_3	VDD33
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTADSM03_0	GTADSM03_1	GTADSM04_0	GTADSM04_1	GTADSM05_0	GTADSM05_1	—	GTADSM06_0
	0x0A	GTIOC03_0A	GTIOC03_0B	GTIOC03_1A	GTIOC03_1B	GTIOC03_2A	GTIOC03_2B	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	ETH3_RXCLK	ETH3_RXD0	ETH3_RXD1	ETH3_RXD2	ETH3_RXD3	ETH3_RXDV	ETH1_RXER	ETH3_TXER
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	ESC_I2CCLK	ESC_I2CDATA	ESC_RESETOUT#
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	RXD3/SCL3/MISO3	TXD3/SDA3/MOSI3	—	—
SCIE	0x15	TXDE06	RXDE06	SCKE07	DEE07	TXDE07	RXDE07	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	IIC_SCL1	IIC_SDA1	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	SPI_MOSI2	SPI_MISO2	SPI_SSL20	SPI_SSL21	SPI_SSL22	SPI_SSL23	SPI_RSPCK3	SPI_MOSI3
	0x1B	—	—	—	—	—	—	—	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	—	—	—	—	—	—	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	—	—	ADTRG0#	ADTRG1#	ADTRG2#	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	ENCIFDO06	ENCIFDI06	ENCIFCK07	ENCIFOE07	ENCIFDO07	ENCIFDI07	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	—	—	—	—	—	—	DUEI08	TST_OUT08
HDSL	0x25	HDSL10_SMPL	HDSL10_CLK1	HDSL10_SEL1	HDSL10_MISO1	HDSL10_MOSI1	HDSL10_CLK2	HDSL10_SEL2	HDSL10_MISO2
ENCOUT	0x26	—	—	—	—	—	—	—	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	SD1_PWEN	SD1_IOVS	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.5.36 Port 35 Function Assignment

Table 17.40 Port 35 function assignment (1 of 2)

Function (PFC)	Port	P35_0	P35_1	P35_2	P35_3	P35_4	P35_5	P35_6	—
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	—
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	—
Interrupt	0x00	—	—	—	—	—	—	—	—
SYS/DBG	0x01	—	—	—	—	—	—	—	—
Clock	0x02	—	—	—	—	—	—	—	—
	0x03	—	—	—	—	—	—	—	—
BSC	0x04	—	—	—	—	—	—	—	—
DMAC	0x05	DACK	TEND	—	—	—	—	—	—
MTU3	0x06	—	—	—	—	—	—	—	—
	0x07	—	—	—	—	—	—	—	—
POE3	0x08	—	—	—	—	—	—	—	—
GPT	0x09	GTADSM06_1	GTADSM07_0	GTADSM07_1	GTADSM08_0	GTADSM08_1	GTADSM09_0	GTADSM09_1	—
	0x0A	—	—	—	—	—	—	—	—
	0x0B	—	—	—	—	—	—	—	—
POEG	0x0C	—	—	—	—	—	—	—	—
CMTW	0x0D	—	—	—	—	—	—	—	—
RTC	0x0E	—	—	—	—	—	—	—	—
ETHSS	0x0F	ETH3_RXER	ETH3_CRS	ETH3_COL	—	—	—	—	—
GMAC	0x10	—	—	—	—	—	—	—	—
ETHSW	0x11	—	—	—	—	—	—	—	—
ESC	0x12	—	—	—	—	—	—	—	—
USB	0x13	—	—	—	—	—	—	—	—
SCI	0x14	—	—	—	—	—	RXD4/SCL4/ MISO4	TXD4/SDA4/ MOSI4	—
SCIE	0x15	—	—	—	—	—	—	—	—
	0x16	—	—	—	—	—	—	—	—
IIC	0x17	—	—	—	—	—	—	—	—
	0x18	—	—	—	—	—	—	—	—
CANFD	0x19	—	—	—	—	—	—	—	—
SPI	0x1A	SPI_MISO3	SPI_SSL30	SPI_SSL31	SPI_SSL32	SPI_SSL33	—	—	—
	0x1B	—	SPI_MISO1	SPI_SSL10	SPI_MOSI1	SPI_SSL11	SPI_RSPCK1	SPI_SSL12	—
xSPI	0x1C	—	—	—	—	—	—	—	—
DSMIF	0x1D	—	MCLK90	MDAT90	MCLK91	MDAT91	MCLK92	MDAT92	—
	0x1E	—	—	—	—	—	—	—	—
ADC	0x1F	—	—	ADTRG2#	ADTRG0#	ADTRG1#	—	—	—
SHOST	0x20	—	—	—	—	—	—	—	—
Mailbox	0x21	—	—	—	—	—	—	—	—
ENCIF	0x22	—	—	—	—	—	—	—	—
	0x23	—	—	—	—	—	—	—	—
ENDAT	0x24	SI08#	DUEI09	TST_OUT09	SI09#	DUEI10	TST_OUT10	SI10#	—
HDSL	0x25	HDSL10_MOSI2	HDSL11_LINK	HDSL11_SMPL	HDSL11_CLK1	HDSL11_SEL1	HDSL11_MISO1	HDSL11_MOSI1	—
ENCOUT	0x26	—	—	—	—	—	—	—	—

Table 17.40 Port 35 function assignment (2 of 2)

Function (PFC)	Port	P35_0	P35_1	P35_2	P35_3	P35_4	P35_5	P35_6	—
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	—
	Voltage domain	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	VDD33	—
PCIE	0x27	—	—	—	—	—	—	—	—
LCDC	0x28	—	—	—	—	—	—	—	—
SDHI	0x29	—	—	—	—	—	—	—	—
Prohibited	0x2A to 0x3F	—	—	—	—	—	—	—	—

17.6 Handling of Unused Pins

Details on the handling of unused pins are given in Table 17.41.

Table 17.41 Handling of unused pins (1 of 2)

Item	Pin name	Handling
Clock	XTAL	Keep this pin open when external clock signal is used.
	EXTAL	Connect this pin to VSS via a resistor (pulling down) when external clock signal is used.
	EXTCLKIN	Connect this pin to VSS via a resistor (pulling down) when a crystal resonator is connected.
Debug	TRST#	Connect these pins to VSS via a resistor (pulling down), or input the same signal as that on the RES# pin.
	TCK (P08_3)	Connect this pin to VSS via a resistor (pulling down).
	TMS (P08_1)	Connect this pin to VDD33 via a resistor (pulling up).
	TDI (P08_2)	Connect this pin to VDD33 via a resistor (pulling up).
	TDO (P08_4)	Keep this pin open.
System	RSTOUT# (P08_5)	Keep this pin open.
	MDX	— (Always connect this pin to VSS via a resister (pull down).)
	BSCANP	— (Use this as a boundary scan enable pin.)
ADC12*1	AN000 to AN003, AN100 to AN103, AN200 to AN214	Keep these pins open.
	AVDDREF_ADC0	Connect this pin to AVDDIO_ADC0.
	AVDDREF_ADC1	Connect this pin to AVDDIO_ADC1.
	AVDDREF_ADC2	Connect this pin to AVDDIO_ADC2.
USB*1	USB_QDP, USB_QDM, USB_OTG_ID	Keep these pins open.
	USB_VUBUSIN, USB_TXRTUNE	Connect these pins to VSS via a resistor (pulling down) or keep these pins open.
PCIE*1	PCIE_REFCLK_P0, PCIE_REFCLK_N0, PCIE_REFCLK_P1, PCIE_REFCLK_N1, PCIE_RXDP_L0, PCIE_RXDN_L0, PCIE_RXDP_L1, PCIE_RXDN_L1, PCIE_TXDP_L0, PCIE_TXDN_L0, PCIE_TXDP_L1, PCIE_TXDN_L1	Keep these pins open.
DDRSS*1	DDR_ZN	Connect this pin to DDR_VDDQ.
	DDR_DTEST, DDR_ATEST, DDR_RESET_N, DDR_CKA_T, DDR_CKA_C, DDR_CKB_T, DDR_CKB_C, DDR_CKEA[1:0], DDR_CKEB[1:0], DDR_CSA[1:0], DDR_CSB[1:0], DDR_CAA[5:0], DDR_CAB[5:0], DDR_DQA[15:0], DDR_DQB[15:0], DDR_DMIA[1:0], DDR_DMIB[1:0], DDR_DQSA_T[1:0], DDR_DQSB_T[1:0], DDR_DQSA_C[1:0], DDR_DQSB_C[1:0]	Keep these pins open.

Table 17.41 Handling of unused pins (2 of 2)

Item	Pin name	Handling
Other	Other Pins*2	Keep these pins open, connect them to VDD33 via a resistor (pulling up), or connect them to VSS via a resistor (pulling down).

Note 1. Set the module to the standby or low power mode. They are in such state after reset released. Do not change the state of the module.
Note 2. When handling them as unused pins, set the corresponding bits of Port m Mode Register (PMm: m = 00 to 35) to "Non-use (Hi-Z input protection)" which is the value after reset release.

17.7 Usage Notes

17.7.1 Release of Register Write Protection

PMCm, PFCm, DRCTLm, RSELPm, SLPORTm, SLRSELP, SLPSR, SLPSRNS, and RSELPSR registers are write protected. To release write protection, follow the steps below.

In case of access to 0x812C_xxxx and RSELPm = 0 for PMCm, PFCm, DRCTLm, and SLPORTm (m = 00 to 12),

In case of access to 0x802B_xxxx and RSELPm = 1 for PMCm, PFCm, DRCTLm, and SLPORTm (m = 00 to 12),

In case of access to 0x812C_xxxx or 0x802B_xxxx for RSELPm, SLRSELP, SLPSR, and RSELPSR:

1. Write 0x0000A504 to PRCRS register (Release write protection by PRC2 bit)
2. Write value to target registers
3. Write 0x0000A500 to PRCRS register (Set write protection by PRC2 bit)

In case of access to 0x802C_xxxx for PMCm, PFCm, DRCTLm, SLPORTm (m = 13 to 35), and SLPSRNS:

1. Write 0x0000A504 to PRCRN register (Release write protection by PRC2 bit)
2. Write value to target registers
3. Write 0x0000A500 to PRCRN register (Set write protection by PRC2 bit)

17.7.2 Safety and Non-safety I/O Port, and Region Select

I/O port consists of two types, Safety I/O and Non-safety I/O. Port 00 to 12 are Safety I/O port and Port 13 to 35 are Non-safety I/O port. Table 17.42 shows the features each I/O port type. Safety I/O port supports access from two separate addresses. Access attribute is controlled by region select register, RSELPm. Access attribute to TrustZone access control registers is controlled by RSELPSR register. Table 17.43 shows the access attribute by region select register.

Table 17.42 I/O Port type

Port	Type	Register address	Features
Port 00 to 12	Safety I/O Port	0x812C_xxxx or 0x802B_xxxx	• Access is possible from two separate addresses. • Selectable access attribute each address and each port. One is R/W and the other is Read only. • DMAC access is not available.
Port 13 to 35	Non-safety I/O Port	0x802C_0000	• DMAC access is available.

Table 17.43 Access attribute by Region Select register

Registers		Region Select	Access attribute each address	
			0x812C_xxxx (Safety region)	0x802B_xxxx (Non-safety region)
Port registers	Pm, PMm, PMCm, PFCm, DRCTLm	RSELPm = 0	R/W	R
		RSELPm = 1	R	R/W
	PINm	—	R	R
TrustZone Access Control registers	SLPORTm, SLRSELP, SLPSR	RSELPSR = 0	R/W	R
		RSELPSR = 1	R	R/W
Region Select registers	RSELPm, RSELPSR	—	R/W	R

Note: m = 00 to 12

17.7.3 Port Register TrustZone Access Control

Access to the Port registers is protected by TrustZone access control. Figure 17.2 shows the overview of the access control. Security level of access to the SLPSR and SLPSRNS register is fixed to 3. It cannot be changed. SLPSR controls the security level of access to the SLPORTm (m = 00 to 12), SLRSELP, and RSELPSR registers. SLPSRNS controls the security level of access to the SLPORTm (m = 13 to 35) registers. SLPORTm registers control the security level of access to the Pm, PMm, PMCm, PFCm, PINm, and DRCTLm registers (m = 00 to 35). SLRSELP register controls the security level of access to the RSELPm register.

SLPSRNS also controls the security level of access to the ELC registers. See section 16, Event Link Controller (ELC) for ELC registers.

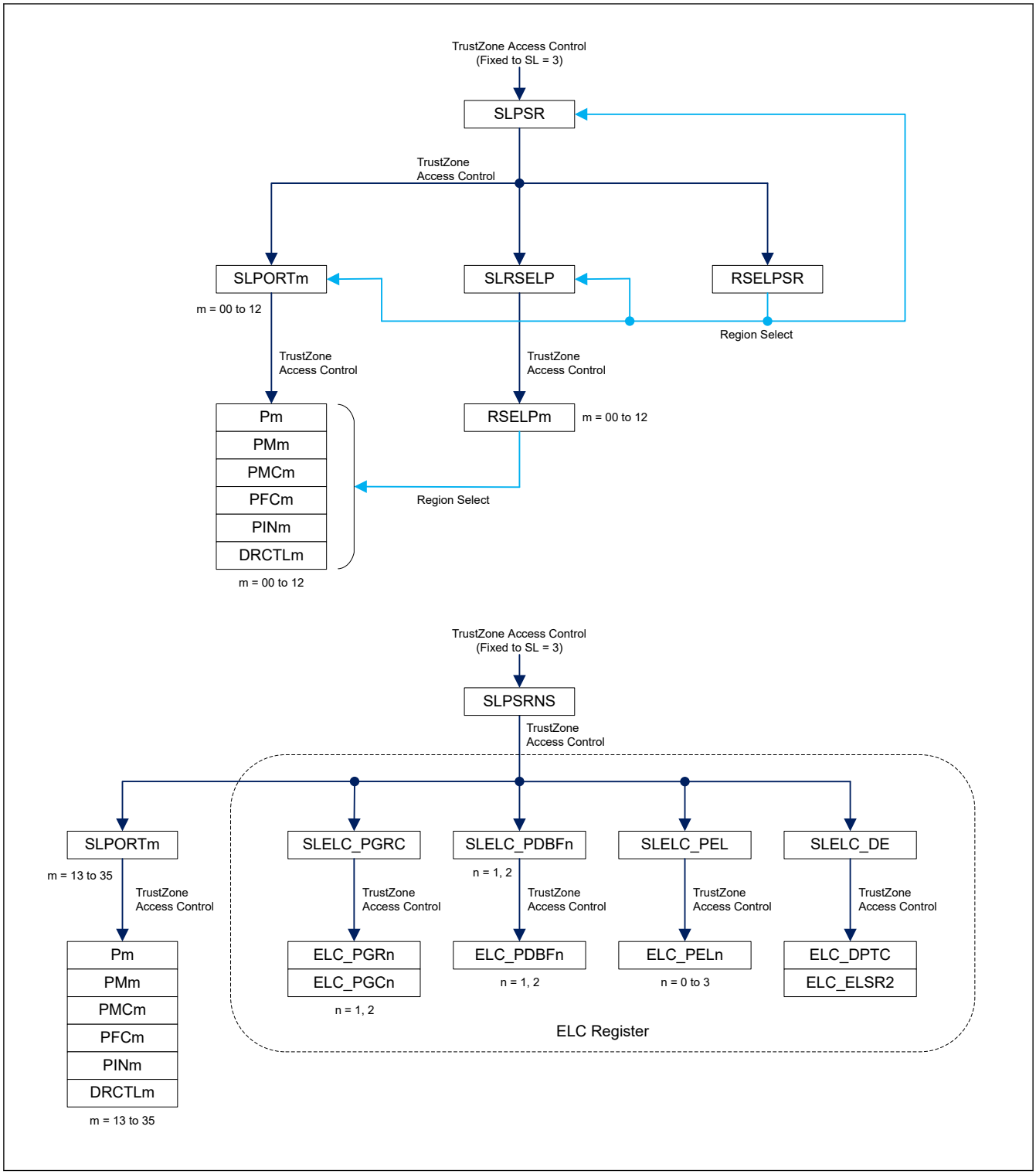


Figure 17.2 Overview of the port register TrustZone access control and Region Select

17.7.4 Examples of Port Setting

The registers to be controlled are shown in Table 17.44.

Table 17.44 Port register configuration

Register name		Bit name	Description
GPIO control register	Pm	POUT_n	Port register
	PMm	PMn	Port mode register
	PINm	PINn	Port input register
Multiplexed function select register	PMCm	PMCn	Port mode control register
	PFCm	PFCn	Port function control register
I/O Buffer function switching register	DRCTLm	DRVn	Port driving ability control
		PUDn	Port Pull-Up/Down control
		SMTn	Port Schmitt trigger control
		SRn	Port slew rate control

Note: m: 00 to 35, n: 0 to 7

Table 17.45 is overview of the mode register settings for the pin function.

Table 17.45 Pin function configuration overview

Mode	PMCm.PMCn	PMm.PMn	Pm.POUT_n	PINm.PINn	I/O
General I/O input	0	01	x	Input value	I
General I/O output	0	1x	Output value	(Input value)	O
I/O port for peripheral functions*1	1	x	—	—	I/O

Note: m: 00 to 35, n: 0 to 7
Note 1. When PMCm.PMCn = 1, select pin function by PFCm.PFCn.

Figure 17.3 is an example flow of setting the Port Pm_n.

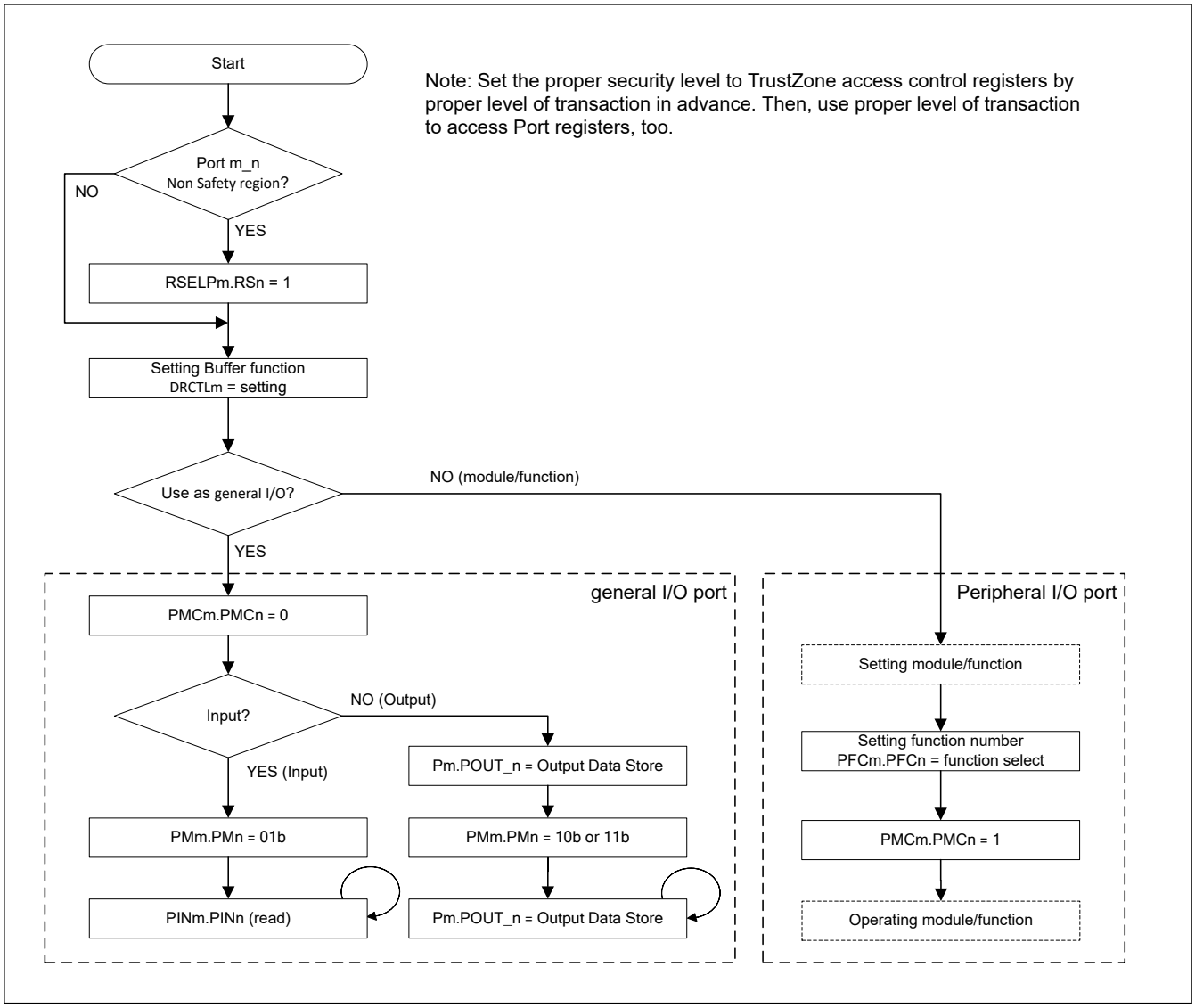


Figure 17.3 Example flow of Safety I/O port setting

17.7.5 Notes on Port Register Setting

1. If the multiplexed function is switched, a spike pulse may be generated in the signal of the on-chip peripheral function multiplexed with port due to the pin state immediately before the switching or other reasons. Take general measures shown below against spike pulses by software.
   - Switch the multiplexed function while the on-chip function is not working.
   - Clear the interrupt request flags, and then release the mask for pins multiplexed with interrupt signals.
   - Determine the output value, and then change the mode.
2. Only the allowed functions should be specified for the PFCm register. If a value that is not allowed for the register is specified, correct operation is not guaranteed.
3. Do not assign a single function to multiple pins through the PFCm registers.

17.7.6 Pin Assignment of each Boot Mode

Functional pin is shared with multiple ports for flexibility of pin assignment. However, pins to be used at booting are uniquely determined each boot mode. These pin assignments are automatically set, and they can't be changed. PMCm and PFCm registers related to function pins at selected boot mode are changed from the initial values and stay after boot processing. If RESTORE_FLG is set to 0x2236_0679 in the boot parameter, these registers are changed back to the initial value after boot processing.

17.7.6.1 Pin Assignment for xSPI Boot Mode

Table 17.46 Pins to be used for xSPI boot mode

	xSPI0			xSPI1
Function pin	x1 boot	x8 boot	Function pin	x1 boot
XSPI0_CLKP	P05_1	P05_1	xSPI1_CLKP	P01_0
XSPI0_CLKN	—	P05_2	xSPI1_CS0#	P01_1
XSPI0_CS0#	P05_3	P05_3	xSPI1_IO0	P01_4
XSPI0_IO0	P05_6	P05_6	xSPI1_IO1	P01_5
XSPI0_IO1	P05_7	P05_7	—	
XSPI0_IO2	—	P06_0		
XSPI0_IO3	—	P06_1		
XSPI0_IO4	—	P06_2		
XSPI0_IO5	—	P06_3		
XSPI0_IO6	—	P06_4		
XSPI0_IO7	—	P06_5		
XSPI0_DS	—	P05_5		
XSPI0_RESET0#	—	P06_6		

17.7.6.2 Pin Assignment for eSD/eMMC Boot Mode

Table 17.47 Pins to be used for eSD/eMMC boot mode

Function pin	eSD boot	Function pin	eMMC boot
SD1_CLK	P16_5	SD0_CLK	P12_0
SD1_CMD	P16_6	SD0_CMD	P12_1
SD1_DATA0	P16_7	SD0_DATA0	P12_2
SD1_DATA1	P17_0	SD0_DATA1	P12_3
SD1_DATA2	P17_1	SD0_DATA2	P12_4
SD1_DATA3	P17_2	SD0_DATA3	P12_5
—		SD0_DATA4	P12_6
		SD0_DATA5	P12_7
		SD0_DATA6	P13_0
		SD0_DATA7	P13_1
		SD0_RST#	P13_2

17.7.6.3 Pin Assignment for SCI Boot Mode

Table 17.48 Pins to be used for SCI boot mode

Function pin	SCI boot
RXD0	P27_4
TXD0	P27_5

18. Multi-Function Timer Pulse Unit 3 (MTU3)

18.1 Overview

This LSI has an on-chip multi-function timer pulse unit 3 (MTU3), consisting of eight 16-bit timer channels and one 32-bit timer channel.

Table 18.1 shows the specifications of the MTU and Table 18.2 lists the functions of the MTU. Figure 18.1 and Figure 18.2 show block diagrams of the MTU.

Table 18.1 MTU specifications

Item	Description
Pulse input/output	28 lines maximum
Pulse input	3 lines
Count clock	11 clocks for each channel (14 clocks for MTU0, 12 clocks for MTU1 and MTU2, 10 clocks for MTU5)
Operating frequency	PCLKH (250 MHz)
Available operations	[MTU0 to MTU4, MTU6, MTU7, MTU8] ● Waveform output on compare match ● Input capture function (noise filter setting available) ● Counter-clearing operation ● Simultaneous writing to multiple timer counters (TCNT) (excluding MTU8) ● Simultaneous clearing on compare match or input capture (excluding MTU8) ● Simultaneous input and output to registers in synchronization with counter operations (excluding MTU8) ● Up to 12-phase PWM output in combination with synchronous operation (excluding MTU8)
	[MTU0, MTU3, MTU4, MTU6, MTU7, MTU8] ● Buffer operation specifiable
	[MTU1, MTU2] ● Phase counting mode can be specified independently ● 32-bit phase counting mode can be specified for interlocked operation of MTU1 and MTU2 (when TMDR3.LWA = 1) ● Cascade connection operation available
	[MTU3, MTU4, MTU6, MTU7] ● Through interlocked operation of MTU3/4 and MTU6/7, the positive and negative signals in six phases (12 phases in total) can be output in complementary PWM and reset-synchronized PWM operation. ● In complementary PWM mode, transfer of values from buffer registers to temporary registers on crests and troughs of the timer-counter values or writing to the buffer registers (MTU4.TGRD and MTU7.TGRD) ● Double-buffering selectable in complementary PWM mode
	[MTU3, MTU4] ● Through interlocking with MTU0, a mode for driving AC synchronous motors (brushless DC motors) by using complementary PWM output and reset PWM output is settable and allows the selection of two types of waveform output (chopping or level)
	[MTU0/MTU5, MTU1, MTU2, MTU8] ● 32-bit phase counting mode specifiable by combining MTU1 and MTU2 and through interlocked operation with MTU0/MTU5 and MTU8
Interrupt skipping function	● In complementary PWM mode, interrupts on crests and troughs of counter values and triggers to start conversion by the A/D converter can be skipped
Interrupt sources	43 sources
Buffer operation	Automatic transfer of register data (transfer from the buffer register to the timer register)
Trigger generation	A/D converter start triggers can be generated
	A/D converter start request delaying function enables A/D converter to be started with any desired timing and to be synchronized with PWM output
Low power consumption function	Module stop mode can be set

Table 18.2 MTU functions (1 of 3)

✓: Possible —: Not possible

Parameter		MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU8
Count clock*1		PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/ 1024 MTCLKA MTCLKB MTCLKC MTCLKD MTIOC1A	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/ 1024 MTCLKA MTCLKB	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/ 1024 MTCLKA MTCLKB MTCLKC	MTCLKA MTCLKB MTCLKC MTCLKD	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/ 1024 MTCLKA MTCLKB	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/ 1024 MTCLKA MTCLKB	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/ 1024 MTIOC1A	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/ 1024 MTCLKA MTCLKB	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/ 1024 MTCLKA MTCLKB	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/ 1024 MTCLKA MTCLKB
General registers (TGR)		TGRA TGRB TGRE	TGRA TGRB TGRALW TGRBLW	TGRA TGRB	TGRALW TGRBLW	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW	TGRA TGRB	TGRA TGRB	TGRA TGRB
General registers/ buffer registers		TGRC TGRD TGRF	—	—	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	TGRC TGRD
I/O pins		MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC1A MTIOC1B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	MTIC5U MTIC5V MTIC5W	MTIOC6A MTIOC6B MTIOC6C MTIOC6D	MTIOC7A MTIOC7B MTIOC7C MTIOC7D	MTIOC8A MTIOC8B MTIOC8C MTIOC8D
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/ TGRBLW compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
	1 output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
	Toggle output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
Input capture function		✓	✓	✓	✓*1	✓	✓	✓	✓	✓	✓*2
Synchronous operation		✓	✓	✓	—	✓	✓	—	✓	✓	—
PWM mode 1		✓	✓	✓	—	✓	✓	—	✓	✓	—
PWM mode 2		✓	✓	✓	—	—	—	—	—	—	—
Complementary PWM mode		—	—	—	—	✓	✓	—	✓	✓	—
Reset-synchronized PWM mode		—	—	—	—	✓	✓	—	✓	✓	—
AC synchronous motor drive mode		✓	—	—	—	✓	✓	—	—	—	—
Phase counting mode		—	✓	✓	✓	—	—	—	—	—	—
Buffer operation		✓	—	—	—	✓	✓	—	✓	✓	✓
DMAC activation		TGRm compare match or input capture (m = A to D)	TGRm compare match or input capture (m = A, B)	TGRm compare match or input capture (m = A, B)	TGRALW/ TGRBLW input capture	TGRm compare match or input capture (m = A to D)	TGRm compare match or input capture (m = A to D) and TCNT overflow/ underflow (only in complemen-tary PWM mode)	TGRm compare match or input capture (m = U, V, W)	TGRm compare match or input capture (m = A to D)	TGRm compare match or input capture (m = A to D) or TCNT overflow/ underflow (only in complementary PWM mode)	TGRm compare match or input capture (m = A to D)

Table 18.2 MTU functions (2 of 3)

✓: Possible —: Not possible

Parameter	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU8
A/D converter start trigger	TGRA compare match or input capture TGRE compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRALW input capture	TGRA compare match or input capture	TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode	—	TGRA compare match or input capture	TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode	—
Interrupt sources	Seven sources	Four sources	Four sources	Four sources	Five sources	Five sources	Three sources	Five sources	Five sources	Five sources
	Overflow	Overflow	Overflow	Overflow	Overflow	Overflow or underflow (only in complementary PWM mode)	—	Overflow	Overflow or underflow (only in complementary PWM mode)	Overflow
	—	Underflow	Underflow	Underflow	—		—	—	—	—
	Compare match or input capture 0A	Compare match or input capture 1A	Compare match or input capture 2A	Input capture 1A	Compare match or input capture 3A	Compare match or input capture 4A	Compare match or input capture 5U	Compare match or input capture 6A	Compare match or input capture 7A	Compare match or input capture 8A
	Compare match or input capture 0B	Compare match or input capture 1B	Compare match or input capture 2B	Input capture 1B	Compare match or input capture 3B	Compare match or input capture 4B	Compare match or input capture 5V	Compare match or input capture 6B	Compare match or input capture 7B	Compare match or input capture 8B
	Compare match or input capture 0C	—	—	—	Compare match or input capture 3C	Compare match or input capture 4C	Compare match or input capture 5W	Compare match or input capture 6C	Compare match or input capture 7C	Compare match or input capture 8C
	Compare match or input capture 0D	—	—	—	Compare match or input capture 3D	Compare match or input capture 4D	—	Compare match or input capture 6D	Compare match or input capture 7D	Compare match or input capture 8D
	Compare match 0E	—	—	—	—	—	—	—	—	—
	Compare match 0F	—	—	—	—	—	—	—	—	—
Event link function (output)	Seven sources	—	—	—	Five sources	Six sources	—	Five sources	Six sources	—
	Overflow				Overflow	Overflow		Overflow	Overflow	
	—				—	Underflow		—	Underflow	
	Compare match 0A				Compare match 3A	Compare match 4A		Compare match 6A	Compare match 7A	
	Compare match 0B				Compare match 3B	Compare match 4B		Compare match 6B	Compare match 7B	
	Compare match 0C				Compare match 3C	Compare match 4C		Compare match 6C	Compare match 7C	
	Compare match 0D				Compare match 3D	Compare match 4D		Compare match 6D	Compare match 7D	
	Compare match 0E				—	—		—	—	
	Compare match 0F				—	—		—	—	
Event link function (input)	Start counting	—	—	—	Start counting	Start counting	—	—	—	—
	Input capture (to be captured in the TGRA)				Input capture (to be captured in the TGRA)	Input capture (to be captured in the TGRA)				
	Restart counting (clear counter)				Restart counting (clear counter)	Restart counting (clear counter)				

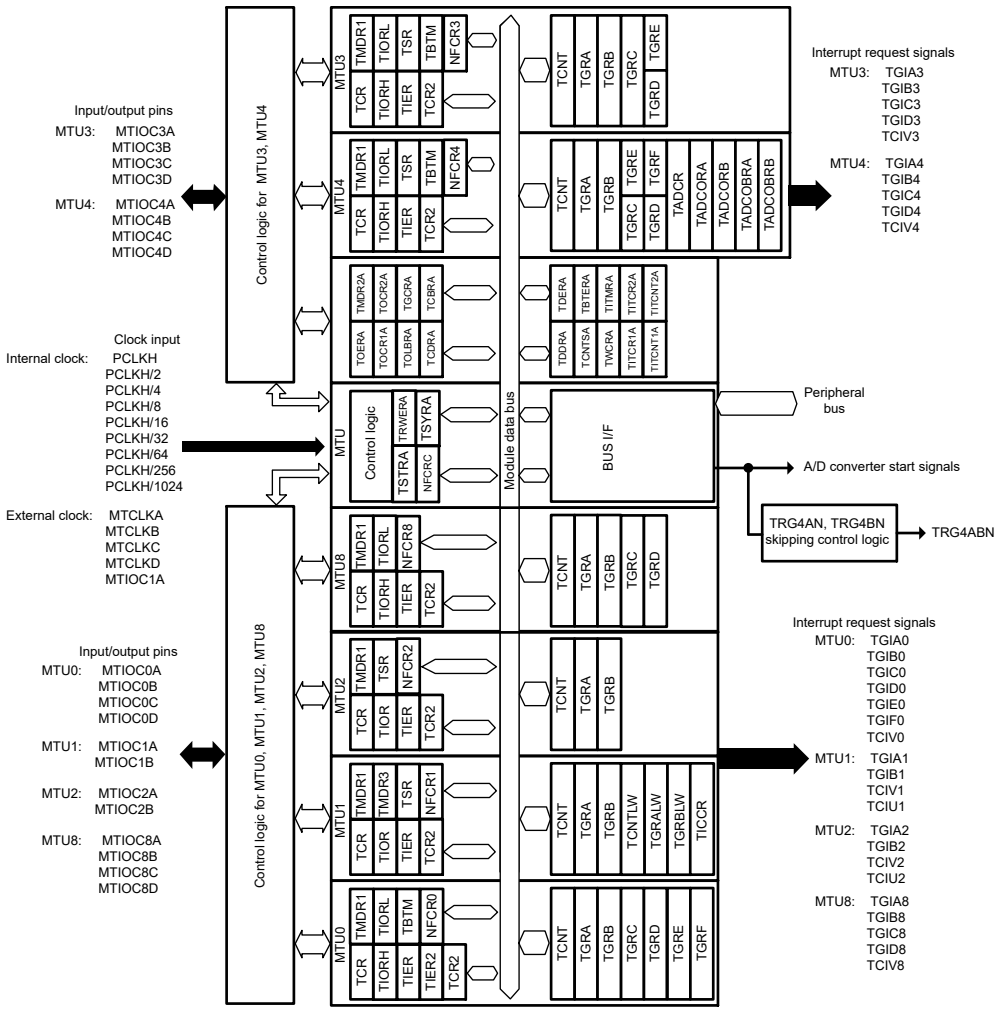
Table 18.2 MTU functions (3 of 3)

✓: Possible —: Not possible

Parameter	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU8
A/D converter start request delaying function	—	—	—	—	—	A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT	—	—	A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT	—
Interrupt skipping 1	—	—	—	—	Skips TGRA compare match interrupts	Skips TCIV interrupts	—	Skips TGRA compare match interrupts	Skips TCIV interrupts	—
Interrupt skipping 2	—	—	—	—	—	Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—	—	Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—
Module stop function	Module stop setting by the MSTPCRC.MSTPCRC00 bit									

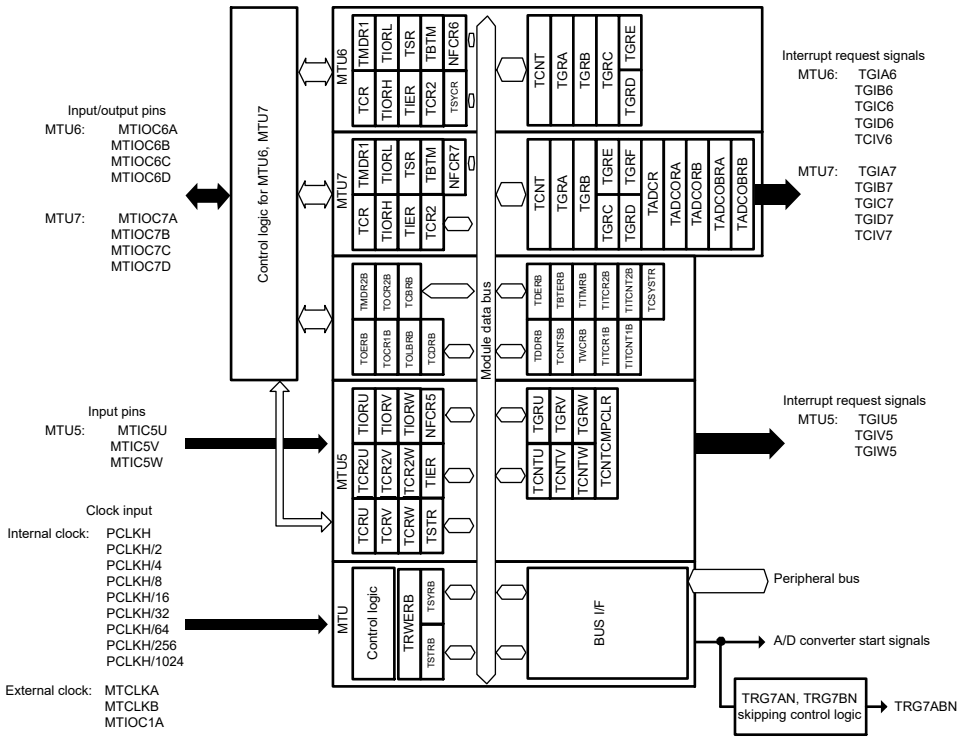
Note 1. When LWA is 1, the TGRALW capture source can be selected from either of the following an input from MTIOC1A or MTU0.TRGA compare match/input capture event. The TGRBLW capture source can be selected from any of the following: an input from MTIOC1B, MTU0.TRGC compare match/input capture event, or MTU8.TGRC compare match event.

Note 2. Capture in MTU8 is supported only in normal mode.



TCR :	Timer control register	TRWERA :	Timer read/write enable register
TCR2 :	Timer control register 2	TOERA :	Timer output master enable register A
TMDR1 :	Timer mode register 1	TOCR1A :	Timer output control register 1A
TMDR2A :	Timer mode register 2A	TOCR2A :	Timer output control register 2A
TMDR3 :	Timer mode register 3	TOLBRA :	Timer output level buffer register A
TIOR :	Timer I/O control register	TGCRA :	Timer gate control register A
TIORH :	Timer I/O control register H	TCNTSA :	Timer subcounter A
TIORL :	Timer I/O control register L	TCDRA :	Timer cycle data register A
TIER :	Timer interrupt enable register	TCBRA :	Timer cycle buffer register A
TIER2 :	Timer interrupt enable register 2	TDDRA :	Timer dead time data register A
TSR :	Timer status register	TDERA :	Timer dead time enable register A
TBTM :	Timer buffer operation transfer mode register	TBTERA :	Timer buffer transfer set register A
TCNT :	Timer counter	TWCRA :	Timer waveform control register A
TCNTLW :	Timer longword counter	TICCR :	Timer input capture control register
TGRA :	Timer general register A	TADCR :	Timer A/D converter start request control register
TGRALW :	Timer longword general register A	TADCORA :	Timer A/D converter start request cycle set register A
TGRB :	Timer general register B	TADCORB :	Timer A/D converter start request cycle set register B
TGRBLW :	Timer longword general register B	TADCOBRA :	Timer A/D converter start request cycle set buffer register A
TGRC :	Timer general register C	TADCOBRB :	Timer A/D converter start request cycle set buffer register B
TGRD :	Timer general register D	TITMRA :	Timer interrupt skipping mode register A
TGRE :	Timer general register E	TITCR1A :	Timer interrupt skipping set register 1A
TGRF :	Timer general register F	TITCR2A :	Timer interrupt skipping set register 2A
TSTRA :	Timer start register A	TITCNT1A :	Timer interrupt skipping counter 1A
TSYRA :	Timer synchronous register A	TITCNT2A :	Timer interrupt skipping counter 2A
TCSYSTR :	Timer counter synchronous start register	NFCRn :	Noise filter control register n

Figure 18.1 Block diagram of MTU (MTU0 to MTU4, MTU8)



TCR : Timer control register
TCR2 : Timer control register 2
TCRU : Timer control register U
TCR2U : Timer control register 2U
TCRV : Timer control register V
TCR2V : Timer control register 2V
TCRW : Timer control register W
TCR2W : Timer control register 2W
TMDR1 : Timer control register 1
TMDR2B : Timer control register 2B
TIORH : Timer I/O control register H
TIORL : Timer I/O control register L
TIORU : Timer I/O control register U
TIORV : Timer I/O control register V
TIORW : Timer I/O control register W
TIER : Timer interrupt enable register
TSR : Timer status register
TBTM : Timer buffer operation transfer mode register
TCNT : Timer counter
TCNTU : Timer counter U
TCNTV : Timer counter V
TCNTW : Timer counter W
TGRA : Timer general register A
TGRB : Timer general register B
TGRC : Timer general register C
TGRD : Timer general register D
TGRE : Timer general register E
TGRF : Timer general register F
TGRU : Timer general register U
TGRV : Timer general register V
TGRW : Timer general register W

TCNTCMPCLR : Timer Compare match clear register
TSTR : Timer start register
TSTRB : Timer start register B
TSYRB : Timer synchronous register B
TSYCR : Timer synchronous clear register
TCSYSTR : Timer counter synchronous start register
TRWERB : Timer read/write enable register B
TOERB: Timer output master enable register B
TCOR1B : Timer output control register 1B
TCOR2B : Timer output control register 2B
TOLBRB : Timer output level buffer register B
TCDRB: Timer cycle data register B
TCBRB : Timer cycle buffer register B
TDDRB : Timer dead time data register B
TDERB : Timer dead time enable register B
TCNTSB : Timer subcounter B
TBTERB : Timer buffer transfer set register B
TWCRB : Timer waveform control register B
TADCR : Timer A/D converter start request control register
TADCORA : Timer A/D converter start request cycle set register A
TADCORB : Timer A/D converter start request cycle set register B
TADCOBRA : Timer A/D converter start request cycle set buffer register A
TADCOBRB : Timer A/D converter start request cycle set buffer register B
TITMRB : Timer interrupt skipping mode register
TITCR1B : Timer interrupt skipping set register 1B
TITCR2B : Timer interrupt skipping set register 2B
TITCNT1B : Timer interrupt skipping counter 1B
TITCNT2B : Timer interrupt skipping counter 2B
NFCRn : Noise filter control register n

Figure 18.2 Block diagram of MTU (MTU5 to MTU7)

Table 18.3 shows the configuration of pins for the MTU.

Table 18.3 Pin configuration of the MTU (1 of 2)

Channel	Pin name	I/O	Function
MTU	MTCLKA	Input	External clock A input pin (MTU1 and MTU2 phase counting mode A phase input)
	MTCLKB	Input	External clock B input pin (MTU1 and MTU2 phase counting mode B phase input)
	MTCLKC	Input	External clock C input pin (MTU2 phase counting mode A phase input)
	MTCLKD	Input	External clock D input pin (MTU2 phase counting mode B phase input)

Table 18.3 Pin configuration of the MTU (2 of 2)

Channel	Pin name	I/O	Function
MTUm (m = 0, 3, 4, 6, 7)	MTIOCmA	I/O	MTUm.TGRA input capture input/output compare output/PWM output pin
	MTIOCmB	I/O	MTUm.TGRB input capture input/output compare output/PWM output pin
	MTIOCmC	I/O	MTUm.TGRC input capture input/output compare output/PWM output pin
	MTIOCmD	I/O	MTUm.TGRD input capture input/output compare output/PWM output pin
MTUn (n = 1, 2)	MTIOCnA	I/O	MTUn.TGRA input capture input/output compare output/PWM output pin
	MTIOCnB	I/O	MTUn.TGRB input capture input/output compare output/PWM output pin
MTU5	MTIC5U	Input	MTU5.TGRU input capture input/external pulse input pin
	MTIC5V	Input	MTU5.TGRV input capture input/external pulse input pin
	MTIC5W	Input	MTU5.TGRW input capture input/external pulse input pin
MTU8	MTIOC8A	I/O	MTU8.TGRA input capture input/output compare output pin
	MTIOC8B	I/O	MTU8.TGRB input capture input/output compare output pin
	MTIOC8C	I/O	MTU8.TGRC input capture input/output compare output pin
	MTIOC8D	I/O	MTU8.TGRD input capture input/output compare output pin

Table 18.4 MTU interrupt sources (1 of 2)

Channel	Name	Interrupt sources	CPU0 GIC request	CPU1 GIC request	DMAC activation
MTU0	TGIA0	MTU0.TGRA input capture/compare match	Possible	Possible	Possible
	TGIB0	MTU0.TGRB input capture/compare match	Possible	Possible	Possible
	TGIC0	MTU0.TGRC input capture/compare match	Possible	Possible	Possible
	TGID0	MTU0.TGRD input capture/compare match	Possible	Possible	Possible
	TCIV0	MTU0.TCNT overflow	Possible	Possible	Not possible
	TGIE0	MTU0.TGRE compare match	Possible	Possible	Not possible
	TGIF0	MTU0.TGRF compare match	Possible	Possible	Not possible
MTU1	TGIA1	MTU1.TGRA input capture/compare match	Possible	Possible	Possible
	TGIB1	MTU1.TGRB input capture/compare match	Possible	Possible	Possible
	TCIV1	MTU1.TCNT overflow	Possible	Possible	Not possible
	TCIU1	MTU1.TCNT underflow	Possible	Possible	Not possible
MTU2	TGIA2	MTU2.TGRA input capture/compare match	Possible	Possible	Possible
	TGIB2	MTU2.TGRB input capture/compare match	Possible	Possible	Possible
	TCIV2	MTU2.TCNT overflow	Possible	Possible	Not possible
	TCIU2	MTU2.TCNT underflow	Possible	Possible	Not possible
MTU3	TGIA3	MTU3.TGRA input capture/compare match	Possible	Possible	Possible
	TGIB3	MTU3.TGRB input capture/compare match	Possible	Possible	Possible
	TGIC3	MTU3.TGRC input capture/compare match	Possible	Possible	Possible
	TGID3	MTU3.TGRD input capture/compare match	Possible	Possible	Possible
	TCIV3	MTU3.TCNT overflow	Possible	Possible	Not possible
MTU4	TGIA4	MTU4.TGRA input capture/compare match	Possible	Possible	Possible
	TGIB4	MTU4.TGRB input capture/compare match	Possible	Possible	Possible
	TGIC4	MTU4.TGRC input capture/compare match	Possible	Possible	Possible
	TGID4	MTU4.TGRD input capture/compare match	Possible	Possible	Possible
	TCIV4	MTU4.TCNT overflow/underflow*1	Possible	Possible	Possible

Table 18.4 MTU interrupt sources (2 of 2)

Channel	Name	Interrupt sources	CPU0 GIC request	CPU1 GIC request	DMAC activation
MTU5	TGIU5	MTU5.TGRU input capture/compare match	Possible	Possible	Possible
	TGIV5	MTU5.TGRV input capture/compare match	Possible	Possible	Possible
	TGIW5	MTU5.TGRW input capture/compare match	Possible	Possible	Possible
MTU6	TGIA6	MTU6.TGRA input capture/compare match	Possible	Possible	Possible
	TGIB6	MTU6.TGRB input capture/compare match	Possible	Possible	Possible
	TGIC6	MTU6.TGRC input capture/compare match	Possible	Possible	Possible
	TGID6	MTU6.TGRD input capture/compare match	Possible	Possible	Possible
	TCIV6	MTU6.TCNT overflow	Possible	Possible	Not possible
MTU7	TGIA7	MTU7.TGRA input capture/compare match	Possible	Possible	Possible
	TGIB7	MTU7.TGRB input capture/compare match	Possible	Possible	Possible
	TGIC7	MTU7.TGRC input capture/compare match	Possible	Possible	Possible
	TGID7	MTU7.TGRD input capture/compare match	Possible	Possible	Possible
	TCIV7	MTU7.TCNT overflow/underflow*1	Possible	Possible	Possible
MTU8	TGIA8	MTU8.TGRA input capture/compare match	Possible	Possible	Possible
	TGIB8	MTU8.TGRB input capture/compare match	Possible	Possible	Possible
	TGIC8	MTU8.TGRC input capture/compare match	Possible	Possible	Possible
	TGID8	MTU8.TGRD input capture/compare match	Possible	Possible	Possible
	TCIV8	MTU8.TCNT overflow	Possible	Possible	Not possible

Note 1. Only in complementary PWM mode.

18.2 Register Map

Table 18.5 MTU3 register map (1 of 5)

Address	Register symbol	Register name	Write protection
0x9000_1290 (MTU_NF)	NFCR0	Noise Filter Control Register 0	—
0x9000_1291 (MTU_NF)	NFCR1	Noise Filter Control Register 1	—
0x9000_1292 (MTU_NF)	NFCR2	Noise Filter Control Register 2	—
0x9000_1293 (MTU_NF)	NFCR3	Noise Filter Control Register 3	—
0x9000_1294 (MTU_NF)	NFCR4	Noise Filter Control Register 4	—
0x9000_1A95 (MTU_NF)	NFCR5	Noise Filter Control Register 5	—
0x9000_1A93 (MTU_NF)	NFCR6	Noise Filter Control Register 6	—
0x9000_1A94 (MTU_NF)	NFCR7	Noise Filter Control Register 7	—
0x9000_1298 (MTU_NF)	NFCR8	Noise Filter Control Register 8	—
0x9000_1299 (MTU_NF)	NFCRC	Noise Filter Control Register C	—
0x9000_1248 (MTU4), 0x9000_1A48 (MTU7)	TADCOBRA	Timer A/D Converter Start Request Cycle Set Buffer Register A	—

Table 18.5 MTU3 register map (2 of 5)

Address	Register symbol	Register name	Write protection
0x9000_124A (MTU4), 0x9000_1A4A (MTU7)	TADCOBRB	Timer A/D Converter Start Request Cycle Set Buffer Register B	—
0x9000_1244 (MTU4), 0x9000_1A44 (MTU7)	TADCORA	Timer A/D Converter Start Request Cycle Set Register A	—
0x9000_1246 (MTU4), 0x9000_1A46 (MTU7)	TADCORB	Timer A/D Converter Start Request Cycle Set Register B	—
0x9000_1240 (MTU4), 0x9000_1A40 (MTU7)	TADCR	Timer A/D Converter Start Request Control Register	—
0x9000_1232 (MTU)	TBTERA	Timer Buffer Transfer Set Register A	—
0x9000_1A32 (MTU)	TBTERB	Timer Buffer Transfer Set Register B	—
0x9000_1238 (MTU3), 0x9000_1239 (MTU4), 0x9000_1326 (MTU0), 0x9000_1A38 (MTU6), 0x9000_1A39 (MTU7)	TBTM	Timer Buffer Operation Transfer Mode Register	—
0x9000_1222 (MTU)	TCBRA	Timer Cycle Buffer Register A	—
0x9000_1A22 (MTU)	TCBRB	Timer Cycle Buffer Register B	—
0x9000_1214 (MTU)	TCDRA	Timer Cycle Data Register A	TRWERA (MTU)
0x9000_1A14 (MTU)	TCDRB	Timer Cycle Data Register B	TRWERB (MTU)
0x9000_1210 (MTU3), 0x9000_1212 (MTU4), 0x9000_1306 (MTU0), 0x9000_1386 (MTU1), 0x9000_1406 (MTU2), 0x9000_1608 (MTU8), 0x9000_1A10 (MTU6), 0x9000_1A12 (MTU7)	TCNT	Timer Counter	TRWERA (MTU3, 4) TRWERB (MTU6, 7)
0x9000_1CB6 (MTU5)	TCNTCMPCLR	Timer Compare Match Clear Register	—
0x9000_13A0 (MTU1)	TCNTLW	Timer Longword Counter	—
0x9000_1220 (MTU)	TCNTSA	Timer Subcounter A	—
0x9000_1A20 (MTU)	TCNTSB	Timer Subcounter B	—
0x9000_1C80 (MTU5)	TCNTU	Timer Counter U	—
0x9000_1C90 (MTU5)	TCNTV	Timer Counter V	—
0x9000_1CA0 (MTU5)	TCNTW	Timer Counter W	—
0x9000_1200 (MTU3), 0x9000_1201 (MTU4), 0x9000_1300 (MTU0), 0x9000_1380 (MTU1), 0x9000_1400 (MTU2), 0x9000_1600 (MTU8), 0x9000_1A00 (MTU6), 0x9000_1A01 (MTU7)	TCR	Timer Control Register	TRWERA (MTU3, 4) TRWERB (MTU6, 7)
0x9000_124C (MTU3), 0x9000_124D (MTU4), 0x9000_1328 (MTU0), 0x9000_1394 (MTU1), 0x9000_140C (MTU2), 0x9000_1606 (MTU8), 0x9000_1A4C (MTU6), 0x9000_1A4D (MTU7)	TCR2	Timer Control Register 2	TRWERA (MTU3, 4) TRWERB (MTU6, 7)
0x9000_1C85 (MTU5)	TCR2U	Timer Control Register 2U	—
0x9000_1C95 (MTU5)	TCR2V	Timer Control Register 2V	—
0x9000_1CA5 (MTU5)	TCR2W	Timer Control Register 2W	—

Table 18.5 MTU3 register map (3 of 5)

Address	Register symbol	Register name	Write protection
0x9000_1C84 (MTU5)	TCRU	Timer Control Register U	—
0x9000_1C94 (MTU5)	TCRV	Timer Control Register V	—
0x9000_1CA4 (MTU5)	TCRW	Timer Control Register W	—
0x9000_1282 (MTU)	TCSYSTR	Timer Counter Synchronous Start Register	—
0x9000_1216 (MTU)	TDDRA	Timer Dead Time Data Register A	TRWERA (MTU)
0x9000_1A16 (MTU)	TDDRB	Timer Dead Time Data Register B	TRWERB (MTU)
0x9000_1234 (MTU)	TDERA	Timer Dead Time Enable Register A	—
0x9000_1A34 (MTU)	TDERB	Timer Dead Time Enable Register B	—
0x9000_120D (MTU)	TGCRA	Timer Gate Control Register A	TRWERA (MTU)
0x9000_1218 (MTU3), 0x9000_121C (MTU4), 0x9000_1308 (MTU0), 0x9000_1388 (MTU1), 0x9000_1408 (MTU2), 0x9000_160C (MTU8), 0x9000_1A18 (MTU6), 0x9000_1A1C (MTU7)	TGRA	Timer General Register A	TRWERA (MTU3, 4) TRWERB (MTU6, 7)
0x9000_13A4 (MTU1)	TGRALW	Timer Longword General Register A	—
0x9000_121A (MTU3), 0x9000_121E (MTU4), 0x9000_130A (MTU0), 0x9000_138A (MTU1), 0x9000_140A (MTU2), 0x9000_1610 (MTU8), 0x9000_1A1A (MTU6), 0x9000_1A1E (MTU7)	TGRB	Timer General Register B	TRWERA (MTU3, 4) TRWERB (MTU6, 7)
0x9000_13A4 (MTU1)	TGRBLW	Timer Longword General Register B	—
0x9000_121A (MTU3), 0x9000_121E (MTU4), 0x9000_130A (MTU0), 0x9000_138A (MTU1), 0x9000_140A (MTU2), 0x9000_1610 (MTU8), 0x9000_1A1A (MTU6), 0x9000_1A1E (MTU7), 0x9000_13A4 (MTU1), 0x9000_1224 (MTU3), 0x9000_1228 (MTU4), 0x9000_130C (MTU0), 0x9000_1614 (MTU8), 0x9000_1A24 (MTU6), 0x9000_1A28 (MTU7)	TGRC	Timer General Register C	—
0x9000_1226 (MTU3), 0x9000_122A (MTU4), 0x9000_130E (MTU0), 0x9000_1618 (MTU8), 0x9000_1A26 (MTU6), 0x9000_1A2A (MTU7)	TGRD	Timer General Register D	—
0x9000_1272 (MTU3), 0x9000_1274 (MTU4), 0x9000_1320 (MTU0), 0x9000_1A72 (MTU6), 0x9000_1A74 (MTU7)	TGRE	Timer General Register E	—
0x9000_1276 (MTU4), 0x9000_1322 (MTU0), 0x9000_1A76 (MTU7)	TGRF	Timer General Register F	—

Table 18.5 MTU3 register map (4 of 5)

Address	Register symbol	Register name	Write protection
0x9000_1C82 (MTU5)	TGRU	Timer General Register U	—
0x9000_1C92 (MTU5)	TGRV	Timer General Register V	—
0x9000_1CA2 (MTU5)	TGRW	Timer General Register W	—
0x9000_1390 (MTU1)	TICCR	Timer Input Capture Control Register	—
0x9000_1208 (MTU3), 0x9000_1209 (MTU4), 0x9000_1304 (MTU0), 0x9000_1384 (MTU1), 0x9000_1404 (MTU2), 0x9000_1604 (MTU8), 0x9000_1A08 (MTU6), 0x9000_1A09 (MTU7), 0x9000_1CB2 (MTU5)	TIER	Timer Interrupt Enable Register	TRWERA (MTU3, 4) TRWERB (MTU6, 7)
0x9000_1324 (MTU0)	TIER2	Timer Interrupt Enable Register 2	—
0x9000_1382 (MTU1), 0x9000_1402 (MTU2)	TIOR	Timer I/O Control Register	—
0x9000_1204 (MTU3), 0x9000_1206 (MTU4), 0x9000_1302 (MTU0), 0x9000_1602 (MTU8), 0x9000_1A04 (MTU6), 0x9000_1A06 (MTU7)	TIORH	Timer I/O Control Register H	TRWERA (MTU3, 4) TRWERB (MTU6, 7)
0x9000_1205 (MTU3), 0x9000_1207 (MTU4), 0x9000_1303 (MTU0), 0x9000_1603 (MTU8), 0x9000_1A05 (MTU6), 0x9000_1A07 (MTU7)	TIORL	Timer I/O Control Register L	TRWERA (MTU3, 4) TRWERB (MTU6, 7)
0x9000_1C86 (MTU5)	TIORU	Timer I/O Control Register U	—
0x9000_1C96 (MTU5)	TIORV	Timer I/O Control Register V	—
0x9000_1CA6 (MTU5)	TIORW	Timer I/O Control Register W	—
0x9000_1231 (MTU)	TITCNT1A	Timer Interrupt Skipping Counter 1A	—
0x9000_1A31 (MTU)	TITCNT1B	Timer Interrupt Skipping Counter 1B	—
0x9000_123C (MTU)	TITCNT2A	Timer Interrupt Skipping Counter 2A	—
0x9000_1A3C (MTU)	TITCNT2B	Timer Interrupt Skipping Counter 2B	—
0x9000_1230 (MTU)	TITCR1A	Timer Interrupt Skipping Set Register 1A	—
0x9000_1A30 (MTU)	TITCR1B	Timer Interrupt Skipping Set Register 1B	—
0x9000_123B (MTU)	TITCR2A	Timer Interrupt Skipping Set Register 2A	—
0x9000_1A3B (MTU)	TITCR2B	Timer Interrupt Skipping Set Register 2B	—
0x9000_123A (MTU)	TITMRA	Timer Interrupt Skipping Mode Register A	—
0x9000_1A3A (MTU)	TITMRB	Timer Interrupt Skipping Mode Register B	—
0x9000_1202 (MTU3), 0x9000_1203 (MTU4), 0x9000_1301 (MTU0), 0x9000_1381 (MTU1), 0x9000_1401 (MTU2), 0x9000_1601 (MTU8), 0x9000_1A02 (MTU6), 0x9000_1A03 (MTU7)	TMDR1	Timer Mode Register 1	TRWERA (MTU3, 4) TRWERB (MTU6, 7)
0x9000_1270 (MTU)	TMDR2A	Timer Mode Register 2A	—
0x9000_1A70 (MTU)	TMDR2B	Timer Mode Register 2B	—

Table 18.5 MTU3 register map (5 of 5)

Address	Register symbol	Register name	Write protection
0x9000_1391 (MTU1)	TMDR3	Timer Mode Register 3	—
0x9000_120E (MTU)	TOCR1A	Timer Output Control Register 1A	TRWERA (MTU)
0x9000_1A0E (MTU)	TOCR1B	Timer Output Control Register 1B	TRWERB (MTU)
0x9000_120F (MTU)	TOCR2A	Timer Output Control Register 2A	TRWERA (MTU)
0x9000_1A0F (MTU)	TOCR2B	Timer Output Control Register 2B	TRWERB (MTU)
0x9000_120A (MTU)	TOERA	Timer Output Master Enable Register A	TRWERA (MTU)
0x9000_1A0A (MTU)	TOERB	Timer Output Master Enable Register B	TRWERB (MTU)
0x9000_1236 (MTU)	TOLBRA	Timer Output Level Buffer Register A	—
0x9000_1A36 (MTU)	TOLBRB	Timer Output Level Buffer Register B	—
0x9000_1284 (MTU)	TRWERA	Timer Read/Write Enable Register A	—
0x9000_1A84 (MTU)	TRWERB	Timer Read/Write Enable Register B	—
0x9000_122C (MTU3), 0x9000_122D (MTU4), 0x9000_1385 (MTU1), 0x9000_1405 (MTU2), 0x9000_1A2C (MTU6), 0x9000_1A2D (MTU7)	TSR	Timer Status Register	—
0x9000_1CB4 (MTU5)	TSTR	Timer Start Register	—
0x9000_1280 (MTU)	TSTRA	Timer Start Register A	—
0x9000_1A80 (MTU)	TSTRB	Timer Start Register B	—
0x9000_1A50 (MTU6)	TSYCR	Timer Synchronous Clear Register	—
0x9000_1281 (MTU)	TSYRA	Timer Synchronous Register A	—
0x9000_1A81 (MTU)	TSYRB	Timer Synchronous Register B	—
0x9000_1260 (MTU)	TWCRA	Timer Waveform Control Register A	—
0x9000_1A60 (MTU)	TWCRB	Timer Waveform Control Register B	—

Table 18.6 MTU3 related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	—	MSTPCRC.MSTPCRC00	SLVACCCTL7.LLPP_SL*1

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

18.3 Register Descriptions

18.3.1 TCR : Timer Control Register

Base address: MTU0 = 0x9000_1300, MTU1 = 0x9000_1380, MTU2 = 0x9000_1400, MTU3 = 0x9000_1100
MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00, MTU8 = 0x9000_1600

Offset address: 0x000 (MTU0, MTU1, MTU2, MTU8), 0x100 (MTU3, MTU6), 0x001 (MTU4, MTU7)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC[2:0]	Time Prescaler Select See section 18.3.3. TCR2 — Timer Control Registers.	R/W

Bit	Symbol	Function	R/W
4:3	CKEG[1:0]	Clock Edge Select 0 0: Count at rising edge 0 1: Count at falling edge 1 x: Count at both edges	R/W
7:5	CCLR[2:0]	Counter Clear Source Select See Table 18.7 and Table 18.8.	R/W

The TCR register controls the TCNT operation for each channel in combination with the TCR2 register. TCR values should be specified only while TCNT operation is stopped.

TPSC[2:0] bits (Time Prescaler Select)

These bits select the TCNT counter clock. The count clock source can be selected independently for each channel. For details, see section 18.3.3. TCR2 — Timer Control Registers.

CKEG[1:0] bits (Clock Edge Select)

The CKEG[1:0] bits select the input clock edge, including the MTIOC1A pin. When the input clock is counted at both edges, the input clock period is halved (e.g. PCLKH/4 at both edges = PCLKH/2 at rising edge). If phase counting mode is used on MTU1 and MTU2, the setting of these bits is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is PCLKH/2 or slower. When PCLKH/1 or the overflow/underflow in another channel is selected for the input clock, a value can be written to these bits but counter operation compiles with the initial value.

CCLR[2:0] bits (Counter Clear Source Select)

These bits select the TCNT counter clearing source. See Table 18.7 and Table 18.8 for details.

Table 18.7 CCLR[2:0] (MTU0, MTU3, MTU4, MTU6, MTU7, MTU8)

Channel	Bit 7 CCLR[2]	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU0 MTU3 MTU4 MTU6 MTU7 MTU8	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare match/input capture*2
	1	1	0	TCNT cleared by TGRD compare match/input capture*2
	1	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYRA.SYNC or TSYRB.SYNC bit to 1 except for MTU8.
Note 2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority and compare match/input capture does not occur.

Table 18.8 CCLR[2:0] (MTU1 and MTU2)

Channel	Bit 7 Reserved*2	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU1 MTU2	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYRA.SYNC and TSYRB.SYNC bits to 1.
Note 2. Bit 7 is reserved in MTU1 and MTU2. It is read as 0. The write value is ignored.

18.3.2 TCRx : Timer Control Register x (x = U, V, W)

Base address: MTU5 = 0x9000_1C00

Offset address: 0x084 (TCRU)
0x094 (TCRV)
0x0A4 (TCRW)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TPSC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TPSC[1:0]	Time Prescaler Select See Table 18.14.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

TPSC[1:0] bits (Time Prescaler Select)

The TPSC[1:0] bits select the TCNT counter clock. For details, see Table 18.14.

18.3.3 TCR2 — Timer Control Registers

18.3.3.1 MTU0.TCR2 : Timer Control Register 2

Base address: MTU0 = 0x9000_1300

Offset address: 0x28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 18.9.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. The count clock source can be selected independently for each channel. For details, see Table 18.9.

Table 18.9 TCR2.TPSC2[2:0], TCR.TPSC[2:0] (MTU0)

Channel	TCR2 register			TCR register			Description
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	
MTU0	0	0	0	0	0	0	Internal clock: counts on PCLKH/1
	0	0	0	0	0	1	Internal clock: counts on PCLKH/4
	0	0	0	0	1	0	Internal clock: counts on PCLKH/16
	0	0	0	0	1	1	Internal clock: counts on PCLKH/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	External clock: counts on MTCLKD pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKH/2
	0	1	0	x	x	x	Internal clock: counts on PCLKH/8
	0	1	1	x	x	x	Internal clock: counts on PCLKH/32
	1	0	0	x	x	x	Internal clock: counts on PCLKH/256
	1	0	1	x	x	x	Internal clock: counts on PCLKH/1024
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	External clock: counts on MTIOC1A pin input

18.3.3.2 MTU1.TCR2 : Timer Control Register 2

Base address: MTU1 = 0x9000_1380

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PCB[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 18.10.	R/W
4:3	PCB[1:0]	Functional Expansion Control for Phase Counting Modes 2, 3, and 5 Controls expansion of functionality to modes 2, 3, and 5 of phase-counting mode	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. The count clock source can be selected independently for each channel. For details, see Table 18.10.

PCB[1:0] bits (Functional Expansion Control for Phase Counting Modes 2, 3, and 5)

The PCB[1:0] bits control extended functions for phase counting modes 2, 3, and 5 in MTU1 and MTU2. For details, see section 18.4.6. Phase Counting Mode.

Table 18.10 TCR2.TPSC2[2:0], TCR.TPSC[2:0] (MTU1)

Channel	TCR2 register			TCR register			Description
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	
MTU1	0	0	0	0	0	0	Internal clock: counts on PCLKH/1
	0	0	0	0	0	1	Internal clock: counts on PCLKH/4
	0	0	0	0	1	0	Internal clock: counts on PCLKH/16
	0	0	0	0	1	1	Internal clock: counts on PCLKH/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	Internal clock: counts on PCLKH/256
	0	0	0	1	1	1	Overflow/underflow of MTU2.TCNT
	0	0	1	x	x	x	Internal clock: counts on PCLKH/2
	0	1	0	x	x	x	Internal clock: counts on PCLKH/8
	0	1	1	x	x	x	Internal clock: counts on PCLKH/32
	1	0	0	x	x	x	Internal clock: counts on PCLKH/1024
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

Note: This setting has no effect when MTU1 is in phase counting mode.

18.3.3.3 MTU2.TCR2 : Timer Control Register 2

Base address: MTU2 = 0x9000_1400

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PCB[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 18.11.	R/W
4:3	PCB[1:0]	Functional Expansion Control for Phase Counting Modes 2, 3, and 5 Controls expansion of functionality to modes 2, 3, and 5 of phase-counting mode.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. The count clock source can be selected independently for each channel. See Table 18.11 for details.

PCB[1:0] bits (Functional Expansion Control for Phase Counting Modes 2, 3, and 5)

The PCB[1:0] bits control extended functions for phase counting modes 2, 3, and 5 in MTU1 and MTU2. See section 18.4.6. Phase Counting Mode for details.

Table 18.11 TCR2.TPSC2[2:0], TCR.TPSC[2:0] (MTU2)

Channel	TCR2 register			TCR register			Description
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	
MTU2	0	0	0	0	0	0	Internal clock: counts on PCLKH/1
	0	0	0	0	0	1	Internal clock: counts on PCLKH/4
	0	0	0	0	1	0	Internal clock: counts on PCLKH/16
	0	0	0	0	1	1	Internal clock: counts on PCLKH/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	Internal clock: counts on PCLKH/1024
	0	0	1	x	x	x	Internal clock: counts on PCLKH/2
	0	1	0	x	x	x	Internal clock: counts on PCLKH/8
	0	1	1	x	x	x	Internal clock: counts on PCLKH/32
	1	0	0	x	x	x	Internal clock: counts on PCLKH/256
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

Note: This setting has no effect when the MTU2 is in phase counting mode.

18.3.3.4 MTUm.TCR2 : Timer Control Register 2 (m = 3, 4, 6, 7)

Base address: MTU3 = 0x9000_1100, MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00

Offset address: 0x4D (MTU4, MTU7)
0x14C (MTU3, MTU6)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 18.12.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. The count clock source can be selected independently for each channel. See Table 18.12 for details.

Table 18.12 TCR2.TPSC2[2:0], TCR.TPSC[2:0] (MTU3, MTU4, MTU6, MTU7)

Channel	TCR2 register			TCR register			Description
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	
MTU3 MTU4 MTU6 MTU7	0	0	0	0	0	0	Internal clock: counts on PCLKH/1
	0	0	0	0	0	1	Internal clock: counts on PCLKH/4
	0	0	0	0	1	0	Internal clock: counts on PCLKH/16
	0	0	0	0	1	1	Internal clock: counts on PCLKH/64
	0	0	0	1	0	0	Internal clock: counts on PCLKH/256
	0	0	0	1	0	1	Internal clock: counts on PCLKH/1024
	0	0	0	1	1	0	External clock: counts on MTCLKA pin input
	0	0	0	1	1	1	External clock: counts on MTCLKB pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKH/2
	0	1	0	x	x	x	Internal clock: counts on PCLKH/8
	0	1	1	x	x	x	Internal clock: counts on PCLKH/32
	1	0	0	x	x	x	Setting prohibited
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

18.3.3.5 MTU8.TCR2 : Timer Control Register 2

Base address: MTU8 = 0x9000_1600

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 18.12.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. The count clock source can be selected independently for each channel. See Table 18.12 for details.

Table 18.13 TCR2.TPSC2[2:0], TCR.TPSC[2:0] (MTU8)

Channel	TCR2 register			TCR register			Description
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	
MTU8	0	0	0	0	0	0	Internal clock: counts on PCLKH/1
	0	0	0	0	0	1	Internal clock: counts on PCLKH/4
	0	0	0	0	1	0	Internal clock: counts on PCLKH/16
	0	0	0	0	1	1	Internal clock: counts on PCLKH/64
	0	0	0	1	0	0	Internal clock: counts on PCLKH/256
	0	0	0	1	0	1	Internal clock: counts on PCLKH/1024
	0	0	0	1	1	0	External clock: counts on MTCLKA pin input
	0	0	0	1	1	1	External clock: counts on MTCLKB pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKH/2
	0	1	0	x	x	x	Internal clock: counts on PCLKH/8
	0	1	1	x	x	x	Internal clock: counts on PCLKH/32
	1	0	0	x	x	x	Setting prohibited
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

18.3.3.6 MTU5.TCR2x : Timer Control Register 2x (x = U, V, W)

Base address: MTU5 = 0x9000_1C00

Offset address: 0x85 (TCR2U)
0x95 (TCR2V)
0xA5 (TCR2W)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CKEG[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 18.14.	R/W
4:3	CKEG[1:0]	Clock Edge Select 0 0: Counts at the rising edge 0 1: Counts at the falling edge 1 x: Counts at both edges	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. See Table 18.14 for details.

CKEG[1:0] bits (Clock Edge Select)

The CKEG[1:0] bits select the edge of the count clock signal input from the MTIOC1A pin.

Table 18.14 TPSC[1:0], CKEG[1:0], TPSC2[2:0] (MTU5)

Channel	TCR2 register					TCR register		Description
	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
	CKEG[1]	CKEG[0]	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[1]	TPSC[0]	
MTU5	x	x	0	0	0	0	0	Internal clock: counts on PCLKH/1
	x	x	0	0	0	0	1	Internal clock: counts on PCLKH/4
	x	x	0	0	0	1	0	Internal clock: counts on PCLKH/16
	x	x	0	0	0	1	1	Internal clock: counts on PCLKH/64
	x	x	0	0	1	x	x	Internal clock: counts on PCLKH/2
	x	x	0	1	0	x	x	Internal clock: counts on PCLKH/8
	x	x	0	1	1	x	x	Internal clock: counts on PCLKH/32
	x	x	1	0	0	x	x	Internal clock: counts on PCLKH/256
	x	x	1	0	1	x	x	Internal clock: counts on PCLKH/1024
	x	x	1	1	0	x	x	Setting prohibited
	0	0	1	1	1	x	x	Counts at the rising edge on MTIOC1A pin input
	0	1	1	1	1	x	x	Counts at the falling edge on MTIOC1A pin input
	1	x	1	1	1	x	x	Counts at the both edges on MTIOC1A pin input

Note: Bits 7 to 2 in the TCR register are reserved for MTU5. These bits are read as 0. The write value should be 0.

18.3.4 TMDR1 : Timer Mode Register 1

Base address: MTU0 = 0x9000_1300

Offset address: 0x001

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	BFE	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU1 = 0x9000_1380, MTU2 = 0x9000_1400

Offset address: 0x001

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU3 = 0x9000_1100, MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00
MTU8 = 0x9000_1600

Offset address: 0x102 (MTU3, MTU6), 0x003 (MTU4, MTU7), 0x001 (MTU8)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	MD[3:0]	Mode Select These bits specify the timer operating mode. See Table 18.15 for details.	R/W

Bit	Symbol	Function	R/W
4	BFA	Buffer Operation A 0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation	R/W
5	BFB	Buffer Operation B 0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation	R/W
6	BFE	Buffer Operation E 0: MTU0.TGRE and MTU0.TGRF operate normally 1: MTU0.TGRE and MTU0.TGRF used together for buffer operation	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

The TMDR1 register specifies the operating mode of each channel. The MTU has a total of eight TMDR1 registers, one each for MTU0 to MTU4, MTU6, MTU7, and MTU8. TMDR1 register values should be specified only while TCNT operation is stopped.

MD[3:0] bits (Mode Select)

Table 18.15 Operating mode setting by MD[3:0] bits (MTU0 to MTU4 and MTU6 to MTU8)

MD[3]	MD[2]	MD[1]	MD[0]	Description	MTU0	MTU1	MTU2	MTU3	MTU4	MTU6	MTU7	MTU8
0	0	0	0	Normal mode	✓	✓	✓	✓	✓	✓	✓	✓
0	0	0	1	Setting prohibited								
0	0	1	0	PWM mode 1	✓	✓	✓	✓	✓	✓	✓	
0	0	1	1	PWM mode 2	✓	✓	✓					
0	1	0	0	Phase counting mode 1		✓	✓					
0	1	0	1	Phase counting mode 2		✓	✓					
0	1	1	0	Phase counting mode 3		✓	✓					
0	1	1	1	Phase counting mode 4		✓	✓					
1	0	0	0	Reset-synchronized PWM mode*1				✓		✓		
1	0	0	1	Phase counting mode 5		✓	✓					
1	0	1	0	Setting prohibited								
1	0	1	1	Setting prohibited								
1	1	0	0	Setting prohibited								
1	1	0	1	Complementary PWM mode 1 (transfer at crest)*1				✓		✓		
1	1	1	0	Complementary PWM mode 2 (transfer at trough)*1				✓		✓		
1	1	1	1	Complementary PWM mode 3 (transfer at crest and trough)*1				✓		✓		

Note: Only set the corresponding operating mode listed above for each channel.
Note 1. Reset-synchronized PWM mode and complementary PWM mode can only be set for MTU3 and MTU6. When MTU3 or MTU6 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 or MTU7 settings become ineffective and automatically conform to the MTU3 or MTU6 setting, respectively. MTU4 and MTU7 should be set to the initial values (normal mode).

BFA bit (Buffer Operation A)

The BFA bit specifies whether to operate TGRA in the normal way or to use TGRA and TGRC together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare does not take place in a mode other than complementary PWM mode, but TGRC compare match takes place in complementary PWM mode. If MTU4 compare match takes place in the Tb interval of complementary PWM mode, clear the TGIEC bit in the timer interrupt enable register (MTU4.TIER) to 0.

In reset synchronized PWM mode, buffer operation in MTU3 and MTU4 (MTU6 and MTU7) depends on the settings in the BFA bit of MTU3.TMDR1. The BFA bit of MTU4.TMDR1 should be set to 0. In complementary PWM mode, buffer

operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the BFA bit setting in MTU3.TMDR1 (MTU6.TMDR1). Set the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) to 0.

In MTU1 and MTU2, which have no TGRC, this bit is reserved. It is read as 0. The write value should be 0. For details about the Tb interval of complementary PWM mode, see Figure 18.50.

BFB bit (Buffer Operation B)

The BFB bit specifies whether to operate TGRB in the normal way or to use TGRB and TGRD together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare does not usually take place, but TGRD compare match takes place in complementary PWM mode. If MTU4 compare match takes place in the Tb interval of complementary PWM mode, clear the TGIED bit in the timer interrupt enable register (MTU4.TIER) to 0.

In reset synchronized PWM mode, buffer operation in MTU3 and MTU4 (MTU6 and MTU7) depends on the settings in the BFB bit of MTU3.TMDR1. The BFB bit of MTU4.TMDR1 should be set to 0. In complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the BFB bit setting in MTU3.TMDR1 (MTU6.TMDR1). Set the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) to 0.

In MTU1 and MTU2, which have no TGRD, this bit is reserved. It is read as 0. The write value should be 0. For details about the Tb interval of complementary PWM mode, see Figure 18.50.

BFE bit (Buffer Operation E)

The BFE bit specifies whether to operate MTU0.TGRE and MTU0.TGRF in the normal way or to use them together for buffer operation. TGRF compare match takes place even when TGRF is used as a buffer register.

In MTU1 to MTU4, MTU6, MTU7, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

18.3.5 TMDR2x : Timer Mode Register 2x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x270 (TMDR2A)
0xA70 (TMDR2B)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DRS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DRS	Double Buffer Select 0: Double buffer function is disabled 1: Double buffer function is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

TMDR2A and TMDR2B specify the double buffer function in complementary PWM mode 3 (transfer at the crest and trough of the counter value). The MTU has two TMDR2 registers, and one each for MTU3 (TMDR2A) and MTU6 (TMDR2B). TMDR2A and TMDR2B values should be specified only while TCNT operation is stopped.

DRS bit (Double Buffer Select)

This bit enables or disables the double buffer function in complementary PWM mode.

18.3.6 TMDR3 : Timer Mode Register 3

Base address: MTU1 = 0x9000_1380

Offset address: 0x011

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PHCK SEL	LWA
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	LWA	MTU1/MTU2 Combination Longword Access Control 0: 16-bit access is enabled. 1: 32-bit access is enabled.	R/W
1	PHCKSEL	External Input Phase Clock Select Selects the external clock pin for phase counting mode. 0: MTCLKA and MTCLKB are selected for the external phase clock. 1: MTCLKC and MTCLKD are selected for the external phase clock.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

The TMDR3 register controls access in longword units to the combination of MTU1 and MTU2, and there is only one register, which is in MTU1. Such access can proceed when the LWA bit is set in the ways listed in Table 18.16.

LWA bit (MTU1/MTU2 Combination Longword Access Control)

This bit selects a 32-bit access in combination of MTU1 and MTU2. Set this bit only in 32-bit phase counting mode.

When LWA is set to 0, the MTU1.TCNTLW, MTU1.TGRALW, and MTU1.TGRBLW registers cannot be accessed. These bits are read as 0x00000000.

When LWA is set to 1, the MTU1.TCNT, MTU2.TCNT, MTU1.TGRA, MTU2.TGRA, MTU1.TGRB, and MTU2.TGRB registers cannot be accessed. These bits are read as 0x0000.

The settings of the control registers (TCR, TCR2, TIOR, and TMDR1) in MTU1 take priority because MTU1 and MTU2 operates together while LWA = 1.

In this case, the settings of the control registers in MTU2 are ignored. Furthermore, MTU2 input capture and compare match are disabled, which in turn disables any linked operation with the ELC.

The cascaded connection of MTU1 and MTU2 with the LWA bit set to 1 can only be used in phase counting mode, but not in normal mode, PWM mode 1, or PWM mode 2. Select phase counting mode when LWA = 1.

Initialize the counters and general registers in MTU1 and MTU2 prior to rewriting to the LWA bit.

PHCKSEL bit (External Input Phase Clock Select)

When the MTU1 and MTU2 registers are combined for 32-bit phase counting mode or MTU2 phase counting mode, this bit selects the external input clock pin on which A- or B-phase signal is to be input. See Table 18.63 for details.

Table 18.16 Setting and combination of the TMDR3 register

Register	TMDR3.LWA = 0		TMDR3.LWA = 1	
	Symbol	Access mode	Symbol	Access mode
Counter in MTU1*1	MTU1.TCNT	Word	MTU1.TCNTLW	Longword
Counter in MTU2	MTU2.TCNT	Word		
General register A in MTU1	MTU1.TGRA	Word	MTU1.TGRALW	Longword
General register A in MTU2	MTU2.TGRA	Word		
General register B in MTU1	MTU1.TGRB	Word	MTU1.TGRBLW	Longword
General register B in MTU2	MTU2.TGRB	Word		

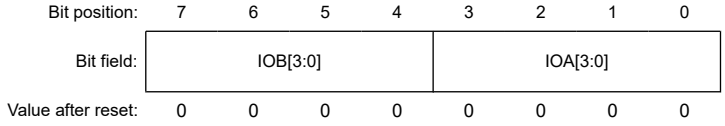
Note 1. When the LWA bit is set to 1, setting the count clock for MTU1 as MTU2.TCNT overflow/underflow is not required.

18.3.7 TIOR — Timer I/O Control Registers

18.3.7.1 MTUm.TIOR : Timer I/O Control Register (m = 1, 2)

Base address: MTU1 = 0x9000_1380
MTU2 = 0x9000_1400

Offset address: 0x002



Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 For MTU1, see Table 18.17. For MTU2, see Table 18.19.	R/W
7:4	IOB[3:0]	I/O Control B*1 For MTU1, see Table 18.18. For MTU2, see Table 18.20.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIOR register controls the TGR register. The TIOR register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIOR is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIOR is affected by the TMDR1 setting.

Table 18.17 IOA[3:0] bits enumerated value and function for MTU1

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC1A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Input capture at generation of MTU0.TGRA compare match/ input capture.

Table 18.18 IOB[3:0] bits enumerated value and function for MTU1

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC1B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	0	0		Input capture at occurrence of compare match or input capture in the MTU0.TGRC register
1	1	1	x		Input capture at occurrence of compare match or input capture in the MTU8.TGRC register

Table 18.19 IOA[3:0] bits enumerated value and function for MTU2

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC2A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Table 18.20 IOB[3:0] bits enumerated value and function for MTU2 (1 of 2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC2B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.

Table 18.20 IOB[3:0] bits enumerated value and function for MTU2 (2 of 2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC2B pin function
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

18.3.7.2 MTU0.TIORH : Timer I/O Control Register H

Base address: MTU0 = 0x9000_1300

Offset address: 0x002

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 18.21.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 18.22.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 18.21 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC0A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	0	0		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*1
1	1	1	x		Input capture on generation of compare match with MTU8.TGRC

Note 1. When PCLKH/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKH/1 as the count clock for MTU1.

Table 18.22 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC0B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*1

Note 1. When PCLKH/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKH/1 as the count clock for MTU1.

18.3.7.3 MTU3.TIORH : Timer I/O Control Register H

Base address: MTU3 = 0x9000_1100

Offset address: 0x104

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 18.23.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 18.24.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 18.23 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC3A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Table 18.24 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC3B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

18.3.7.4 MTU4.TIORH : Timer I/O Control Register H

Base address: MTU4 = 0x9000_1200

Offset address: 0x006

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 18.25.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 18.26.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 18.25 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC4A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Table 18.26 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC4B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

18.3.7.5 MTU6.TIORH : Timer I/O Control Register H

Base address: MTU6 = 0x9000_1900

Offset address: 0x104

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 18.27.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 18.28.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 18.27 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC6A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Table 18.28 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC6B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

18.3.7.6 MTU7.TIORH : Timer I/O Control Register H

Base address: MTU7 = 0x9000_1A00

Offset address: 0x006

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 18.29.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 18.30.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 18.29 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC7A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Table 18.30 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC7B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

18.3.7.7 MTU8.TIORH : Timer I/O Control Register H

Base address: MTU8 = 0x9000_1600

Offset address: 0x002

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 18.31.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 18.32.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 18.31 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC8A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Table 18.32 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC8B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).

18.3.7.8 MTU0.TIORL : Timer I/O Control Register L

Base address: MTU0 = 0x9000_1300

Offset address: 0x003

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 18.33.	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 18.34.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 18.33 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC0C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*2

Note 1. When the MTU0.TMDR1.BFA bit is set to 1 and MTU0.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.
Note 2. When PCLKH/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKH/1 as the count clock for MTU1.

Table 18.34 IOD[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC0D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*2

Note 1. When the MTU0.TMDR1.BFB bit is set to 1 and MTU0.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.
Note 2. When PCLKH/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKH/1 as the count clock for MTU1.

18.3.7.9 MTU3.TIORL : Timer I/O Control Register L

Base address: MTU3 = 0x9000_1100

Offset address: 0x105

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 18.35.	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 18.36.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 18.35 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC3C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU3.TMDR1.BFA bit is set to 1 and MTU3.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.36 IOD[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC3D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU3.TMDR1.BFB bit is set to 1 and MTU3.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

18.3.7.10 MTU4.TIORL : Timer I/O Control Register L

Base address: MTU4 = 0x9000_1200

Offset address: 0x007

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 18.37.	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 18.38.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 18.37 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC4C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.38 IOD[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC4D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

18.3.7.11 MTU6.TIORL : Timer I/O Control Register L

Base address: MTU6 = 0x9000_1900

Offset address: 0x105

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 18.39.	R/W

Bit	Symbol	Function	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 18.40.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 18.39 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC6C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU6.TMDR1.BFA bit is set to 1 and MTU6.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.40 IOD[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC6D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU6.TMDR1.BFB bit is set to 1 and MTU6.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

18.3.7.12 MTU7.TIORL : Timer I/O Control Register L

Base address: MTU7 = 0x9000_1A00

Offset address: 0x007

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 18.41.	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 18.42.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 18.41 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC7C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU7.TMDR1.BFA bit is set to 1 and MTU7.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.42 IOD[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC7D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU7.TMDR1.BFB bit is set to 1 and MTU7.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

18.3.7.13 MTU8.TIORL : Timer I/O Control Register L

Base address: MTU8 = 0x9000_1600

Offset address: 0x003

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 18.43.	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 18.44.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 18.43 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC8C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU8.TMDR1.BFA bit is set to 1 and MTU8.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 18.44 IOD[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC8D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU8.TMDR1.BFB bit is set to 1 and MTU8.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

18.3.7.14 MTU5.TIORx : Timer I/O Control Register x (x = U, V, W)

Base address: MTU5 = 0x9000_1C00

Offset address: 0x086 (TIORU)
0x096 (TIORV)
0x0A6 (TIORW)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	IOC[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	IOC[4:0]	I/O Control C For details, see Table 18.45.	R/W

Bit	Symbol	Function	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The TIORx register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORx is valid when the counter is stopped (the CST5 bit in TSTR is set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORx is affected by the TMDR1 setting.

Table 18.45 IOC[4:0] bits enumerated value and function for MTU5

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[4]	IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRU, TGRV, TGRW register function	MTIC5U, MTIC5V, MTIC5W pin function
0	0	0	0	0	Output compare register	No function
0	0	0	0	1		Setting prohibited
0	0	0	1	x		Setting prohibited
0	0	1	x	x		Setting prohibited
0	1	x	x	x		Setting prohibited
1	0	0	0	0	Input capture register	Setting prohibited
1	0	0	0	1		Input capture at rising edge.
1	0	0	1	0		Input capture at falling edge.
1	0	0	1	1		Input capture at both edges.
1	0	1	x	x		Input capture on generation of compare match with MTU8.TGRC
1	1	0	0	0		Setting prohibited
1	1	0	0	1		Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	0	1	0		Measurement of low pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	0	1	1		Measurement of low pulse width of external input signal. Capture at crest and trough of complementary PWM mode.
1	1	1	0	0		Setting prohibited
1	1	1	0	1		Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	1	1	0		Measurement of high pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	1	1	1		Measurement of high pulse width of external input signal. Capture at crest and trough of complementary PWM mode.

18.3.8 TCNTCMPCLR : Timer Compare Match Clear Register

Base address: MTU5 = 0x9000_1C00

Offset address: 0x0B6

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPCLR5W	TCNT Compare Clear 5W 0: Disables MTU5.TCNTW to be cleared to 0x0000 at MTU5.TCNTW and MTU5.TGRW compare match or input capture 1: Enables MTU5.TCNTW to be cleared to 0x0000 at MTU5.TCNTW and MTU5.TGRW compare match or input capture	R/W
1	CMPCLR5V	TCNT Compare Clear 5V 0: Disables MTU5.TCNTV to be cleared to 0x0000 at MTU5.TCNTV and MTU5.TGRV compare match or input capture 1: Enables MTU5.TCNTV to be cleared to 0x0000 at MTU5.TCNTV and MTU5.TGRV compare match or input capture	R/W
2	CMPCLR5U	TCNT Compare Clear 5U 0: Disables MTU5.TCNTU to be cleared to 0x0000 at MTU5.TCNTU and MTU5.TGRU compare match or input capture 1: Enables MTU5.TCNTU to be cleared to 0x0000 at MTU5.TCNTU and MTU5.TGRU compare match or input capture	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

TCNTCMPCLR specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW. The MTU has one TCNTCMPCLR (on MTU5).

18.3.9 TIER — Timer Interrupt Enable Registers

18.3.9.1 MTUm.TIER : Timer Interrupt Enable Register (m = 0 to 4, 6 to 8)

Base address: MTU1 = 0x9000_1380, MTU2 = 0x9000_1400

Offset address: 0x004 (MTU1, MTU2)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU0 = 0x9000_1300, MTU3 = 0x9000_1100, MTU6 = 0x9000_1900

Offset address: 0x004 (MTU0), 0x108 (MTU3, MTU6)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU4 = 0x9000_1200, MTU7 = 0x9000_1A00

Offset address: 0x009 (MTU4, MTU7)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU8 = 0x9000_1600

Offset address: 0x004 (MTU8)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TGIEA	TGR Interrupt Enable A 0: Interrupt requests (TGIA) disabled 1: Interrupt requests (TGIA) enabled	R/W
1	TGIEB	TGR Interrupt Enable B 0: Interrupt requests (TGIB) disabled 1: Interrupt requests (TGIB) enabled	R/W
2	TGIEC	TGR Interrupt Enable C 0: Interrupt requests (TGIC) disabled 1: Interrupt requests (TGIC) enabled	R/W
3	TGIED	TGR Interrupt Enable D 0: Interrupt requests (TGID) disabled 1: Interrupt requests (TGID) enabled	R/W
4	TCIEV	Overflow Interrupt Enable 0: Interrupt requests (TCIV) disabled 1: Interrupt requests (TCIV) enabled	R/W
5	TCIEU	Underflow Interrupt Enable 0: Interrupt requests (TCIU) disabled 1: Interrupt requests (TCIU) enabled	R/W
6	TTGE2	A/D Converter Start Request Enable 2 0: A/D converter start request generation by MTUn.TCNT underflow (trough) disabled 1: A/D converter start request generation by MTUn.TCNT underflow (trough) enabled	R/W
7	TTGE	A/D Converter Start Request Enable 0: A/D converter start request generation disabled 1: A/D converter start request generation enabled	R/W

Note: n = 4 or 7

The TIER register enables or disables interrupt requests from each channel. The MTU has a total of ten TIER registers, two for MTU0 and one each for MTU1 to MTU8.

TGIEA and TGIEB bits (TGR Interrupt Enable A and TGR Interrupt Enable B)

Each bit enables or disables interrupt requests (TGIn) (n = A, B).

TGIEC and TGIED bits (TGR Interrupt Enable C and TGR Interrupt Enable D)

Each bit enables or disables interrupt requests (TGIn) (n = C, D).

In MTU1 and MTU2, these bits are reserved. They are read as 0. The write value should be 0.

TCIEV bit (Overflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIV).

TCIEU bit (Underflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIU).

In MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

TTGE2 bit (A/D Converter Start Request Enable 2)

This bit enables or disables generation of A/D converter start requests by MTUn.TCNT underflow (trough) in complementary PWM mode (n = 4 or 7).

In MTU0 to MTU3, MTU6, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

TTGE bit (A/D Converter Start Request Enable)

This bit enables or disables generation of A/D converter start requests by TGRA input capture/compare match.

MTU8 is a reserved bit. It is read as 0. The write value should be 0.

18.3.9.2 MTU0.TIER2 : Timer Interrupt Enable Register 2

Base address: MTU0 = 0x9000_1300

Offset address: 0x024

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TTGE2	—	—	—	—	—	TGIEF	TGIEE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TGIEE	TGR Interrupt Enable E 0: Interrupt requests (TGIE) disabled 1: Interrupt requests (TGIE) enabled	R/W
1	TGIEF	TGR Interrupt Enable F 0: Interrupt requests (TGIF) disabled 1: Interrupt requests (TGIF) enabled	R/W
6:2	—	The read values are undefined. The write value should be 0.	R/W
7	TTGE2	A/D Converter Start Request Enable 2 0: A/D converter start request generation by compare match between MTU0.TCNT and MTU0.TGRE disabled 1: A/D converter start request generation by compare match between MTU0.TCNT and MTU0.TGRE enabled	R/W

TGIEE bit (TGR Interrupt Enable E)

The TGIEE bit enables or disables interrupt requests by compare match between MTU0.TCNT and MTU0.TGRE.

TGIEF bit (TGR Interrupt Enable F)

The TGIEF bit enables or disables interrupt requests by compare match between MTU0.TCNT and MTU0.TGRF.

TTGE2 bit (A/D Converter Start Request Enable 2)

The TTGE2 bit enables or disables A/D converter start requests by compare match between MTU0.TCNT and MTU0.TGRE.

18.3.9.3 MTU5.TIER : Timer Interrupt Enable Register

Base address: MTU5 = 0x9000_1C00

Offset address: 0x0B2

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TGIE5W	TGR Interrupt Enable 5W 0: Interrupt requests TGIW5 disabled 1: Interrupt requests TGIW5 enabled	R/W
1	TGIE5V	TGR Interrupt Enable 5V 0: Interrupt requests TGIV5 disabled 1: Interrupt requests TGIV5 enabled	R/W
2	TGIE5U	TGR Interrupt Enable 5U 0: Interrupt requests TGIU5 disabled 1: Interrupt requests TGIU5 enabled	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

TGIE5x bits (TGR Interrupt Enable 5x)

Each bit enables or disables interrupt requests (TGIx5) (x = U, V, W).

18.3.10 TSR : Timer Status Register

Base address: MTU1 = 0x9000_1380, MTU2 = 0x9000_1400
MTU3 = 0x9000_1100, MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00

Offset address: 0x005 (MTU1, MTU2), 0x02D (MTU4, MTU7), 0x12C (MTU3, MTU6)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TGFA	Input Capture/Output Compare Flag A 0: TGRA input capture or compare match has not occurred 1: TGRA input capture or compare match has occurred	R/W*1
1	TGFB	Input Capture/Output Compare Flag B 0: TGRB input capture or compare match has not occurred 1: TGRB input capture or compare match has occurred	R/W*1
2	TGFC	Input Capture/Output Compare Flag C 0: TGRC input capture or compare match has not occurred 1: TGRC input capture or compare match has occurred	R/W*1
3	TGFD	Input Capture/Output Compare Flag D 0: TGRD input capture or compare match has not occurred 1: TGRD input capture or compare match has occurred	R/W*1
4	TCFV	Overflow flag 0: TCNT overflow has not occurred 1: TCNT overflow has occurred	R/W*1
5	TCFU	Underflow flag 0: TCNT underflow has not occurred when MTU1 and MTU2 are in phase counting mode 1: TCNT underflow has occurred when MTU1 and MTU2 are in phase counting mode	R/W*1
6	—	This bit is read as 1. The write value should be 1.	R/W
7	TCFD	Count Direction Flag 0: TCNT counts down 1: TCNT counts up	R

Note 1. Only writing 0 to this bit after reading 1 from this bit in order to clear the flag is possible. After 1 is read from this bit, if the next flag is set before 0 is written to this bit, writing 0 to this bit does not clear the flag. In this case, read 1 again from this bit, and then write 0 to this bit.

TSR indicates the states of each of the channels. The MTU has a total of six TSR registers, one each for MTU1 to MTU4, MTU6, and MTU7.

TGFA flag (Input Capture/Output Compare Flag A)

The TGFA flag indicates generation of TGRA input capture or compare match. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TGFA = 1, the TGFA flag is read, and then 0 is written to the TGFA flag.

[Setting conditions]

- TCNT = TGRA when the TGRA register functions as an output compare register.
- The value of TCNT is transferred to TGRA due to the input capture signal when the TGRA register functions as an input capture register.

TGFB flag (Input Capture/Output Compare Flag B)

The TGFB flag indicates generation of TGRB input capture or compare match. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TGFB = 1, the TGFB flag is read, and then 0 is written to the TGFB flag.

[Setting conditions]

- TCNT = TGRB when the TGRB register functions as an output compare register.
- The value of TCNT is transferred to the TGRB register due to the input capture signal when the TGRB register functions as an input capture register.

TGFC flag (Input Capture/Output Compare Flag C)

The TGFC flag indicates generation of TGRC input capture or compare match for MTU3, MTU4, MTU6, or MTU7. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TGFC = 1, the TGFC flag is read, and then 0 is written to the TGFC flag.

[Setting conditions]

- TCNT = TGRC when the TGRC register functions as an output compare register.
- The value of TCNT is transferred to the TGRC register due to the input capture signal when the TGRC register functions as an input capture register.

This bit is reserved for MTU1 and MTU2. This bit is read as 0. The write value should be 0.

TGFD flag (Input Capture/Output Compare Flag D)

The TGFD flag indicates generation of TGRD input capture or compare match for MTU3, MTU4, MTU6, or MTU7. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TGFD = 1, the TGFD flag is read, and then 0 is written to the TGFD flag.

[Setting conditions]

- TCNT = TGRD when the TGRD register functions as the output compare register.
- The value of TCNT is transferred to the TGRD register due to the input capture signal when the TGRD register functions as an input capture register.

This bit is reserved for MTU1 and MTU2. This bit is read as 0. The write value should be 0.

TCFV flag (Overflow flag)

The TCFV flag indicates generation of TCNT overflow. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TCFV = 1, the TCFV flag is read, and then 0 is written to the TCFV flag.

[Setting conditions]

- The value of TCNT overflows (changed from 0xFFFF to 0x0000).

For MTU4 or MTU7, the TCFV flag is also set to 1 when the value of TCNT of MTU4 or MTU7 underflows (changed from 0x0001 to 0x0000) in complementary PWM mode.

TCFU flag (Underflow flag)

The TCFU flag indicates generation of TCNT underflow when MTU1 and MTU2 are in phase counting mode. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TCFU = 1, TCFU is read, and then 0 is written to TCFU.

[Setting conditions]

- The value of TCNT underflows (changed from 0x0000 to 0xFFFF).

This bit is reserved for MTU3, MTU4, MTU6, and MTU7. This bit is read as 0. The write value should be 0.

TCFD flag (Count Direction Flag)

The TCFD flag indicates the direction in which TCNT is counting in MTU1 to MTU4, MTU6, and MTU7.

18.3.11 TBTM : Timer Buffer Operation Transfer Mode Register

Base address: MTU0 = 0x9000_1300

Offset address: 0x026

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TTSE	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU3 = 0x9000_1100, MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00

Offset address: 0x138 (MTU3, MTU6), 0x039 (MTU4, MTU7)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TTSA	Timing Select A 0: When compare match A occurs in each channel, data is transferred from TGRC to TGRA 1: When TCNT is cleared in each channel, data is transferred from TGRC to TGRA	R/W
1	TTSB	Timing Select B 0: When compare match B occurs in each channel, data is transferred from TGRD to TGRB 1: When TCNT is cleared in each channel, data is transferred from TGRD to TGRB	R/W
2	TTSE	Timing Select E 0: When compare match E occurs in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE 1: When MTU0.TCNT is cleared in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE	R/W
7:37:2	—	These bits are read as 0. The write value should be 0.	R/W

TBTM specifies the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU has a total of five TBTM registers, one each for MTU0, MTU3, MTU4, MTU6, and MTU7.

TTSA bit (Timing Select A)

This bit specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSA bit in the channel to 1.

TTSB bit (Timing Select B)

This bit specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSB bit in the channel to 1.

TTSE bit (Timing Select E)

This bit specifies the timing for transferring data from MTU0.TGRF to MTU0.TGRE when they are used together for buffer operation.

In MTU3, MTU4, MTU6, and MTU7, this bit is reserved. It is read as 0 and the write value should be 0. When a channel is not set to PWM mode, do not set the TTSE bit in the channel to 1.

18.3.12 TICCR : Timer Input Capture Control Register

Base address: MTU1 = 0x9000_1380

Offset address: 0x010

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	I2BE	I2AE	I1BE	I1AE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	I1AE	Input Capture Enable 0: Does not include the MTIOC1A pin in the MTU2.TGRA input capture conditions 1: Includes the MTIOC1A pin in the MTU2.TGRA input capture conditions	R/W
1	I1BE	Input Capture Enable 0: Does not include the MTIOC1B pin in the MTU2.TGRB input capture conditions 1: Includes the MTIOC1B pin in the MTU2.TGRB input capture conditions	R/W
2	I2AE	Input Capture Enable 0: Does not include the MTIOC2A pin in the MTU1.TGRA input capture conditions 1: Includes the MTIOC2A pin in the MTU1.TGRA input capture conditions	R/W
3	I2BE	Input Capture Enable 0: Does not include the MTIOC2B pin in the MTU1.TGRB input capture conditions 1: Includes the MTIOC2B pin in the MTU1.TGRB input capture conditions	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

TICCR specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded. The MTU has one TICCR for MTU1.

18.3.13 TSYCR : Timer Synchronous Clear Register

Base address: MTU6 = 0x9000_1900

Offset address: 0x150

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CE2B	Clear Enable 2B 0: Disables counter clearing by the MTU2.TGIB2 interrupt generation timing. 1: Enables counter clearing by the MTU2.TGIB2 interrupt generation timing.	R/W
1	CE2A	Clear Enable 2A 0: Disables counter clearing by the MTU2.TGIA2 interrupt generation timing.*1 1: Enables counter clearing by the MTU2.TGIA2 interrupt generation timing.*1	R/W
2	CE1B	Clear Enable 1B 0: Disables counter clearing by the MTU1.TGIB1 interrupt generation timing.*1 1: Enables counter clearing by the MTU1.TGIB1 interrupt generation timing.*1	R/W

Bit	Symbol	Function	R/W
3	CE1A	Clear Enable 1A 0: Disables counter clearing by the MTU1.TGIA1 interrupt generation timing.*1 1: Enables counter clearing by the MTU1.TGIA1 interrupt generation timing.*1	R/W
4	CE0D	Clear Enable 0D 0: Disables counter clearing by the MTU0.TGID0 interrupt generation timing.*1 1: Enables counter clearing by the MTU0.TGID0 interrupt generation timing.*1	R/W
5	CE0C	Clear Enable 0C 0: Disables counter clearing by the MTU0.TGIC0 interrupt generation timing.*1 1: Enables counter clearing by the MTU0.TGIC0 interrupt generation timing.*1	R/W
6	CE0B	Clear Enable 0B 0: Disables counter clearing by the MTU0.TGIB0 interrupt generation timing.*1 1: Enables counter clearing by the MTU0.TGIB0 interrupt generation timing.*1	R/W
7	CE0A	Clear Enable 0A 0: Disables counter clearing by the MTU0.TGIA0 interrupt generation timing.*1 1: Enables counter clearing by the MTU0.TGIA0 interrupt generation timing.*1	R/W

Note 1. This does not depend on the MTUn.TIERn.TGIEm bit setting. (n = 0, 1, 2; m = A, B, C, D)

TSYCR specifies synchronous clear conditions for MTU6.TCNT and MTU7.TCNT. The MTU has one TSYCR for MTU6.

18.3.14 TCNT — Timer Counters

18.3.14.1 MTUn.TCNT : Timer Counter (n = 0 to 4, 6, 7)

Base address: MTU0 = 0x9000_1300, MTU1 = 0x9000_1380, MTU2 = 0x9000_1400, MTU3 = 0x9000_1100
MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00

Offset address: 0x006 (MTU0, MTU1, MTU2), 0x110 (MTU3, MTU6), 0x012 (MTU4, MTU7)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15	n/a	Timer Counter	R/W

Note: The TCNT counters should be accessed in 16-bit units.

18.3.14.2 MTU5.TCNTx : Timer Counter x (x = U, V, W)

Base address: MTU5 = 0x9000_1C00

Offset address: 0x080 (TCNTU)
0x090 (TCNTV)
0x0A0 (TCNTW)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

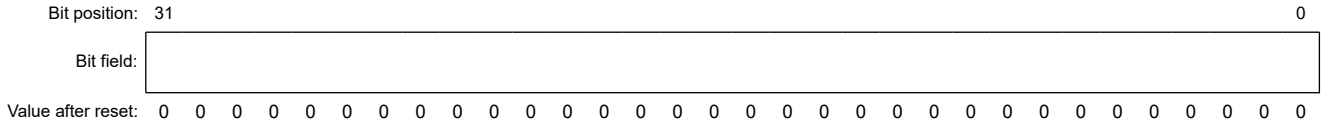
Bit	Symbol	Function	R/W
15	n/a	Timer Counter	R/W

Note: The TCNT counters should be accessed in 16-bit units.

18.3.14.3 MTU8.TCNT : Timer Counter

Base address: MTU8 = 0x9000_1600

Offset address: 0x008



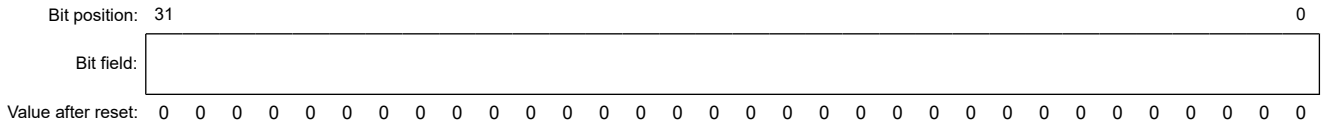
Bit	Symbol	Function	R/W
31:0	n/a	Timer Counter	R/W

Note: The TCNT counters should be accessed in 32-bit units.

18.3.15 TCNTLW : Timer Longword Counter

Base address: MTU1 = 0x9000_1380

Offset address: 0x020



Bit	Symbol	Function	R/W
31:0	n/a	The TCNTLW counter is a 32-bit readable/writable counter. Only one counter of this type is provided, and is formed by combining MTU1.TCNT and MTU2.TCNT. Such operation is only effective when TMDR3.LWA = 1. The TCNTLW counter is initialized to 0x00000000 by a reset. See section 18.3.6. TMDR3 : Timer Mode Register 3 for details. This register can only be used in 32-bit phase counting mode.	R/W

18.3.16 TGR — Timer General Registers

18.3.16.1 MTU0.TGRx : Timer General Register x (x = A, B, C, D, E, F)

Base address: MTU0 = 0x9000_1300

Offset address: 0x008 (TGRA), 0x00A (TGRB), 0x00C (TGRC), 0x00E (TGRD), 0x020 (TGRE), 0x022 (TGRF)

Bit	Symbol	Function	R/W
15:0	n/a	The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD. TGRE and TGRF function as compare registers. When the MTU0.TCNT count matches the TGRE value, an A/D converter start request can be issued. The TGRF register can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF.	R/W

Note: TGR should be accessed in 16-bit units.

18.3.16.2 MTU1/2.TGRx : Timer General Register x (x = A, B)

Base address: MTU1 = 0x9000_1380, MTU2 = 0x9000_1400

Offset address: 0x008 (TGRA), 0x00A (TGRB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	The TGRA and TGRB registers function as either output compare or input capture registers. The TGRA and TGRB registers are read as 0x0000 when TMDR3.LWA = 1. For details, see section 18.3.6. TMDR3 : Timer Mode Register 3.	R/W

Note: TGR should be accessed in 16-bit units.

18.3.16.3 MTU3/6.TGRx : Timer General Register x (x = A, B, C, D, E)

Base address: MTU3 = 0x9000_1100, MTU6 = 0x9000_1900

Offset address: 0x118 (TGRA), 0x11A (TGRB), 0x124 (TGRC), 0x126 (TGRD), 0x172 (TGRE)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers for MTU3 and MTU6 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.	R/W

Note: TGR should be accessed in 16-bit units.

18.3.16.4 MTU4/7.TGRx : Timer General Register x (x = A, B, C, D, E, F)

Base address: MTU4 = 0x9000_1200, MTU7 = 0x9000_1A00

Offset address: 0x01C (TGRA), 0x01E (TGRB), 0x028 (TGRC), 0x02A (TGRD), 0x074 (TGRE), 0x076 (TGRF)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

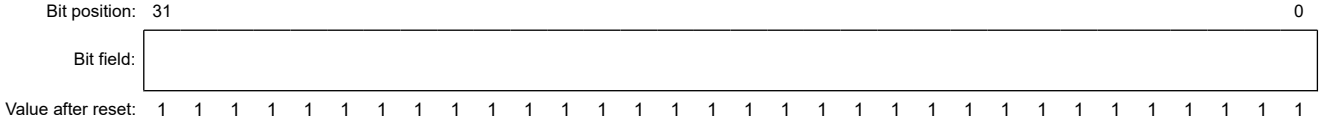
Bit	Symbol	Function	R/W
15:0	n/a	The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers for MTU4 and MTU7 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.	R/W

Note: TGR should be accessed in 16-bit units.

18.3.16.5 MTU8.TGRx : Timer General Register x (x = A, B, C, D)

Base address: MTU8 = 0x9000_1600

Offset address: 0x00C (TGRA), 0x010 (TGRB), 0x014 (TGRC), 0x018 (TGRD)



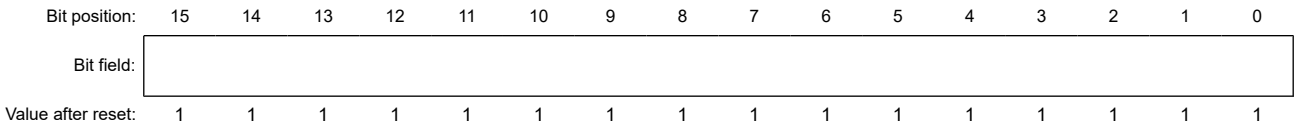
Bit	Symbol	Function	R/W
31:0	n/a	The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.	R/W

Note: TGR should be accessed in 32-bit units.

18.3.16.6 MTU5.TGRx : Timer General Register x (x = U, V, W)

Base address: MTU5 = 0x9000_1C00

Offset address: 0x082 (TGRU), 0x092 (TGRV), 0x0A2 (TGRW)



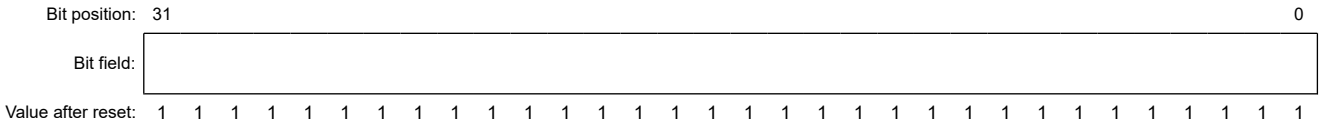
Bit	Symbol	Function	R/W
15:0	n/a	MTU5.TGRU, MTU5.TGRV, and MTU5.TGRW function as compare match, input capture, or external pulse width measurement registers.	R/W

Note: TGR should be accessed in 16-bit units.

18.3.17 TGRxLW : Timer Longword General Register x (x = A, B)

Base address: MTU1 = 0x9000_1380

Offset address: 0x024 (TGRALW)
0x028 (TGRBLW)



Bit	Symbol	Function	R/W
31:0	n/a	The TGRnLW register is a 32-bit readable/writable register. Two general registers of this type are provided, and are formed by combining MTU1.TGRn and MTU2.TGRn. Such operation is only effective when TMDR3.LWA = 1. See section 18.3.6. TMDR3 : Timer Mode Register 3 for details. The TGRnLW register functions as an output compare or input capture register when TMDR3.LWA = 1. This register can only be used in 32-bit phase counting mode.	R/W

18.3.18 TSTR — Timer Start Registers

18.3.18.1 TSTRA : Timer Start Register A

Base address: MTU = 0x9000_1000

Offset address: 0x280

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CST4	CST3	—	—	CST8	CST2	CST1	CST0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CST0	Counter Start 0 0: MTU0.TCNT counting is stopped 1: MTU0.TCNT performs count operation	R/W
1	CST1	Counter Start 1 0: MTU1.TCNT counting is stopped 1: MTU1.TCNT performs count operation	R/W
2	CST2	Counter Start 2 0: MTU2.TCNT counting is stopped 1: MTU2.TCNT performs count operation	R/W
3	CST8	Counter Start 8 0: MTU8.TCNT counting is stopped 1: MTU8.TCNT performs count operation	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	CST3	Counter Start 3 0: MTU3.TCNT counting is stopped 1: MTU3.TCNT performs count operation	R/W
7	CST4	Counter Start 4 0: MTU4.TCNT counting is stopped 1: MTU4.TCNT performs count operation	R/W

Note: When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRA is also set to 1 automatically.

CSTn bits (Counter Start n (n = 0 to 4, 8))

Each bit starts or stops TCNT in the corresponding channel.

If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops. At this time the MTIOC pin outputs the initial output level set in the TOCR1A or TOCR2A register in complementary PWM mode or reset-synchronized PWM mode, but the output compare signal level from the MTIOC pin is retained in the other modes. If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

18.3.18.2 TSTRB : Timer Start Register B

Base address: MTU = 0x9000_1000

Offset address: 0xA80

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CST7	CST6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	CST6	Counter Start 6 0: MTU6.TCNT counting is stopped 1: MTU6.TCNT performs count operation	R/W
7	CST7	Counter Start 7 0: MTU7.TCNT counting is stopped 1: MTU7.TCNT performs count operation	R/W

Note: When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRB is also set to 1 automatically.

CSTn bits (Counter Start n (n = 6, 7))

Each bit starts or stops TCNT in the corresponding channel.

If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops. At this time the MTIOC pin outputs the initial output level set in the TOCR1B or TOCR2B register in complementary PWM mode or reset-synchronized PWM mode, but the output compare signal level from the MTIOC pin is retained in the other modes. If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

18.3.18.3 MTU5.TSTR : Timer Start Register

Base address: MTU5 = 0x9000_1C00

Offset address: 0x0B4

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CSTU5	CSTV5	CSTW5
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CSTW5	Counter Start W5 0: MTU5.TCNTW counting is stopped 1: MTU5.TCNTW performs count operation	R/W
1	CSTV5	Counter Start V5 0: MTU5.TCNTV counting is stopped 1: MTU5.TCNTV performs count operation	R/W
2	CSTU5	Counter Start U5 0: MTU5.TCNTU counting is stopped 1: MTU5.TCNTU performs count operation	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

18.3.19 TSYRA : Timer Synchronous Register A

Base address: MTU = 0x9000_1000

Offset address: 0x281

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SYNC0	Timer Synchronous Operation 0 0: MTU0.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU0.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W

Bit	Symbol	Function	R/W
1	SYNC1	Timer Synchronous Operation 1 0: MTU1.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU1.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W
2	SYNC2	Timer Synchronous Operation 2 0: MTU2.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU2.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	SYNC3	Timer Synchronous Operation 3 0: MTU3.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W
7	SYNC4	Timer Synchronous Operation 4 0: MTU4.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W

TSYRA selects independent operation or synchronous operation of TCNT in MTU0 to MTU4. A channel performs synchronous operation when the corresponding bit in TSYRA is set to 1.

SYNCn bits (Timer Synchronous Operation n)

Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous presetting of multiple channels and synchronous clearing by counter clearing on another channel are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of the TCR.CCLR[2:0] bits.

18.3.20 TSYRB : Timer Synchronous Register B

Base address: MTU = 0x9000_1000

Offset address: 0xA81

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SYNC 7	SYNC 6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	SYNC6	Timer Synchronous Operation 6 0: MTU6.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU6.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W
7	SYNC7	Timer Synchronous Operation 7 0: MTU7.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU7.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W

TSYRB selects independent operation or synchronous operation of TCNT in MTU6 and MTU7. A channel performs synchronous operation when the corresponding bit in TSYRB is set to 1.

SYNCn bits (Timer Synchronous Operation n)

Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous presetting of multiple channels and synchronous clearing by counter clearing on another channel are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of the TCR.CCLR[2:0] bits.

18.3.21 TCSYSTR : Timer Counter Synchronous Start Register

Base address: MTU = 0x9000_1000

Offset address: 0x282

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH6	SCH7
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SCH7	Synchronous Start 7 0: Does not specify synchronous start for MTU7.TCNT 1: Specifies synchronous start for MTU7.TCNT	R/W*1
1	SCH6	Synchronous Start 6 0: Does not specify synchronous start for MTU6.TCNT 1: Specifies synchronous start for MTU6.TCNT	R/W*1
2	—	This bit is read as 0. The write value should be 0.	R/W
3	SCH4	Synchronous Start 4 0: Does not specify synchronous start for MTU4.TCNT 1: Specifies synchronous start for MTU4.TCNT	R/W*1
4	SCH3	Synchronous Start 3 0: Does not specify synchronous start for MTU3.TCNT 1: Specifies synchronous start for MTU3.TCNT	R/W*1
5	SCH2	Synchronous Start 2 0: Does not specify synchronous start for MTU2.TCNT 1: Specifies synchronous start for MTU2.TCNT	R/W*1
6	SCH1	Synchronous Start 1 0: Does not specify synchronous start for MTU1.TCNT 1: Specifies synchronous start for MTU1.TCNT	R/W*1
7	SCH0	Synchronous Start 0 0: Does not specify synchronous start for MTU0.TCNT 1: Specifies synchronous start for MTU0.TCNT	R/W*1

Note 1. Only 1 can be written to this bit, and doing so sets the register. TCSYSTR is automatically cleared to 0 after 1 is written to.

TCSYSTR specifies synchronous start of the counters.

SCH7 bit (Synchronous Start 7)

This bit controls synchronous start of MTU7.TCNT.

[Clearing condition]

- When 1 is set to the TSTRB.CST7 bit while SCH7 = 1

SCH6 bit (Synchronous Start 6)

This bit controls synchronous start of MTU6.TCNT.

[Clearing condition]

- When 1 is set to the TSTRB.CST6 bit while SCH6 = 1

SCH4 bit (Synchronous Start 4)

This bit controls synchronous start of MTU4.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST4 bit while SCH4 = 1

SCH3 bit (Synchronous Start 3)

This bit controls synchronous start of MTU3.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST3 bit while SCH3 = 1

SCH2 bit (Synchronous Start 2)

This bit controls synchronous start of MTU2.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST2 bit while SCH2 = 1

SCH1 bit (Synchronous Start 1)

This bit controls synchronous start of MTU1.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST1 bit while SCH1 = 1

SCH0 bit (Synchronous Start 0)

This bit controls synchronous start of MTU0.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST0 bit while SCH0 = 1

18.3.22 TRWERx : Timer Read/Write Enable Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x284 (TRWERA)
0xA84 (TRWERB)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RWE
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	RWE	Read/Write Enable 0: Read/write access to the registers is disabled 1: Read/write access to the registers is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

TRWERA enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.

TRWERB enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU6 and MTU7.

RWE bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.

[Clearing condition]

- When 0 is written to the RWE bit after reading RWE = 1

Note:
- **Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERA)** 24 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTU.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA, and MTUn.TCNT (n = 3, 4)
- **Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERB)** 23 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTUn.MTU.TOERB, MTU.TOCR1B, MTU.TOCR2B, MTU.TCDRB, MTU.TDDRB, and MTUn.TCNT (n = 6, 7)

18.3.23 TOERA : Timer Output Master Enable Register A

Base address: MTU = 0x9000_1000

Offset address: 0x20A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OE3B	Master Enable MTIOC3B 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
1	OE4A	Master Enable MTIOC4A 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
2	OE4B	Master Enable MTIOC4B 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
3	OE3D	Master Enable MTIOC3D 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
4	OE4C	Master Enable MTIOC4C 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
5	OE4D	Master Enable MTIOC4D 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
7:6	—	These bits are read as 1. The write value should be 1.	R/W

Note 1. The non-active level depends on the setting of the timer output control register 1 or 2 (TOCR 1/2). See section 18.3.25. TOCR1x : Timer Output Control Register 1x (x = A, B) and section 18.3.26. TOCR2x : Timer Output Control Register 2x (x = A, B) for details. When MTU output is in a mode other than complementary PWM mode or reset-synchronized PWM mode, set the relevant bit to 1. If the setting is 0, the non-active level will be output due to this setting of the timer output control register.

TOERA enables or disables output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output correctly if the bits in the TOERA register have not been set. In MTU3 and MTU4, set TOERA prior to setting TIOR.

Set MTU.TOERA after setting the CST3 and CST4 bits in MTU.TSTRA to 0 (see Figure 18.44 and Figure 18.48).

18.3.24 TOERB : Timer Output Master Enable Register B

Base address: MTU = 0x9000_1000

Offset address: 0xA0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	OE7D	OE7C	OE6D	OE7B	OE7A	OE6B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OE6B	Master Enable MTIOC6B 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
1	OE7A	Master Enable MTIOC7A 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
2	OE7B	Master Enable MTIOC7B 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
3	OE6D	Master Enable MTIOC6D 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
4	OE7C	Master Enable MTIOC7C 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
5	OE7D	Master Enable MTIOC7D 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
7:6	—	These bits are read as 1. The write value should be 1.	R/W

Note 1. The non-active level depends on the setting of the timer output control register 1 or 2 (TOCR 1/2). See section 18.3.25. TOCR1x : Timer Output Control Register 1x (x = A, B) and section 18.3.26. TOCR2x : Timer Output Control Register 2x (x = A, B) for details. When MTU output is in a mode other than complementary PWM mode or reset-synchronized PWM mode, set the relevant bit to 1. If the setting is 0, the non-active level will be output due to this setting of the timer output control register.

TOERB enables or disables output settings for output pins MTIOC7D, MTIOC7C, MTIOC6D, MTIOC7B, MTIOC7A, and MTIOC6B.

These pins do not output correctly if the bits in the TOERB register have not been set. In MTU6 and MTU7, set TOERB prior to setting TIOR.

Set MTU.TOERB after setting the CST6 and CST7 bits in MTU.TSTRB to 0 (see Figure 18.44 and Figure 18.48).

18.3.25 TOCR1x : Timer Output Control Register 1x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x20E (TOCR1A)
0xA0E (TOCR1B)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OLSP	Output Level Select P*1 *3 See Table 18.46.	R/W

Bit	Symbol	Function	R/W
1	OLSN	Output Level Select N*1 *3 See Table 18.47.	R/W
2	TOCS	TOC Select 0: TOCR1j setting is selected (j = A, B). 1: TOCR2j setting is selected (j = A, B).	R/W
3	TOCL	TOC Register Write Protection*2 *4 0: Write access to the TOCS, OLSN, and OLSP bits is enabled. 1: Write access to the TOCS, OLSN, and OLSP bits is disabled.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	PSYE	PWM Synchronous Output Enable 0: Toggle output is disabled. 1: Toggle output is enabled.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. Setting the TOCR1j.TOCS bit to 0 makes this bit setting valid.
Note 2. Setting the TOCR1j.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
Note 3. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSP bit is valid.
Note 4. This bit can be set to 1 only once after a reset. After 1 is written, 0 cannot be written to the bit.

TOCR1A and TOCR1B enable or disable PWM-synchronized toggle output in complementary PWM mode and reset synchronized PWM mode, and control inversion of PWM output level.

OLSP bit (Output Level Select P)

This bit selects the positive-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

OLSN bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

TOCS bit (TOC Select)

This bit selects either the TOCR1j or TOCR2j (j = A, B) setting to be used for the output level in complementary PWM mode and reset-synchronized PWM mode.

TOCL bit (TOC Register Write Protection)

This bit enables or disables write access to the TOCS, OLSN, and OLSP bits in TOCR1j (j = A, B).

PSYE bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM period.

Table 18.46 Output level select function

Bit 0	Function			
			Compare match output	
OLSP	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 18.47 Output level select function

Bit 1	Function			
			Compare match output	
OLSN	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Figure 18.3 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

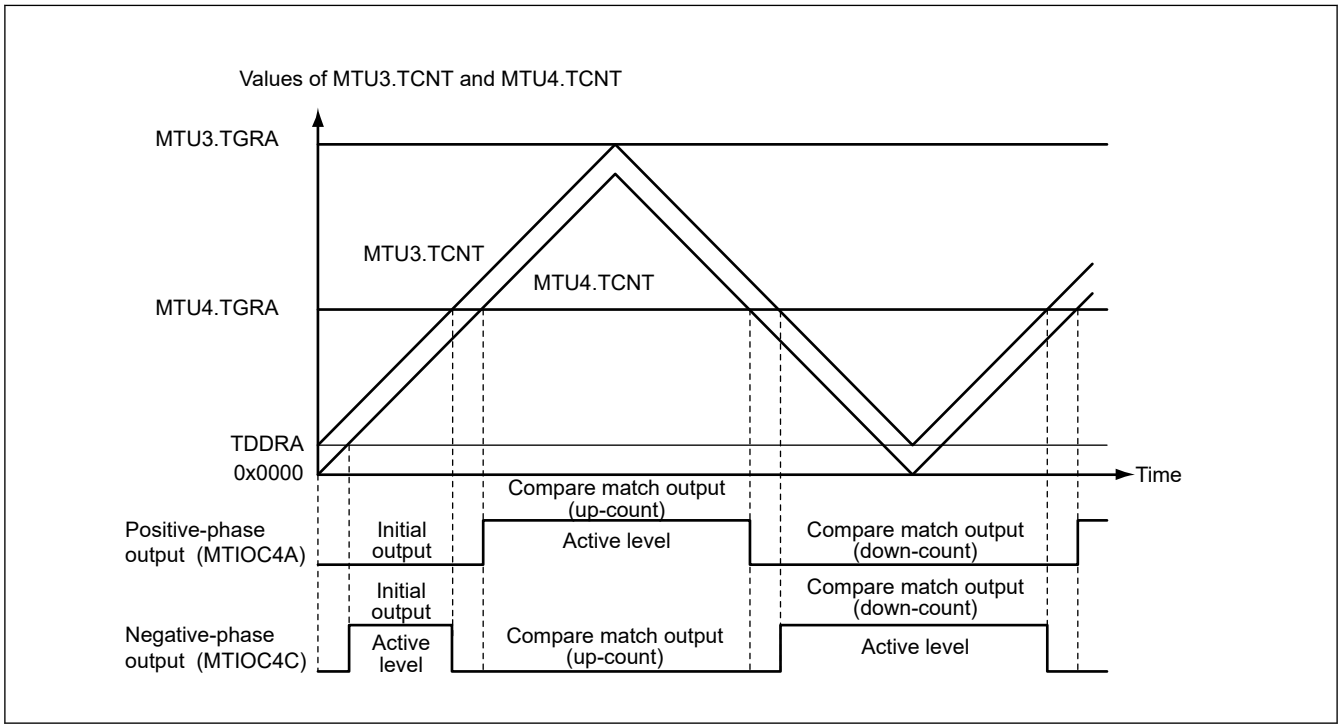


Figure 18.3 Example of output in complementary PWM mode

18.3.26 TOCR2x : Timer Output Control Register 2x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x20F (TOCR2A)
0xA0F (TOCR2B)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OLS1P	Output Level Select 1P*1 *2 This bit selects the output level on MTIOC3B or MTIOC6B in reset-synchronized PWM mode and complementary PWM mode. See Table 18.48.	R/W
1	OLS1N	Output Level Select 1N*1 *2 This bit selects the output level on MTIOC3D or MTIOC6D in reset-synchronized PWM mode and complementary PWM mode. See Table 18.49.	R/W
2	OLS2P	Output Level Select 2P*1 *2 This bit selects the output level on MTIOC4A or MTIOC7A in reset-synchronized PWM mode and complementary PWM mode. See Table 18.50.	R/W
3	OLS2N	Output Level Select 2N*1 *2 This bit selects the output level on MTIOC4C or MTIOC7C in reset-synchronized PWM mode and complementary PWM mode. See Table 18.51.	R/W

Bit	Symbol	Function	R/W
4	OLS3P	Output Level Select 3P*1 *2 This bit selects the output level on MTIOC4B or MTIOC7B in reset-synchronized PWM mode and complementary PWM mode. See Table 18.52.	R/W
5	OLS3N	Output Level Select 3N*1 *2 This bit selects the output level on MTIOC4D or MTIOC7D in reset-synchronized PWM mode and complementary PWM mode. See Table 18.53.	R/W
7:6	BF[1:0]	TOLBR Buffer Transfer Timing Select These bits select the timing for transferring data from TOLBRj to TOCR2j. See Table 18.54 for details.	R/W

Note 1. Setting the TOCR1j.TOCS bit to 1 makes this bit setting valid (j = A, B).
Note 2. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSiP bits are valid (i = 1 to 3).

TOCR2A and TOCR2B control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

The initial output is selected while the counter is stopped.

Table 18.48 MTIOCmB output level select function

Bit 0	Function			
			Compare match output	
OLS1P	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Note: m = 3, 6

Table 18.49 MTIOCmD output level select function

Bit 1	Function			
			Compare match output	
OLS1N	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: m = 3, 6
Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 18.50 MTIOCmA output level select function

Bit 2	Function			
			Compare match output	
OLS2P	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Note: m = 4, 7

Table 18.51 MTIOCmC output level select function

Bit 3	Function			
			Compare match output	
OLS2N	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: m = 4, 7

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 18.52 MTIOCmB output level select function

Bit 4	Function			
			Compare match output	
OLS3P	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Note: m = 4, 7

Table 18.53 MTIOCmD output level select function

Bit 5	Function			
			Compare match output	
OLS3N	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: m = 4, 7
Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 18.54 Setting of TOCR2j.BF[1:0] bits

Bit 7	Bit 6	Description	
BF[1]	BF[0]	Complementary PWM mode	Reset-synchronized PWM mode
0	0	Does not transfer data from the buffer register (TOLBRj) to TOCR2j.	Does not transfer data from the buffer register (TOLBRj) to TOCR2j.
0	1	Transfers data from the buffer register (TOLBRj) to TOCR2j at the crest of the MTUn.TCNT count.	Transfers data from the buffer register (TOLBRj) to TOCR2j when MTUm.TCNT or MTUn.TCNT is cleared.
1	0	Transfers data from the buffer register (TOLBRj) to TOCR2j at the trough of the MTUn.TCNT count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBRj) to TOCR2j at the crest and trough of the MTUn.TCNT count.	Setting prohibited

Note: n = 4, 7; m = 3, 6; j = A, B

18.3.27 TOLBRx : Timer Output Level Buffer Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x236 (TOLBRA)
0xA36 (TOLBRB)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OLS1P	Output Level Select 1P Specify the buffer value to be transferred to the OLS1P bit in TOCR2A and TOCR2B.	R/W
1	OLS1N	Output Level Select 1N Specify the buffer value to be transferred to the OLS1N bit in TOCR2A and TOCR2B.	R/W
2	OLS2P	Output Level Select 2P Specify the buffer value to be transferred to the OLS2P bit in TOCR2A and TOCR2B.	R/W

Bit	Symbol	Function	R/W
3	OLS2N	Output Level Select 2N Specify the buffer value to be transferred to the OLS2N bit in TOCR2A and TOCR2B.	R/W
4	OLS3P	Output Level Select 3P Specify the buffer value to be transferred to the OLS3P bit in TOCR2A and TOCR2B.	R/W
5	OLS3N	Output Level Select 3N Specify the buffer value to be transferred to the OLS3N bit in TOCR2A and TOCR2B.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

TOLBRA and TOLBRB are buffer registers for TOCR2A and TOCR2B and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 18.4 shows an example of the PWM output level setting procedure in buffer operation.

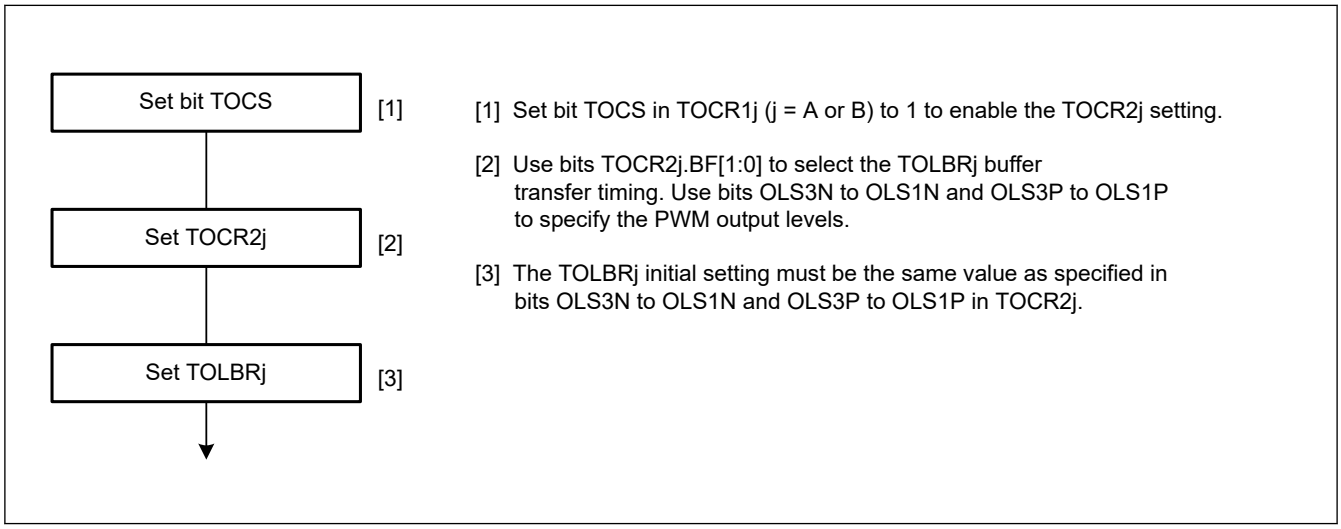


Figure 18.4 Example of PWM output level setting procedure in buffer operation

18.3.28 TGCRA : Timer Gate Control Register A

Base address: MTU = 0x9000_1000

Offset address: 0x20D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	BDC	N	P	FB	WF	VF	UF
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UF	Output Phase Switch These bits turn on or off the positive-phase/negative-phase output. The setting of these bits is valid only when FB = 1. In this case, the setting of bit 0 to bit 2 is used instead of the external input. See Table 18.55.	R/W
1	VF	Output Phase Switch See Table 18.55.	R/W
2	WF	Output Phase Switch See Table 18.55.	R/W
3	FB	External Feedback Signal Enable 0: Output is switched by external input (input sources are TGRA, TGRB, and TGRC input capture signals in MTU0) 1: Output is switched by software (UF, VF, and WF settings)	R/W

Bit	Symbol	Function	R/W
4	P	Positive-Phase Output (P) Control 0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
5	N	Negative-Phase Output (N) Control 0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
6	BDC	Brushless DC Motor 0: Ordinary output 1: Functions of this register are made effective	R/W
7	—	This bit is read as 1. The write value should be 1.	R/W

TGCRA controls the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. TGCRA register settings are ineffective for anything other than complementary PWM mode and reset-synchronized PWM mode.

UF, VF, and WF bits (Output Phase Switch)

The setting of these bits is valid only when FB = 1. In this case, the setting of bit 0 to bit 2 is used instead of the external input. For details, see Table 18.55.

Table 18.55 Output level select function

Bit 2	Bit 1	Bit 0	Function					
			MTIOC3B	MTIOC4A	MTIOC4B	MTIOC3D	MTIOC4C	MTIOC4D
WF	VF	UF	U phase	V phase	W phase	U phase	V phase	W phase
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
0	0	1	ON	OFF	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	ON	OFF	OFF
0	1	1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
1	0	1	ON	OFF	OFF	OFF	ON	OFF
1	1	0	OFF	OFF	ON	ON	OFF	OFF
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF

FB bit (External Feedback Signal Enable)

This bit selects whether the positive-/negative-phase output is switched automatically with the TGRA, TGRB, and TGRC input capture signals in MTU0 or by writing 0 or 1 to bits 2 to 0 in TGCRA. When the TGCRA.FB bit is 0, output of MTU3 and MTU4 can be switched with the TGRA, TGRB, and TGRC input capture signals in MTU0.

P bit (Positive-Phase Output (P) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the positive-phase output pins (MTIOC3B, MTIOC4A, and MTIOC4B pins).

N bit (Negative-Phase Output (N) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the negative-phase output pins (MTIOC3D, MTIOC4C, and MTIOC4D pins).

BDC bit (Brushless DC Motor)

This bit selects whether to make the functions of TGCRA effective or ineffective.

18.3.29 TCNTSx : Timer Subcounter x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x220 (TCNTSA)
0xA20 (TCNTSB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	TCNTSA and TCNTSB are 16-bit read-only counters used only in complementary PWM mode.	R

Note: TCNTSA and TCNTSB should be accessed in 16-bit units.

18.3.30 TCDRx : Timer Cycle Data Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x214 (TCDRA)
0xA14 (TCDRB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	TCDRA and TCDRB are 16-bit readable/writable registers used only in complementary PWM mode. Set half the PWM carrier period as the TCDRA and TCDRB values. The TCDRA and TCDRB registers are constantly compared with the TCNTSA and TCNTSB counters in complementary PWM mode, respectively. When a match occurs, the TCNTSA and TCNTSB counters switch the count direction (down-count to up-count).	R/W

Note: TCDRA and TCDRB should be accessed in 16-bit units.

18.3.31 TCBRx : Timer Cycle Buffer Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x222 (TCBRA)
0xA22 (TCBRB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	TCBRA and TCBRB are 16-bit readable/writable registers, used only in complementary PWM mode, that function as buffer registers for TCDRA and TCDRB. The TCBRA and TCBRB values are transferred to TCDRA and TCDRB with the transfer timing set in TMDR1.	R/W

Note: TCBRA and TCBRB should be accessed in 16-bit units.

18.3.32 TDDRx : Timer Dead Time Data Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x216 (TDDRA)
0xA16 (TDDRB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	TDDRA and TDDRB are 16-bit readable/writable registers, used only in complementary PWM mode, that specify the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counter offset value. In complementary PWM mode, when the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counters are cleared and then restarted, the TDDRA (TDDRB) value is loaded into the MTU3.TCNT (MTU6.TCNT) counter and the count operation starts.	R/W

Note: TDDRA and TDDRB should be accessed in 16-bit units.

18.3.33 TDERx : Timer Dead Time Enable Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x234 (TDERA)
0xA34 (TDERB)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TDER
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TDER	Dead Time Enable 0: No dead time is generated 1: Dead time is generated*1	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. TDDRA and TDDRB should be set to 1 or a larger value.

TDERA and TDERB control dead time generation in complementary PWM mode. The MTU has one TDER each for MTU3 and MTU6. TDERA and TDERB should be modified only while TCNT stops.

TDER bit (Dead Time Enable)

This bit specifies whether to generate dead time.

[Clearing condition]

- When 0 is written to TDER after reading TDER = 1

18.3.34 TBTERx : Timer Buffer Transfer Set Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x232 (TBTERA)
0xA32 (TBTERB)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	BTE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	BTE[1:0]	Buffer Transfer Disable and Interrupt Skipping Link Setting These bits enable or disable transfer from the buffer registers*1 used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping function 1. For details, see Table 18.56.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Applicable buffer registers (TBTERA):
MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and TCBRA
Applicable buffer registers (TBTERB):
MTU6.TGRC, MTU6.TGRD, MTU7.TGRC, MTU7.TGRD, and TCBRB

TBTERA and TBTERB enable or disable transfer from the buffer registers used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping 1 operation.

Table 18.56 Setting of TBTERA.BTE[1:0] bits and TBTERB.BTE[1:0] bits

Bit 1	Bit 0	
BTE[1]	BTE[0]	Description
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping function 1.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping function 1.*2
1	1	Setting prohibited

Note 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR1. For details, see section 18.4.8. Complementary PWM Mode.
Note 2. When interrupt skipping is disabled the T3AEN and T4VEN (T6AEN and T7VEN) bits are set to 0 in the timer interrupt skipping set register (TITCR1A (TITCR1B)) or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), be sure to disable link of buffer transfer with interrupt skipping (set the BTE1 bit in the timer buffer transfer set register (TBTERA (TBTERB)) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

18.3.35 TWCRx : Timer Waveform Control Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x260 (TWCRA)
0xA60 (TWCRB)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CCE	—	—	—	—	—	SCC	WRE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WRE*3	Waveform Retain Enable 0: Initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are output 1: Initial output is inhibited	R/W
1	SCC*1*3	Synchronous Clearing Control (Only valid in TWCRB) 0: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2 to MTU6, and MTU7 is enabled. 1: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2 to MTU6, and MTU7 is disabled.	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	CCE*2	Compare Match Clear Enable 0: Counters are not cleared at MTU3.TGRA (MTU6.TGRA) compare match 1: Counters are cleared at MTU3.TGRA (MTU6.TGRA) compare match	R/W

Note 1. This bit is only valid in register TWCRB and is a reserved bit in register TWCRA.
Note 2. Do not set to 1 when complementary PWM mode 1 is not selected.

Note 3. Do not set to 1 when complementary PWM mode is not selected.

TWCRA and TWCRB control the output waveform when synchronous counter clearing occurs in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA (MTU6.TGRA) compare match.

The CCE bit and WRE bit in TWCRA and TWCRB should be modified only while TCNT stops.

WRE bit (Waveform Retain Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.

The initial output is inhibited with this function only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial values specified in TOCR1A and TOCR2A are output regardless of the WRE bit setting. The initial values specified in TOCR1A and TOCR2A are also output when synchronous clearing occurs in the Tb interval at the trough immediately after MTU3.TCNT and MTU4.TCNT start operation.

For the Tb interval at the trough in complementary PWM mode, see Figure 18.50.

[Setting condition]

- When 1 is written to the WRE bit after reading WRE = 0

SCC bit (Synchronous Clearing Control)

The setting of this bit selects whether MTU6.TCNT and MTU7.TCNT are or are not cleared when counter-synchronous clearing is generated for MTU0, MTU1, MTU2 to MTU6, and MTU7 in complementary PWM mode.

Make the complementary PWM mode settings for MTU6 and MTU7 when this function is in use. When writing a new value to the SCC bit while the counter is operating, do so in such a way that the values of the CCE and WRE bits are not changed.

Synchronous clearing from the MTU module only becomes disabled due to the setting of the SCC bit when synchronous clearing is generated outside the Tb interval in the trough. If synchronous clearing is generated within the Tb interval in the trough including immediately after the value at which MTU6.TCNT and MTU7.TCNT start, MTU6.TCNT and MTU7.TCNT are cleared.

Regarding the Tb interval in the trough in complementary PWM mode, see Figure 18.50.

[Setting condition]

- Writing of 1 to the SCC bit after reading SCC = 0

The corresponding bit in register TWCRA is reserved and is read as 0. When writing to TWCRA, write 0 to this bit.

CCE bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at MTU3.TGRA (MTU6.TGRA) compare match in complementary PWM mode.

[Setting condition]

- When 1 is written to the CCE bit after reading CCE = 0

18.3.36 NFCR — Noise Filter Control Registers

18.3.36.1 NFCRn : Noise Filter Control Register n (n = 0, 3, 4, 6 to 8)

Base address: MTU_NF = 0x9000_1290

Offset address: 0x000 + 0x01 × n (n = 0 to 4, 8)
0x803 + 0x01 × (n - 6) (n = 6, 7)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NFAEN	Noise Filter A Enable 0: The noise filter for the MTIOCnA pin is disabled 1: The noise filter for the MTIOCnA pin is enabled.	R/W
1	NFBEN	Noise Filter B Enable 0: The noise filter for the MTIOCnB pin is disabled 1: The noise filter for the MTIOCnB pin is enabled	R/W
2	NFCEN	Noise Filter C Enable 0: The noise filter for the MTIOCnC pin is disabled 1: The noise filter for the MTIOCnC pin is enabled	R/W
3	NFDEN	Noise Filter D Enable 0: The noise filter for the MTIOCnD pin is disabled 1: The noise filter for the MTIOCnD pin is enabled	R/W
5:4	NFCS[1:0]	Noise Filter Clock Select 0 0: PCLKH/1 0 1: PCLKH/8 1 0: PCLKH/32 1 1: Clock source for counting	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The NFCRn register controls noise filtering for the input pins of MTUn.

NFAEN bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCnA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFBEN bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCnB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCEN bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTIOCnC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFDEN bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTIOCnD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCS[1:0] bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function. When the NFCS[1:0] bits are set to 11b, i.e. selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input capture function.

18.3.36.2 NFCRm : Noise Filter Control Register m (m = 1, 2)

Base address: MTU_NF = 0x9000_1290

Offset address: 0x001 (NFCR1), 0x002 (NFCR2)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	NFCS[1:0]		—	—	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NFAEN	Noise Filter A Enable 0: The noise filter for the MTIOCmA pin is disabled 1: The noise filter for the MTIOCmA pin is enabled.	R/W
1	NFBEN	Noise Filter B Enable 0: The noise filter for the MTIOCmB pin is disabled 1: The noise filter for the MTIOCmB pin is enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	NFCS[1:0]	Noise Filter Clock Select 0 0: PCLKH/1 0 1: PCLKH/8 1 0: PCLKH/32 1 1: Clock source for counting	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The NFCRm register (m = 1, 2) controls noise filtering for the input pins of MTUm.

NFAEN bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCmA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFBEN bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCmB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCS[1:0] bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function. When the NFCS[1:0] bits are set to 11b, i.e. selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input capture function.

18.3.36.3 NFCRC : Noise Filter Control Register C

Base address: MTU_NF = 0x9000_1290

Offset address: 0x009

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NFAEN	Noise Filter A Enable 0: The noise filter for the MTCLKA pin is disabled 1: The noise filter for the MTCLKA pin is enabled.	R/W
1	NFBEN	Noise Filter B Enable 0: The noise filter for the MTCLKB pin is disabled 1: The noise filter for the MTCLKB pin is enabled	R/W
2	NFCEN	Noise Filter C Enable 0: The noise filter for the MTCLKC pin is disabled 1: The noise filter for the MTCLKC pin is enabled	R/W
3	NFDEN	Noise Filter D Enable 0: The noise filter for the MTCLKD pin is disabled 1: The noise filter for the MTCLKD pin is enabled	R/W
5:4	NFCS[1:0]	Noise Filter Clock Select 0 0: PCLKH/1 0 1: PCLKH/2 1 0: PCLKH/8 1 1: PCLKH/32	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The NFCRC register controls noise filtering for the external clock input pins of the MTU.

NFAEN bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTCLKA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFBEN bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTCLKB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCEN bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTCLKC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFDEN bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTCLKD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCS[1:0] bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. After setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval to set the counting start.

18.3.36.4 NFCR5 : Noise Filter Control Register 5

Base address: MTU_NF = 0x9000_1290

Offset address: 0x805

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	NFCS[1:0]		—	NFWEN	NFVEN	NFUEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NFUEN	Noise Filter U Enable 0: The noise filter for the MTIC5U pin is disabled 1: The noise filter for the MTIC5U pin is enabled	R/W

Bit	Symbol	Function	R/W
1	NFVEN	Noise Filter V Enable 0: The noise filter for the MTIC5V pin is disabled 1: The noise filter for the MTIC5V pin is enabled	R/W
2	NFWEN	Noise Filter W Enable 0: The noise filter for the MTIC5W pin is disabled 1: The noise filter for the MTIC5W pin is enabled	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	NFCS[1:0]	Noise Filter Clock Select 0 0: PCLKH/1 0 1: PCLKH/8 1 0: PCLKH/32 1 1: Clock source for counting	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The NFCR5 register controls noise filtering for the input pins of MTU5.

NFUEN bit (Noise Filter U Enable)

This bit disables or enables the noise filter for input from the MTIC5U pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFVEN bit (Noise Filter V Enable)

This bit disables or enables the noise filter for input from the MTIC5V pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFWEN bit (Noise Filter W Enable)

This bit disables or enables the noise filter for input from the MTIC5W pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFCS[1:0] bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function.

18.3.37 TADCR — Timer A/D Converter Start Request Control Registers

18.3.37.1 MTU4.TADCR : Timer A/D Converter Start Request Control Register

Base address: MTU4 = 0x9000_1200

Offset address: 0x040

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BF[1:0]		—	—	—	—	—	—	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ITB4VE	TCIV4 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG4BN and TCIV4 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG4BN and TCIV4 interrupt skipping 1 are linked.	R/W

Bit	Symbol	Function	R/W
1	ITB3AE	TGIA3 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG4BN and TGIA3 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG4BN and TGIA3 interrupt skipping 1 are linked.	R/W
2	ITA4VE	TCIV4 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG4AN and TCIV4 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG4AN and TCIV4 interrupt skipping 1 are linked.	R/W
3	ITA3AE	TGIA3 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG4AN and TGIA3 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG4AN and TGIA3 interrupt skipping 1 are linked.	R/W
4	DT4BE	Down-Count TRG4BN Enable*3 0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT down-count operation. 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT down-count operation.	R/W
5	UT4BE	Up-Count TRG4BN Enable 0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT up-count operation. 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT up-count operation.	R/W
6	DT4AE	Down-Count TRG4AN Enable*3 0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT down-count operation. 1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT down-count operation.	R/W
7	UT4AE	Up-Count TRG4AN Enable 0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT up-count operation. 1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT up-count operation.	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W
15:14	BF[1:0]	MTU4.TADCOBRA/TADCOBRB Transfer Timing Select These bits specify the transfer timing from MTU4.TADCOBRA and MTU4.TADCOBRB to MTU4.TADCORA and MTU4.TADCORB. For detail, see Table 18.57.	R/W

Note: MTU4.TADCR should be accessed in 16-bit units.
Note 1. When interrupt skipping is disabled (the T3AEN and T4VEN bits in TITCR1A are cleared to 0 or the T3ACOR and T4VCOR bits in TITCR1A are cleared to 0), do not link A/D converter start requests with interrupt skipping function 1 (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in MTU4.TADCR to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
Note 3. Do not set to 1 when complementary PWM mode is not selected.

TADCR enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping function. The MTU has one TADCR each for MTU4 and MTU7.

Table 18.57 Setting of transfer timing by TADCR.BF[1:0] bits (MTU4) (1 of 2)

BF[1]	BF[0]	Description
0	0	Does not transfer data from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when the crest of the MTU4.TCNT count is reached or the MTU4.TGRD is written to in complementary PWM mode, when a compare match occurs between MTU3.TCNT and MTU3.TGRA in reset-synchronized PWM mode, or when a compare match occurs between MTU4.TCNT and MTU4.TGRA in PWM mode 1 or normal operation mode.

Table 18.57 Setting of transfer timing by TADCR.BF[1:0] bits (MTU4) (2 of 2)

BF[1]	BF[0]	Description
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the MTU4.TCNT count. These settings are prohibited when complementary PWM mode is not selected.
1	1	Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the MTU4.TCNT count. These settings are prohibited when complementary PWM mode is not selected.

18.3.37.2 MTU7.TADCR : Timer A/D Converter Start Request Control Register

Base address: MTU7 = 0x9000_1A00

Offset address: 0x040

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BF[1:0]		—	—	—	—	—	—	UT7AE	DT7AE	UT7BE	DT7BE	ITA6AE	ITA7VE	ITB6AE	ITB7VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ITB7VE	TCIV7 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG7BN and TCIV7 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG7BN and TCIV7 interrupt skipping 1 are linked.	R/W
1	ITB6AE	TGIA6 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG7BN and TGIA6 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG7BN and TGIA6 interrupt skipping 1 are linked.	R/W
2	ITA7VE	TCIV7 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG7AN and TCIV7 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG7AN and TCIV7 interrupt skipping 1 are linked.	R/W
3	ITA6AE	TGIA6 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG7AN and TGIA6 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG7AN and TGIA6 interrupt skipping 1 are linked.	R/W
4	DT7BE	Down-Count TRG7BN Enable*3 0: A/D converter start requests (TRG7BN) disabled during MTU7.TCNT down-count operation. 1: A/D converter start requests (TRG7BN) enabled during MTU7.TCNT down-count operation.	R/W
5	UT7BE	Up-Count TRG7BN Enable 0: A/D converter start requests (TRG7BN) disabled during MTU7.TCNT up-count operation. 1: A/D converter start requests (TRG7BN) enabled during MTU7.TCNT up-count operation.	R/W
6	DT7AE	Down-Count TRG7AN Enable*3 0: A/D converter start requests (TRG7AN) disabled during MTU7.TCNT down-count operation. 1: A/D converter start requests (TRG7AN) enabled during MTU7.TCNT down-count operation.	R/W

Bit	Symbol	Function	R/W
7	UT7AE	Up-Count TRG7AN Enable 0: A/D converter start requests (TRG7AN) disabled during MTU7.TCNT up-count operation. 1: A/D converter start requests (TRG7AN) enabled during MTU7.TCNT up-count operation.	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W
15:14	BF[1:0]	MTU7.TADCOBRA/TADCOBRB Transfer Timing Select These bits specify the transfer timing from MTU7.TADCOBRA and MTU7.TADCOBRB to MTU7.TADCORA and MTU7.TADCORB. For detail, see Table 18.58.	R/W

Note: MTU7.TADCR should be accessed in 16-bit units.
Note 1. When interrupt skipping is disabled (the T6AEN and T7VEN bits in TITCR1B are cleared to 0 or the T6ACOR and T7VCOR bits in TITCR1B are cleared to 0), do not link A/D converter start requests with interrupt skipping function 1 (clear the ITA6AE, ITA7VE, ITB6AE, and ITB7VE bits in MTU7.TADCR to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests are not issued.
Note 3. Do not set to 1 when complementary PWM mode is not selected.

TADCR enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping function.

Table 18.58 Setting of transfer timing by TADCR.BF[1:0] bits (MTU7)

BF[1]	BF[0]	Description
0	0	Does not transfer data from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) when the crest of the MTU7.TCNT count is reached or the MTU7.TGRD is written to in complementary PWM mode, when a compare match occurs between MTU6.TCNT and MTU6.TGRA in reset-synchronized PWM mode, or when a compare match occurs between MTU7.TCNT and MTU7.TGRA in PWM mode 1 or normal operation mode.
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the MTU7.TCNT count. These settings are prohibited when complementary PWM mode is not selected.
1	1	Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the MTU7.TCNT count. These settings are prohibited when complementary PWM mode is not selected.

18.3.38 TADCORx : Timer A/D Converter Start Request Cycle Set Register x (x = A, B)

Base address: MTU4 = 0x9000_1200, MTU7 = 0x9000_1A00

Offset address: 0x044 (TADCORA)
0x046 (TADCORB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	TADCORA and TADCORB are 16-bit readable/writable registers that issue a corresponding A/D converter start request when the MTUn.TCNT (n = 4, 7) count reaches the value in TADCORA or TADCORB.	R/W

Note: The TADCORA and TADCORB registers should be accessed in 16-bit units.

When the A/D converter start request delaying function linked with skipping function 1 (for details, see section 18.4.9.4. A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 1) is used, the value of this register should be 0x0002 to TCDRA setting − 2 in MTU4 and 0x0002 to TCDRB setting − 2 in MTU7.

When interrupt skipping function 2 is used and the difference between the TADCORA value and the TADCORB value is small, the skipping count may not be counted correctly and the A/D converter start request may not be generated with the expected timing in some cases. The TADCORA and TADCORB values should satisfy the following conditions.

1. When skipping function 2 is specified with the skipping count set to 0
   - The difference between the TADCORA and TADCORB values should be equal to or greater than 4.
   - The TADCORA compare interval should be equal to or greater than 4 PCLKH cycles (the TADCORA update value should be the previous value + 4 or greater, or previous value − 4 or smaller).
   - The TADCORB compare interval should be equal to or greater than 4 PCLKH cycles (the TADCORB update value should be the previous value + 4 or greater, or previous value − 4 or smaller).
2. When skipping function 2 is specified with the skipping count set to 1 or greater
   - The difference between the TADCORA and TADCORB values should be equal to or greater than 2.
   - The TADCORB compare interval should be equal to or greater than 2 PCLKH cycles (the TADCORB update value should be the previous value + 2 or greater, or previous value − 2 or smaller)

18.3.39 TADCOBRx : Timer A/D Converter Start Request Cycle Set Buffer Register x (x = A, B)

Base address: MTU4 = 0x9000_1200, MTU7 = 0x9000_1A00

Offset address: 0x048 (TADCOBRA)
0x04A (TADCOBRB)

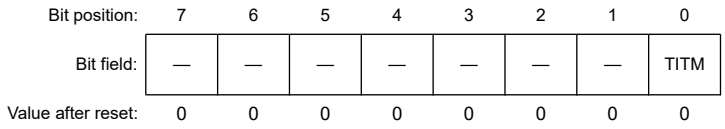
Bit	Symbol	Function	R/W
15:0	n/a	TADCOBRA and TADCOBRB are 16-bit readable/writable registers whose values are transferred to TADCORA and TADCORB, respectively, when the crest or trough of the MTUn.TCNT count is reached.	R/W

Note: TADCOBRA and TADCOBRB should be accessed in 16-bit units.

18.3.40 TITMRx : Timer Interrupt Skipping Mode Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x23A (TITMRA)
0xA3A (TITMRB)



Bit	Symbol	Function	R/W
0	TITM	Interrupt Skipping Function Select Selects one of the two types of interrupt skipping functions. 0: Selects interrupt skipping function 1 Setting the TITCR1A or TITCR1B register enables interrupt skipping function 1. 1: Selects interrupt skipping function 2 Setting the TITCR2A or TITCR2B register enables interrupt skipping function 2.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

TITMRA and TITMRB are used to select either of two skipping functions.

18.3.41 TITCR1A : Timer Interrupt Skipping Set Register 1A

Base address: MTU = 0x9000_1000

Offset address: 0x230

Bit position:	7	6	5	4	3	2	1	0
Bit field:	T3AEN	T3ACOR[2:0]			T4VEN	T4VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	T4VCOR[2:0]	TCIV4 Interrupt Skipping Count Setting These bits specify the TCIV4 interrupt skipping count within the range from 0 to 7.*1 0 0 0: Does not skip TCIV4 interrupts. 0 0 1: Sets the TCIV4 interrupt skipping count to 1. 0 1 0: Sets the TCIV4 interrupt skipping count to 2. 0 1 1: Sets the TCIV4 interrupt skipping count to 3. 1 0 0: Sets the TCIV4 interrupt skipping count to 4. 1 0 1: Sets the TCIV4 interrupt skipping count to 5. 1 1 0: Sets the TCIV4 interrupt skipping count to 6. 1 1 1: Sets the TCIV4 interrupt skipping count to 7.	R/W
3	T4VEN	TCIV4 Interrupt Skipping Enable 0: Disabled 1: Enabled	R/W
6:4	T3ACOR[2:0]	TGIA3 Interrupt Skipping Count Setting These bits specify the TGIA3 interrupt skipping count within the range from 0 to 7.*1 0 0 0: Does not skip TGIA3 interrupts. 0 0 1: Sets the TGIA3 interrupt skipping count to 1. 0 1 0: Sets the TGIA3 interrupt skipping count to 2. 0 1 1: Sets the TGIA3 interrupt skipping count to 3. 1 0 0: Sets the TGIA3 interrupt skipping count to 4. 1 0 1: Sets the TGIA3 interrupt skipping count to 5. 1 1 0: Sets the TGIA3 interrupt skipping count to 6. 1 1 1: Sets the TGIA3 interrupt skipping count to 7.	R/W
7	T3AEN	TGIA3 Interrupt Skipping Enable 0: Disabled 1: Enabled	R/W

Note 1. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed. Before changing the interrupt skipping count, be sure to set the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0 to clear the skipping counter (TITCNT1A).

The TITCR1A register enables or disables interrupt skipping and specifies the interrupt skipping count. This setting is valid only while the TITMRA.TITM bit is set to 0; when the TITMRA.TITM bit is set to 1, the setting in the TITCR1A register is cleared.

18.3.42 TITCR1B : Timer Interrupt Skipping Set Register 1B

Base address: MTU = 0x9000_1000

Offset address: 0xA30

Bit position:	7	6	5	4	3	2	1	0
Bit field:	T6AEN	T6ACOR[2:0]			T7VEN	T7VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	T7VCOR[2:0]	TCIV7 Interrupt Skipping Count Setting These bits specify the TCIV7 interrupt skipping count within the range from 0 to 7.*1 0 0 0: Does not skip TCIV7 interrupts. 0 0 1: Sets the TCIV7 interrupt skipping count to 1. 0 1 0: Sets the TCIV7 interrupt skipping count to 2. 0 1 1: Sets the TCIV7 interrupt skipping count to 3. 1 0 0: Sets the TCIV7 interrupt skipping count to 4. 1 0 1: Sets the TCIV7 interrupt skipping count to 5. 1 1 0: Sets the TCIV7 interrupt skipping count to 6. 1 1 1: Sets the TCIV7 interrupt skipping count to 7.	R/W
3	T7VEN	TCIV7 Interrupt Skipping Enable 0: Disabled 1: Enabled	R/W
6:4	T6ACOR[2:0]	TGIA6 Interrupt Skipping Count Setting These bits specify the TGIA6 interrupt skipping count within the range from 0 to 7.*1 0 0 0: Does not skip TGIA6 interrupts. 0 0 1: Sets the TGIA6 interrupt skipping count to 1. 0 1 0: Sets the TGIA6 interrupt skipping count to 2. 0 1 1: Sets the TGIA6 interrupt skipping count to 3. 1 0 0: Sets the TGIA6 interrupt skipping count to 4. 1 0 1: Sets the TGIA6 interrupt skipping count to 5. 1 1 0: Sets the TGIA6 interrupt skipping count to 6. 1 1 1: Sets the TGIA6 interrupt skipping count to 7.	R/W
7	T6AEN	TGIA6 Interrupt Skipping Enable 0: Disabled 1: Enabled	R/W

Note 1. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed.
Before changing the interrupt skipping count, be sure to set the TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0 to clear the skipping counter (TITCNT1B).

The TITCR1B register enables or disables interrupt skipping and specifies the interrupt skipping count. This setting is valid only while the TITMRB.TITM bit is set to 0; when the TITMRB.TITM bit is set to 1, the setting in the TITCR1B register is cleared.

18.3.43 TITCNT1A : Timer Interrupt Skipping Counter 1A

Base address: MTU = 0x9000_1000

Offset address: 0x231

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	T3ACNT[2:0]			—	T4VCNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	T4VCNT[2:0]	TCIV4 Interrupt Counter While the T4VEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TCIV4 interrupt occurs.	R
3	—	This bit is read as 0.	R
6:4	T3ACNT[2:0]	TGIA3 Interrupt Counter While the T3AEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TGIA3 interrupt occurs.	R
7	—	This bit is read as 0.	R

Note: To clear the TITCNT1A, set the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0.

TITCNT1A is 8-bit readable/writable counters. TITCNT1A retains their values even after stopping the count operation of MTU3.TCNT and MTU4.TCNT.

T4VCNT[2:0] bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T4VEN bit in TITCR1A is set to 0
- When the T4VCOR[2:0] bits in TITCR1A are set to 000b
- When the T4VCNT[2:0] bits in TITCNT1A match the T4VCOR[2:0] bits in TITCR1A

T3ACNT[2:0] bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T3AEN bit in TITCR1A is set to 0
- When the T3ACOR[2:0] bits in TITCR1A are set to 000b
- When the T3ACNT[2:0] bits in TITCNT1A match the T3ACOR[2:0] bits in TITCR1A

18.3.44 TITCNT1B : Timer Interrupt Skipping Counter 1B

Base address: MTU = 0x9000_1000

Offset address: 0xA31

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	T6ACNT[2:0]			—	T7VCNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	T7VCNT[2:0]	TCIV7 Interrupt Counter While the T7VEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TCIV7 interrupt occurs.	R
3	—	This bit is read as 0.	R
6:4	T6ACNT[2:0]	TGIA6 Interrupt Counter While the T6AEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TGIA6 interrupt occurs.	R
7	—	This bit is read as 0.	R

Note: To clear the TITCNT1B, set the TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0.

TITCNT1B is 8-bit readable/writable counters. TITCNT1B retains their values even after stopping the count operation of MTU6.TCNT and MTU7.TCNT.

T7VCNT[2:0] bits (TCIV7 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T7VEN bit in TITCR1B is set to 0
- When the T7VCOR[2:0] bits in TITCR1B are set to 000b
- When the T7VCNT[2:0] bits in TITCNT1B match the T7VCOR[2:0] bits in TITCR1B

T6ACNT[2:0] bits (TGIA6 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T6AEN bit in TITCR1B is set to 0
- When the T6ACOR[2:0] bits in TITCR1B are set to 000b

- When the T6ACNT[2:0] bits in TITCNT1B match the T6ACOR[2:0] bits in TITCR1B

18.3.45 TITCR2A : Timer Interrupt Skipping Set Register 2A

Base address: MTU = 0x9000_1000

Offset address: 0x23B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TRG4COR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TRG4COR[2:0]	TRG4AN/TRG4BN Interrupt Skipping Count Setting These bits specify the TRG4AN/TRG4BN interrupt skipping count within the range from 0 to 7. 0 0 0: Does not skip TRG4AN and TRG4BN interrupts. 0 0 1: Sets the TRG4AN and TRG4BN interrupt skipping count to 1. 0 1 0: Sets the TRG4AN and TRG4BN interrupt skipping count to 2. 0 1 1: Sets the TRG4AN and TRG4BN interrupt skipping count to 3. 1 0 0: Sets the TRG4AN and TRG4BN interrupt skipping count to 4. 1 0 1: Sets the TRG4AN and TRG4BN interrupt skipping count to 5. 1 1 0: Sets the TRG4AN and TRG4BN interrupt skipping count to 6. 1 1 1: Sets the TRG4AN and TRG4BN interrupt skipping count to 7.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

TITCR2A specifies the interrupt skipping count for TRG4AN and TRG4BN.

This setting is valid only while TITMRA is set to 1.

18.3.46 TITCR2B : Timer Interrupt Skipping Set Register 2B

Base address: MTU = 0x9000_1000

Offset address: 0xA3B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TRG7COR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TRG7COR[2:0]	TRG7AN/TRG7BN Interrupt Skipping Count Setting These bits specify the TRG7AN/TRG7BN interrupt skipping count within the range from 0 to 7. 0 0 0: Does not skip TRG7AN and TRG7BN interrupts. 0 0 1: Sets the TRG7AN and TRG7BN interrupt skipping count to 1. 0 1 0: Sets the TRG7AN and TRG7BN interrupt skipping count to 2. 0 1 1: Sets the TRG7AN and TRG7BN interrupt skipping count to 3. 1 0 0: Sets the TRG7AN and TRG7BN interrupt skipping count to 4. 1 0 1: Sets the TRG7AN and TRG7BN interrupt skipping count to 5. 1 1 0: Sets the TRG7AN and TRG7BN interrupt skipping count to 6. 1 1 1: Sets the TRG7AN and TRG7BN interrupt skipping count to 7.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

TITCR2B specifies the interrupt skipping count for TRG7AN and TRG7BN.

This setting is valid only while TITMRB is set to 1.

18.3.47 TITCNT2A : Timer Interrupt Skipping Counter 2A

Base address: MTU = 0x9000_1000

Offset address: 0x23C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TRG4CNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TRG4CNT[2:0]	TRG4AN/TRG4BN Interrupt Counter These bits start counting from the value set in TRG4COR[2:0] and the count decrements every time TRG4AN or TRG4BN is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.	R
7:3	—	These bits are read as 0.	R

TITCNT2A starts counting from the values set in the TRG4COR[2:0] bits and the count decrements every time TRG4AN or TRG4BN is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.

TRG4CNT[2:0] bits (TRG4AN/TRG4BN Interrupt Counter)

These bits start counting from the value set in the TRG4COR[2:0] bits and the count decrements every time a TRG4AN or TRG4BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRA is 0
- When the TRG4COR[2:0] bits in TITCR2A are set to 000b
- When the count of TRG4AN and TRG4BN occurrence matches the TRG4COR[2:0] value in TITCR2A

18.3.48 TITCNT2B : Timer Interrupt Skipping Counter 2B

Base address: MTU = 0x9000_1000

Offset address: 0xA3C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TRG7CNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TRG7CNT[2:0]	TRG7AN/TRG7BN Interrupt Counter These bits start counting from the value set in TRG7COR[2:0] and the count decrements every time TRG7AN or TRG7BN is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.	R
7:3	—	These bits are read as 0.	R

TITCNT2B starts counting from the values set in the TRG7COR[2:0] bits and the count decrements every time TRG7AN or TRG7BN is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.

TRG7CNT[2:0] bits (TRG7AN/TRG7BN Interrupt Counter)

These bits start counting from the value set in the TRG7COR[2:0] bits and the count decrements every time a TRG7AN or TRG7BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRB is 0

- When the TRG7COR[2:0] bits in TITCR2B are set to 000b
- When the count of TRG7AN and TRG7BN occurrence matches the TRG7COR[2:0] value in TITCR2B

18.3.49 Bus Master Interface

The timer counter (MTU8.TCNT), general registers (MTU8.TGRn) for MTU8, and MTU1.TCNTLW, MTU1.TGRALW, and MTU1.TGRBLW registers when TMDR3.LWA = 1 are 32-bit registers. A 32-bit data bus to the bus master enables 32-bit read/write access. 8- and 16-bit read/write are not allowed. Always access these registers in 32-bit units.

The timer counters (MTU0.TCNT to MTU7.TCNT) excluding MTU8, general registers (MTU0.TGRn to MTU7.TGRn), timer subcounters (TCNTSA and TCNTSB), timer cycle buffer registers (TCBRA and TCBRB), timer dead time data registers (TDDRA and TDDRB), timer cycle data registers (TCDRA and TCDRB), timer A/D converter start request control registers (MTU4.TADCR and MTU7.TADCR), timer A/D converter start request cycle set registers (MTU4.TADCORA, MTU4.TADCORB, MTU7.TADCORA, and MTU7.TADCORB), and timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA, MTU4.TADCOBRB, MTU7.TADCOBRA, and MTU7.TADCOBRB) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/write access. 8-bit read/write is not allowed. Always access the registers in 16-bit units.

All registers other than the above registers are 8-bit registers. A 16-bit data bus to the CPU enables 16-bit read/write access. These registers can also be accessed in 8-bit units.

18.4 Operations

18.4.1 Basic Functions

Each channel has TCNT and TGR. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.

Each TGR can be used as an input capture register or an output compare register.

18.4.1.1 Counter Operation

When one of bits CST0 to CST4 and CST8 in the TSTRA register, bits CST6 and CST7 in the TSTRB register, and bits CSTU5, CSTV5, and CSTW5 in the MTU5.TSTR register is set to 1, TCNT for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

(1) Example of Count Operation Setting Procedure

Figure 18.5 shows an example of the count operation setting procedure.

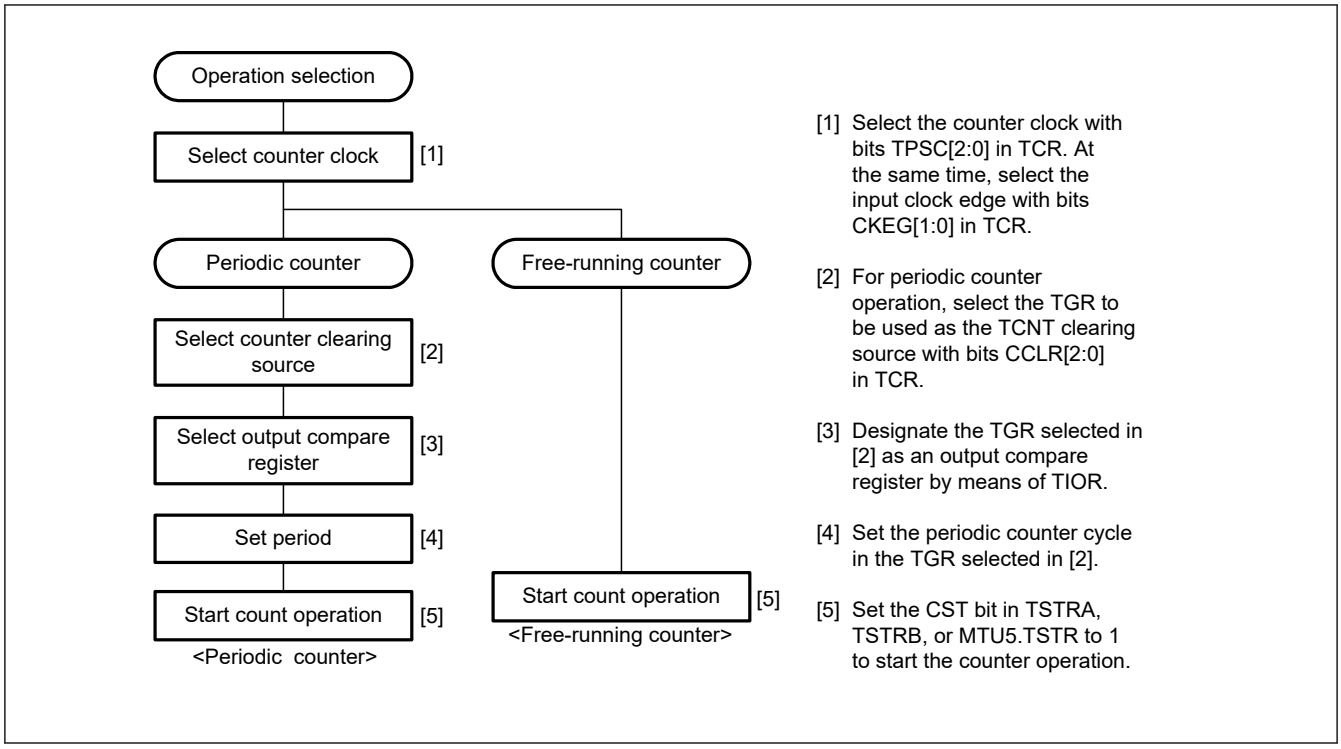


Figure 18.5 Example of count operation setting procedure

(2) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the MTU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTRA, TSTRB, or MTU5.TSTR is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from 0xFFFF to 0x0000), if the corresponding TIER.TCIEV bit is 1, the MTU requests an interrupt. After an overflow, TCNT starts counting up again from 0x0000.

Figure 18.6 shows free-running counter operation.

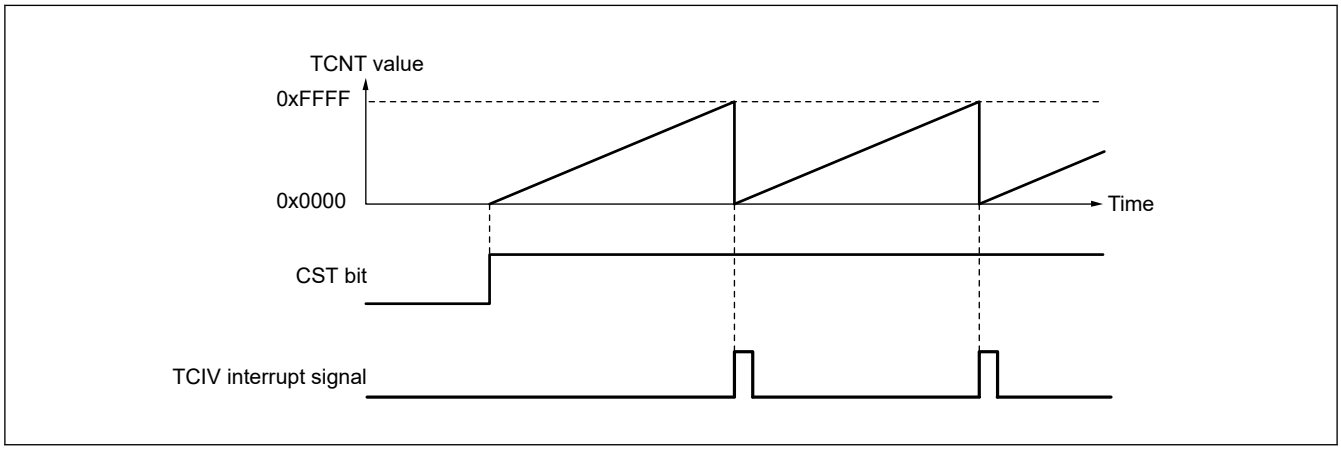


Figure 18.6 Free-running counter operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the cycle is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR[2:0] in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTRA, TSTRB or MTU5.TSTR is set to 1. When the count matches the value in TGR, TCNT is cleared to 0x0000.

If the value of the corresponding TIER.TGIE bit is 1 at this point, the MTU requests an interrupt. After a compare match, TCNT starts counting up again from 0x0000.

Figure 18.7 shows periodic counter operation.

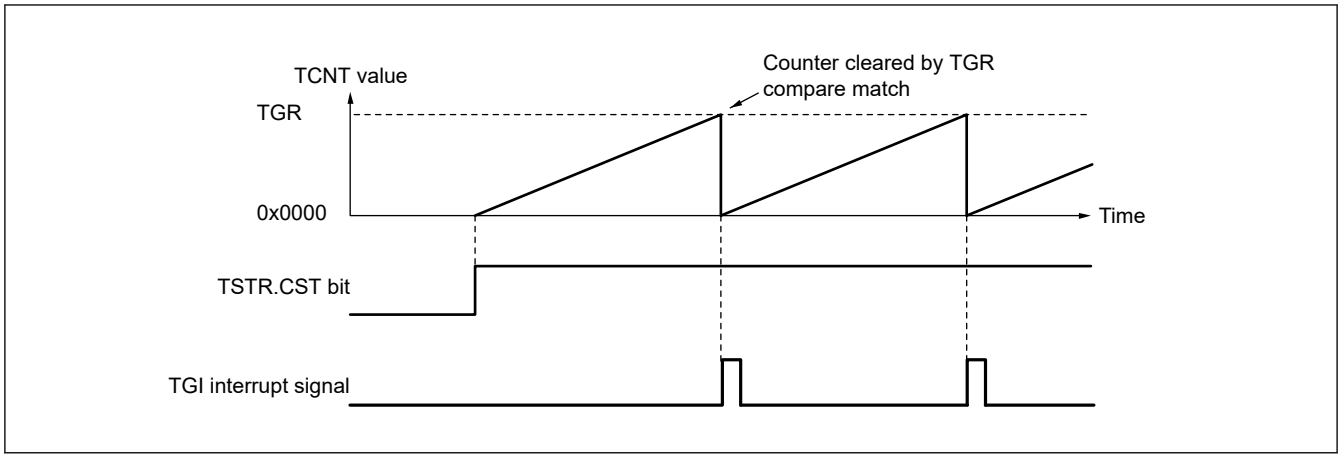


Figure 18.7 Periodic counter operation

18.4.1.2 Waveform Output by Compare Match

The MTU can output low or high or toggles output from the corresponding output pin using compare match. Note that MTU5 cannot output compare-match signals.

(1) Example of Procedure for Setting Waveform Output by Compare Match

Figure 18.8 shows an example of the procedure for setting waveform output by compare match.

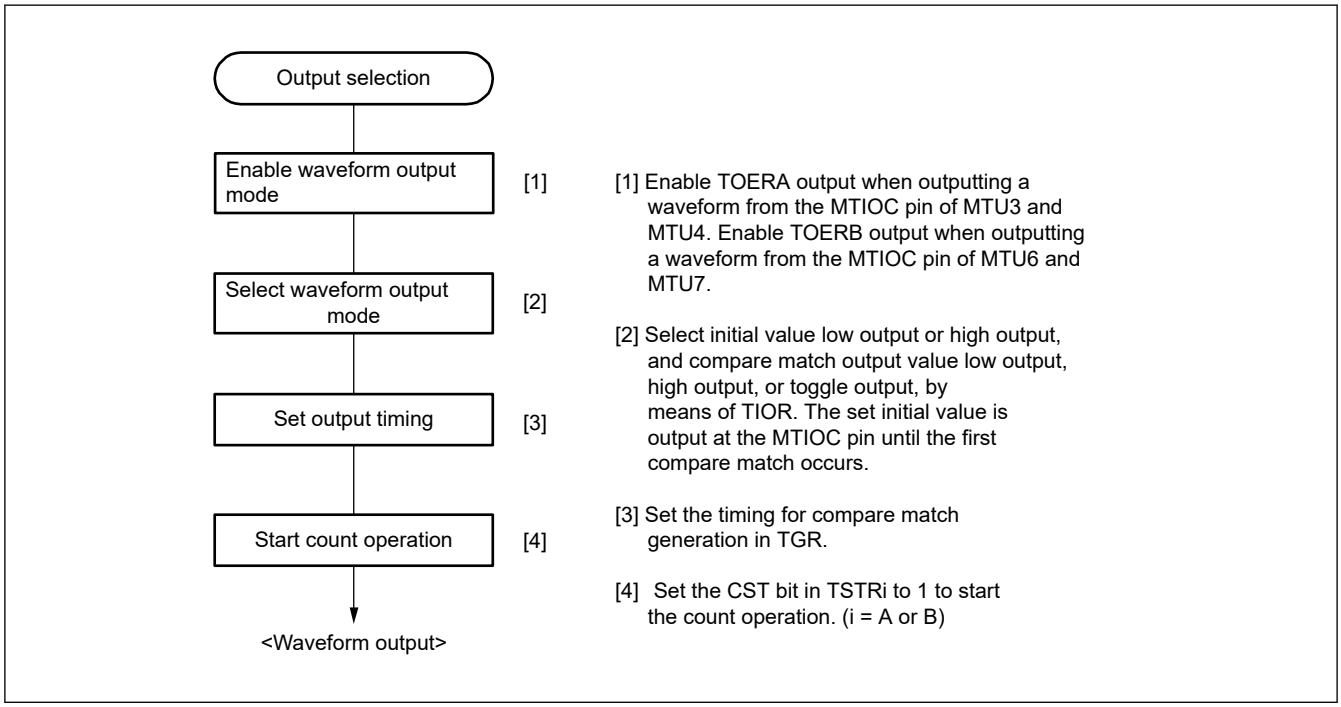


Figure 18.8 Example of procedure for setting waveform output by compare match

(2) Examples of Waveform Output Operation

Figure 18.9 shows an example of low output and high output.

In this example, TCNT has been designated as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the pin level is the same as the specified level, the pin level does not change.

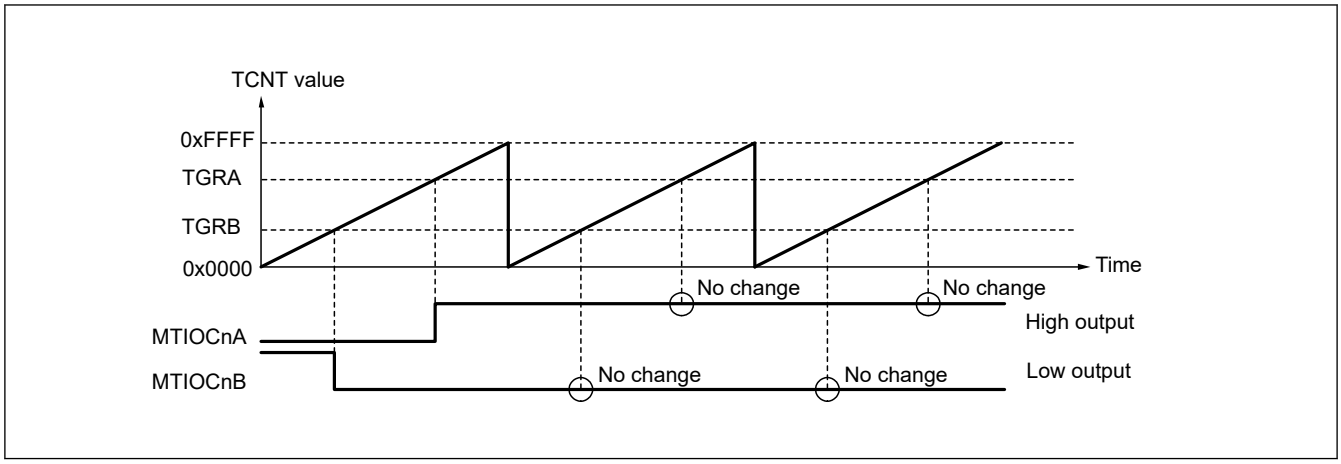


Figure 18.9 Example of low output and high output operation

Figure 18.10 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made so that the output is toggled by both compare match A and compare match B.

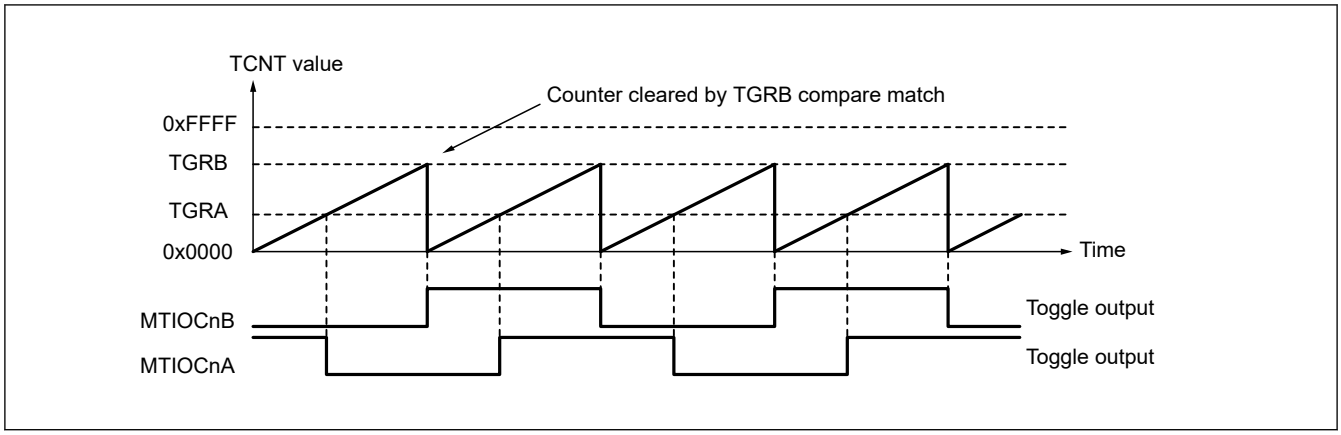


Figure 18.10 Example of toggle output operation

18.4.1.3 Input Capture Function

The TCNT value can be transferred to TGR on detection of an edge input on the MTIOCnm (n = 0 to 4, 6, 7, or 8, m = A to D), MTIC5U, MTIC5V, or MTIC5W pin.

The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0 and MTU1, a count clock of another channel or compare match signal can also be specified as the input capture source.

Note: When a count input clock of another channel is used as the input capture input for MTU0 and MTU1, PCLKH/1 should not be selected as the count input clock used for input capture input. Input capture is not generated if PCLKH/1 is selected.

(1) Example of Input Capture Operation Setting Procedure

Figure 18.11 shows an example of the input capture operation setting procedure.

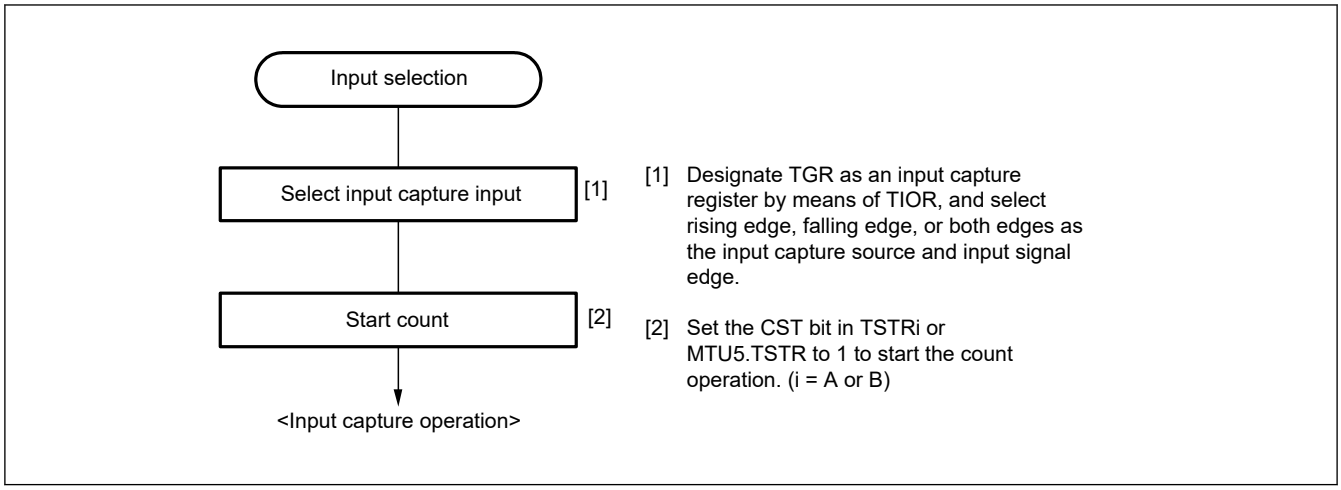


Figure 18.11 Example of input capture operation setting procedure

(2) Example of Input Capture Operation

Figure 18.12 shows an example of input capture operation.

In this example, both rising and falling edges have been selected as the MTIOCnA pin input capture input edge, the falling edge has been selected as the MTIOCnB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT. (n = 0 to 4, 6, 7, 8)

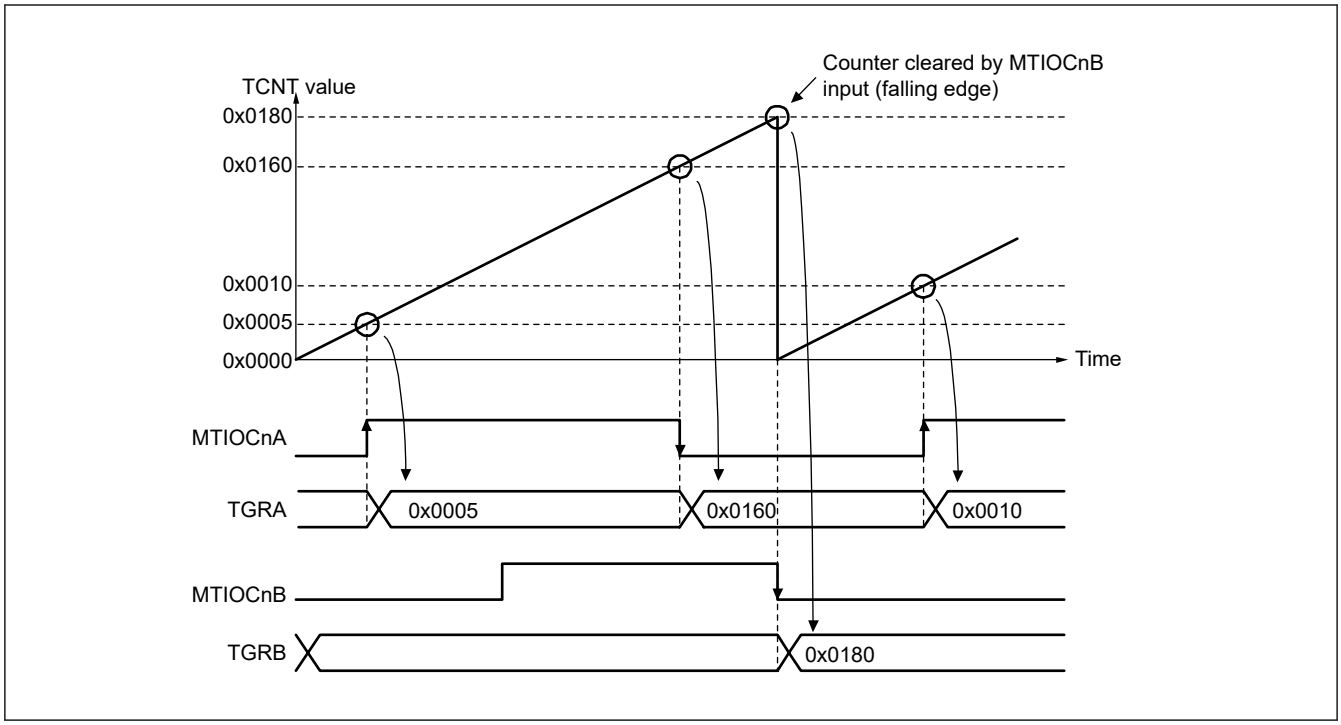


Figure 18.12 Example of input capture operation (n = 0 to 4, 6, 7, 8)

18.4.2 Synchronous Operation

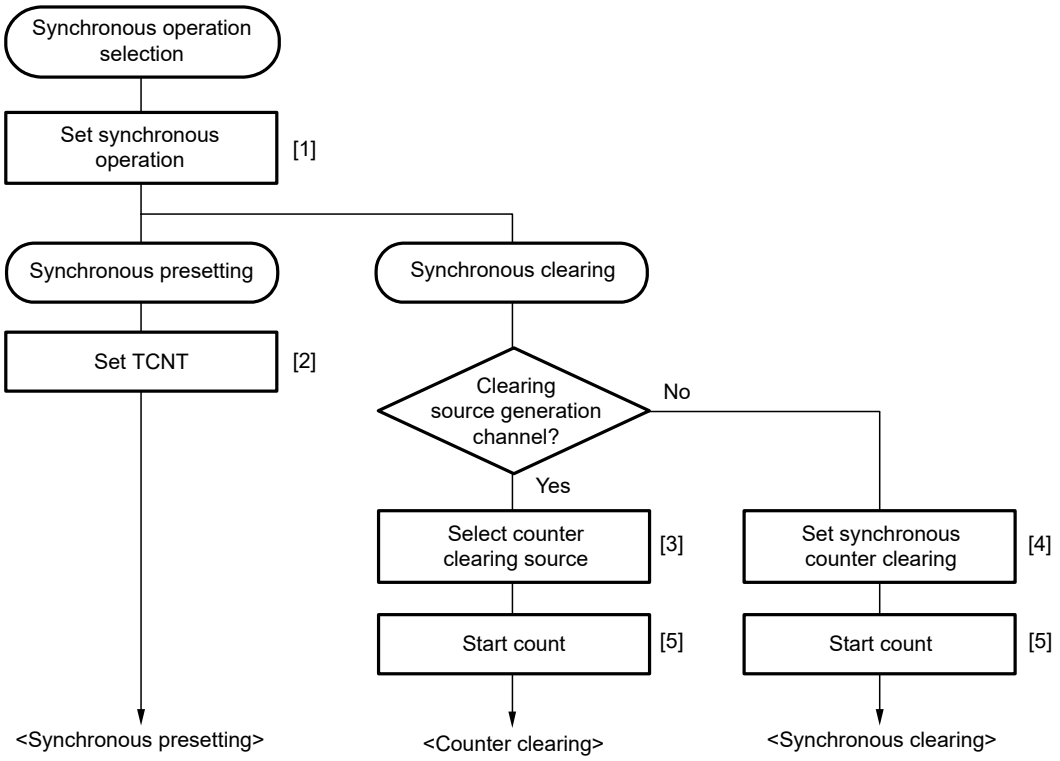
In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous presetting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.

Synchronous operation increases the number of TGR registers assigned to a single time base.

MTU0 to MTU4, MTU6, and MTU7 can all be designated for synchronous operation. MTU5 and MTU8 cannot be used for synchronous operation.

18.4.2.1 Example of Synchronous Operation Setting Procedure

Figure 18.13 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYRj corresponding to the channels to be designated for synchronous operation. (j = A or B)
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use bits CCLR[2:0] in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use bits CCLR[2:0] in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 the CST bits in TSTRj for the relevant channels, to start the count operation. (j = A or B)

Figure 18.13 Example of synchronous operation setting procedure

18.4.2.2 Example of Synchronous Operation

Figure 18.14 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU2, MTU0.TGRB compare match has been set as the counter clearing source in MTU0, and synchronous clearing has been set for the counter clearing source in MTU1 and MTU2.

Three-phase PWM waveforms are output from pins MTIOC0A, MTIOC1A, and MTIOC2A. At this time, synchronous presetting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in MTU0 to MTU2, and the data set in MTU0.TGRB is used as the PWM cycle.

For details of PWM modes, see section 18.4.5. PWM Modes.

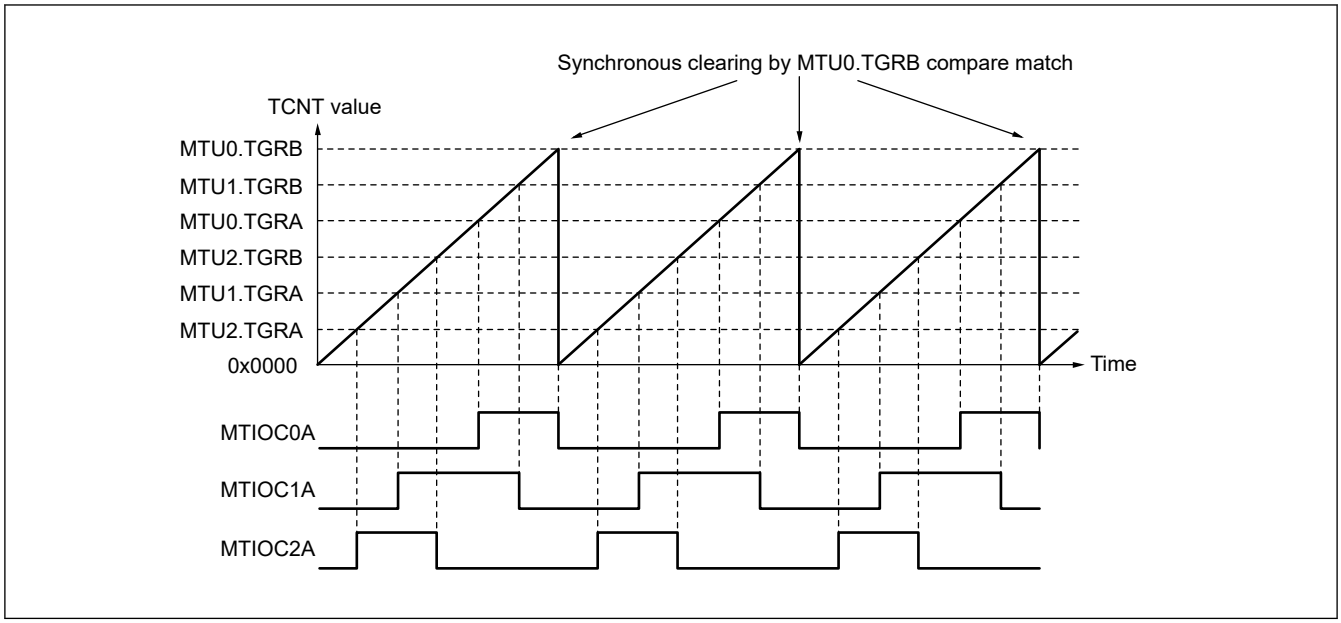


Figure 18.14 Example of synchronous operation

18.4.3 Buffer Operation

Buffer operation, provided for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, enables TGRC and TGRD to be used as buffer registers. In MTU0, TGRF can also be used as a buffer register.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Note: MTU0.TGRE cannot be designated as an input capture register and can only operate as a compare match register.

Table 18.59 shows the register combinations used in buffer operation.

Table 18.59 Register combinations in buffer operation

Channel	Timer general register	Buffer register
MTU0	TGRA	TGRC
	TGRB	TGRD
	TGRE	TGRF
MTU3	TGRA	TGRC
	TGRB	TGRD
MTU4	TGRA	TGRC
	TGRB	TGRD
MTU6	TGRA	TGRC
	TGRB	TGRD
MTU7	TGRA	TGRC
	TGRB	TGRD
MTU8	TGRA	TGRC
	TGRB	TGRD

(1) When TGR is an output compare register

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

This operation is illustrated in Figure 18.15.

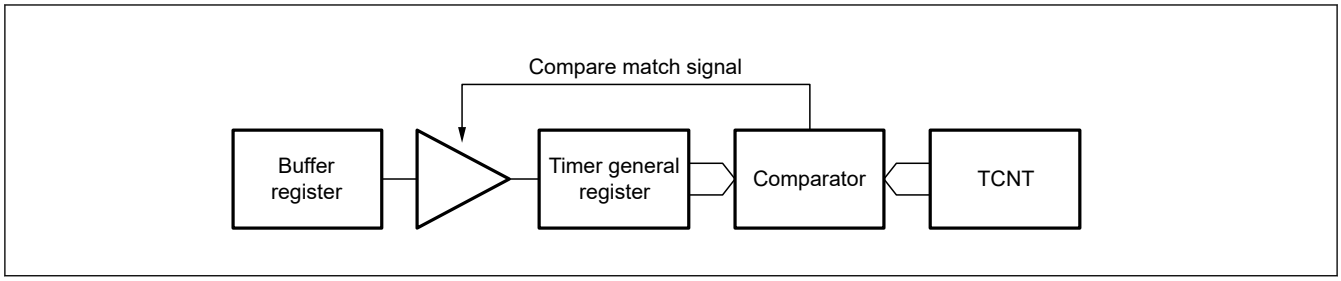


Figure 18.15 Compare match buffer operation

(2) When TGR is an input capture register

When an input capture occurs, the value in TCNT is transferred to TGR and the value previously held in TGR is transferred to the buffer register.

This operation is illustrated in Figure 18.16.

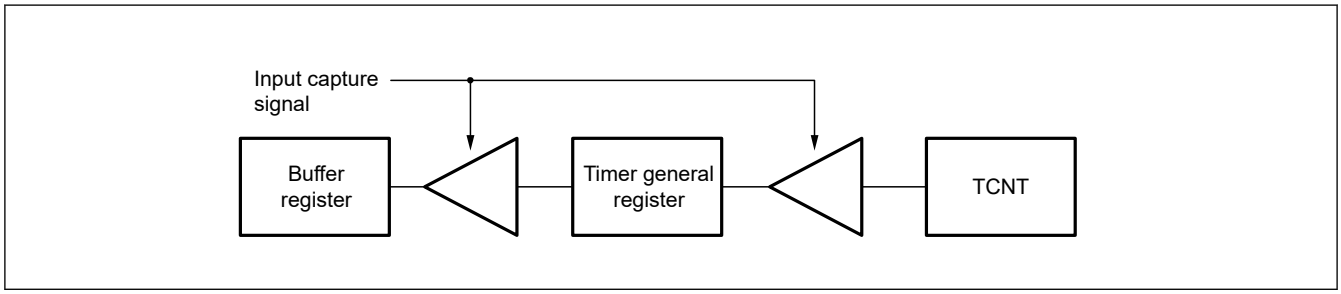


Figure 18.16 Input capture buffer operation

18.4.3.1 Example of Buffer Operation Setting Procedure

Figure 18.17 shows an example of the buffer operation setting procedure.

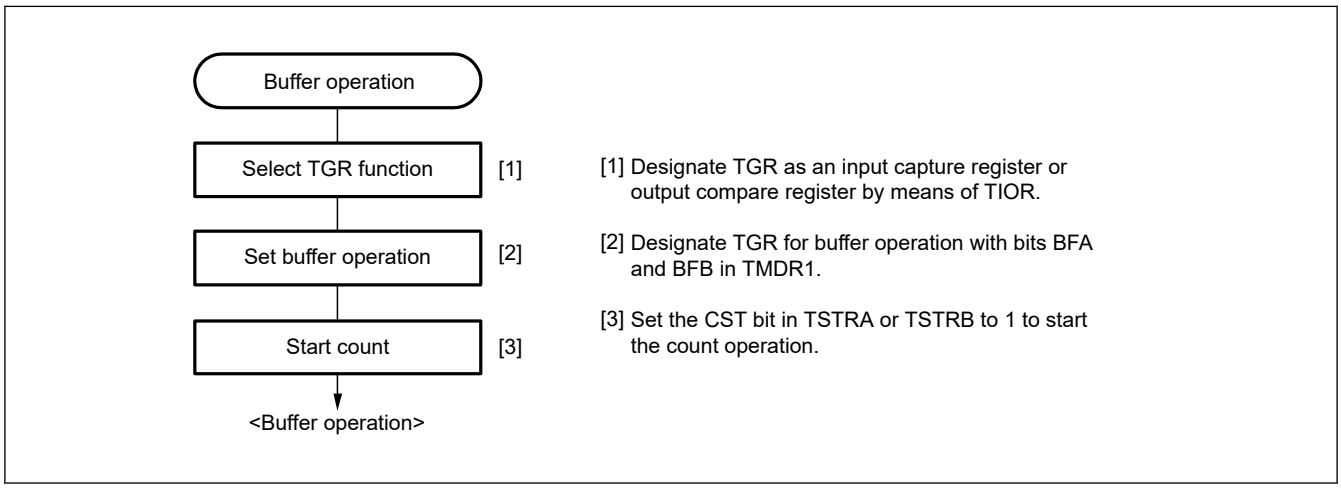


Figure 18.17 Example of buffer operation setting procedure

18.4.3.2 Examples of Buffer Operation

(1) When TGR is an Output Compare Register

Figure 18.18 shows an operation example in which PWM mode 1 has been designated for MTU0, and TGRC has been designated as a buffer register for TGRA. The settings used in this example are TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. In this example, the TTSA bit in TBTM is set to 0.

As buffer operation has been set, when compare match A occurs, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

For details of PWM modes, see section 18.4.5. PWM Modes.

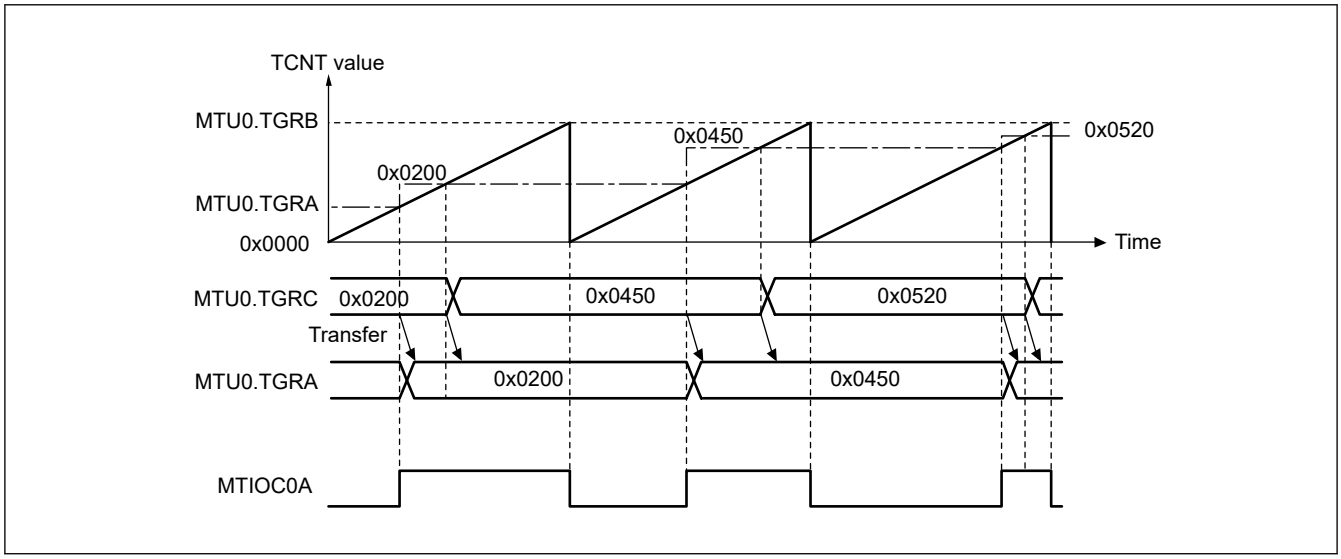


Figure 18.18 Example of buffer operation (1)

(2) When TGR is an Input Capture Register

Figure 18.19 shows an operation example in which TGRA has been designated as an input capture register, and TGRC has been designated as a buffer register for TGRA.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the MTIOCnA pin input capture input edge. (n = 0 to 4, 6, 7, 8)

As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

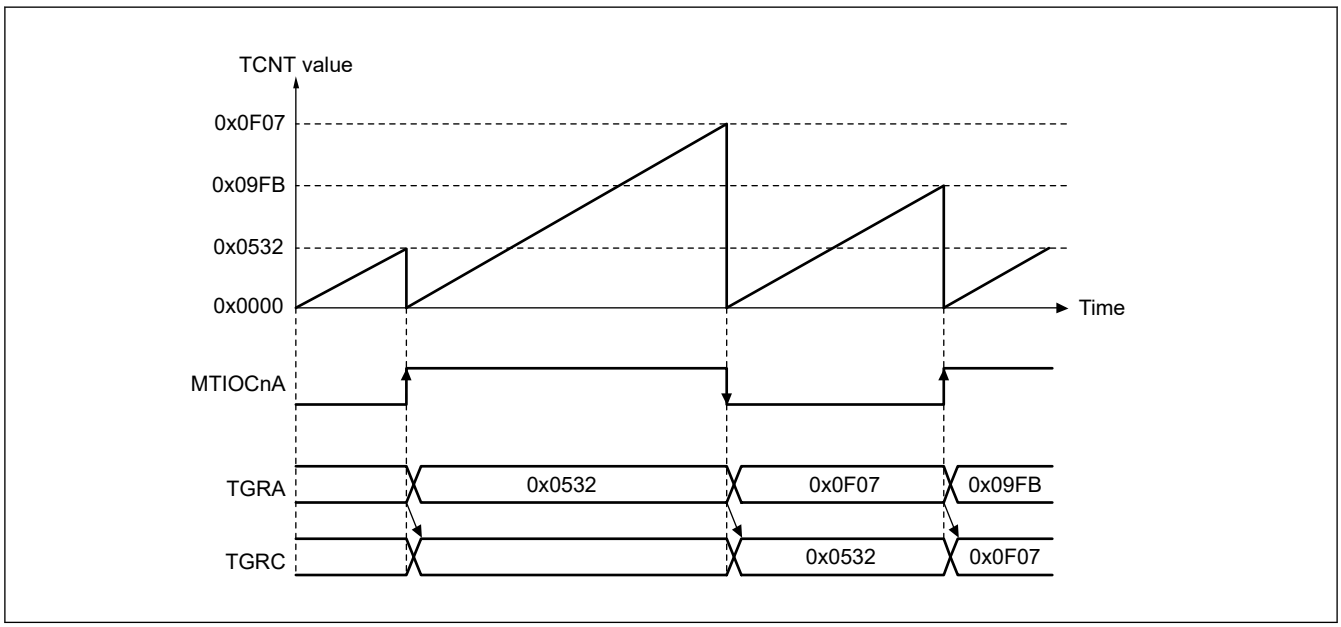


Figure 18.19 Example of buffer operation (2) (n = 0 to 4, 6, 7, 8)

18.4.3.3 Selecting Timing for Transfer from Buffer Registers to Timer General Registers in PWM Mode

The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for MTU0 or in PWM mode 1 for MTU3, MTU4, MTU6, and MTU7 by setting the buffer operation transfer mode registers (MTUn.TBTM

(n = 0, 3, 4, 6, 7)). Either compare match (value after reset) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (0xFFFF to 0x0000)
- When 0x0000 is written to TCNT during counting
- When TCNT is set to 0x0000 under the condition specified in the CCLR[2:0] bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 18.20 shows an operation example in which PWM mode 1 is designated for MTU0 and buffer operation is designated for MTU0.TGRA and MTU0.TGRC. The settings used in this example are MTU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. The TTSA bit in MTU0.TBTM is set to 1.

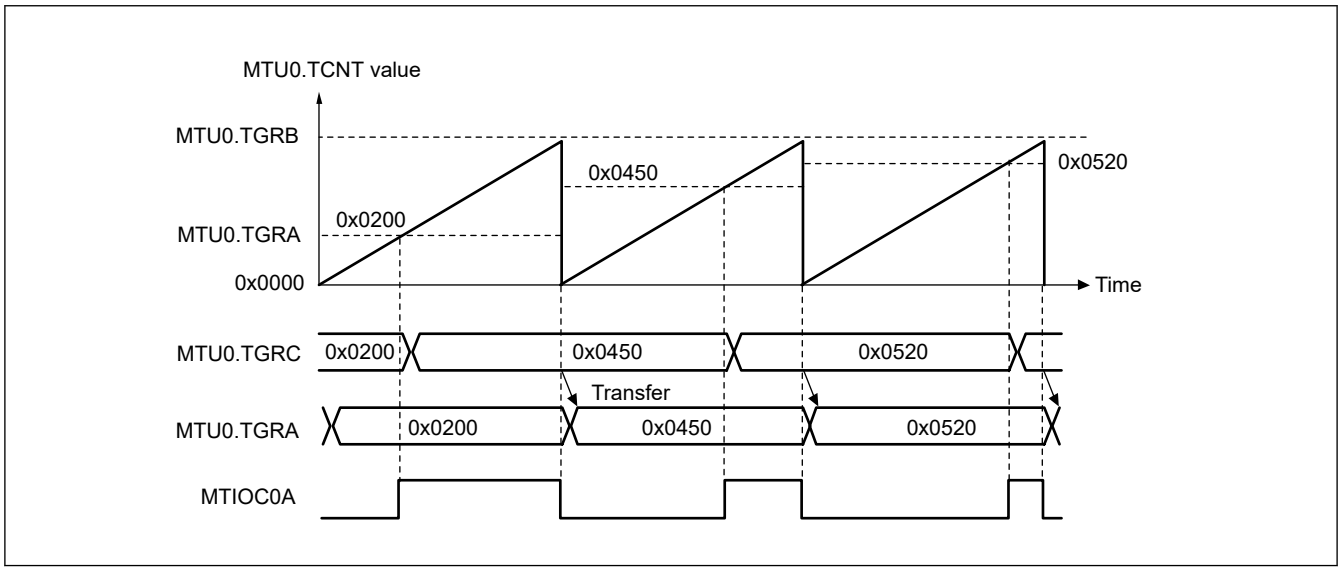


Figure 18.20 Example of buffer operation when MTU0.TCNT clearing is selected for MTU0.TGRC to MTU0.TGRA transfer timing

18.4.4 Cascaded Operation

In cascaded operation, two 16-bit counters in different channels are used together as a 32-bit counter.

There are two functions for connecting MTU1 and MTU2 to use as a 32-bit counter: cascade connection to be set when the MTU1.TMDR3.LWA bit is 0, and cascade connection 32-bit phase counting mode to be set when the MTU1.TMDR3.LWA bit is 1. For details on cascade connection 32-bit phase counting mode, see section 18.4.6.2. Cascade Connection 32-Bit Phase Counting Mode. This section describes the cascade connection function to be set when the MTU1.TMDR3.LWA bit is 0.

This function operates when the MTU1.TMDR3.LWA bit is set to 0 and the MTU1.TCR.TPSC[2:0] bits are set so that MTU1.TCNT counts at an overflow/underflow of MTU2.TCNT. Underflow occurs only when the MTU2 to which the lower 16 bits allocated is in phase counting mode.

Table 18.60 shows the register combinations used in cascaded operation.

When phase counting mode is set for MTU1, the count clock setting is invalid and the counters operate independently in phase counting mode.

Table 18.60 Cascaded combinations

Combination	Upper 16 bits	Lower 16 bits
MTU1 and MTU2	MTU1.TCNT	MTU2.TCNT

For simultaneous input capture of MTU1.TCNT and MTU2.TCNT during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). The input-capture condition is of edges in the signal produced by taking the logical OR of the input level on the main input pin and the input level on the added input pin. Accordingly, if either is at the high level, a change in the level of the other will not produce an edge for detection.

For details, see section 18.4.4.4. Cascaded Operation Example (c). For input capture in cascade connection, see section 18.7.21. Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.

Table 18.61 shows the TICCR setting and input capture input pins.

Table 18.61 TICCR setting and input capture input pins

Target input capture	TICCR setting	Input capture input pin
Input capture from MTU1.TCNT to MTU1.TGRA	I2AE bit = 0 (Initial value)	MTIOC1A
	I2AE bit = 1	MTIOC1A, MTIOC2A
Input capture from MTU1.TCNT to MTU1.TGRB	I2BE bit = 0 (Initial value)	MTIOC1B
	I2BE bit = 1	MTIOC1B, MTIOC2B
Input capture from MTU2.TCNT to MTU2.TGRA	I1AE bit = 0 (Initial value)	MTIOC2A
	I1AE bit = 1	MTIOC2A, MTIOC1A
Input capture from MTU2.TCNT to MTU2.TGRB	I1BE bit = 0 (Initial value)	MTIOC2B
	I1BE bit = 1	MTIOC2B, MTIOC1B

18.4.4.1 Example of Cascaded Operation Setting Procedure

Figure 18.21 shows an example of the cascaded operation setting procedure.

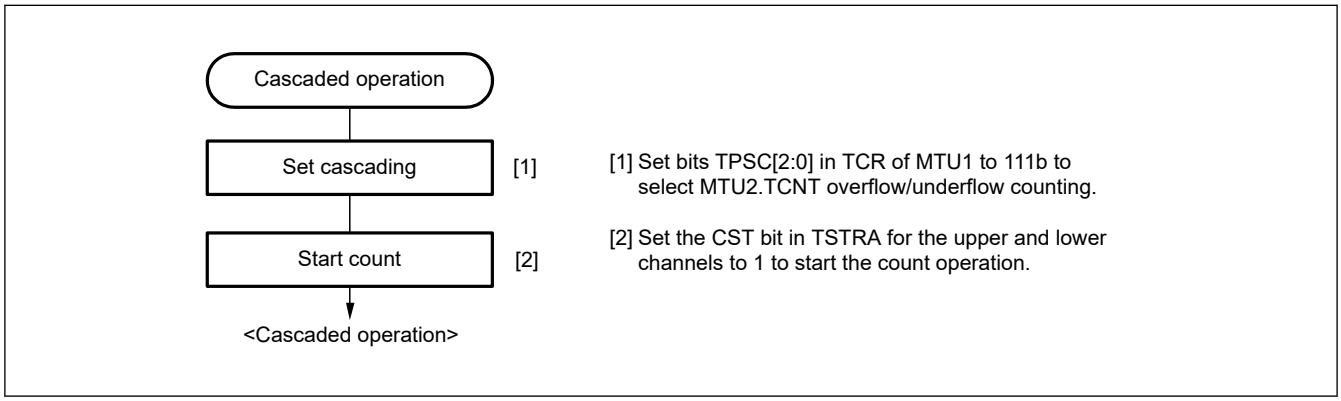


Figure 18.21 Cascaded operation setting procedure

18.4.4.2 Cascaded Operation Example (a)

Figure 18.22 shows the operation when MTU1.TCNT is set for counting at MTU2.TCNT overflow/underflow and MTU2 is set for phase counting mode while MTU1.TCNT and MTU2.TCNT are cascaded.

MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

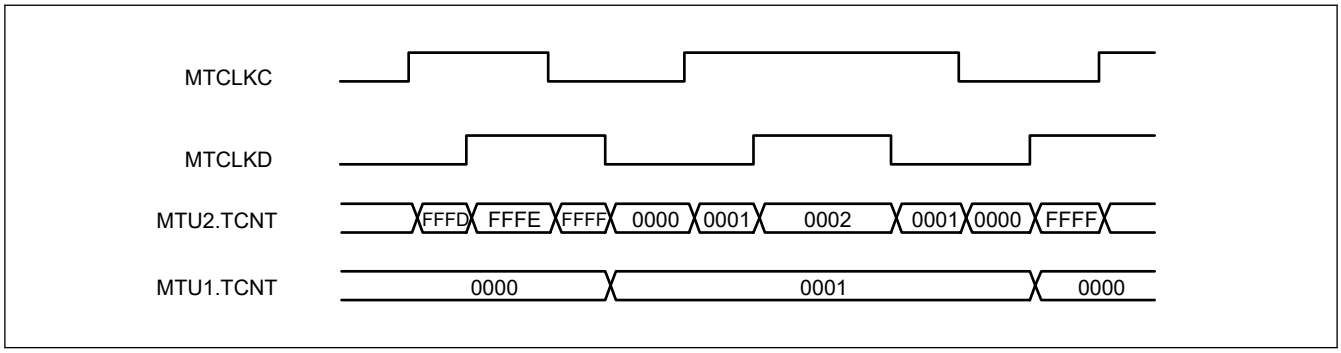


Figure 18.22 Cascaded operation example (a)

18.4.4.3 Cascaded Operation Example (b)

Figure 18.23 shows the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in TICCR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the MTU1.TIOR.IOA[3:0] bits have selected the MTIOC1A rising edge for the input capture timing while the MTU2.TIOR.IOA[3:0] bits have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, the rising edge of both MTIOC1A and MTIOC2A is used for the MTU1.TGRA input capture condition. For the MTU2.TGRA input capture condition, the MTIOC2A rising edge is used.

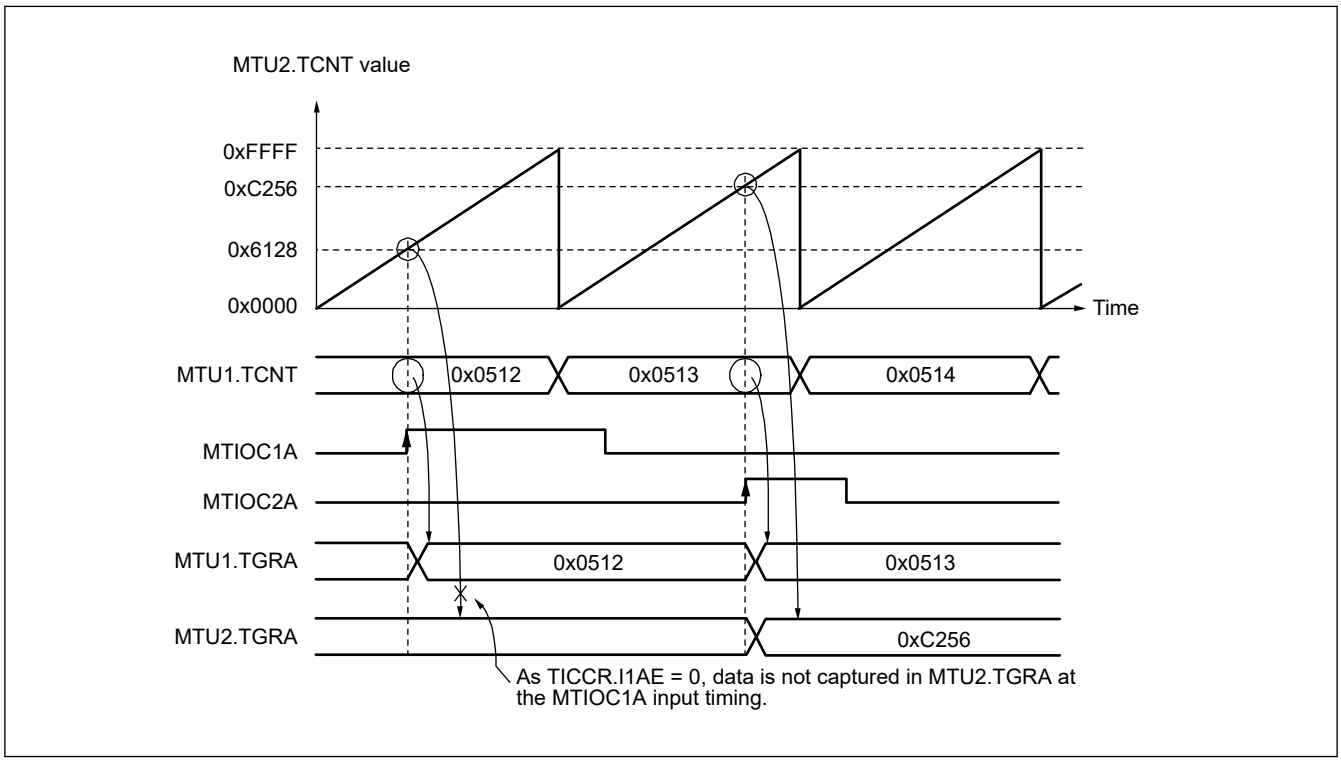
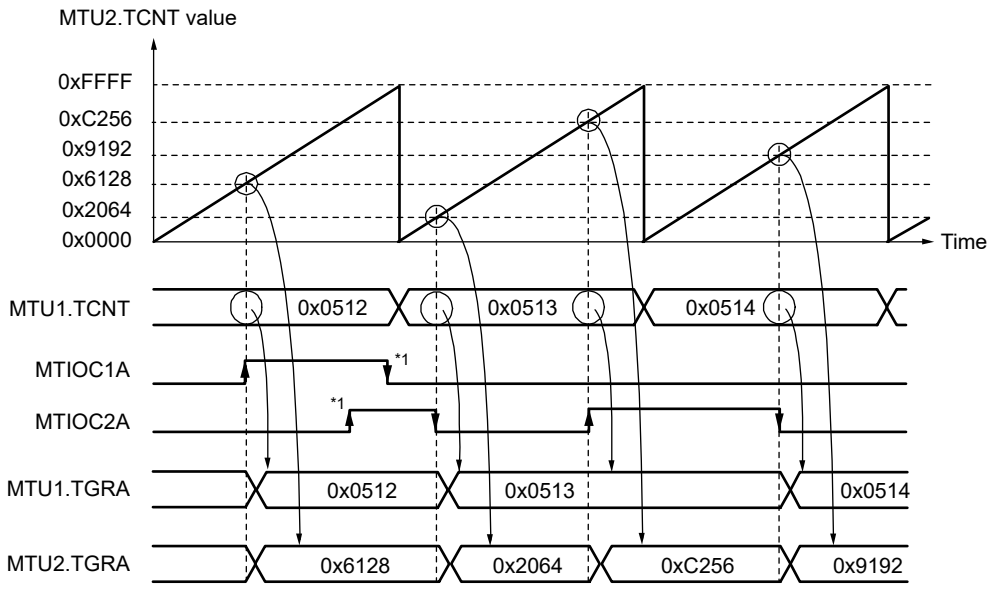


Figure 18.23 Cascaded operation example (b)

18.4.4.4 Cascaded Operation Example (c)

Figure 18.24 shows the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE and I1AE bits have been set to 1 to include the MTIOC2A and MTIOC1A pins in the MTU1.TGRA and MTU2.TGRA input capture conditions, respectively. In this example, the IOA[3:0] bits in both MTU1.TIOR and MTU2.TIOR have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.



Note 1. When either of the input signals is driven high, the selected edge of the other cannot act as the input capture condition.

Figure 18.24 Cascaded operation example (c)

18.4.4.5 Cascaded Operation Example (d)

Figure 18.25 shows the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA[3:0] bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA[3:0] bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as MTU1.TIOR has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the I2AE bit in TICCR has been set to 1.

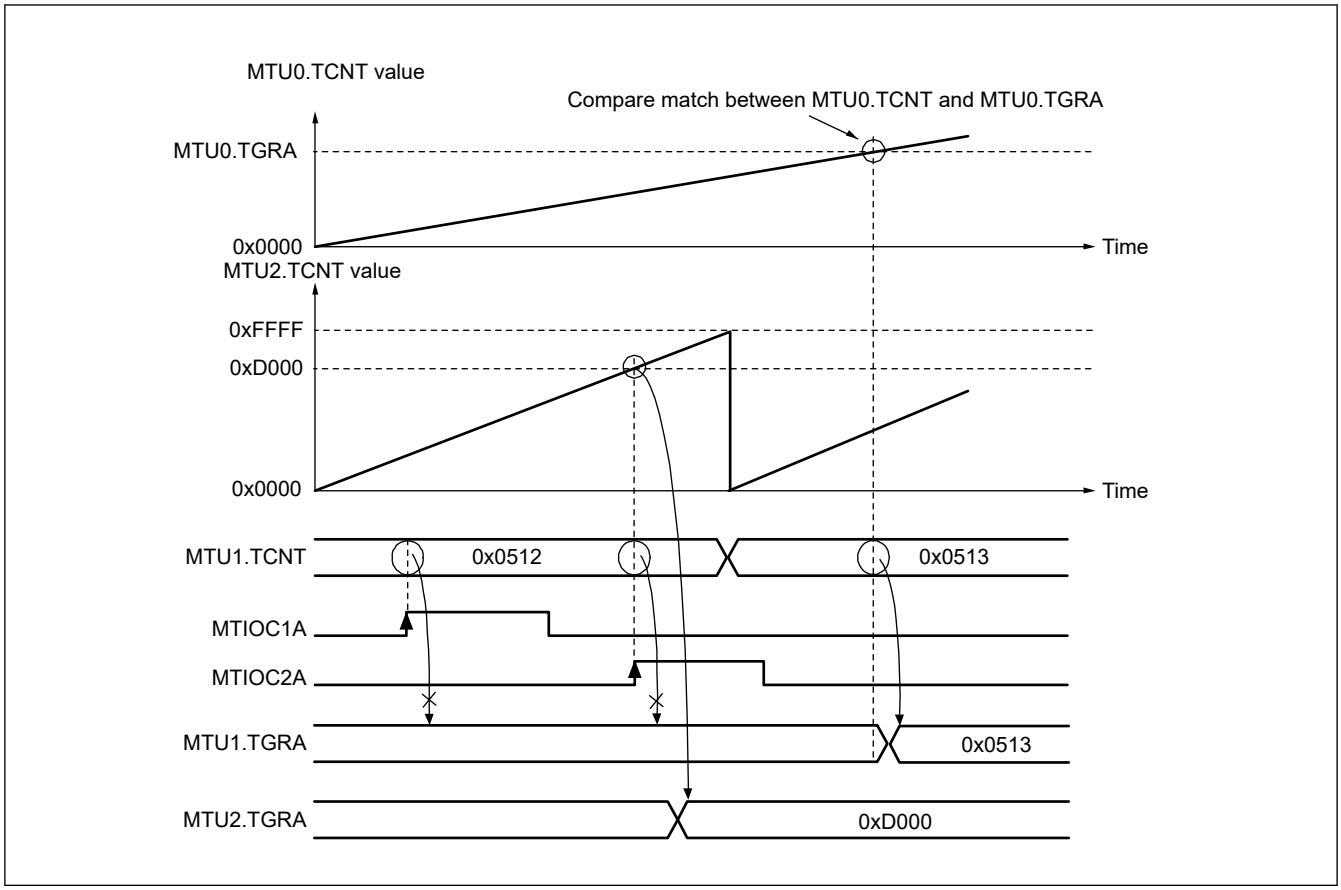


Figure 18.25 Cascaded operation example (d)

18.4.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output level can be selected as low, high, or toggle output in response to a compare match of each TGR.

PWM waveforms in the range of 0% to 100% duty cycle can be output according to the TGR settings.

By designating TGR compare match as the counter clearing source, the PWM period can be specified in that register.

Every channel except MTU5 and MTU8 can be set to PWM mode independently. Channels set to PWM mode can perform synchronous operation with each other or other channels set to any other mode.

There are two PWM modes as described below.

(1) PWM Mode 1

PWM waveforms are output from the MTIOCnA and MTIOCnC pins by pairing TGRA with TGRB and TGRC with TGRD. The levels specified by the TIOR.IOA[3:0] and IOC[3:0] bits are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the level specified by the TIOR.IOB[3:0] and IOD[3:0] bits are output at compare matches B and D (n = 0 to 4, 6, 7). The initial output value is set in TGRA or TGRC. If the values set in paired TGRs are identical, the output value does not change even when a compare match occurs.

In PWM mode 1, PWM waveforms in up to 12 phases can be output.

(2) PWM Mode 2

PWM waveform output is generated using one TGR as the period register and the others as duty registers. The level specified in TIOR is output at compare matches. Upon counter clearing by a period register compare match, the initial value set in TIOR is output from each pin. If the values set in the period and duty registers are identical, the output value does not change even when a compare match occurs.

Up to eight phases of PWM waveforms can be output by combining synchronous clearing of channels that cannot be set to PWM mode 2 as synchronous operation.

The correspondence between PWM output pins and registers is shown in Table 18.62.

Table 18.62 PWM output registers and output pins

Channel	Register	Output pins	
		PWM mode 1	PWM mode 2
MTU0	TGRA	MTIOC0A	MTIOC0A
	TGRB		MTIOC0B
	TGRC	MTIOC0C	MTIOC0C
	TGRD		MTIOC0D
MTU1	TGRA	MTIOC1A	MTIOC1A
	TGRB		MTIOC1B
MTU2	TGRA	MTIOC2A	MTIOC2A
	TGRB		MTIOC2B
MTU3	TGRA	MTIOC3A	Setting prohibited
	TGRB		
	TGRC	MTIOC3C	
	TGRD		
MTU4	TGRA	MTIOC4A	
	TGRB		
	TGRC	MTIOC4C	
	TGRD		
MTU6	TGRA	MTIOC6A	
	TGRB		
	TGRC	MTIOC6C	
	TGRD		
MTU7	TGRA	MTIOC7A	
	TGRB		
	TGRC	MTIOC7C	
	TGRD		

Note: In PWM mode 2, PWM waveform output is not possible for the TGR register in which the PWM period is set.

18.4.5.1 Example of PWM Mode Setting Procedure

Figure 18.26 shows an example of the PWM mode setting procedure.

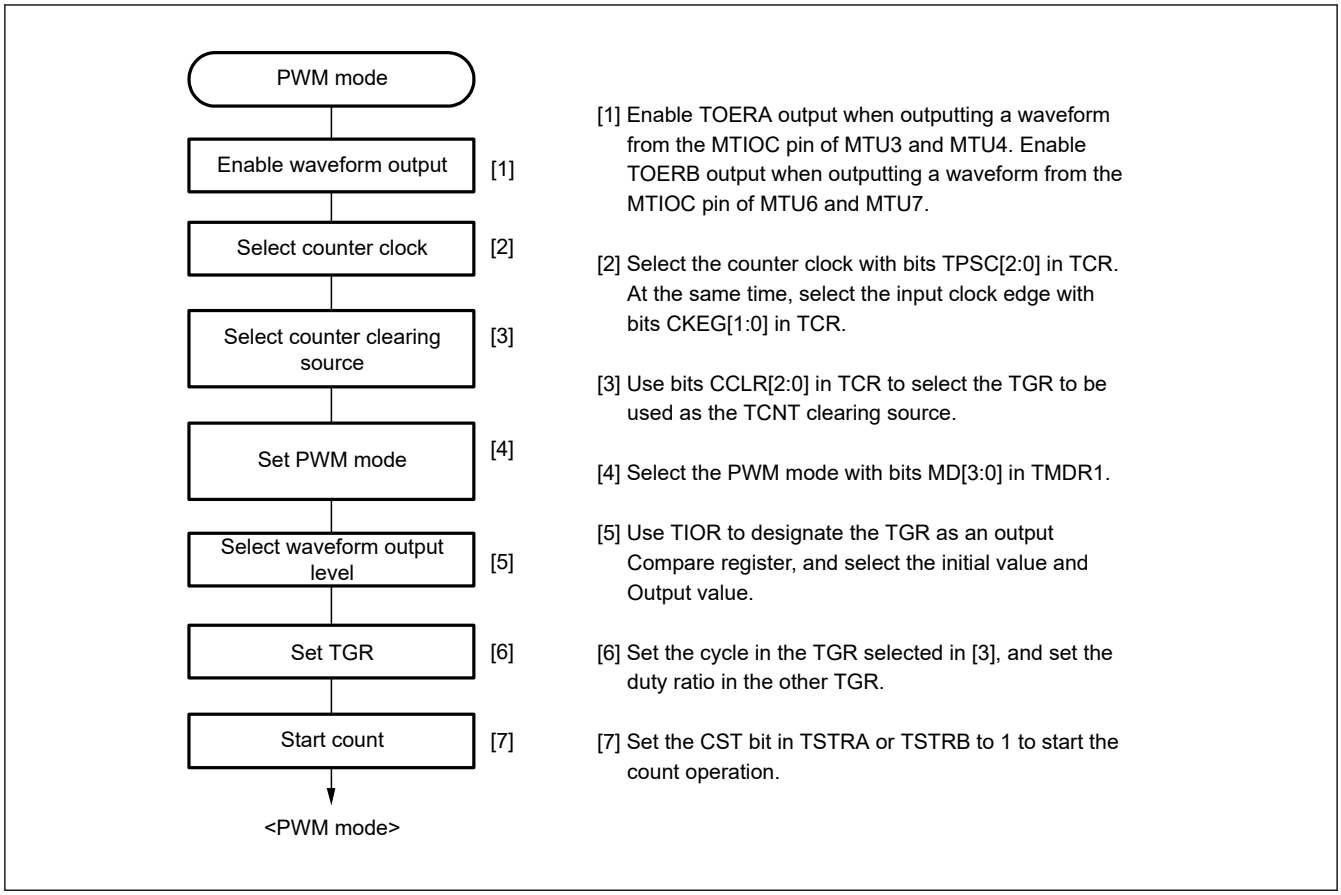


Figure 18.26 Example of PWM mode setting procedure

18.4.5.2 Examples of PWM Mode Operation

Figure 18.27 shows an example of operation in PWM mode 1.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set as the initial output value and output value for TGRA, and 1 is set as the output value for TGRB.

In this case, the value set in TGRA is used as the period, and the value set in TGRB is used as the duty ratio.

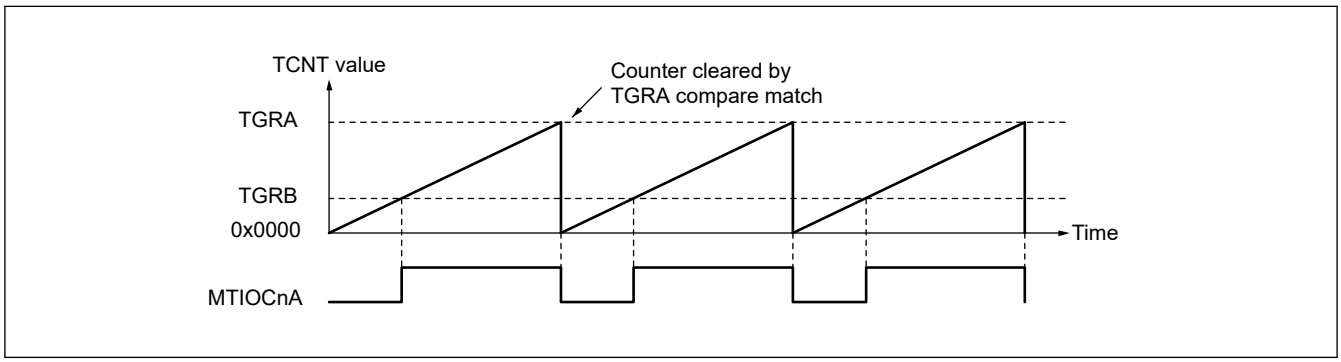


Figure 18.27 Example of PWM mode 1 operation (n = 0 to 4, 6, 7)

Figure 18.28 shows an example of operation in PWM mode 2.

In this example, synchronous operation is designated for MTU0 and MTU1, MTU1.TGRB compare match is set as the TCNT clearing source, and low is set as the initial output value and high as the output value for the other TGR registers (MTU0.TGRA to MTU0.TGRD and MTU1.TGRA), outputting 5-phase PWM waveforms.

In this case, the value set in MTU1.TGRB is used as the period, and the values set in the other TGRs are used as the duty ratio.

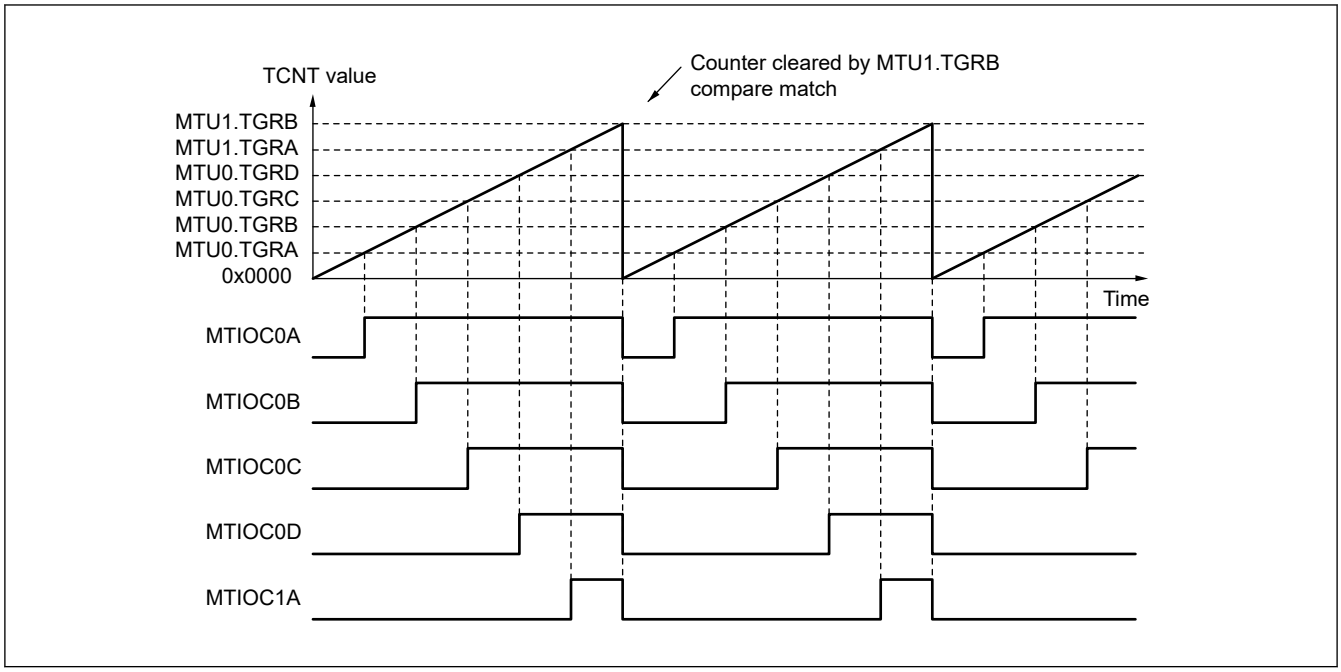


Figure 18.28 Example of PWM mode 2 operation

Figure 18.29 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode 1.

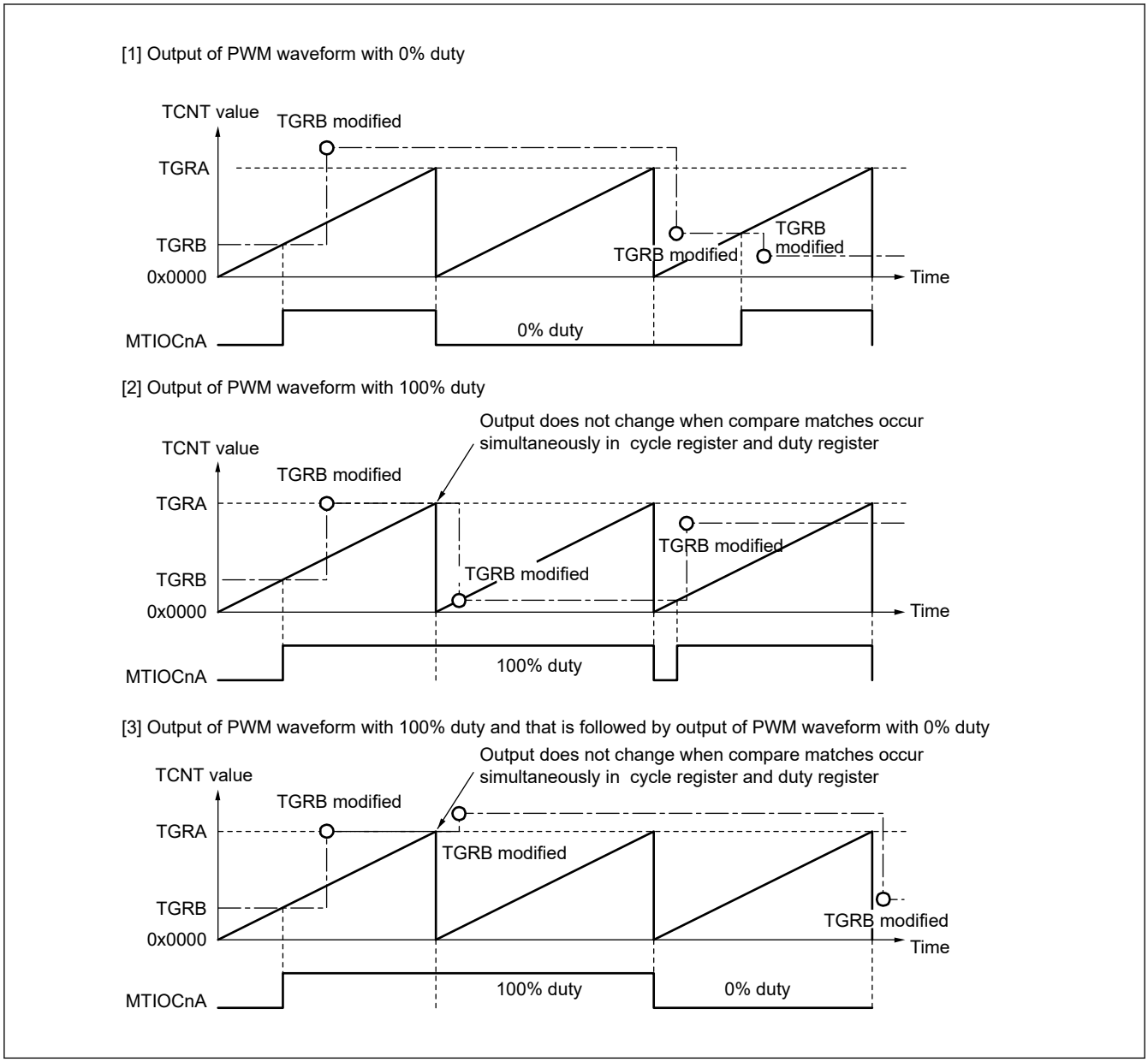


Figure 18.29 Examples of PWM mode operation (PWM waveform output with 0% duty and 100% duty) (n = 0 to 4, 6, 7)

18.4.6 Phase Counting Mode

There are two phase counting modes: 16-bit phase counting mode in which MTU1 and MTU2 operate independently, and cascade connection 32-bit phase counting mode in which MTU1 and MTU2 are cascaded.

In phase counting mode, the phase difference between two external input clocks is detected and the corresponding TCNT is incremented or decremented.

Two external clock input pins for each phase counting mode are not affected by the settings of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. The two external clock input pins used in 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2 can be selected by MTU1.TMDR3.PHCKSEL. In a phase counting mode other than 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2, MTCLKA and MTCLKB are selected for A-phase and B-phase, respectively. In phase counting mode, the external clock pins MTCLKA, MTCLKB, MTCLKC, and MTCLKD are used for two-phase encoder pulse input.

Table 18.63 lists the external clock input pins to be connected in each phase counting mode.

Table 18.63 Clock input pins in phase counting mode

Phase counting mode	TMDR3.PHCKSEL bit	External clock input pins	
		A-phase	B-phase
MTU1 16-bit phase counting mode	x (Don't care)	MTCLKA	MTCLKB
MTU2 16-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD
Cascade connection 32-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD

18.4.6.1 16-Bit Phase Counting Mode

When the MTU1.TMDR3.LWA is 0, 16-bit phase counting mode can be set individually for MTU1 and MTU2.

In 16-bit phase counting mode, the phase difference between two external input clocks is detected and the 16-bit counter TCNT of the corresponding channel is incremented or decremented.

When 16-bit phase counting mode is specified, an external clock is selected as the count clock and TCNT operates as an up-counter/down-counter regardless of the setting of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. However, the functions of TCR.CCLR[1:0], TIOR, TIER, and TGR are enabled, and input capture/compare match and interrupt functions can be used.

These external input pins can be used for two-phase encoder pulse input.

When an overflow occurs while TCNT is counting up and the corresponding TIER.TCIEV bit is 1, a TCIV interrupt is generated. When an underflow occurs while TCNT is counting down and the corresponding TIER.TCIEU bit is 1, a TCIU interrupt is generated.

The TSR.TCFD flag is the count direction flag. Read the TCFD flag to check whether TCNT is counting up and down.

18.4.6.1.1 Example of 16-Bit Phase Counting Mode Setting Procedure

Figure 18.30 shows an example of the 16-bit phase counting mode setting procedure.

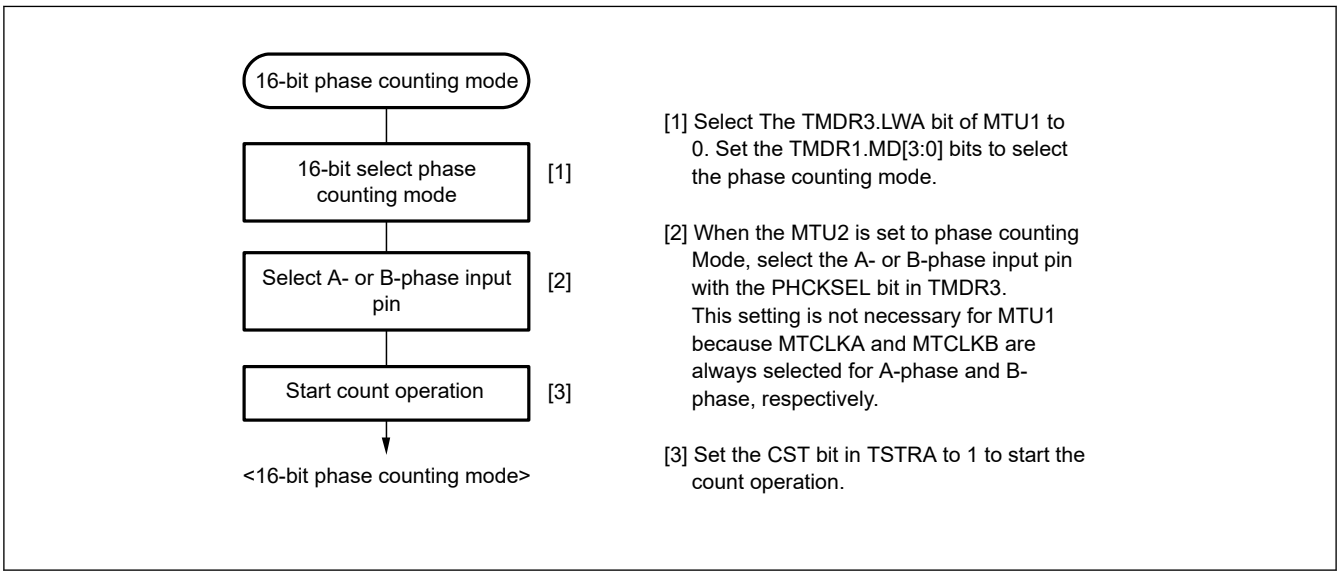


Figure 18.30 Example of 16-bit phase counting mode setting procedure

18.4.6.1.2 Examples of 16-Bit Phase Counting Mode Operation

In phase counting mode, TCNT is incremented or decremented according to the phase difference between two external clocks. There are five modes according to the count conditions. Each mode operates under the condition PHCKSEL = 1, which means the phase clock for MTU1 is input from MTCLKA or MTCLKB and that for MTU2 is input from MTCLKC or MTCLKD.

(1) Phase Counting Mode 1

Figure 18.31 shows an example of operation in phase counting mode 1, and Table 18.64 summarizes the TCNT upcounting and down-counting conditions.

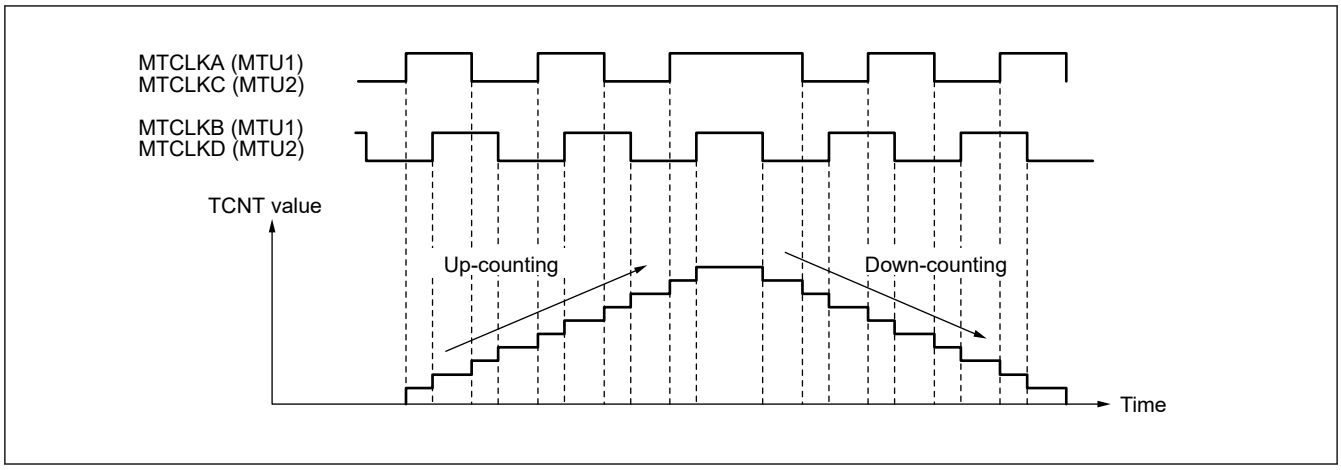


Figure 18.31 Example of operation in phase counting mode 1

Table 18.64 Up-counting and down-counting conditions in phase counting mode 1

: Rising edge

: Falling edge

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	
↓	Low	

(2) Phase Counting Mode 2

Figure 18.32 to Figure 18.34 show the examples of operation in phase counting mode 2 and Table 18.65 summarizes the TCNT up-counting and down-counting conditions.

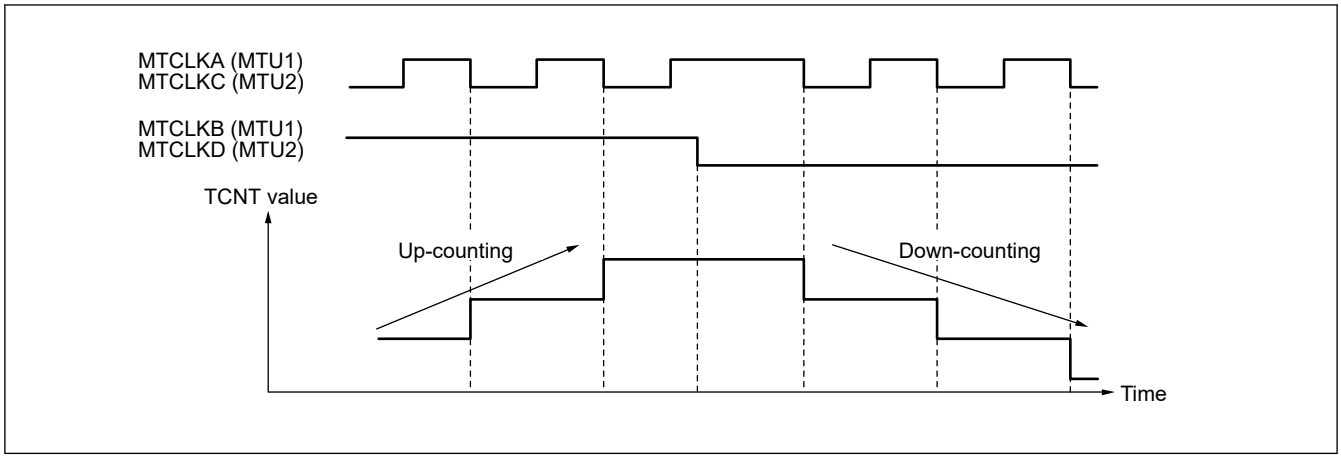


Figure 18.32 Example of operation in phase counting mode 2 (when MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

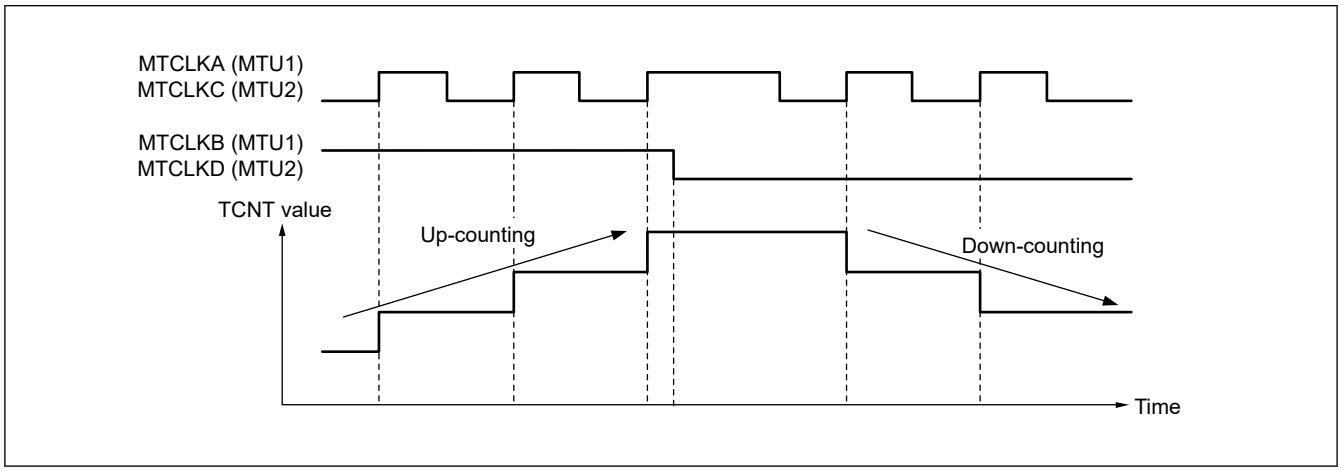


Figure 18.33 Example of operation in phase counting mode 2 (when MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

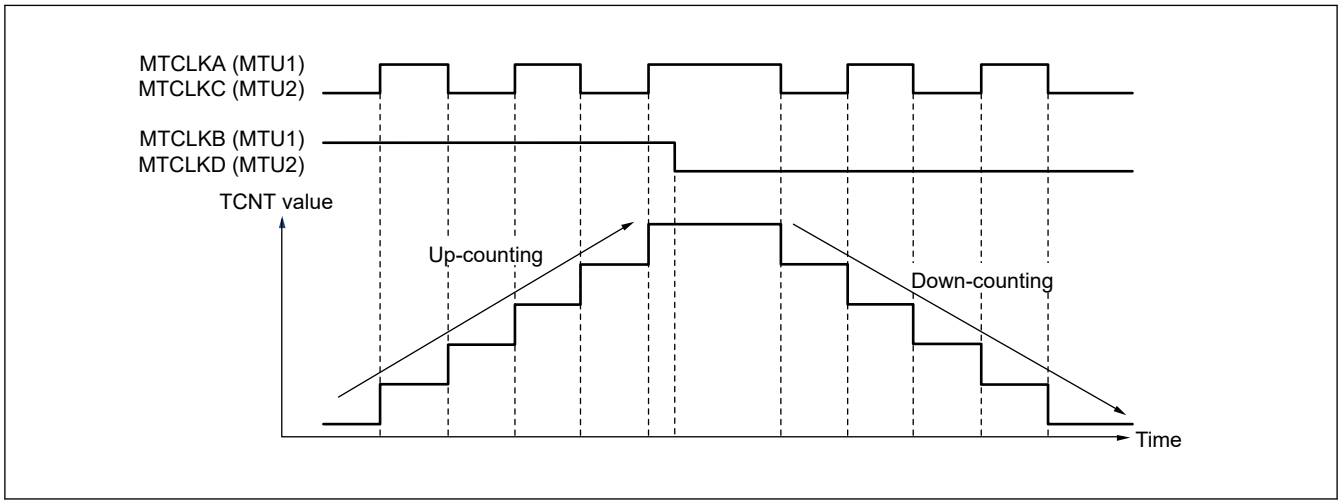


Figure 18.34 Example of operation in phase counting mode 2 (when MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 18.65 Up-counting and down-counting conditions in phase counting mode 2

: Rising edge

: Falling edge

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High		Not counted (Don't care)
	Low		
		Low	
		High	Up-counting
	High		Not counted (Don't care)
	Low		
		High	
		Low	Down-counting
01b	High		Not counted (Don't care)
	Low		
		Low	Down-counting
		High	Not counted (Don't care)
	High		
	Low		
		High	Up-counting
		Low	Not counted (Don't care)
1xb	High		Not counted (Don't care)
	Low		
		Low	Down-counting
		High	Up-counting
	High		Not counted (Don't care)
	Low		
		High	Up-counting
		Low	Down-counting

(3) Phase Counting Mode 3

Figure 18.35 to Figure 18.37 show the examples of operation in phase counting mode 3 and Table 18.66 summarizes the TCNT up-counting and down-counting conditions.

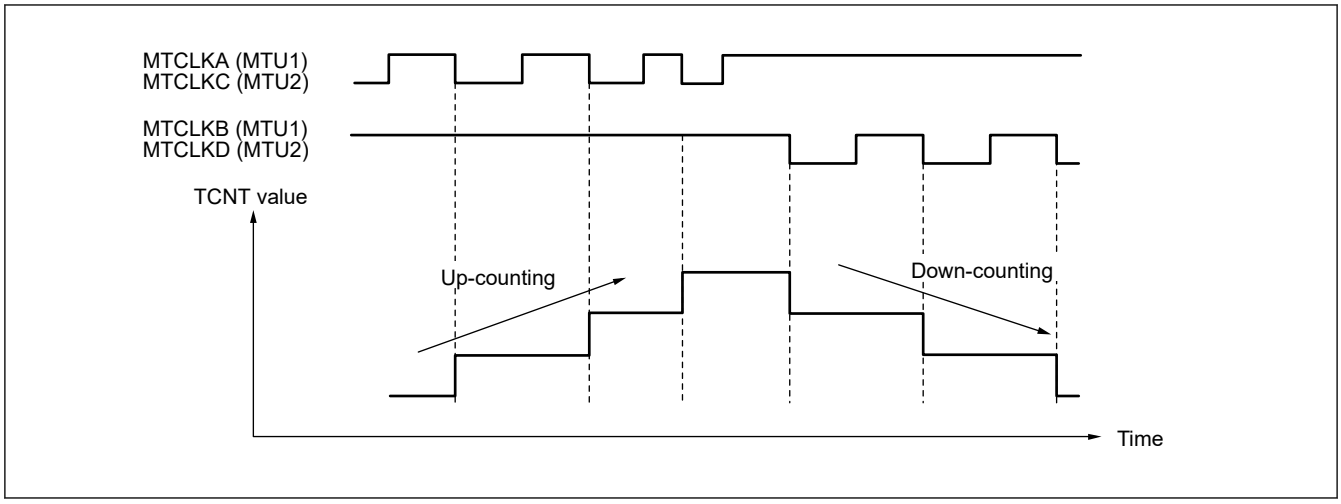


Figure 18.35 Example of operation in phase counting mode 3 (when MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

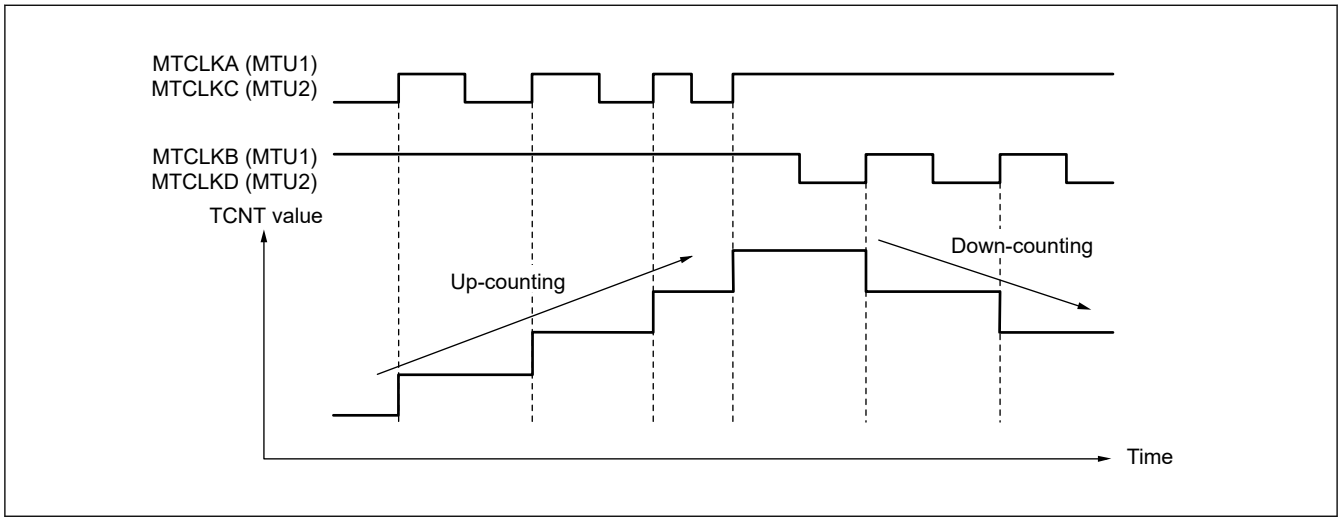


Figure 18.36 Example of operation in phase counting mode 3 (when MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

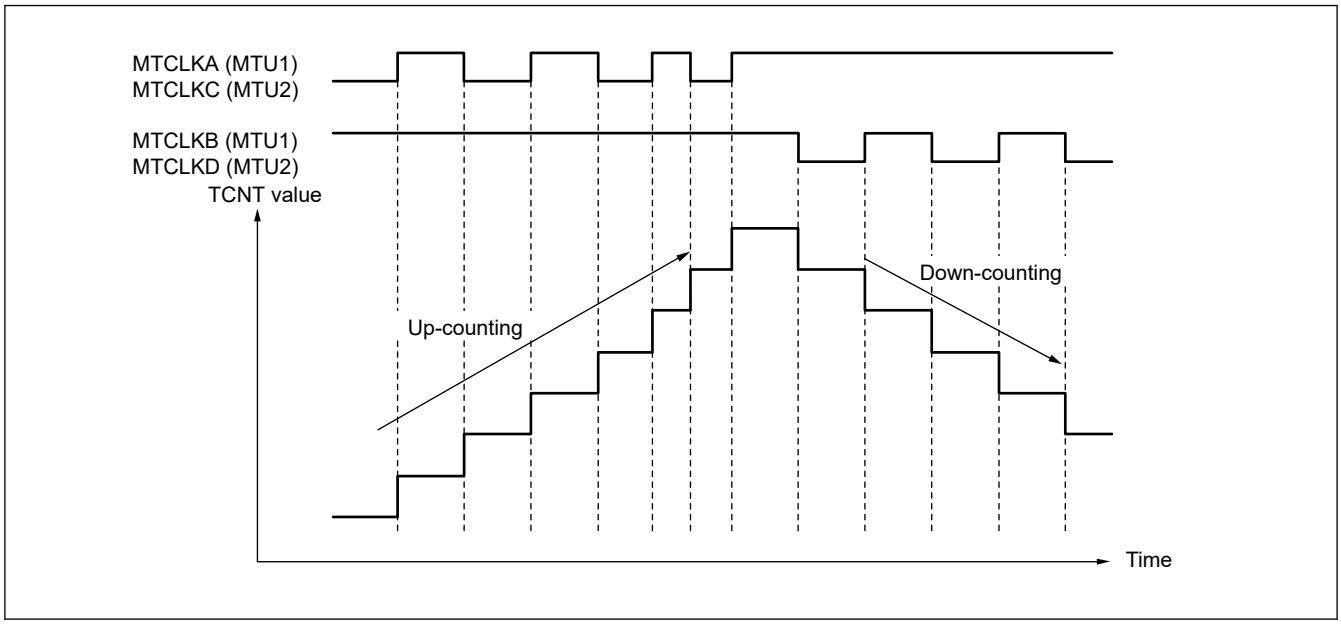


Figure 18.37 Example of operation in phase counting mode 3 (when MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 18.66 Up-counting and down-counting conditions in phase counting mode 3

: Rising edge

: Falling edge

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High		Not counted (Don't care)
	Low		
		Low	
		High	Up-counting
	High		Down-counting
	Low		Not counted (Don't care)
		High	
		Low	
01b	High		Down-counting
	Low		Not counted (Don't care)
		Low	
		High	
	High		
	Low		
		High	Up-counting
		Low	Not counted (Don't care)
1xb	High		Down-counting
	Low		Not counted (Don't care)
		Low	
		High	Up-counting
	High		Down-counting
	Low		Not counted (Don't care)
		High	Up-counting
		Low	Not counted (Don't care)

(4) Phase Counting Mode 4

Figure 18.38 shows an example of operation in phase counting mode 4, and Table 18.67 summarizes the TCNT up-counting and down-counting conditions.

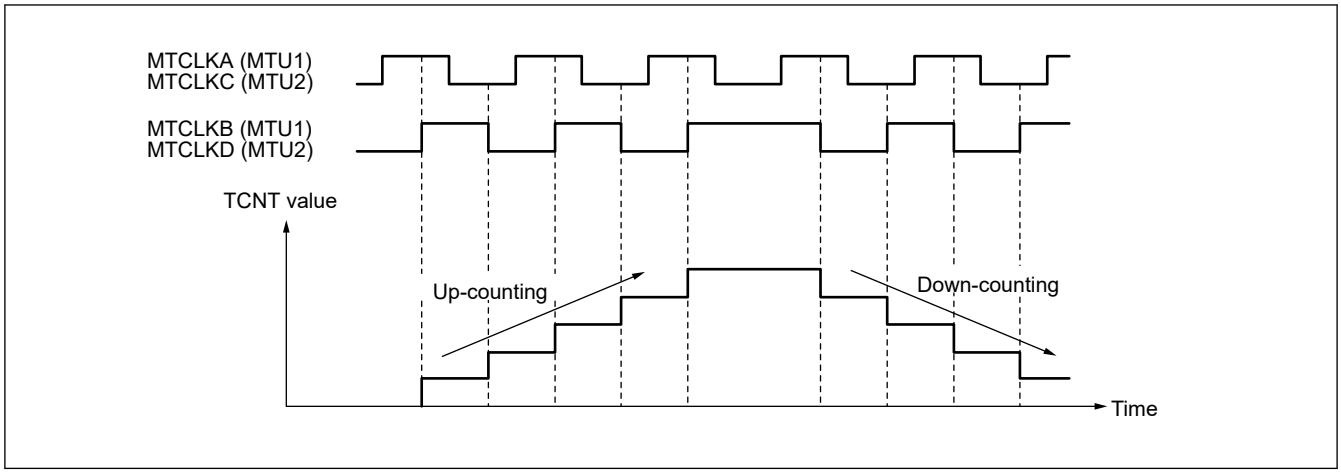


Figure 18.38 Example of operation in phase counting mode 4

Table 18.67 Up-counting and down-counting conditions in phase counting mode 4

: Rising edge

: Falling edge

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	Not counted (Don't care)
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	Not counted (Don't care)
↓	Low	

(5) Phase Counting Mode 5

Figure 18.39 and Figure 18.40 show the examples of operation in phase counting mode 5 and Table 18.68 summarizes the TCNT up-counting and down-counting conditions.

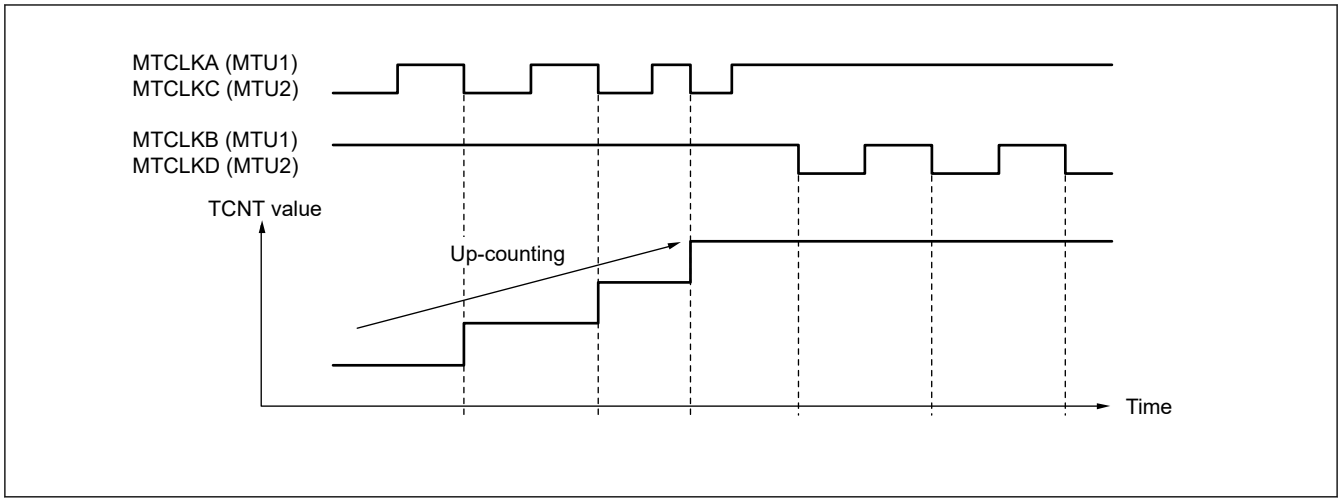


Figure 18.39 Example of operation in phase counting mode 5 (when MTUn.TCR2.PCB[1:0] = 0xb (n = 1, 2))

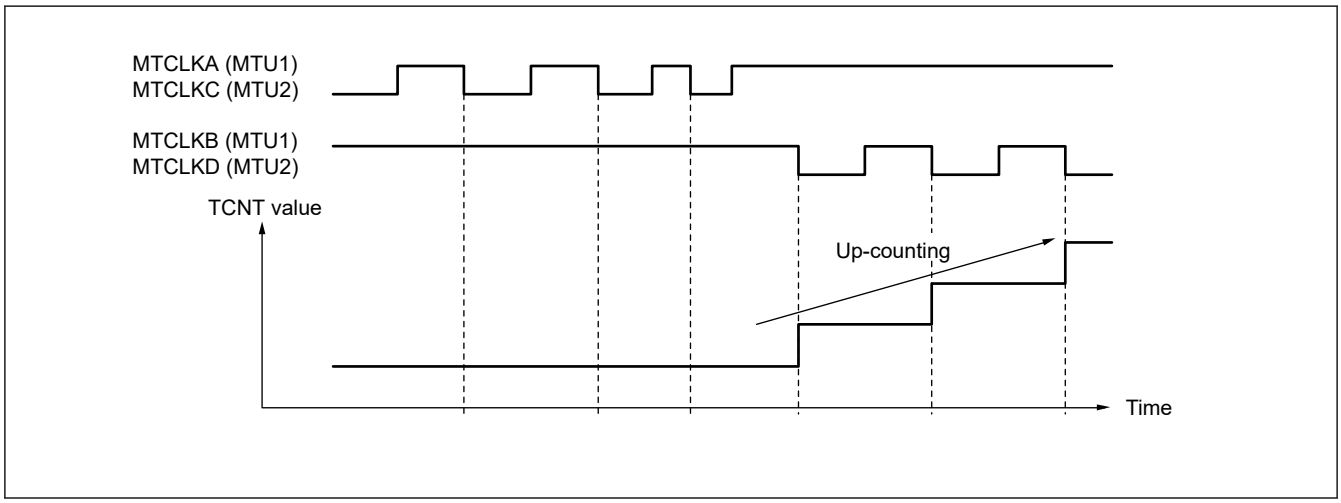


Figure 18.40 Example of operation in phase counting mode 5 (when MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 18.68 Up-counting and down-counting conditions in phase counting mode 5 (1 of 2)

: Rising edge

: Falling edge

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
0xb	High		Not counted (Don't care)
	Low		
		Low	
		High	Up-counting
	High		Not counted (Don't care)
	Low		
		High	
		Low	Up-counting

Table 18.68 Up-counting and down-counting conditions in phase counting mode 5 (2 of 2)

: Rising edge

: Falling edge

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
1xb	High		Not counted (Don't care)
	Low		Up-counting
		Low	Not counted (Don't care)
		High	
	High		Up-counting
	Low		Not counted (Don't care)
		High	
		Low	

18.4.6.1.3 16-Bit Phase Counting Mode Application Example

Figure 18.41 shows an example in which MTU1 is in phase counting mode, and MTU1 is coupled with MTU0 to input 2-phase encoder pulses of a servo motor in order to detect position or speed.

MTU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to MTCLKA and MTCLKB.

In MTU0, MTU0.TGRC compare match is specified as the TCNT clearing source and MTU0.TGRA and MTU0.TGRC are used for the compare match function and are set with the speed control period and position control period.

MTU0.TGRB is used for input capture, with MTU0.TGRB and MTU0.TGRD operating in buffer mode. The MTU1 count clock is designated as the MTU0.TGRB input capture source, and the widths of 2-phase encoder 4-multiplication pulses are detected.

MTU1.TGRA and MTU1.TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and MTU0.TGRC compare matches in MTU0 are selected as the input capture sources to store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.

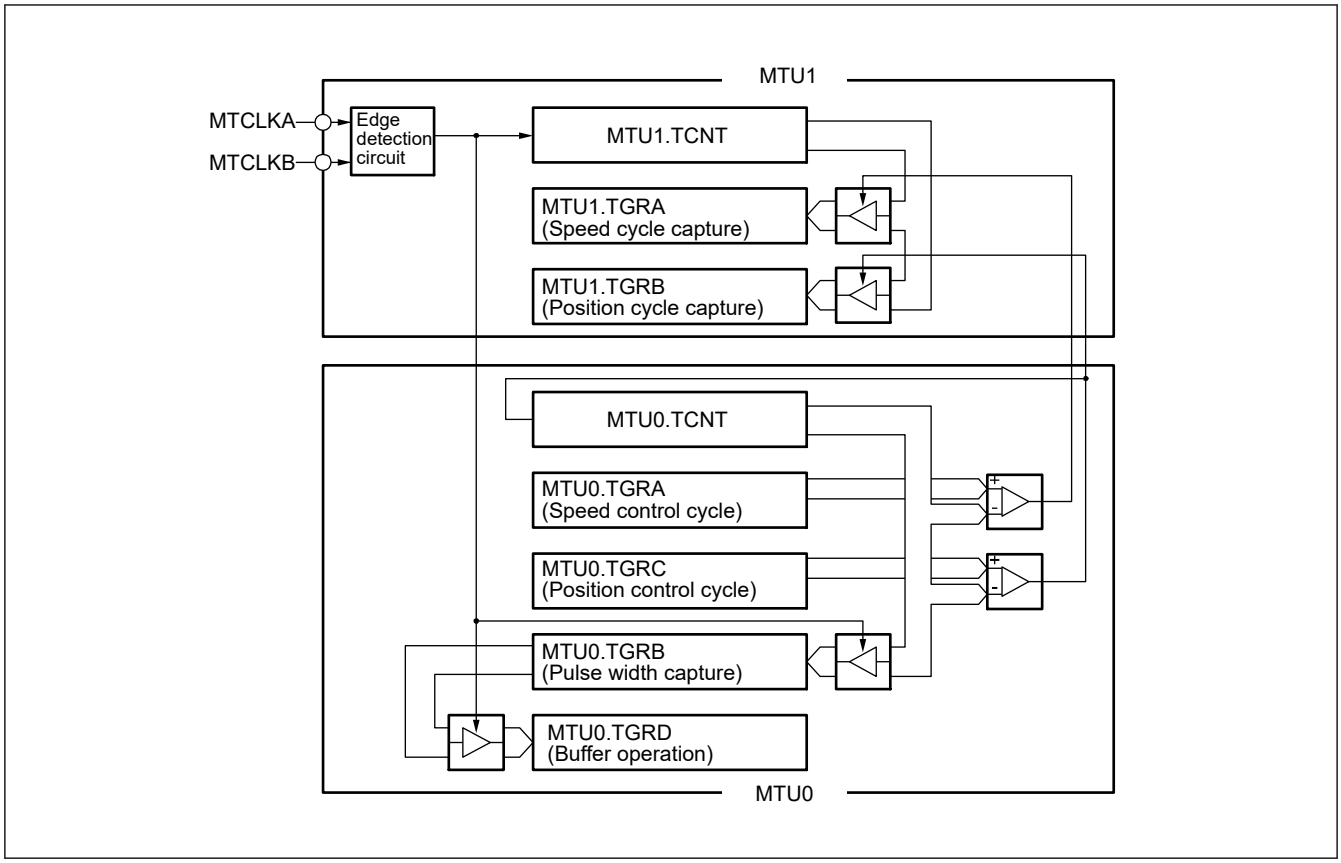


Figure 18.41 16-bit phase counting mode application example

18.4.6.2 Cascade Connection 32-Bit Phase Counting Mode

When MTU1 is set to phase counting mode by setting MTU1.TMDR3.LWA = 1, MTU1 and MTU2 are connected to operate in cascade connection 32-bit phase counting mode as shown in Figure 18.42. When this mode is used, the TCR, TCR2, TIOR, TIER, TGR, and TSR registers are controlled by MTU1 and the settings of MTU2 are disabled. See Figure 18.43 for the procedure for setting cascade connection 32-bit phase counting mode.

In this mode, three-phase (A, B, and Z) signals can be input. As an encoder pulse signal, the external input phase clocks MTCLKA and MTCLKB or MTCLKC and MTCLKD can be selected for A-phase and B-phase, and MTIOC1A can be selected for Z-phase, respectively. See Table 18.63 for selecting external clock input of A-phase and B-phase. A counter event is generated using an A-phase or B-phase pulse and counted by the 32-bit counter MTU1.TCNTLW.

An input capture can also be generated using a Z-phase signal; thus angular velocity can be measured using the captured value in the general register.

Furthermore, MTU8 can be used as a channel for measuring a control period interval, and a compare match signal can be output at a control period interval to the MTU1 and MTU2, which operate in cascade connection 32-bit phase counting mode. That is, a compare match signal of MTU8 is used as a capture signal of MTU1 and MTU2, and the number of A-phase and B-phase pulses for a control period can be measured.

When MTU0 or MTU5 is specified as the channel for measuring a Z-phase signal pulse, the compare match signal of the MTU8.TGRC register can be output as a capture signal or clear signal to MTU0 or MTU5, thus the Z-phase count at a control period interval can be measured.

In addition, a counter event signal of combined MTU1 and MTU2 can be used as a capture signal of the MTU8.TGRD register, and measurement can be performed including the intervals of A, B, or both phase pulses. In this case, the MTU8.TGRD register should be set to buffer operation.

See section 18.4.4. Cascaded Operation, for details on the cascade connection function for connecting MTU1 and MTU2 in a mode other than cascade connection 32-bit phase counting mode.

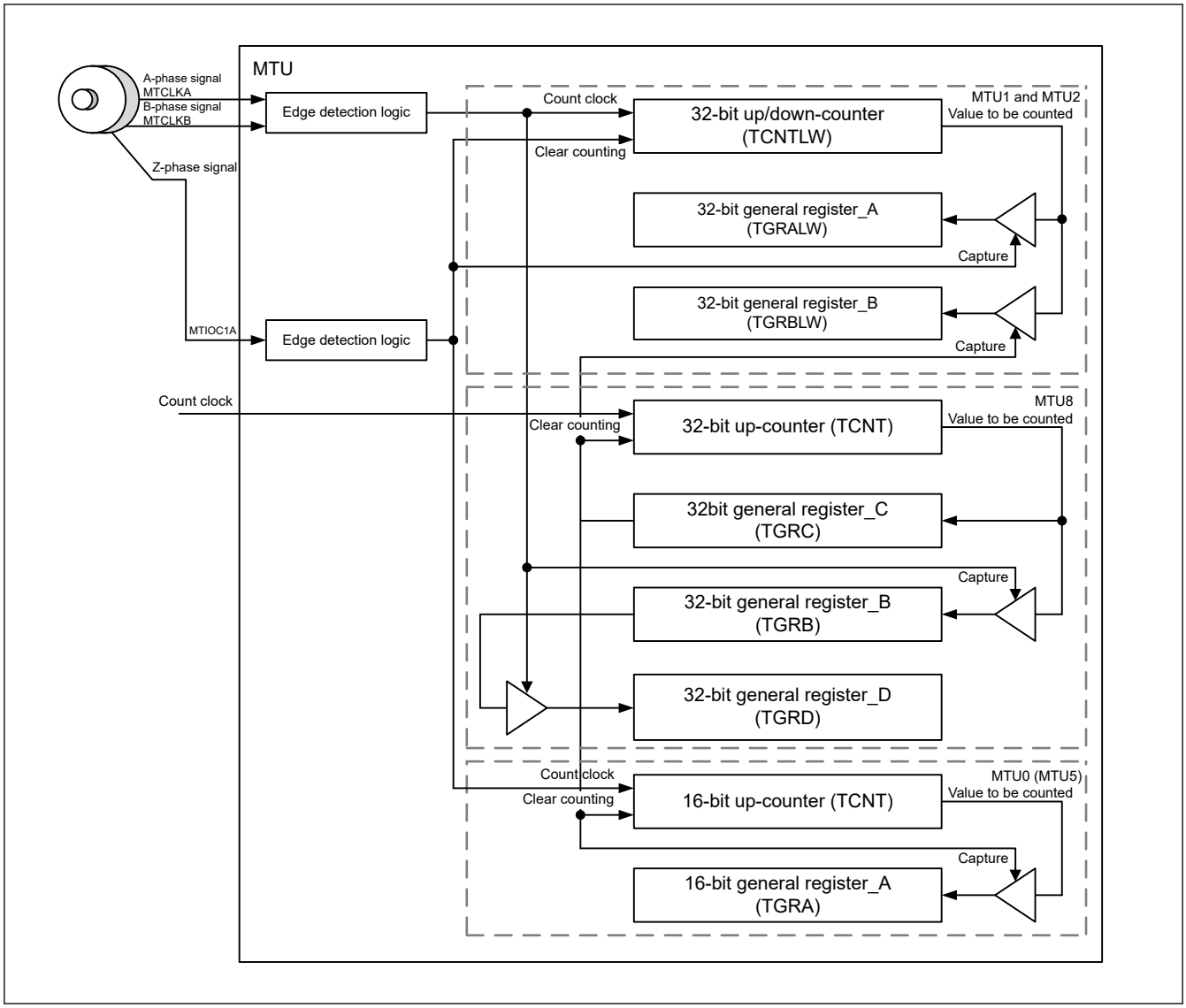
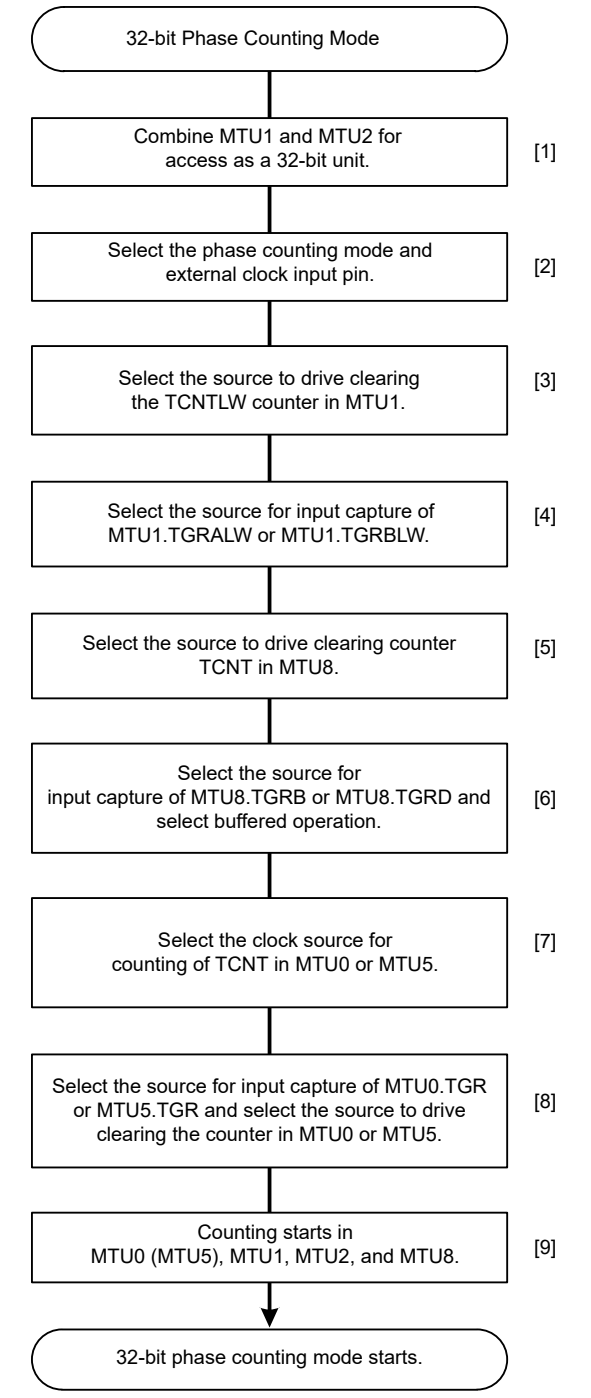


Figure 18.42 Block diagram for operation in cascade connection 32-bit phase counting mode

18.4.6.2.1 Example of Setting Cascade Connection 32-Bit Phase Counting Mode

Figure 18.43 shows an example of the procedure for setting cascade connection 32-bit phase counting mode.



[1] Set the LWA bit in MTU1.TMDR3 to 1 and combine MTU1 and MTU2 for access as a 32-bit unit.

[2] Select phase counting mode with the MD[3:0] bits in MTU1.TMDR1 and select the external clock input pin with the PHCKSEL bits in MTU1.TMDR3.

[3] Set the CCLR[2:0] bits in MTU1.TCR to 001b to select input capture of MTU1.TGRALW as the source of the signal to drive clearing counter TCNTLW in MTU1.

[4] Set the IOA[3:0] bits in MTU1.TIOR to select the effective edge (rising, falling or both) of the signal produced by input capture of TGRALW in MTU1. After that, set the IOB[3:0] bits in MTU1.TIOR to 1110b to select the compare match signal of MTU8.TGRC as the source to drive input capture of MTU1.TGRBLW.

[5] Set the CCLR[2:0] bits in MTU8.TCR to 101b to select the compare match signal of MTU8.TGRC as the source to drive clearing counter TCNT in MTU8.

[6] Select the source for input capture of MTU8.TGRB as the clock source for counting of TCNTLW in MTU1 by setting the IOB[3:0] bits in MTU8.TIORH to 1110b. At the same time, select buffered operation for TGRB and TGRD by setting the BFB bit in MTU8.TMDR1 to 1.

[7] Set the TPSC2[2:0] bits in MTU0.TCR2 to 111b to select the MTIOC1A pin as the clock source for counting of TCNT in MTU0.
(When using TCNT in MTU5, set the TPSC2[2:0] bits in MTU5.TCR2 to 111b).

[8] Select the compare match signal of MTU8.TGRC as the source to drive input capture of MTU0.TGRA by setting the IOA[3:0] bits in MTU0.TIORH to 1110b.
(When using TCNT in MTU5, set the TIORU, TIORV, or TIORW bit in MTU5 is set to 10100b).
After that, select input capture of MTU0.TGRA as the source to drive clearing counter TCNT in MTU0 by setting the CCLR[2:0] bits in MTU0.TCR to 001b.

[9] Start counting in MTU0, MTU1, MTU2, and MTU8 by setting the TSTRA bit to 00001111b.
(When using TCNT in MTU5, set the counter corresponding to TSTR in MTU5 to 1).

Note: When the LWA bit in MTU1.TMDR3 is set to 1, the TCR, TCR2, TMDR1, and TIOR registers in MTU1 are effective in controlling cascaded operation. In this case, the settings of these registers in MTU2 are ignored.

Figure 18.43 Procedure for setting cascade connection 32-bit phase counting mode

18.4.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, six phases of positive and negative PWM waveforms (12 phases in total) that share a common wave transition point can be output by combining MTU3 and MTU4 and MTU6 and MTU7.

When set for reset-synchronized PWM mode, the MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D pins function as PWM output pins and timer counters 3 and 6 (MTU3.TCNT and MTU6.TCNT) functions as an up-counter.

Table 18.69 shows the PWM output pins used. Table 18.70 shows the settings of the registers.

Table 18.69 Output pins for reset-synchronized PWM mode

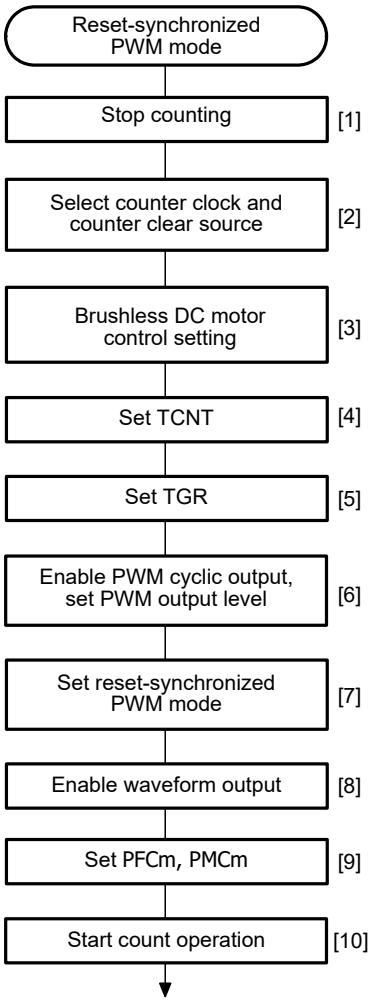
Channel	Output pin	Description
MTU3	MTIOC3B	PWM output pin 1
	MTIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)
MTU6	MTIOC6B	PWM output pin 4
	MTIOC6D	PWM output pin 4' (negative-phase waveform of PWM output 4)
MTU7	MTIOC7A	PWM output pin 5
	MTIOC7C	PWM output pin 5' (negative-phase waveform of PWM output 5)
	MTIOC7B	PWM output pin 6
	MTIOC7D	PWM output pin 6' (negative-phase waveform of PWM output 6)

Table 18.70 Register settings for reset-synchronized PWM mode

Register	Setting
MTU3.TCNT	Initial setting (0x0000)
MTU4.TCNT	Initial setting (0x0000)
MTU3.TGRA	Set the count period for MTU3.TCNT
MTU3.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC3B and MTIOC3D pins
MTU4.TGRA	Set the transition point of the PWM waveform to be output from the MTIOC4A and MTIOC4C pins
MTU4.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC4B and MTIOC4D pins
MTU6.TCNT	Initial setting (0x0000)
MTU7.TCNT	Initial setting (0x0000)
MTU6.TGRA	Set the count period for MTU6.TCNT
MTU6.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC6B and MTIOC6D pins
MTU7.TGRA	Set the transition point of the PWM waveform to be output from the MTIOC7A and MTIOC7C pins
MTU7.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC7B and MTIOC7D pins

18.4.7.1 Example of Procedure for Setting Reset-Synchronized PWM Mode

Figure 18.44 shows an example of procedure for setting the reset-synchronized PWM mode.



[1] Clear the CST3 and CST4 bits in TSTRA to 0 to stop the TCNT operation. Specify the reset-synchronized PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set bits TPSC[2:0] and CKEG[1:0] in MTU3.TCR to select the counter clock and clock edge for MTU3. Set bits CCLR[2:0] to select TGRA compare match as a counter clear source.

[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCRA) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Set MTU3.TCNT and MTU4.TCNT to 0x0000.

[5] MTU3.TGRA is the cycle register. Set the waveform cycle value in MTU3.TGRA. Set the transition timing of the PWM output waveforms in MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB. Set times within the compare match range of MTU3.TCNT.

X = MTU3.TGRA (X: Set value)

[6] Enable or disable toggle output synchronized with the PMW cycle using bit PSYE in TOCR1A, and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBRA as a buffer for TOCR2A, see Figure 17.4.

[7] Set bits MD[3:0] in MTU3.TMDR1 to 1000b to select the reset-synchronized PWM mode. In addition, specify the buffer operation of the MTU3.TGRA (BFA = 1) and MTU3.TGRB (BFB = 1) when necessary. Set the MTU4.TMDR1 to the initial value (normal operation; BFA = 0 and BFB = 0).

[8] Set the enabling/disabling of the PWM waveform output pin in TOERA.

[9] Set the Port Function Control register (PFCm) and the Port Mode Control registers (PMCm) of the I/O ports.

[10] Set the CST3 bit in the TSTRA to 1 to start the count operation.

Note: The output waveform starts to toggle at the point of MTU3.TCNT = MTU3.TGRA = X by setting X = MTU3.TGRA for example, period = duty ratio.

Figure 18.44 Procedure for selecting reset-synchronized PWM mode

18.4.7.2 Example of Reset-Synchronized PWM Mode Operation

Figure 18.45 shows an example of operation in the reset-synchronized PWM mode.

MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) operate as up-counters. The counters are cleared when a compare match occurs between MTU3.TCNT (MTU6.TCNT) and MTU3.TGRA (MTU6.TGRA), and then begin incrementing from 0x0000. The output from the PWM pins toggles every time a compare match occurs in MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB) and the counters are cleared.

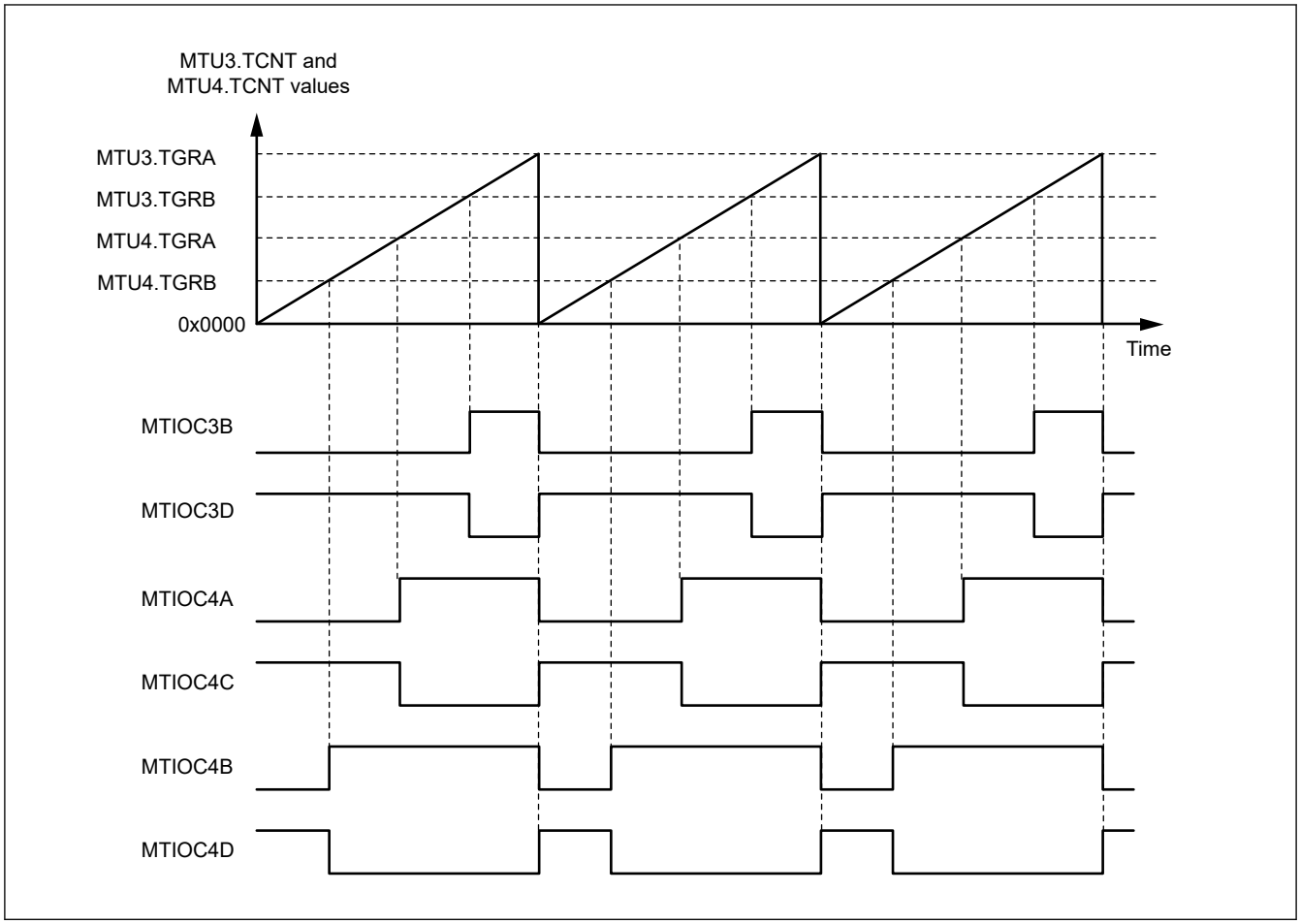


Figure 18.45 Example of reset-synchronized PWM mode operation (when OLSN = 1 and OLSP = 1 in MTU3.TOCR1A and MTU4.TOCR1A)

18.4.8 Complementary PWM Mode

In complementary PWM mode, dead time can be set for PWM waveforms to be output. The dead time is the period during which the upper and lower arm transistors are set to the inactive level in order to prevent short-circuiting of the arms.

Six positive-phase and six negative-phase PWM waveforms (12 phases in total) with dead time can be output by combining MTU3/ MTU4 and MTU6/MTU7. PWM waveforms without dead time can also be output.

In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D pins function as PWM output pins, and the MTIOC3A and MTIOC6A pins can be set for toggle output synchronized with the PWM period.

MTU3.TCNT, MTU4.TCNT, MTU6.TCNT, and MTU7.TCNT function as up/down-counters.

Table 18.71 shows the PWM output pins used. Table 18.72 shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 18.71 Output pins for complementary PWM mode (1 of 2)

Channel	Output pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC3B	PWM output pin 1
	MTIOC3C	I/O port*1
	MTIOC3D	PWM output pin 1' (non-overlapping negative-phase waveform of PWM output 1 — PWM output without non-overlapping interval is also available)

Table 18.71 Output pins for complementary PWM mode (2 of 2)

Channel	Output pin	Description
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM output 2 — PWM output without non-overlapping interval is also available)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM output 3 — PWM output without non-overlapping interval is also available)
MTU6	MTIOC6A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC6B	PWM output pin 4
	MTIOC6C	I/O port*1
	MTIOC6D	PWM output pin 4' (non-overlapping negative-phase waveform of PWM output 4 — PWM output without non-overlapping interval is also available)
MTU7	MTIOC7A	PWM output pin 5
	MTIOC7C	PWM output pin 5' (non-overlapping negative-phase waveform of PWM output 5 — PWM output without non-overlapping interval is also available)
	MTIOC7B	PWM output pin 6
	MTIOC7D	PWM output pin 6' (non-overlapping negative-phase waveform of PWM output 6 — PWM output without non-overlapping interval is also available)

Note 1. Avoid setting the MTIOC3C and MTIOC6C pins as timer I/O pins in complementary PWM mode.

Table 18.72 Register settings for complementary PWM mode (1 of 3)

Channel	Counter/ register	Description	Read/write from CPU
MTU3	TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWERA setting*1
	TGRA	Set MTU3.TCNT upper limit value (1/2 carrier period + dead time)	Maskable by TRWERA setting*1
	TGRB	PWM output 1 compare register	Maskable by TRWERA setting*1
	TGRC	MTU3.TGRA buffer register	Readable/writable
	TGRD	PWM output 1/MTU3.TGRB buffer register	Readable/writable
	TGRE	MTU3.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU4	TCNT	Starts up-counting after being initialized to 0x0000	Maskable by TRWERA setting*1
	TGRA	PWM output 2 compare register	Maskable by TRWERA setting*1
	TGRB	PWM output 3 compare register	Maskable by TRWERA setting*1
	TGRC	PWM output 2/MTU4.TGRA buffer register	Readable/writable
	TGRD	PWM output 3/MTU4.TGRB buffer register	Readable/writable
	TGRE	MTU4.TGRA buffer register B (when double buffer function is used)	Readable/writable
	TGRF	MTU4.TGRB buffer register B (when double buffer function is used)	Readable/writable

Table 18.72 Register settings for complementary PWM mode (2 of 3)

Channel	Counter/ register	Description	Read/write from CPU
MTU6	TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWERB setting*2
	TGRA	Set MTU6.TCNT upper limit value (1/2 carrier period + dead time)	Maskable by TRWERB setting*2
	TGRB	PWM output 4 compare register	Maskable by TRWERB setting*2
	TGRC	MTU6.TGRA buffer register	Readable/writable
	TGRD	PWM output 4/MTU6.TGRB buffer register	Readable/writable
	TGRE	MTU6.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU7	TCNT	Starts up-counting after being initialized to 0x0000	Maskable by TRWERB setting*2
	TGRA	PWM output 5 compare register	Maskable by TRWERB setting*2
	TGRB	PWM output 6 compare register	Maskable by TRWERB setting*2
	TGRC	PWM output 5/MTU7.TGRA buffer register	Readable/writable
	TGRD	PWM output 6/MTU7.TGRB buffer register	Readable/writable
	TGRE	MTU7.TGRA buffer register B (when double buffer function is used)	Readable/writable
	TGRF	MTU7.TGRB buffer register B (when double buffer function is used)	Readable/writable
Timer dead time data register A (TDDRA)		Set MTU4.TCNT and MTU3.TCNT offset value (dead time value)	Maskable by TRWERA setting*1
Timer dead time data register B (TDDRB)		Set MTU7.TCNT and MTU6.TCNT offset value (dead time value)	Maskable by TRWERB setting*2
Timer cycle data register A (TCDRA)		Set MTU4.TCNT upper limit value (1/2 carrier period)	Maskable by TRWERA setting*1
Timer cycle data register B (TCDRB)		Set MTU7.TCNT upper limit value (1/2 carrier period)	Maskable by TRWERB setting*2
Timer cycle buffer register A (TCBRA)		TCDRA buffer register	Readable/writable
Timer cycle buffer register B (TCBRB)		TCDRB buffer register	Readable/writable
Timer subcounter A (TCNTSA)		Subcounter A for dead time generation	Read-only
Timer subcounter B (TCNTSB)		Subcounter B for dead time generation	Read-only
Temporary register 1A (TEMP1A)		PWM output 1/MTU3.TGRB temporary register A	Not readable/writable
Temporary register 1B (TEMP1B)		PWM output 1/MTU3.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 2A (TEMP2A)		PWM output 2/MTU4.TGRA temporary register A	Not readable/writable
Temporary register 2B (TEMP2B)		PWM output 2/MTU4.TGRA temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 3A (TEMP3A)		PWM output 3/MTU4.TGRB temporary register A	Not readable/writable
Temporary register 3B (TEMP3B)		PWM output 3/MTU4.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 4A (TEMP4A)		PWM output 4/MTU6.TGRB temporary register A	Not readable/writable

Table 18.72 Register settings for complementary PWM mode (3 of 3)

Channel	Counter/ register	Description	Read/write from CPU
Temporary register 4B (TEMP4B)		PWM output 4/MTU6.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 5A (TEMP5A)		PWM output 5/MTU7.TGRA temporary register A	Not readable/writable
Temporary register 5B (TEMP5B)		PWM output 5/MTU7.TGRA temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 6A (TEMP6A)		PWM output 6/MTU7.TGRB temporary register A	Not readable/writable
Temporary register 6B (TEMP6B)		PWM output 6/MTU7.TGRB temporary register B (when double buffer function is used)	Not readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).
Note 2. Access can be enabled or disabled according to the setting in TRWERB (timer read/write enable register B).

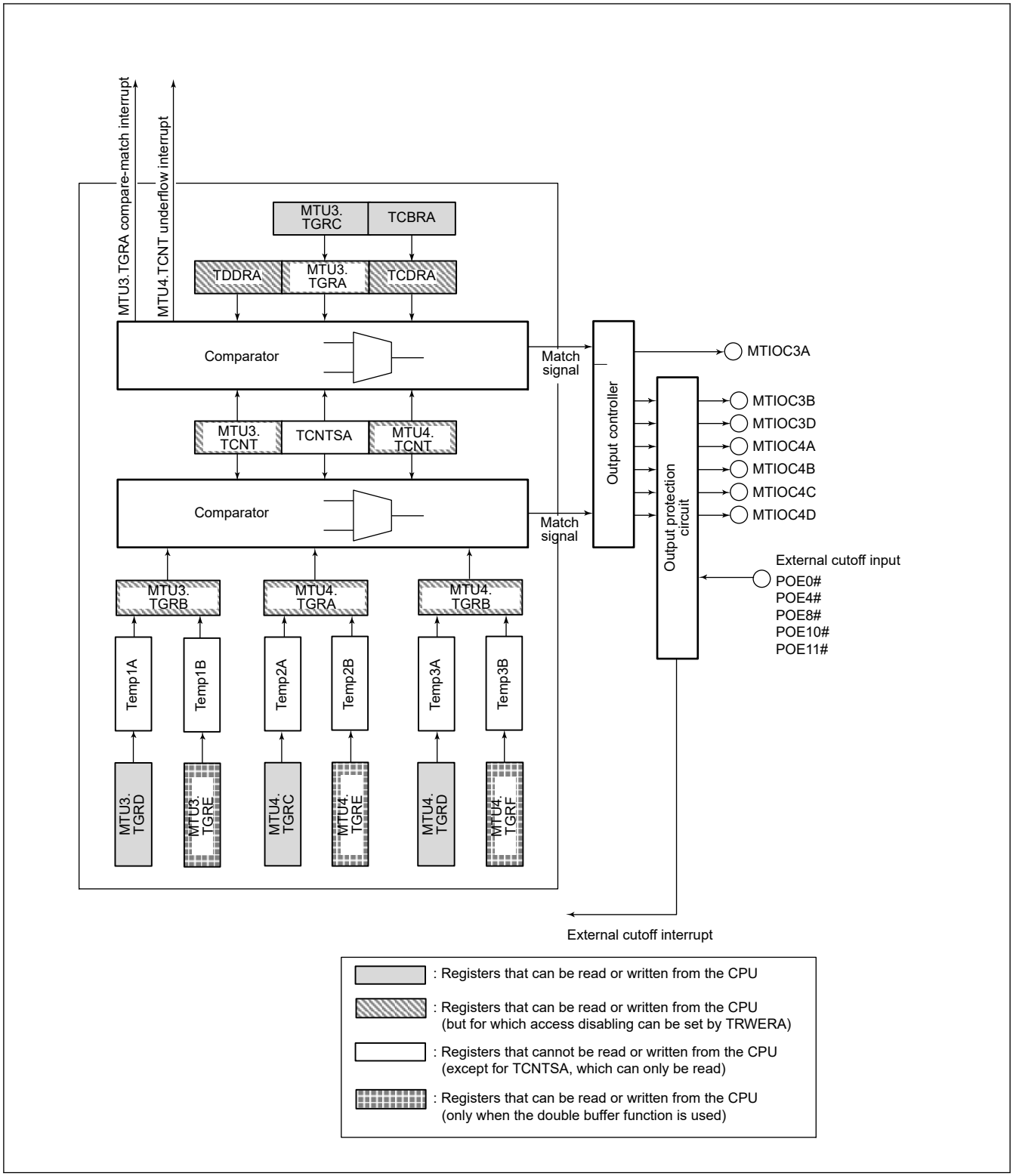


Figure 18.46 Block diagram of MTU3 and MTU4 in complementary PWM mode

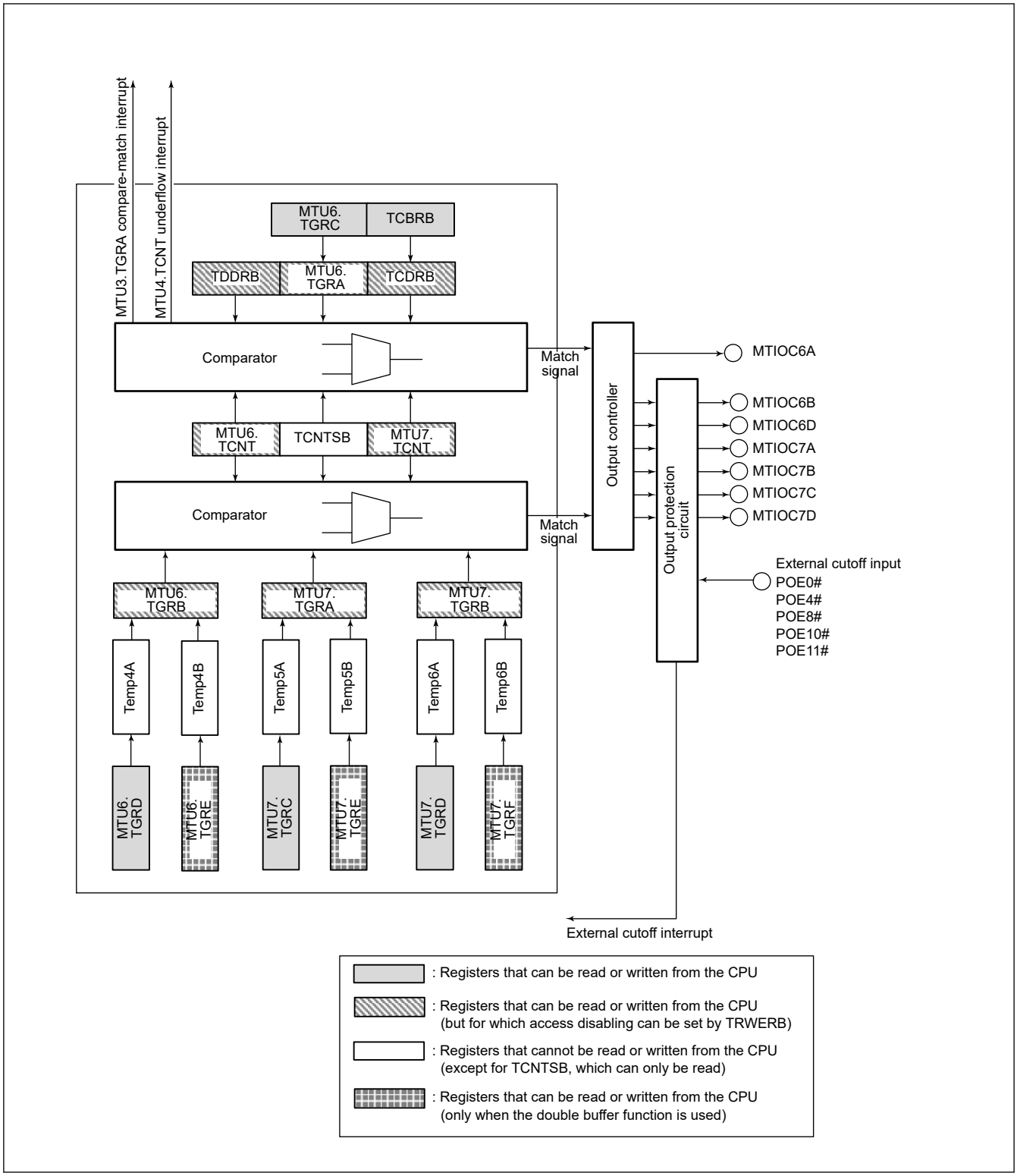
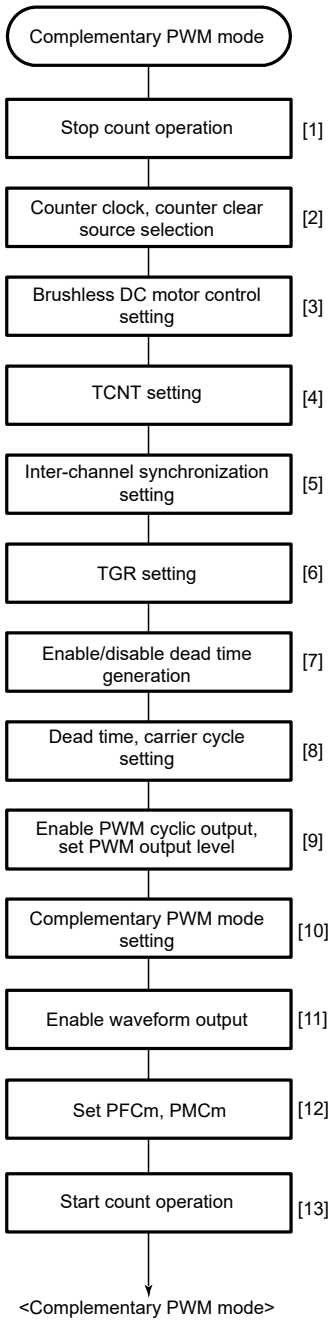


Figure 18.47 Block diagram of MTU6 and MTU7 in complementary PWM mode

18.4.8.1 Example of Complementary PWM Mode Setting Procedure

Figure 18.48 shows an example of the complementary PWM mode setting procedure.



[1] Clear bits CST3 and CST4 in the Timer Start register (TSTRA) to 0 to stop timer counter (TCNT) operation. Specify complementary PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the same counter clock and clock edge for MTU3 and MTU4 with bits TPSC[2:0] and bits CKEG[1:0] in the Timer Control Register (TCR). Use bits CCLR[2:0] to set synchronous clearing only when restarting by synchronous clearing with another channel during complementary PWM mode operation.

[3] When performing brushless DC motor control, set bit BDC in the Timer Gate Control register (TGCRA) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Set the dead time in MTU3.TCNT. Set MTU4.TCNT to 0x0000.

[5] Set only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with MTU3 and MTU4 using the Timer Synchronous register (TSYRA).

[6] Set the output PWM duty cycles in the Duty registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) and Buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD). Be sure to set the same value in each pair of Duty register and Buffer register.
Only when using the double buffer function, set the output PWM duty value - 1 in the Buffer registers (MTU3.TGRE, MTU4.TGRE, MTU4.TGRF).

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the Timer Dead Time Enable register (TDERA) so that no dead time is generated.

[8] Set the dead time in the Timer Dead Time register (TDDRA), 1/2 the carrier cycle in the Timer Cycle Data register (TCDRA) and Timer Cycle Buffer register (TCBRA), and 1/2 the carrier cycle plus the dead time in MTU3.
TGRA and MTU3.TGRC. When no dead time generation is selected, set 1 in TDDRA and 1/2 the carrier cycle + 1 in MTU3.TGRA and MTU3.TGRC.

[9] Enable or disable toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register 1 (TOCR1A), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBRA as a buffer for TOCR2A, see Figure 17.4.

[10] Select complementary PWM mode in the Timer Mode register 1 (MTU3.TMDR1). In addition, specify the buffer operation of the MTU3.TGRA (BFA = 1) and MTU3.TGRB (BFB = 1) when necessary. Enable the double buffer operation with the TMDR2 register only when the double buffer function is used.

[11] Set enabling/disabling of PWM waveform output pin output in the Timer Output Master Enable register (TOERA).

[12] Set the Port Function Control register (PFCm) and Port Mode Control registers (PMCm) of the I/O ports.

[13] Set bits CST3 and CST4 in TSTRA to 1 simultaneously to start the count operation.

Figure 18.48 Example of complementary PWM mode setting procedure

18.4.8.2 Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Figure 18.49 shows counter operation in complementary PWM mode (MTU3 and MTU4), and Figure 18.50 shows an example of operation in complementary PWM mode.

(1) Counter Operation

In complementary PWM mode, three counters—MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) —in each unit perform up-/down-count operations.

MTU3.TCNT (MTU6.TCNT) is automatically initialized to the value set in TDDRA (TDDRB) when complementary PWM mode is selected and the CST bit in TSTRA (TSTRB) is 0. When the CST bit is set to 1, MTU3.TCNT (MTU6.TCNT)

counts up to the value set in MTU3.TGRA (MTU6.TGRA), then switches to down-counting when it matches MTU3.TGRA (MTU6.TGRA). When the MTU4.TCNT (MTU7.TCNT) value matches 0x0000, MTU3.TCNT (MTU6.TCNT) switches to up-counting, and the operation is repeated in this way.

MTU4.TCNT (MTU7.TCNT) should be initialized to 0x0000 after a reset. When the CST4 bit is set to 1, MTU4.TCNT (MTU7.TCNT) counts up in synchronization with MTU3.TCNT (MTU6.TCNT), and switches to down-counting when MTU3.TCNT (MTU6.TCNT) matches MTU3.TGRA (MTU6.TGRA). On reaching 0x0000, MTU4.TCNT (MTU7.TCNT) switches to up-counting, and the operation is repeated in this way.

MTU4.TCNT (MTU7.TCNT) should be initialized to 0x0000 after a reset. When the CST4 bit is set to 1, MTU4.TCNT (MTU7.TCNT) counts up in synchronization with MTU3.TCNT (MTU6.TCNT), and switches to down-counting when MTU3.TCNT (MTU6.TCNT) matches MTU3.TGRA (MTU6.TGRA).

On reaching 0x0000, MTU4.TCNT (MTU7.TCNT) switches to up-counting, and the operation is repeated in this way. TCNTSA (TCNTSB) is a read-only counter. It does not need to be initialized after a reset.

In counting up by MTU3.TCNT and MTU4.TCNT (or MTU6.TCNT and MTU7.TCNT), MTU3.TCNT (or MTU6.TCNT) starts counting up when it matches TCDRA (or TCDRB) and switches to counting down when it matches MTU3.TGRA (or MTU6.TGRA). Furthermore, when MTU4.TCNT (or MTU7.TCNT) matches TDDRA (or TDDRB), TCNTSA (or TCNTSB) is set to the value in MTU3.TGRA (or MTU6.TGRA) and counting is stopped.

When MTU4.TCNT (MTU7.TCNT) matches TDDRA (TDDRB) during down-counting of MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT), TCNTSA (TCNTSB) starts up-counting, and when MTU4.TCNT (MTU7.TCNT) matches 0x0000, the operation switches to down-counting. When MTU3.TCNT (MTU6.TCNT) matches TCDRA (TCDRB), TCNTSA (TCNTSB) becomes 0x0000 and stops counting.

TCNTSA (TCNTSB) is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

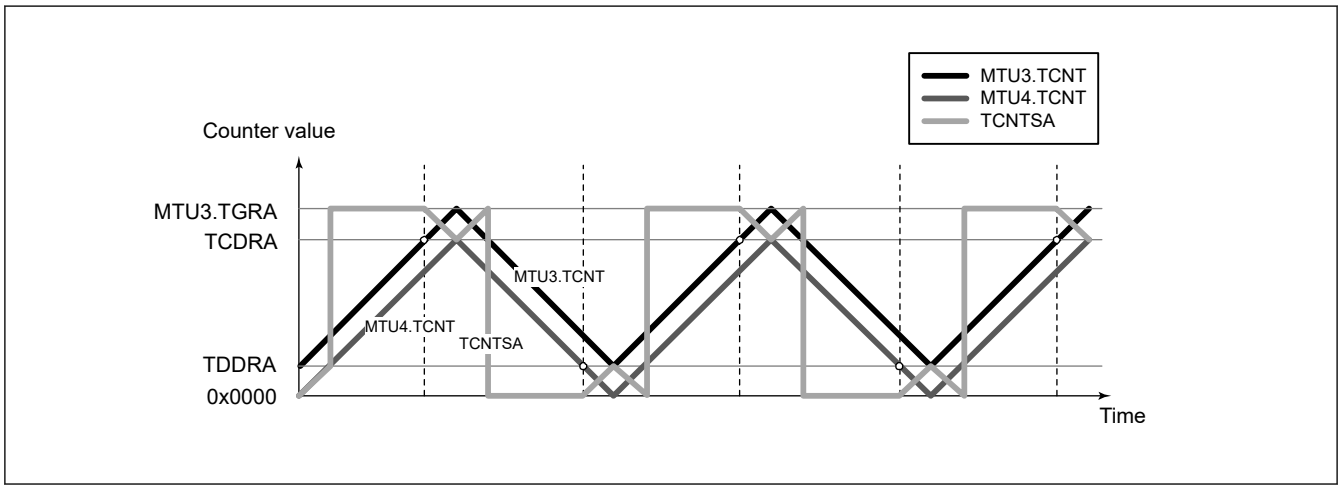


Figure 18.49 Count operation in complementary PWM mode (MTU3 and MTU4)

(2) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used for each unit. Figure 18.50 shows an example of operation in complementary PWM mode (MTU3 and MTU4).

MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB) are compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in the OLSN and OLSP bits in the timer output control registers (TOCR1A and TOCR1B) is output from the PWM output pin. MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD) are buffer registers for these compare registers.

When the double buffer function is used, MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) are also used as buffer registers B. For details of double buffer operation, see (19) Double Buffer Function in Complementary PWM Mode.

Data in a compare register can be changed by writing new data to the corresponding buffer register. The buffer registers can be read or written at any time.

When modifying data in a buffer register, write to MTU4.TGRD (MTU7.TGRD) last and enable data transfer from the buffer register to a temporary register. At this time, transfer from the TCBRA (TCBRB) register and MTU3.TGRC (MTU6.TGRC) register, which operate as buffer registers for the timer period registers, to temporary registers is also enabled. Data is transferred to all five temporary registers at the same time.

When transfer is enabled in the Ta interval, data written to a buffer register is transferred to the temporary register. The data is not transferred to the temporary register in the Tb1 and Tb2 intervals. Data enabled for transfer in this interval is transferred to the temporary register at the end of this interval.

The value transferred to a temporary register is transferred to the compare register when TCNTSA (TCNTSB) for which the Tb interval ends matches MTU3.TGRA (MTU6.TGRA) while TCNTSA (TCNTSB) is counting up, or 0x0000 while counting down.

The timing for transfer from the temporary register to the compare register can be selected with bits MD[3:0] in the timer mode register 1 (TMDR1). Figure 18.50 shows an example in which the trough is selected for the transfer timing.

In the Tb interval in which data is not transferred to the temporary register (Tb1 in Figure 18.50), the temporary register has the same function as the compare register and is compared with the counter. In this interval, therefore, there are two compare match registers for one output phase; the compare register contains the pre-change data and the temporary register contains new data. In this interval, three counters MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

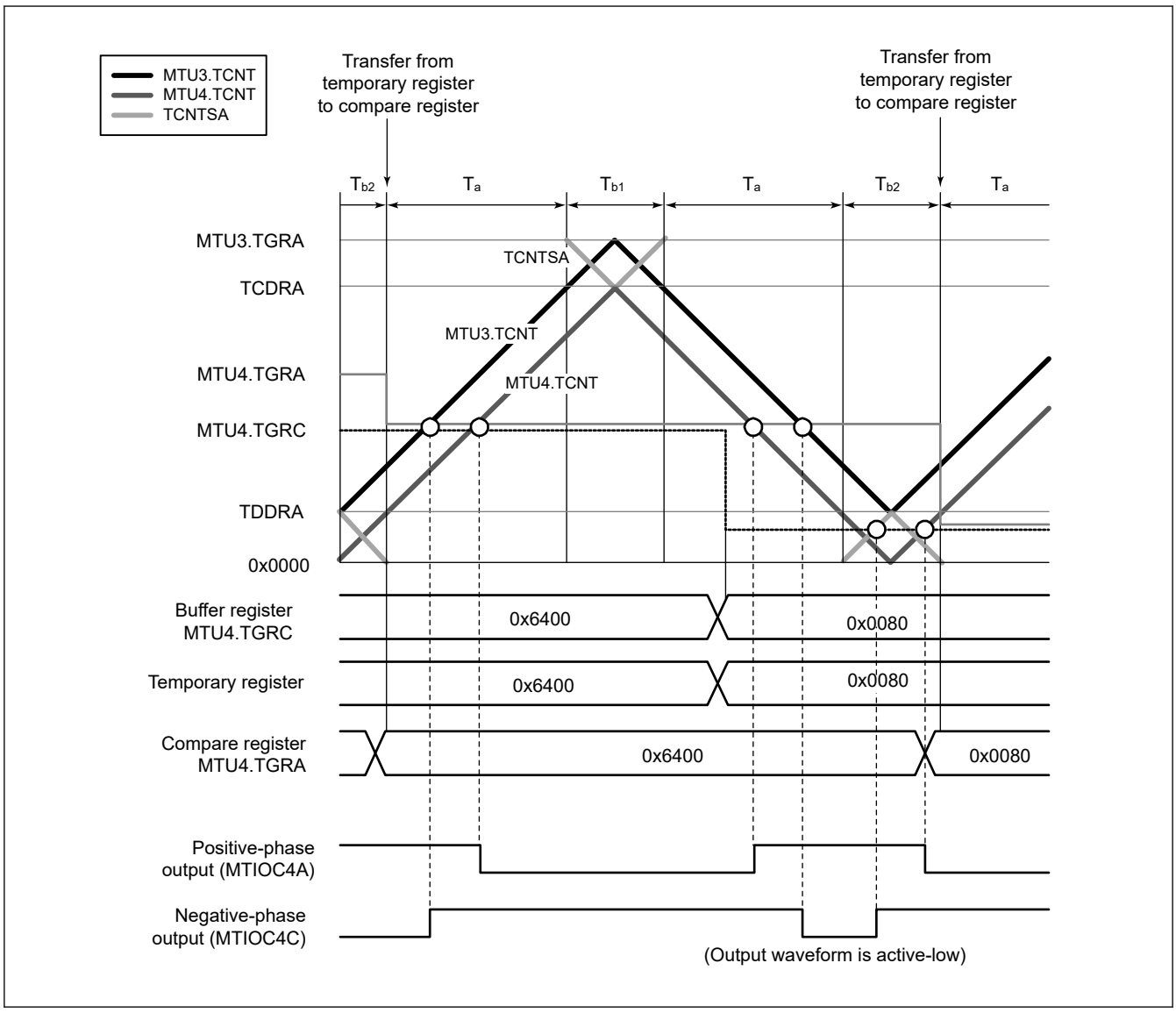


Figure 18.50 Example of operation in complementary PWM mode (MTU3 and MTU4)

(3) Initial Setting

In complementary PWM mode, there are nine registers that require initial setting. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).

Before setting complementary PWM mode with MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits, initial values should be set in the following registers.

MTU3.TGRC (MTU6.TGRC) operates as the buffer register for MTU3.TGRA (MTU6.TGRA), and should be set with 1/2 the PWM carrier period + dead time Td. The timer cycle buffer register (TCBRA or TCBRB) operates as the buffer register for the timer cycle data register (TCDRA or TCDRB), and should be set with 1/2 the PWM carrier period. Set dead time Td in the timer dead time data register (TDDRA or TDDRB).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDERA or TDERB) should be set to 0, MTU3.TGRC and MTU3.TGRA (MTU6.TGRC and MTU6.TGRA) should be set to 1/2 the PWM carrier period + 1, and TDDRA (TDDRB) should be set to 1.

Set the respective initial PWM duty values in three buffer registers A (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD)).

Set the respective (initial PWM duty − 1) values in three buffer registers B (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF)) only when the double buffer function is used.

The values set in the five buffer registers excluding TDDRA (TDDRB) are transferred to the corresponding compare registers as soon as complementary PWM mode is set.

Set MTU4.TCNT (MTU7.TCNT) to 0x0000 before setting complementary PWM mode.

Table 18.73 Registers and counters requiring initial setting

Register and counter	Setting
TOCR1A, TOCR2A, TOCR1B, TOCR2B	PWM output level
MTU3.TGRC MTU6.TGRC	1/2 PWM carrier period + dead time Td (1/2 PWM carrier period + 1 when dead time generation is disabled by TDERA or TDERB)
TDDRA, TDDRB	Dead time Td (1 when dead time generation is disabled by TDERA or TDERB)
TCBRA, TCBRB	1/2 PWM carrier period
MTU3.TGRD, MTU4.TGRC, MTU4.TGRD MTU6.TGRD, MTU7.TGRC, MTU7.TGRD	Initial PWM duty ratio value for each phase
MTU3.TGRE, MTU4.TGRE, MTU4.TGRF MTU6.TGRE, MTU7.TGRE, MTU7.TGRF	Initial PWM duty ratio − 1 value for each phase (only when double buffer function is used)
MTU4.TCNT MTU7.TCNT	0x0000

Note: The value set in MTU3.TGRC (MTU6.TGRC) should be the sum of 1/2 the PWM carrier period set in TCBRA (TCBRB) and dead time Td set in TDDRA (TDDRB). When dead time generation is disabled by TDERA (TDERB), TGRC should be set to 1/2 the PWM carrier period + 1.

(4) PWM Output Level Setting

In complementary PWM mode, the PWM output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1A or TOCR1B) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2A or TOCR2B).

The output level can be set for each of the three positive phases and three negative phases of 6-phase output.

Complementary PWM mode should be cleared before setting or changing output levels.

(5) Dead Time Setting

In complementary PWM mode, dead time can be set for PWM output.

The dead time is set in the timer dead time data register (TDDRA or TDDRB). The value set in TDDRA (TDDRB) is used as the MTU3.TCNT (MTU6.TCNT) counter start value and creates a non-overlapping interval between MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT). Complementary PWM mode should be cleared before changing the contents of TDDRA (TDDRB).

(6) Dead Time Suppressing

Dead time generation is suppressed by setting the TDER bit in the timer dead time enable register (TDERA or TDERB) to 0. TDERA (TDERB) can be set to 0 only when 0 is written to it after reading TDER = 1.

MTU3.TGRA and MTU3.TGRC (MTU6.TGRA and MTU6.TGRC) should be set to 1/2 PWM carrier period + 1 and the timer dead time data register (TDDRA or TDDRB) should be set to 1.

By the above settings, PWM waveforms without dead time can be obtained. Figure 18.51 shows an example of operation without dead time (MTU3 and MTU4).

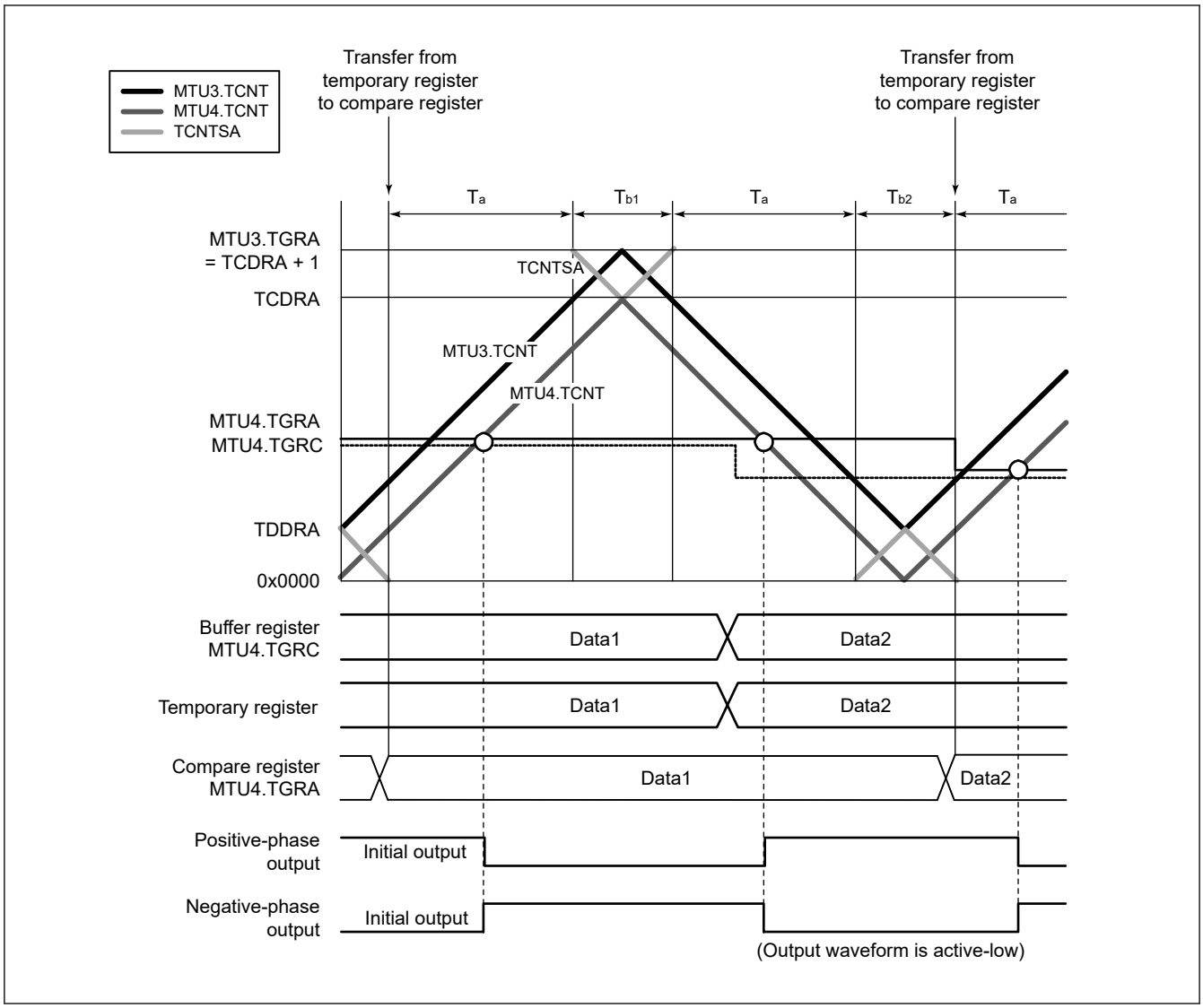


Figure 18.51 Example of operation without dead time (MTU3 and MTU4)

(7) PWM Cycle Setting

In complementary PWM mode, the PWM period is set in two registers—MTU3.TGRA (MTU6.TGRA), in which the MTU3.TCNT (MTU6.TCNT) upper limit value is set, and TCDRA (TCDRB), in which the MTU4.TCNT (MTU7.TCNT) upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + TDDRA (TDDRB) setting

Without dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + 1

In addition, the settings should be made so as to achieve the following relationship between the TCDRA (TCDRB) register and the TDDRA (TDDRB) register:

TCDRA (TCDRB) setting > TDDRA (TDDRB) setting × 2 + 2

The MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) settings are made by setting values in buffer registers MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB). The values set in MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB) are transferred simultaneously to MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) with the transfer timing selected with bits MD[3:0] in the timer mode register 1 (TMDR1).

The new PWM period is reflected from the next cycle when data is updated at the crest, or from the current cycle when updated in the trough. Figure 18.52 shows the operation when the PWM period is updated at the crest.

See the following section, (8) Register Data Updating, for the method of updating the data in each buffer register.

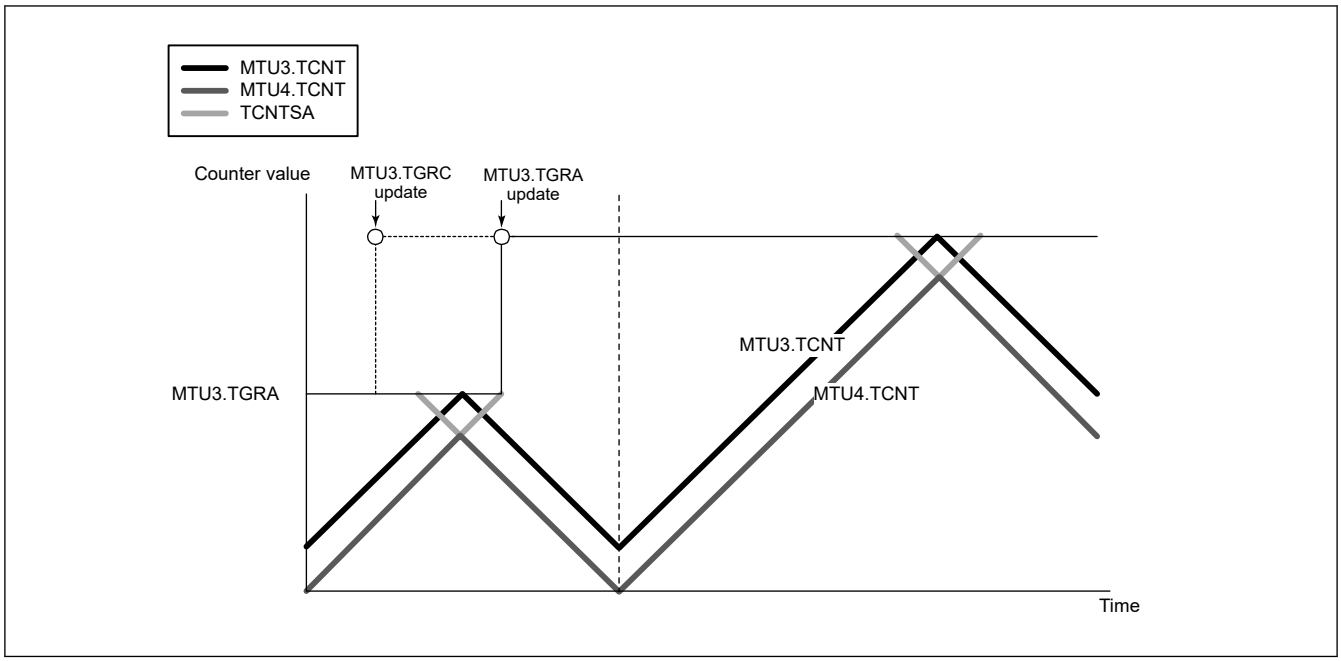


Figure 18.52 Example of PWM period updating (MTU3 and MTU4)

(8) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five registers (PWM duty and carrier cycle registers) that have buffer registers and can be updated during operation.

There is a temporary register between each of these registers and its buffer register. While subcounter TCNTSA (TCNTSB) is not counting, if buffer register data is updated, the temporary register value also changes. Data is not transferred from buffer registers to temporary registers while TCNTSA (TCNTSB) is counting; in this case, the value written to a buffer register is transferred after TCNTSA (TCNTSB) halts.

The temporary register value is transferred to the compare register at the data update timing set with bits MD[3:0] in the timer mode register 1 (TMDR1). Figure 18.53 shows an example of data updating in complementary PWM mode (MTU3 and MTU4). This example shows the mode in which data is updated at both the counter crest and trough.

When updating buffer register data, be sure to write to MTU4.TGRD (MTU7.TGRD) at the end of the update. Data is transferred from buffer registers to the temporary registers simultaneously for all five registers after the write to MTU4.TGRD (MTU7.TGRD).

Even when not updating all five registers or when not updating the MTU4.TGRD (MTU7.TGRD) data, be sure to write to MTU4.TGRD (MTU7.TGRD) after writing data to the registers to be updated. In this case, the data written to MTU4.TGRD (MTU7.TGRD) should be the same as the data prior to the write operation.

See (19) Double Buffer Function in Complementary PWM Mode, for data updating when the double buffer function is used.

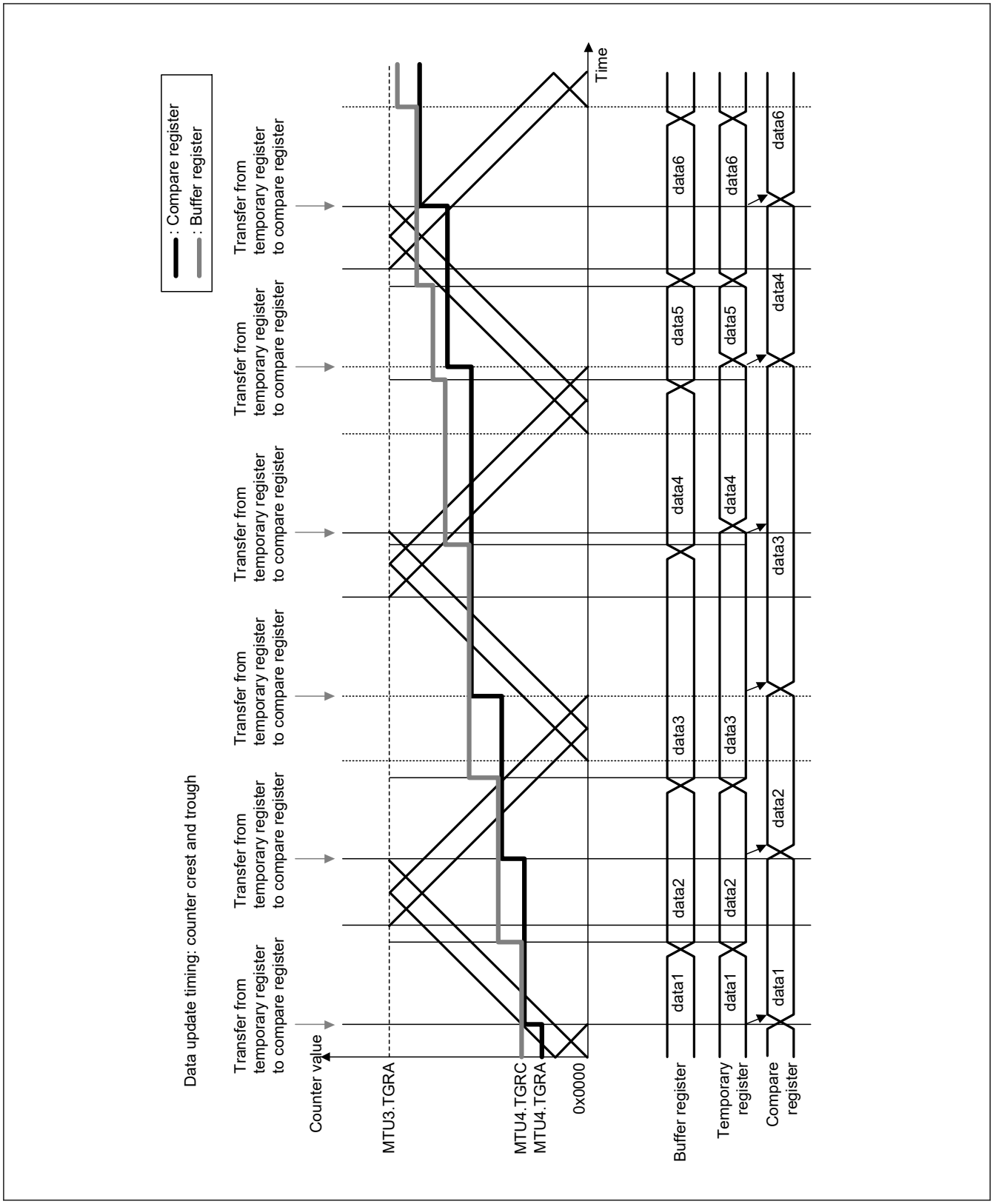


Figure 18.53 Example of data updating in complementary PWM mode (MTU3 and MTU4)

(9) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in timer output control register 1 (TOCR1A or TOCR1B) or bits OLS1N to OLS3N and OLS1P to OLS3P in timer output control register 2 (TOCR2A or TOCR2B).

This initial output is the non-active level of the PWM pulse and continues from when complementary PWM mode is set with the timer mode register 1 (TMDR1) until MTU4.TCNT (MTU7.TCNT) exceeds the value set in the Timer dead time data register (TDDRA or TDDRB). Figure 18.54 shows an example of the initial output in complementary PWM mode.

An example of the waveform when the initial PWM duty ratio value is smaller than the TDDRA value is shown in Figure 18.55.

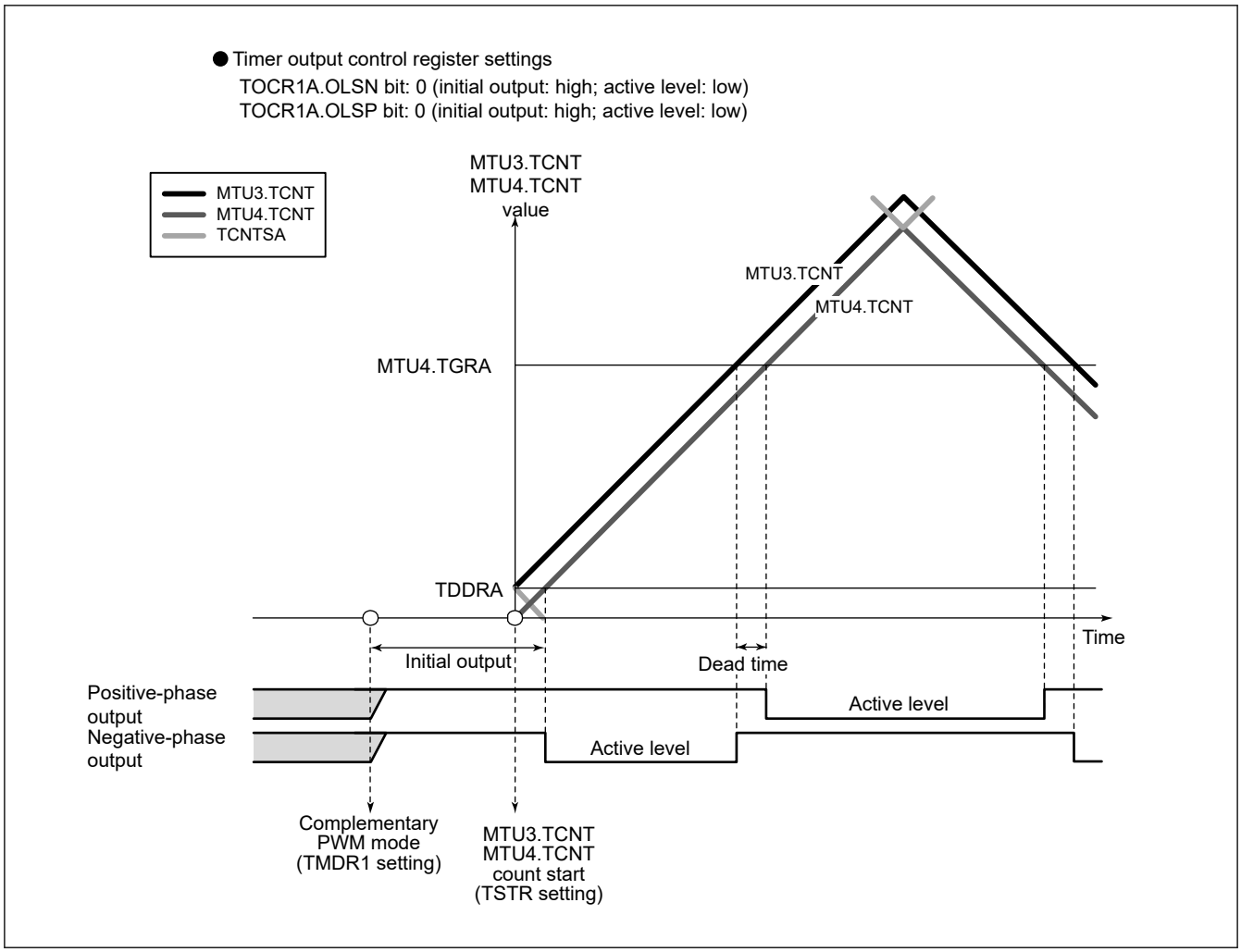


Figure 18.54 Example of initial output in complementary PWM mode (MTU3 and MTU4) (1)

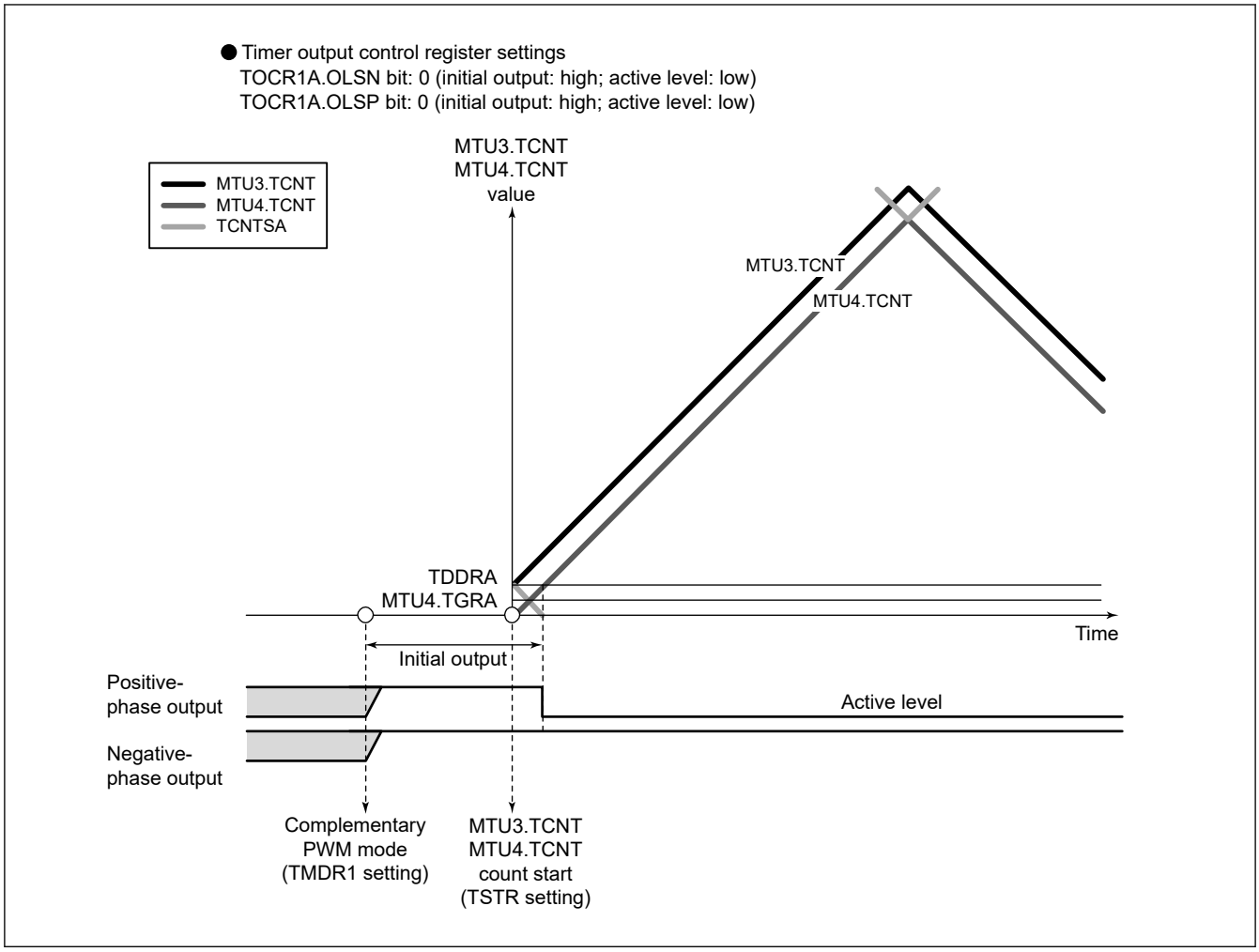


Figure 18.55 Example of initial output in complementary PWM mode (MTU3 and MTU4) (2)

(10) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six waveforms of three-phase PWM waveforms can be output. Dead time can be set for PWM waveforms to be output.

A PWM waveform is generated by output of the level selected in the Timer Output Control register in the event of a compare match between a counter and a compare register. While TCNTSA (TCNTSB) is counting, the compare register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100% duty ratio. The relative timing of turn-on and turn-off compare match occurrence may vary, but the compare match that turns off each phase takes precedence to secure the dead time and ensure that the positive-phase and negative-phase turn-on times do not overlap. Figure 18.56 to Figure 18.58 show examples of waveform generation in complementary PWM mode.

The positive-phase and negative-phase turn-off timing is generated by a compare match with the counter indicated by a solid line, and the turn-on timing is generated by a compare match with the counter indicated by a dotted line, which operates with a delay equal to the dead time behind the counter indicated by a solid line. In the T1 period, compare match a that turns off the negative phase has the highest priority, and compare matches before a are ignored. In the T2 period, compare match c that turns off the positive phase has the highest priority, and compare matches before c are ignored.

In most cases, compare matches occur in the order a → b → c → d (or c → d → a' → b') as shown in Figure 18.56.

If compare matches deviate from the a → b → c → d order, since the time for which the negative phase is off is shorter than twice the dead time, the positive phase is not turned on. If compare matches deviate from the c → d → a' → b' order, since the time for which the positive phase is off is shorter than twice the dead time, the negative phase is not turned on.

As shown in Figure 18.57, if compare match c follows compare match a before compare match b, compare match b is ignored and the negative phase is turned on by compare match d. This is because turning off the positive phase has priority due to the occurrence of compare match c (positive-phase off timing) before compare match b (positive-phase on timing) (consequently, the waveform does not change because the positive phase goes from off to off).

Similarly, in the example in Figure 18.58, compare match a' with new data in the temporary register occurs before compare-match c, but until compare-match c, which turns off the positive phase, other compare matches are ignored. As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare matches at turn-off timings take precedence, and turn-on timing compare matches that occur before a turn-off timing compare match are ignored.

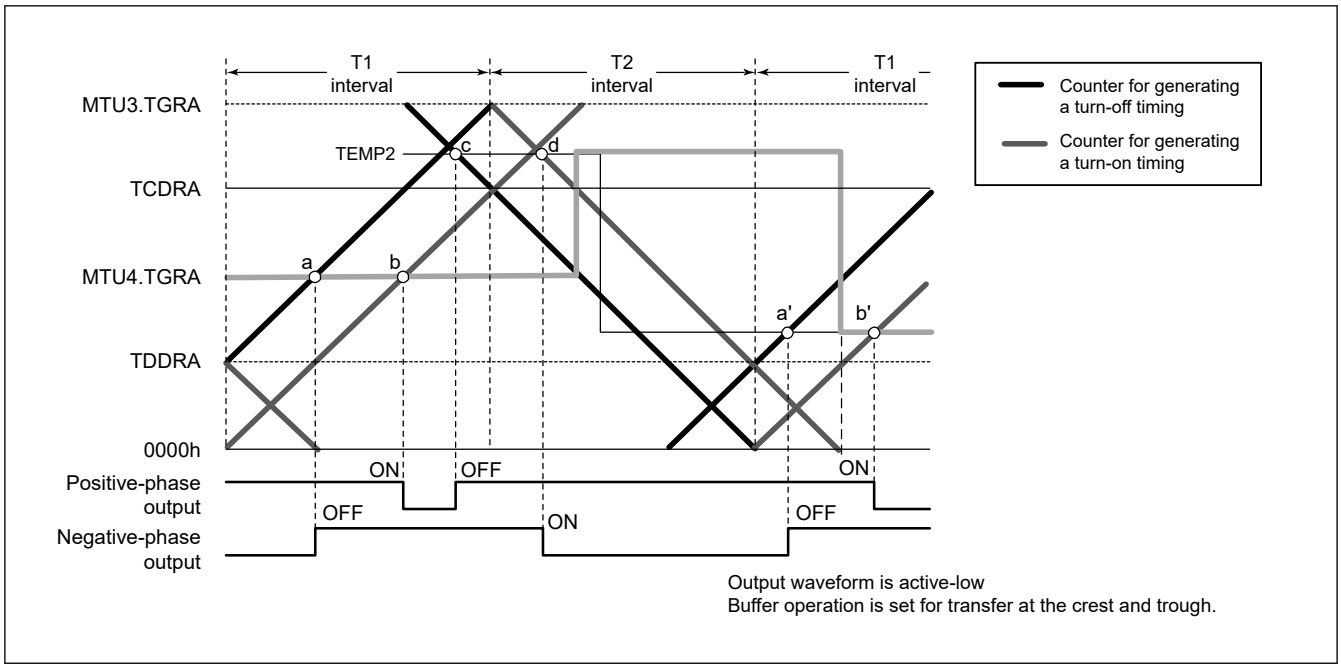


Figure 18.56 Example of waveform output in complementary PWM mode (MTU3 and MTU4) (1)

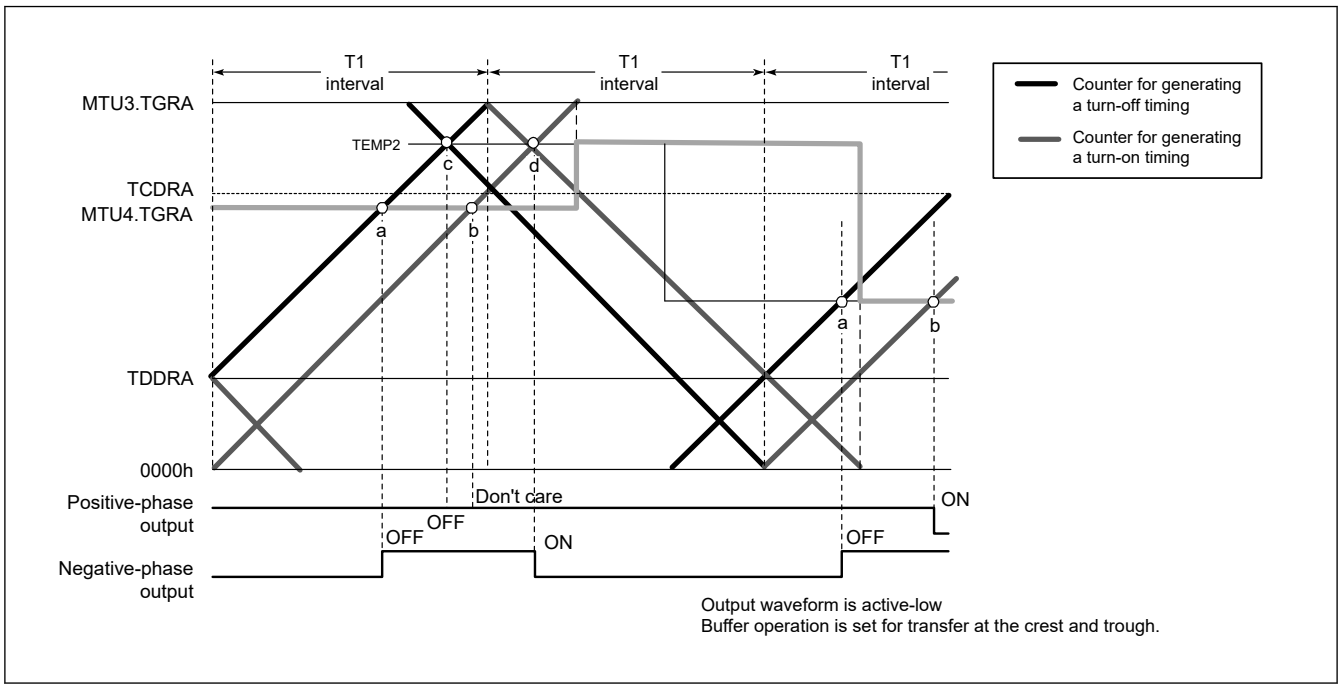


Figure 18.57 Example of waveform output in complementary PWM mode (MTU3 and MTU4) (2)

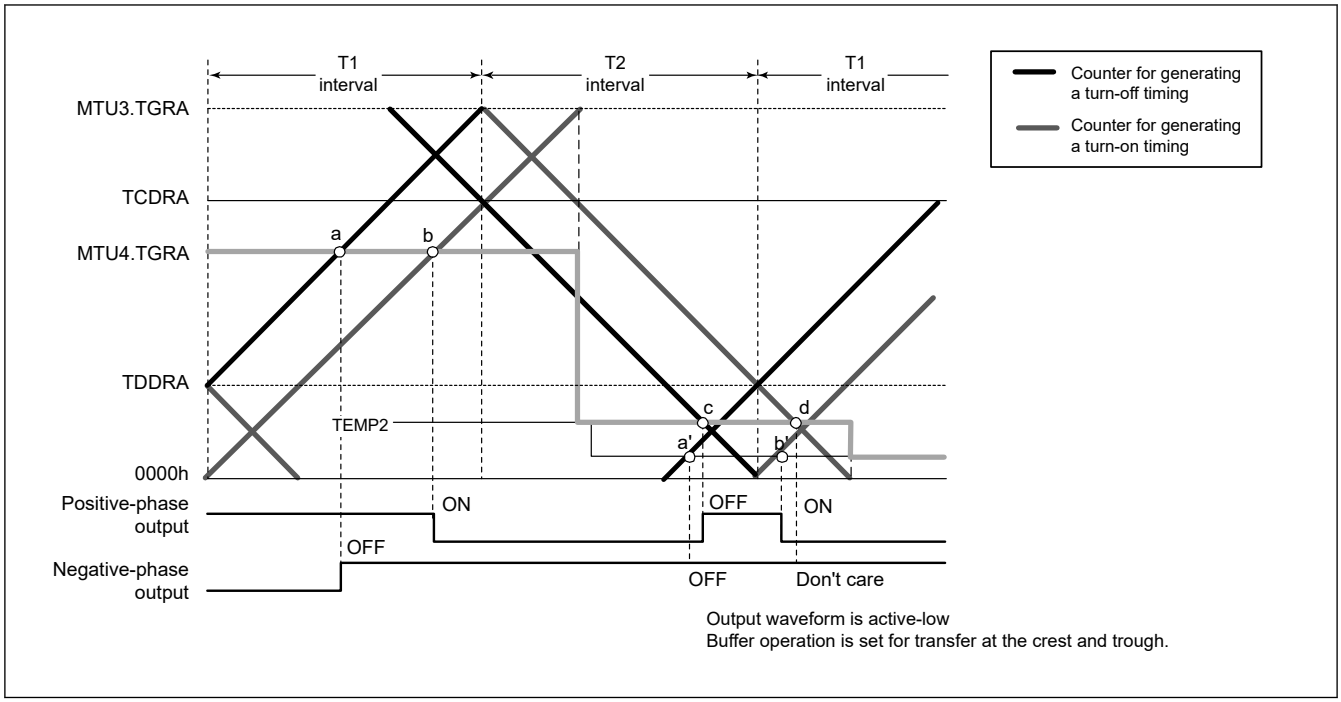


Figure 18.58 Example of waveform output in complementary PWM mode (MTU3 and MTU4) (3)

(11) 0% and 100% Duty Ratio Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty PWM waveforms can be output as required. Figure 18.59 to Figure 18.63 show output examples.

A 100% duty waveform is output when the compare register value is set to 0x0000. The waveform in this case has a positive phase with a 100% on-state. A 0% duty waveform is output when the compare register value is set to the same value as MTU3.TGRA (MTU6.TGRA). The waveform in this case has a positive phase with a 100% off-state.

Turn-on and turn-off compare matches occur simultaneously, but if a turn-on compare match and turn-off compare match for the same phase occur simultaneously, both compare matches are ignored and the waveform does not change.

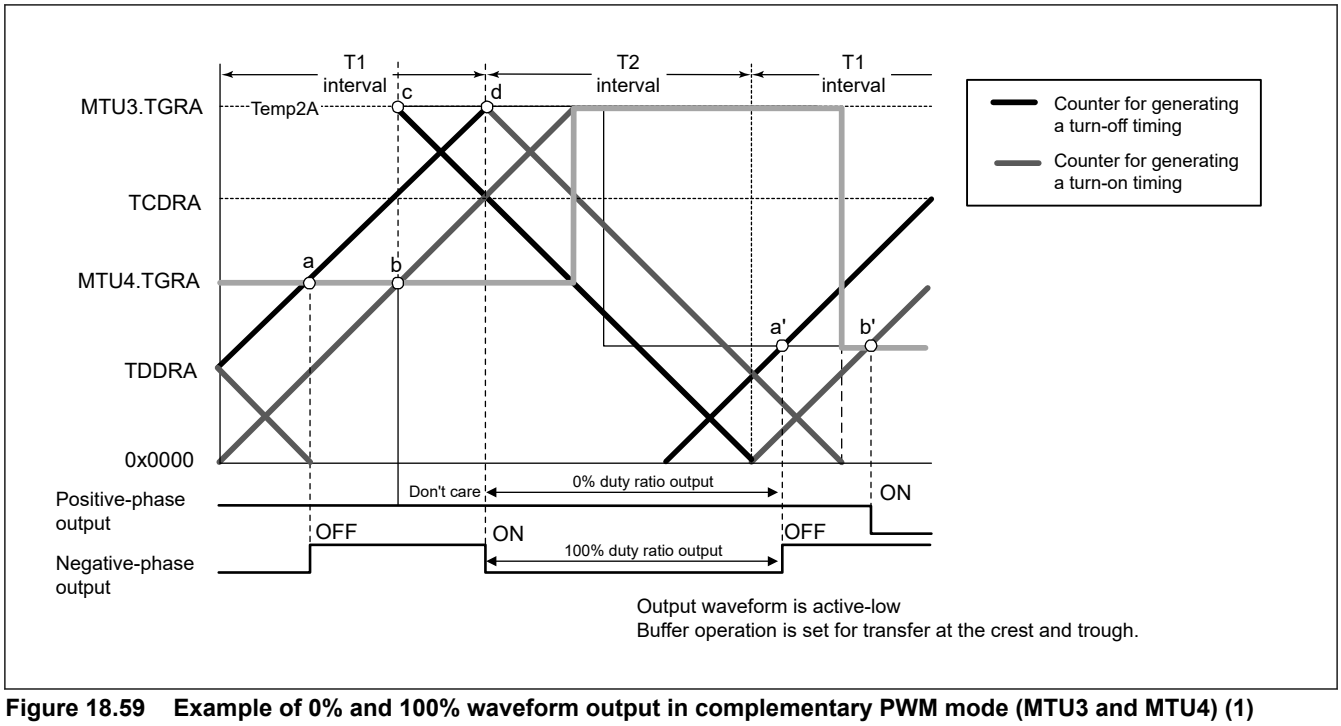


Figure 18.59 Example of 0% and 100% waveform output in complementary PWM mode (MTU3 and MTU4) (1)

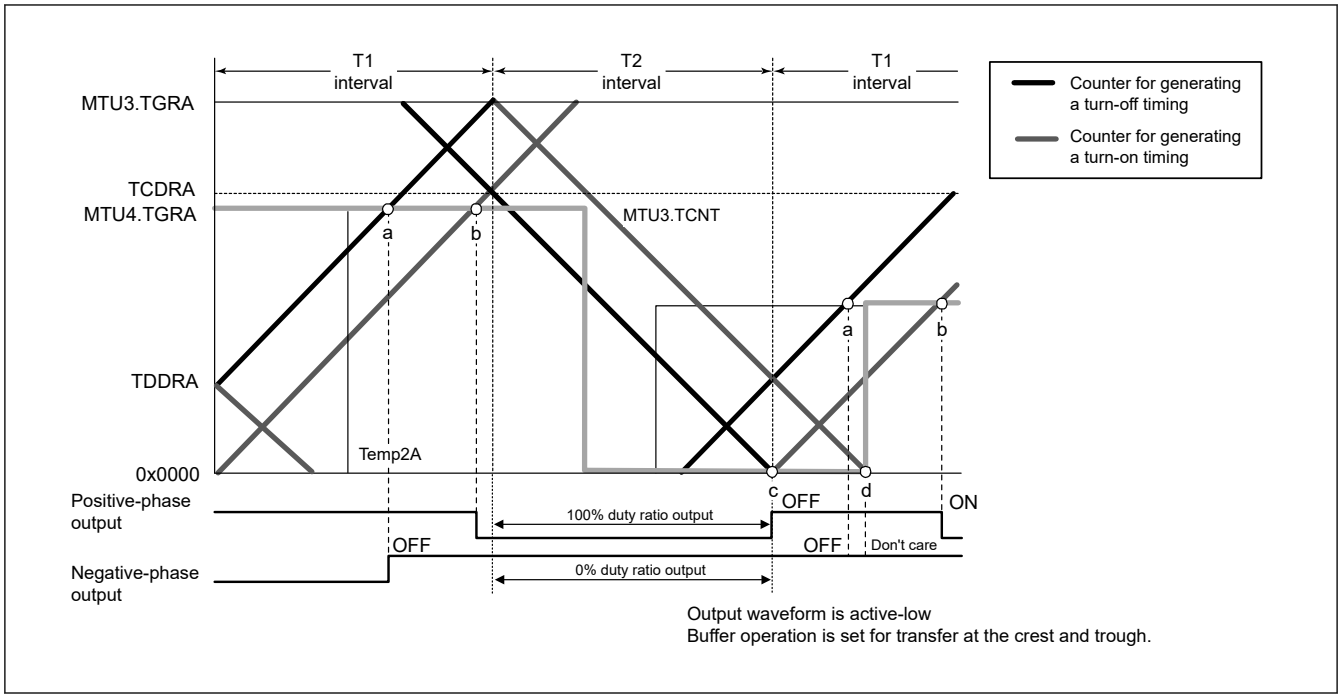


Figure 18.60 Example of 0% and 100% waveform output in complementary PWM mode (MTU3 and MTU4) (2)

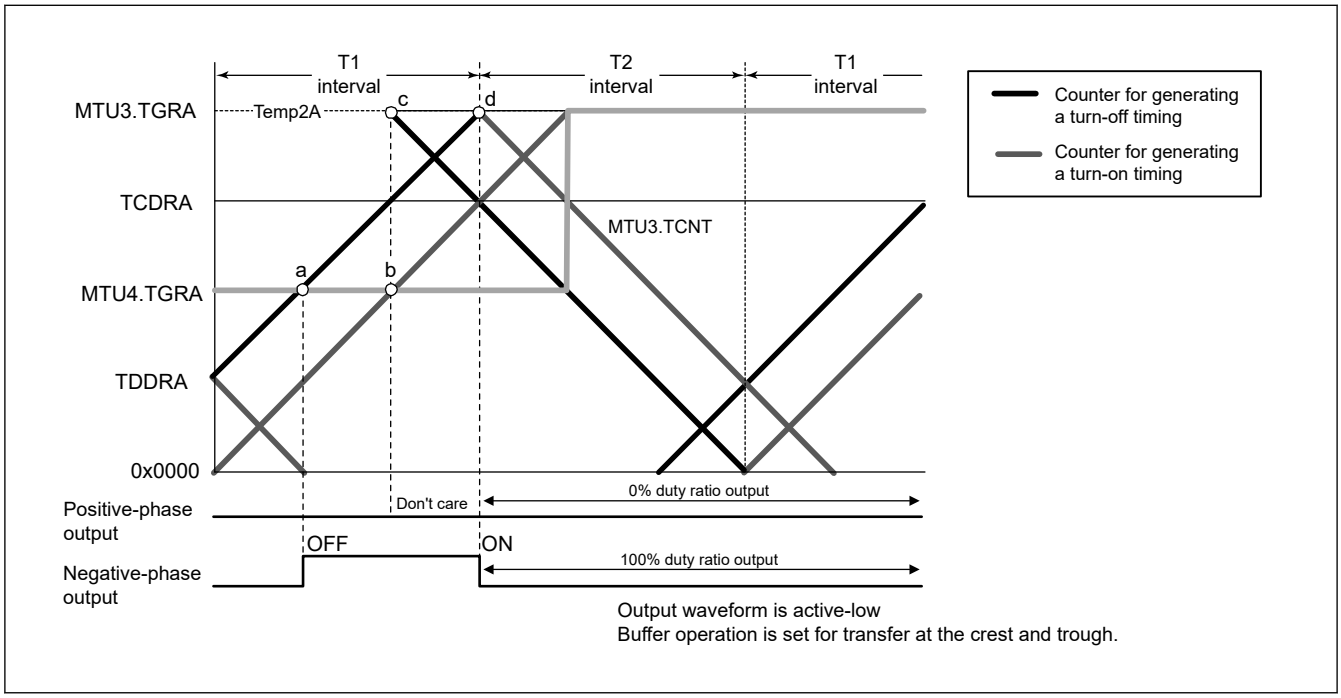


Figure 18.61 Example of 0% and 100% waveform output in complementary PWM mode (MTU3 and MTU4) (3)

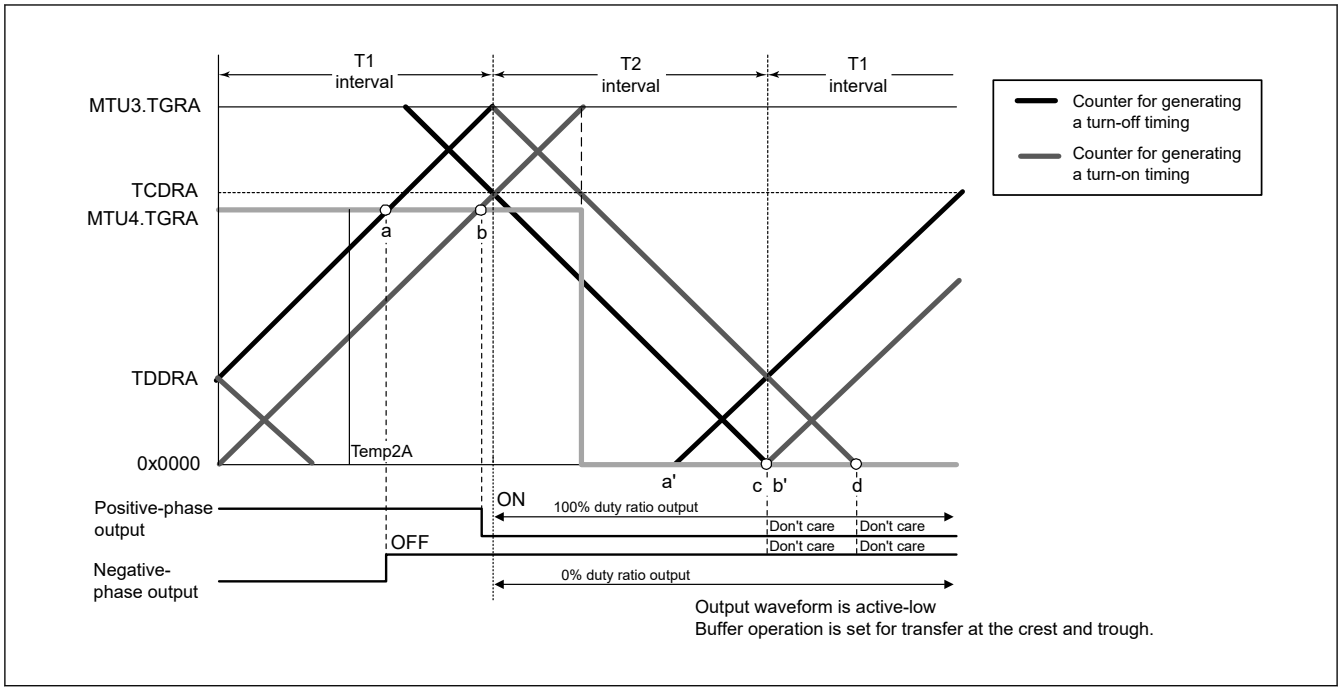


Figure 18.62 Example of 0% and 100% waveform output in complementary PWM mode (MTU3 and MTU4) (4)

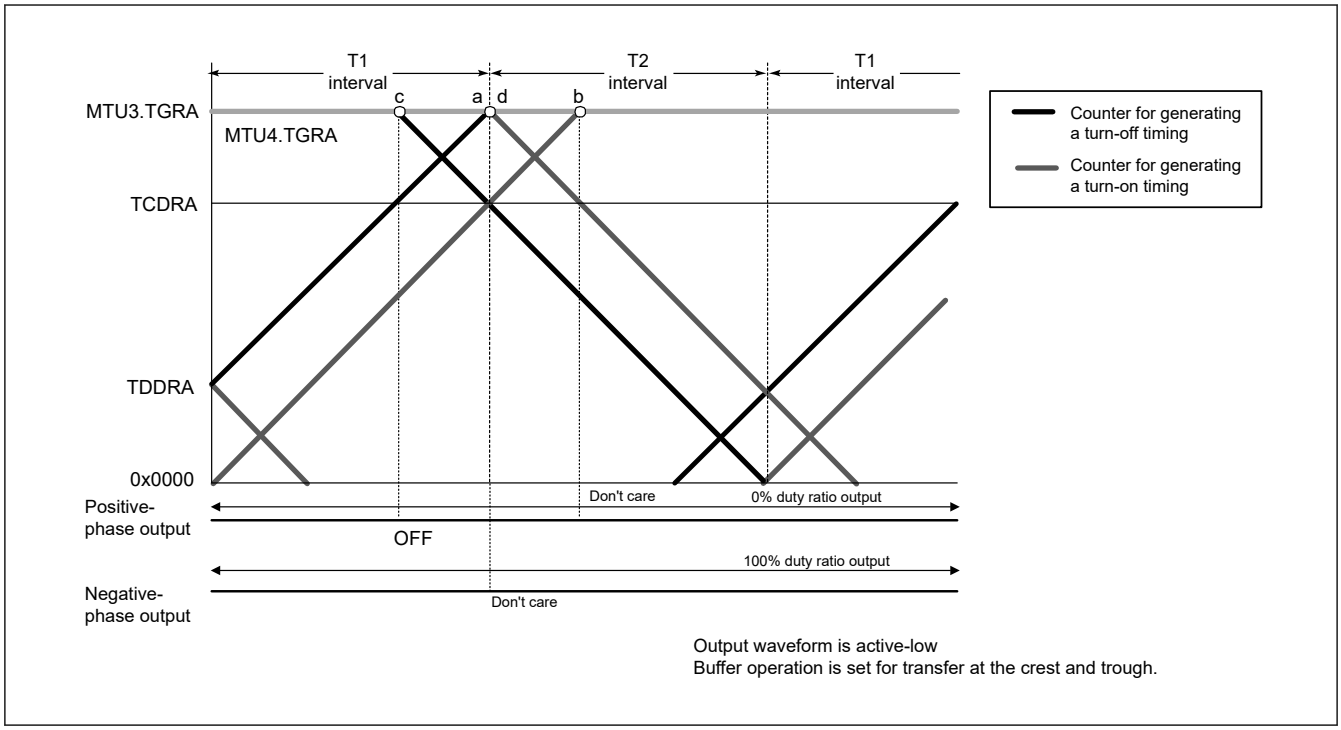


Figure 18.63 Example of 0% and 100% waveform output in complementary PWM mode (MTU3 and MTU4) (5)

(12) Toggle Output Synchronized with PWM Period

In complementary PWM mode, toggle output from the PWM output pin in synchronization with the PWM carrier period can be generated by setting the PSYE bit in the TOCR1A (TOCR1B) register to 1. An example of a toggle output waveform is shown in Figure 18.64.

This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRA (MTU6.TCNT and MTU6.TGRA) and a compare match between MTU4.TCNT (MTU7.TCNT) and 0x0000.

The MTIOC3A (MTIOC6A) pin is assigned for this toggle output. The initial output is high-level.

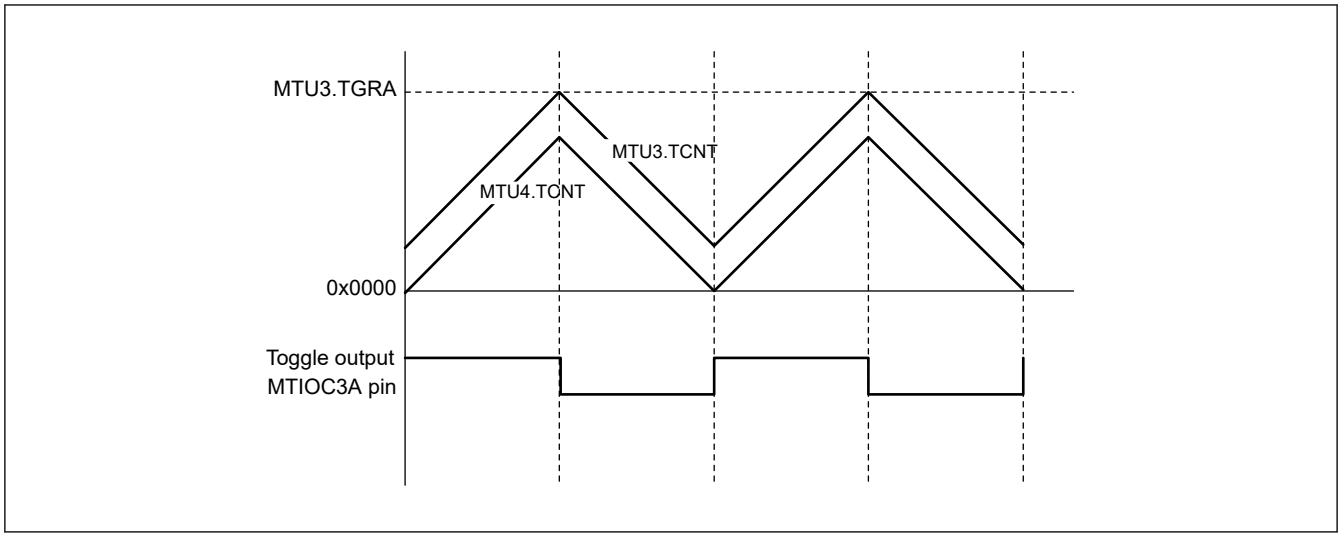


Figure 18.64 Example of toggle output waveform synchronized with PWM output (MTU3 and MTU4)

(13) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by another channel when a mode for synchronization with another channel is specified through the timer synchronous register (TSYRA or TSYRB)) and synchronous clearing is selected with bits CCLR[2:0] in the timer control register (TCR).

Figure 18.65 shows an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

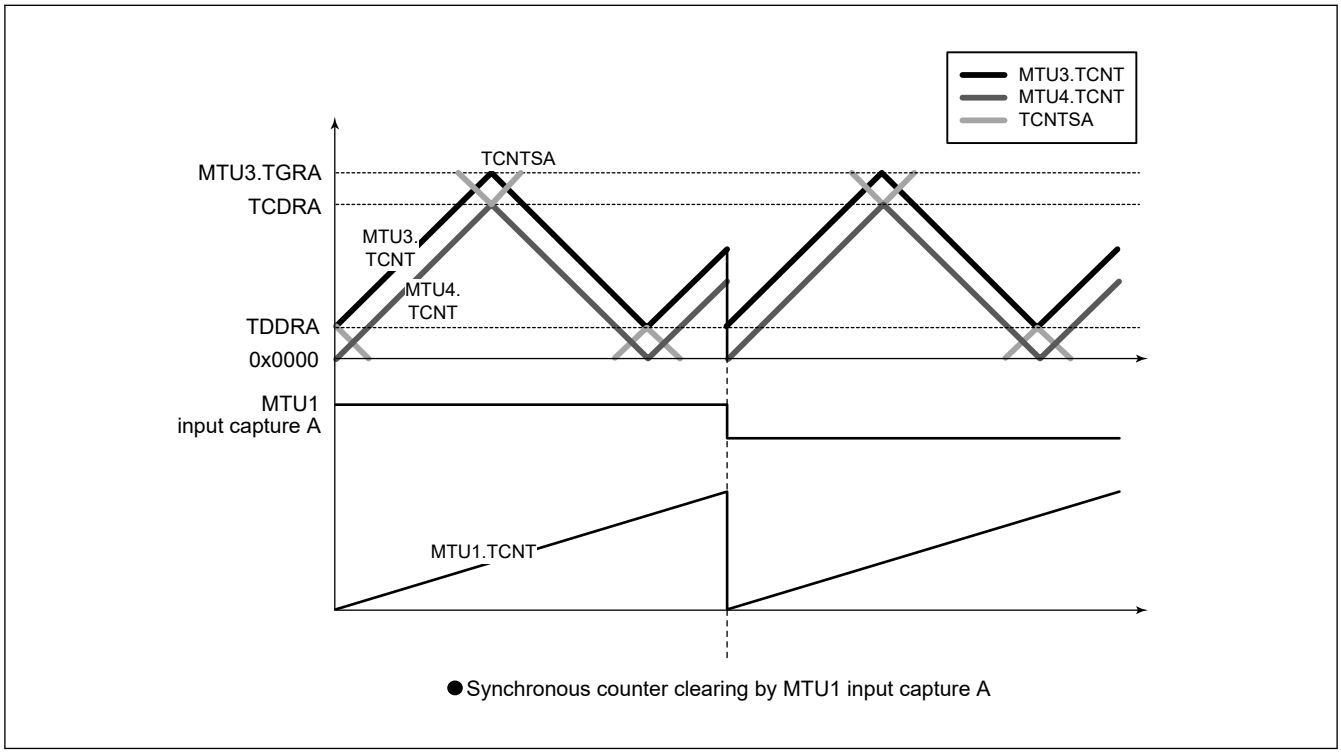


Figure 18.65 Counter clearing synchronized with another channel (MTU3 and MTU4)

(14) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCRA (TWCRB) to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression through the WRE bit = 1 is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in Figure 18.66. When synchronous clearing occurs outside that interval, the initial value specified by the OLSN and OLSP bits in TOCR1A (TOCR1B) is output. Even in the Tb interval at the trough, if synchronous clearing occurs in the initial value output period (indicated by (1) in Figure 18.66) immediately after the counters start operation, initial value output is not suppressed.

This function can be used in both channel combinations of MTU3 and MTU4, and MTU6 and MTU7. In MTU3 and MTU4, synchronous clearing generated in MTU0 to MTU2 can cause counter clearing; in MTU6 and MTU7, compare match or input capture in any of MTU0 to MTU2 can cause counter clearing.

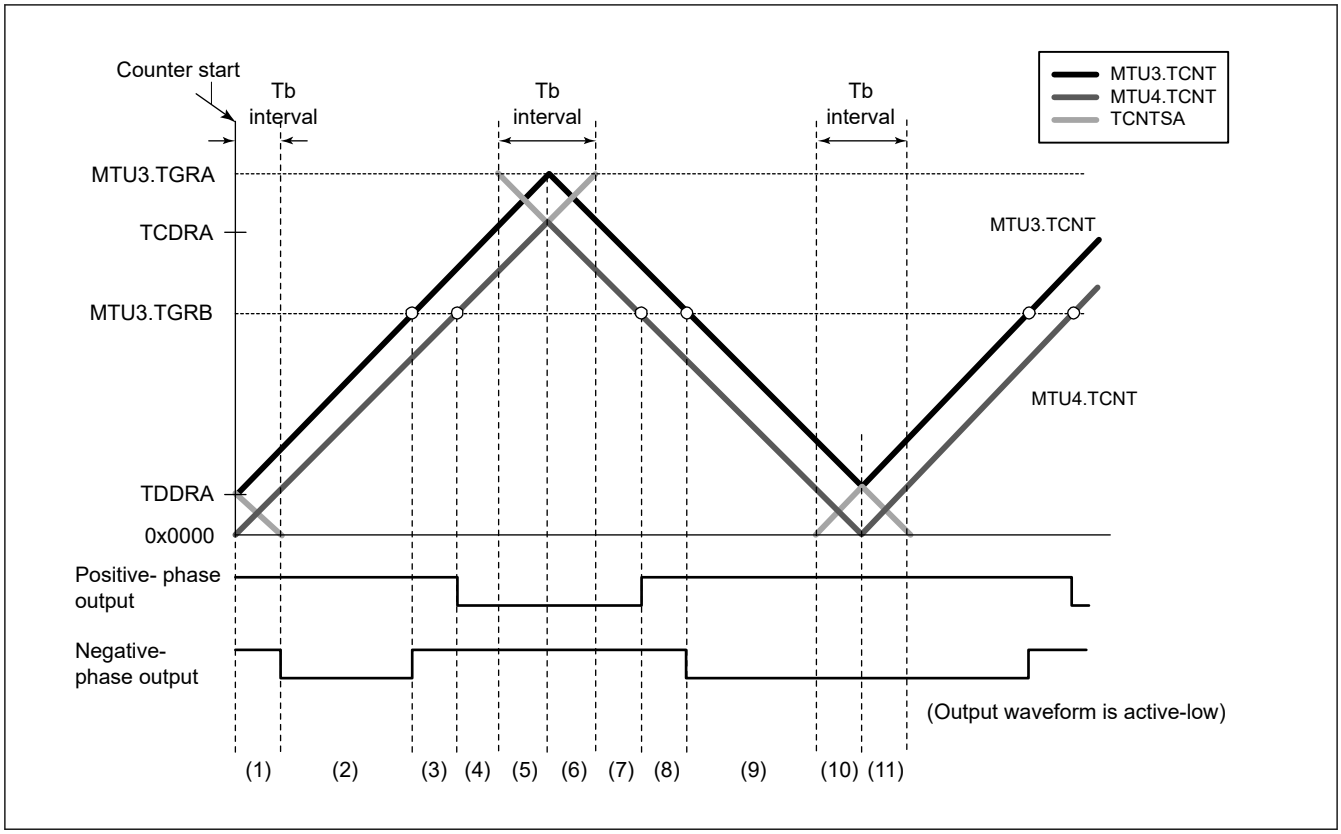


Figure 18.66 Timing for synchronous counter clearing (MTU3 and MTU4)

Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in Figure 18.67.

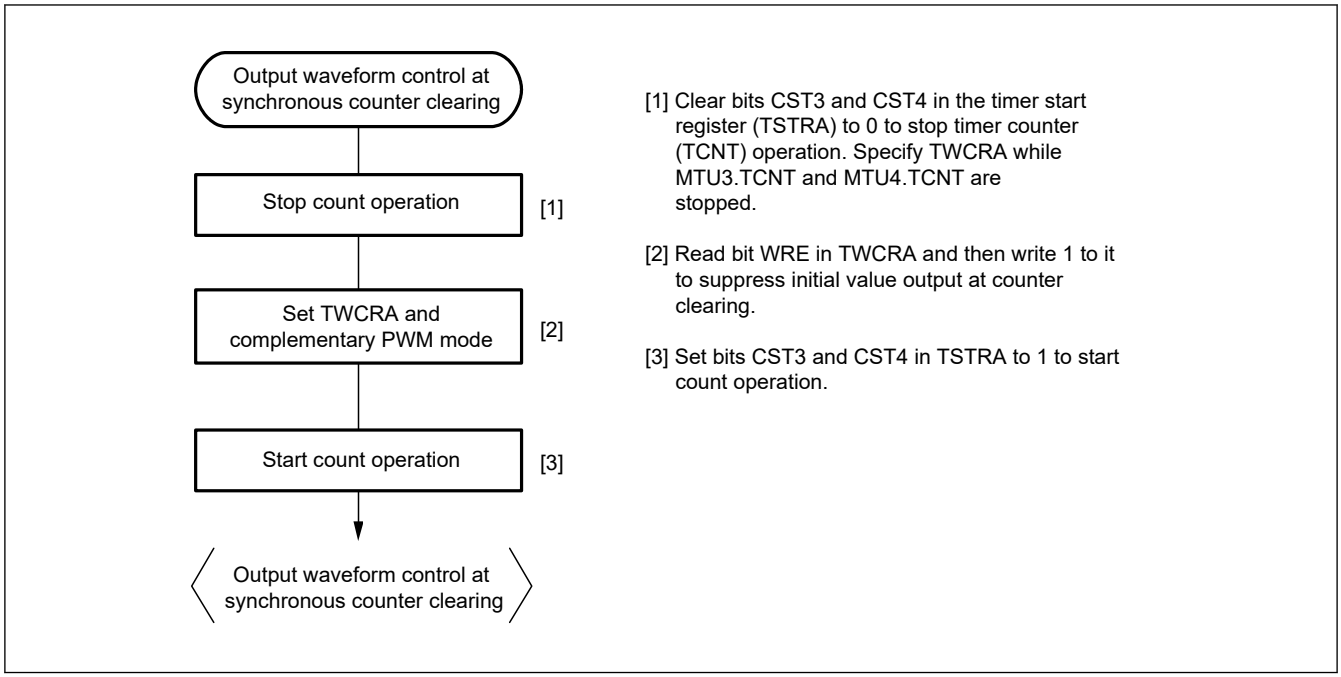
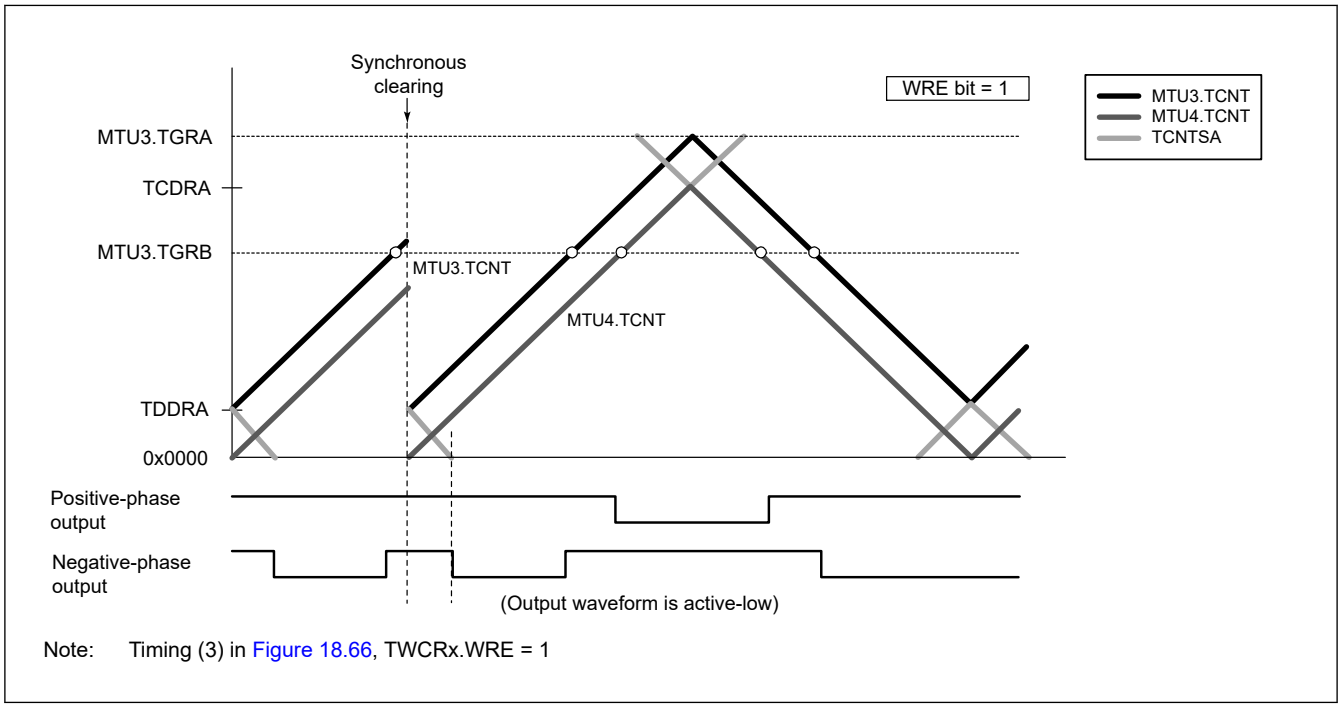


Figure 18.67 Example of procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode (MTU3 and MTU4)

Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

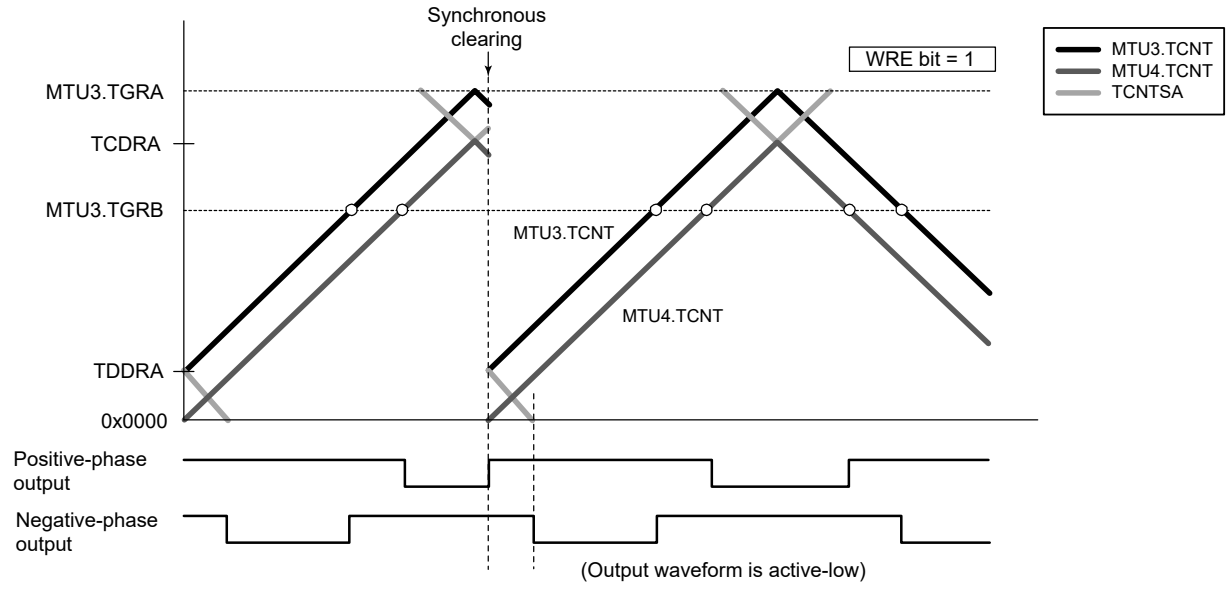
Figure 18.68 to Figure 18.71 show examples of output waveform control in which MTU3 and MTU4 operate in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCRA is set to 1. In the examples shown in Figure 18.68 to Figure 18.71, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 18.66, respectively.

In MTU6 and MTU7, these examples are equivalent to the cases when MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing is generated while the SCC bit is set to 0 and the WRE bit is set to 1 in TWCRB.



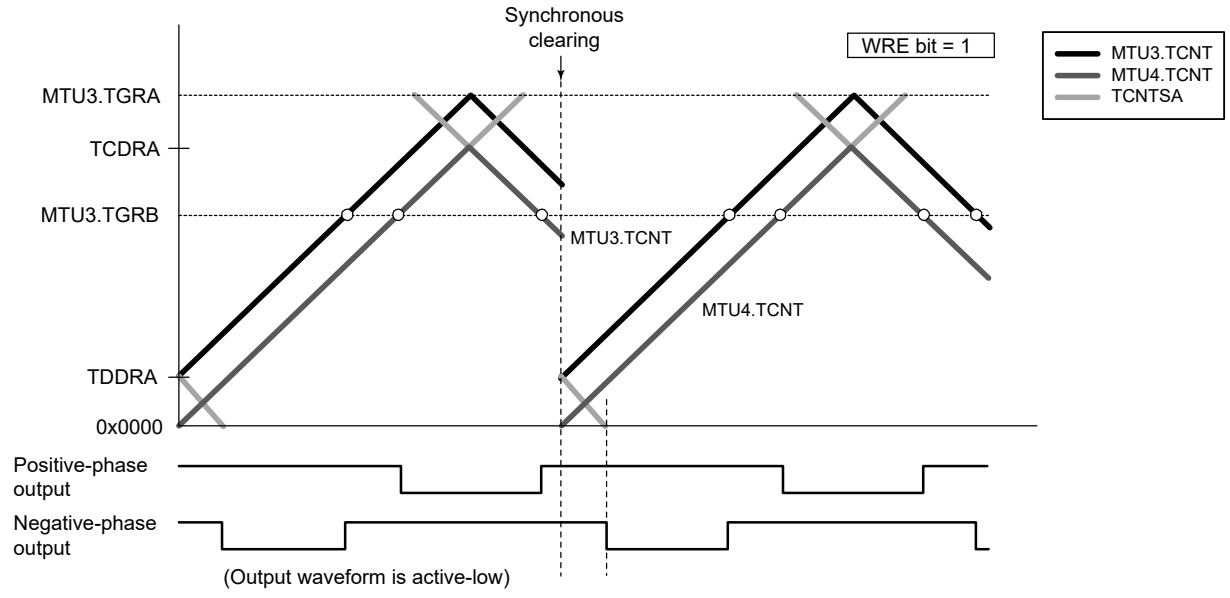
Note: Timing (3) in Figure 18.66, TWCRx.WRE = 1

Figure 18.68 Example of synchronous clearing in dead time during up-counting



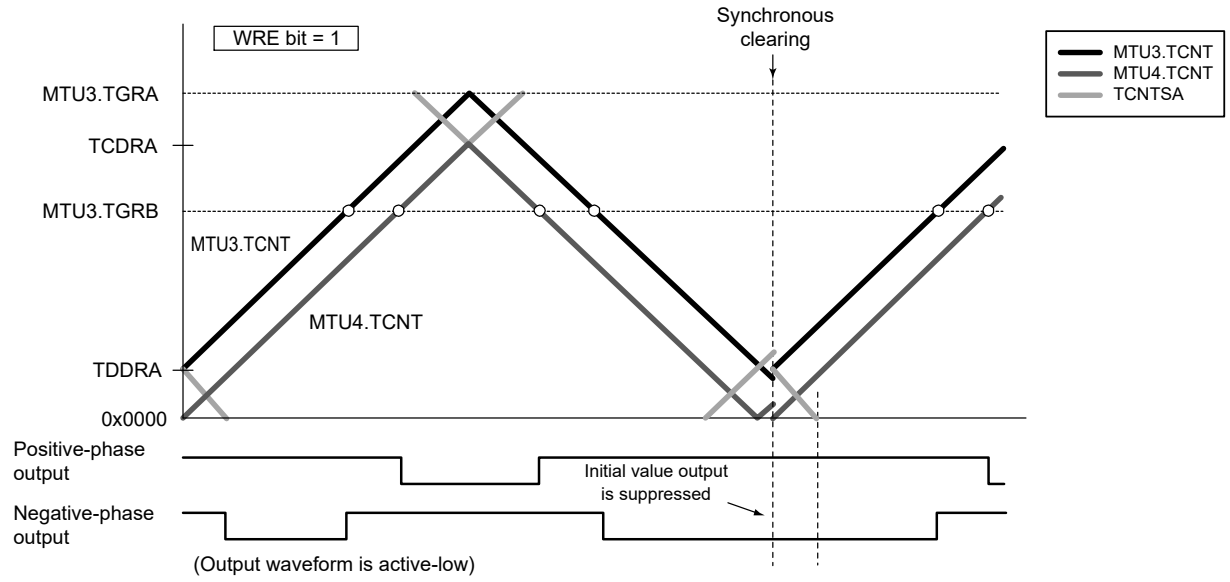
Note: Timing (6) in Figure 18.66, TWCRx.WRE = 1

Figure 18.69 Example of synchronous clearing in interval Tb at crest



Note: Timing (8) in Figure 18.66, TWCRx.WRE = 1

Figure 18.70 Example of synchronous clearing in dead time during down-counting



Note: Timing (11) in Figure 18.66, TWCRx.WRE = 1

Figure 18.71 Example of synchronous clearing in interval Tb at trough

(15) Suppressing Synchronous Counter Clearing by MTU0 to MTU2 in MTU6 and MTU7

In MTU6 and MTU7, setting the SCC bit in TWCRB to 1 suppresses synchronous counter clearing caused by MTU0 to MTU2.

Synchronous counter clearing caused by MTU0 to MTU2 is suppressed only within the interval shown in Figure 18.72. When using this function, MTU6 and MTU7 should be set to complementary PWM mode.

For details of synchronous clearing caused by MTU0 to MTU2, see section 18.4.10.2. Synchronous Counter Clearing for MTU6 and MTU7.

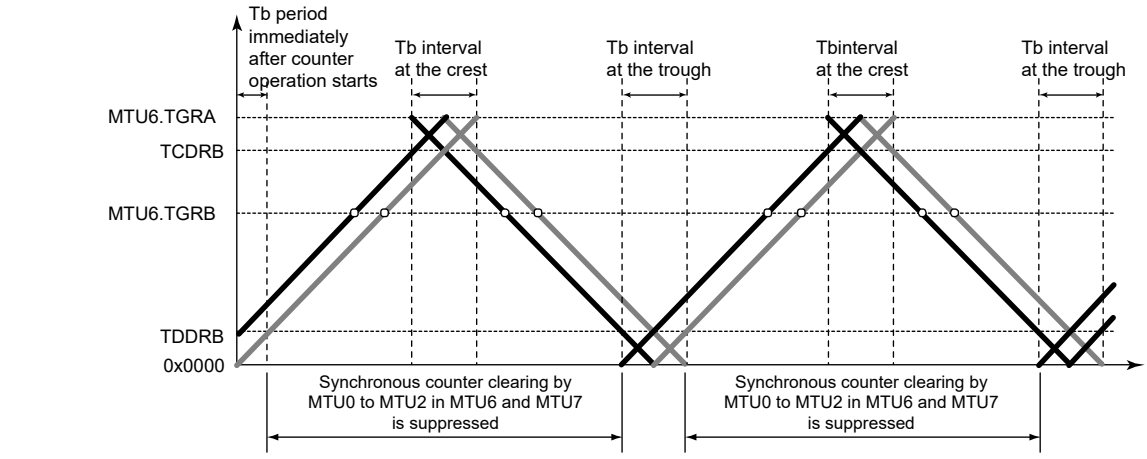
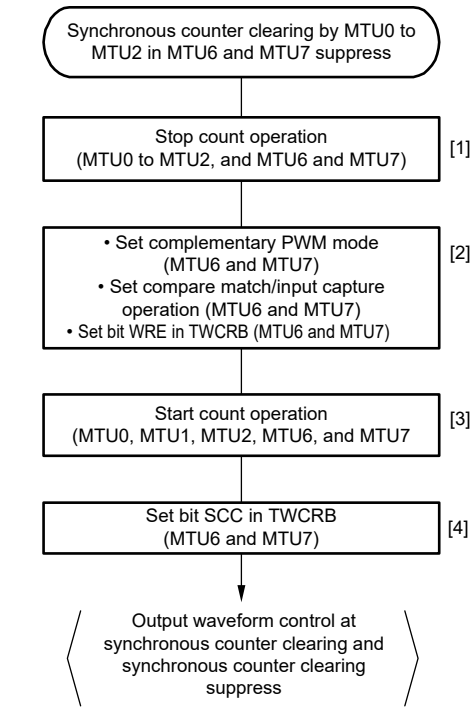


Figure 18.72 Synchronous clearing-suppressed interval specified by TWCRB.SCC bit by MTU0 to MTU2 in MTU6 and MTU7

Example of Procedure for Suppressing Synchronous Counter Clearing by MTU0 to MTU2 in MTU6 and MTU7

An example of the procedure for suppressing synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7 is shown in Figure 18.73.



[1] Clear bits CST of the timer start register (TSTRB) in MTU6 and MTU7 to 0, and stop count operation. Clear bits CST of TSTRA in MTU0, MTU1, and MTU2 to 0, and stop count operation.

[2] Set the complementary PWM mode in MTU6 and MTU7 and compare match/input capture operation in MTU0, MTU1, and MTU2.
When bit WRE in TWCRA or TWCRB should be set, make appropriate setting here.

[3] Set bits CST6 and CST7 in MTU6 and MTU7 to 1 to start count operation.
For synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7, set bits CST of TSTRA in MTU0, MTU1, and MTU2 to 1 to start count operation in any one of MTU0.TCNT to MTU2.TCNT.

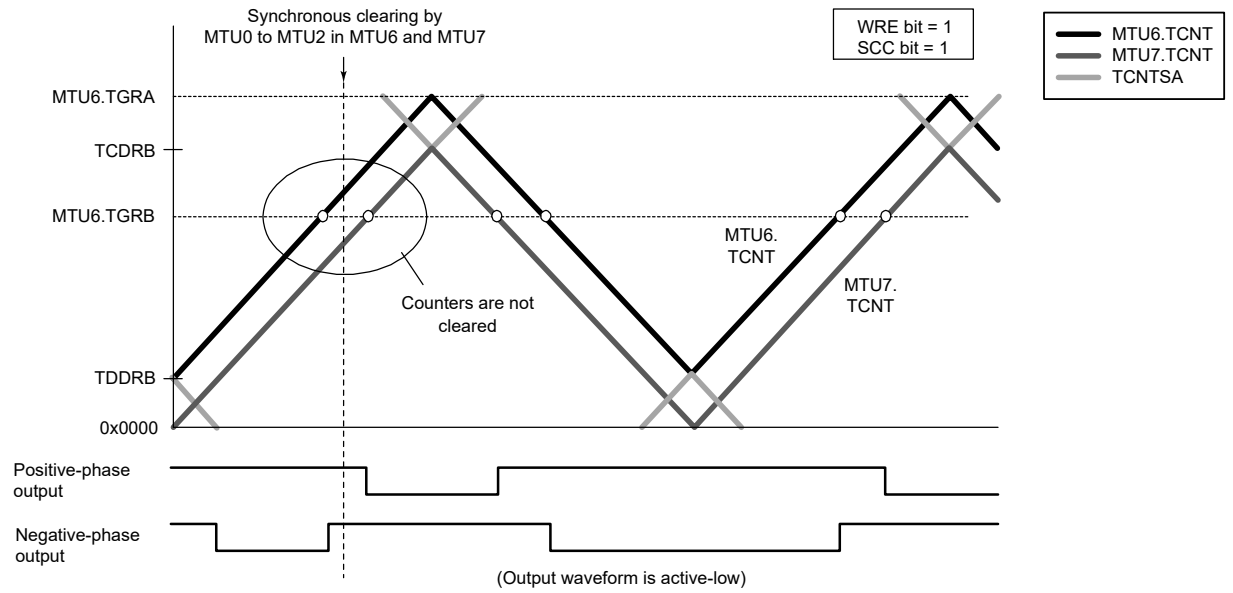
[4] Read TWCRB and then set bit SCC in TWCRB to 1 to suppress synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7*1.
Here, do not modify the CCE and WRE bit values in TWCRB of MTU6 and MTU7. Synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7 is suppressed in the synchronous counter clearing-suppressed periods.

Note 1. The TWCRB.SCC bit value can be modified during counter operation.
However, if a synchronous clearing occurs when bit TWCRB.SCC is modified from 0 to 1, the synchronous clearing may not be suppressed.
If a synchronous clearing occurs when bit TWCRB.SCC is modified from 1 to 0, the synchronous clearing may be suppressed.

Figure 18.73 Example of procedure for suppressing synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7

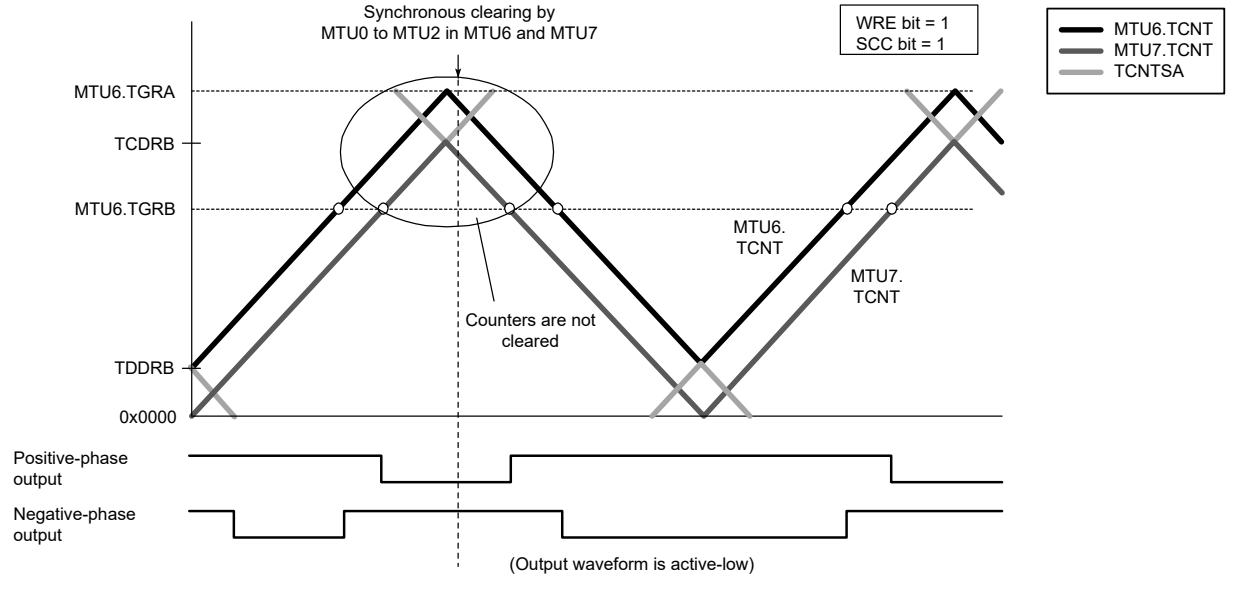
Examples of Suppression of Synchronous Counter Clearing by MTU0 to MTU2 in MTU6 and MTU7

Figure 18.74 to Figure 18.77 show examples of operation in which MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7 is suppressed by setting the SCC bit in TWCRB in MTU6 and MTU7 to 1. In the examples shown in Figure 18.74 to Figure 18.77, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 18.66, respectively. In these examples, the WRE bit in TWCRB is set to 1.



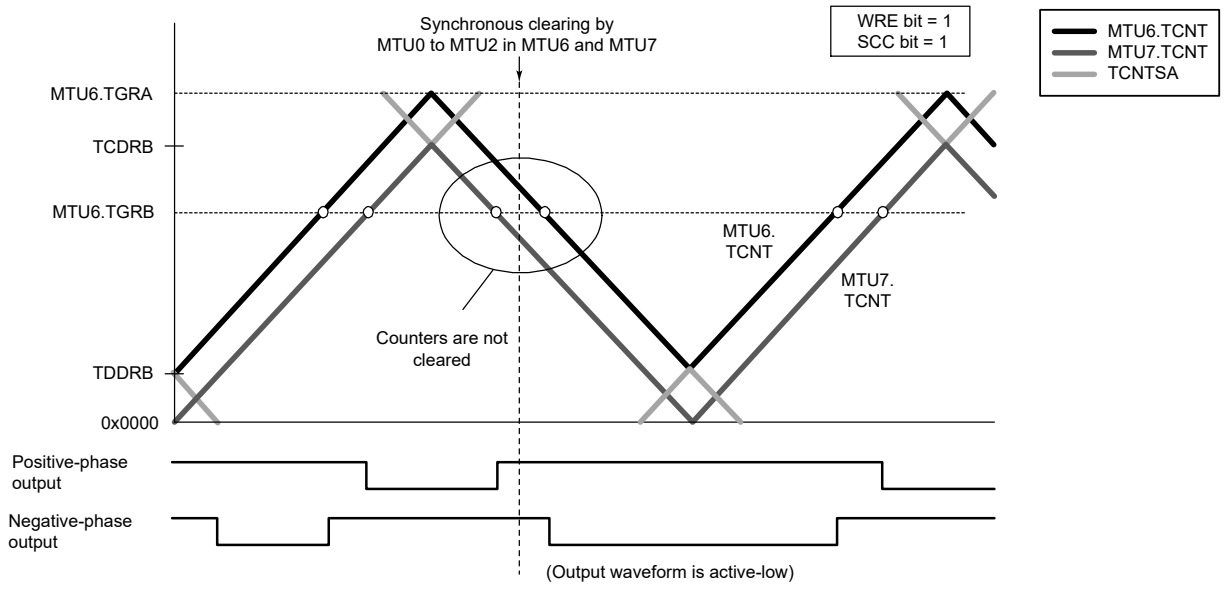
Note: Timing (3) in Figure 18.66, TWCRB.WRE = 1 and TWCRB.SCC = 1 for MTU6 and MTU7

Figure 18.74 Example of synchronous clearing in dead time during up-counting



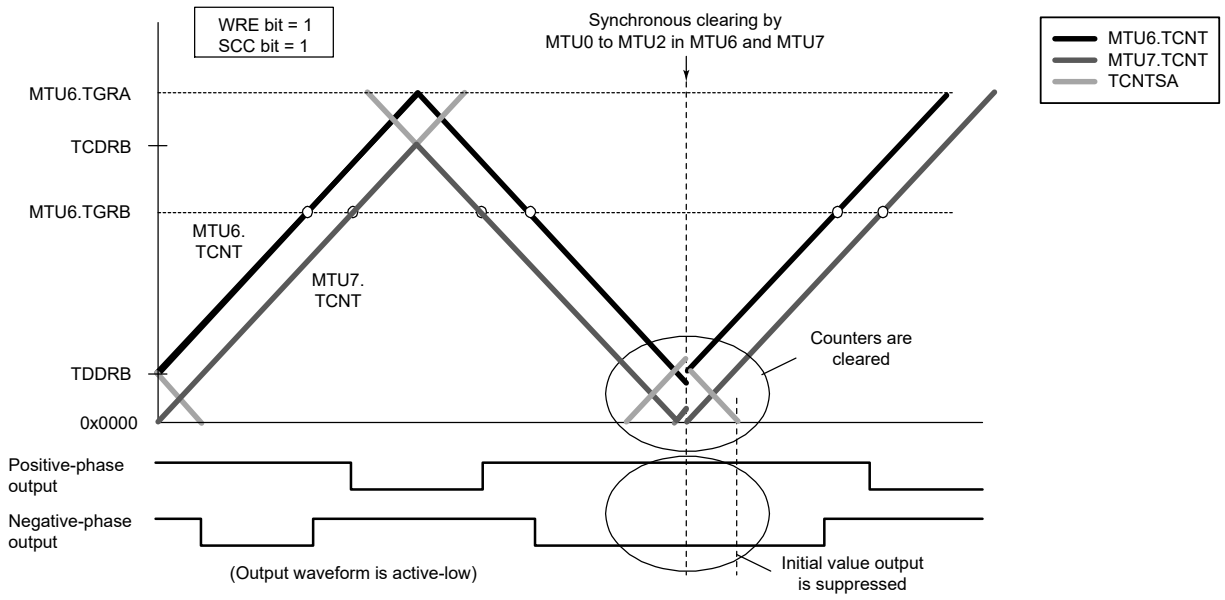
Note: Timing (6) in Figure 18.66, TWCRB.WRE = 1 and TWCRB.SCC = 1 for MTU6 and MTU7

Figure 18.75 Example of synchronous clearing in interval Tb at crest



Note: Timing (8) in Figure 18.66, TWCRB.WRE = 1 and TWCRB.SCC = 1 for MTU6 and MTU7

Figure 18.76 Example of synchronous clearing in dead time during down-counting



Note: Timing (11) in Figure 18.66, TWCRB.WRE = 1 and TWCRB.SCC = 1 for MTU6 and MTU7

Figure 18.77 Example of synchronous clearing in interval Tb at trough

(16) Counter Clearing by MTU3.TGRA (MTU6.TGRA) Compare Match

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by MTU3.TGRA (MTU6.TGRA) compare match when the TWCRA.CCE (TWCRB.CCE) bit is set.

Figure 18.78 shows an operation example.

Note:
- Use this function only in complementary PWM mode 1 (transfer at crest).
- Do not specify synchronous clearing by another channel (do not set 1 in the SYNC0 to SYNC4, or SYNC6 and SYNC7 bits in the timer synchronous register (TSYRA or TSYRB) and the CE0A to CE0D, CE1A, and CE1B, or CE2A and CE2B bits in TSYCR).

- Do not set the PWM duty value to 0x0000.
- Do not set the PSYE bit in timer output control register 1 (TOCR1A or TOCR1B) to 1.

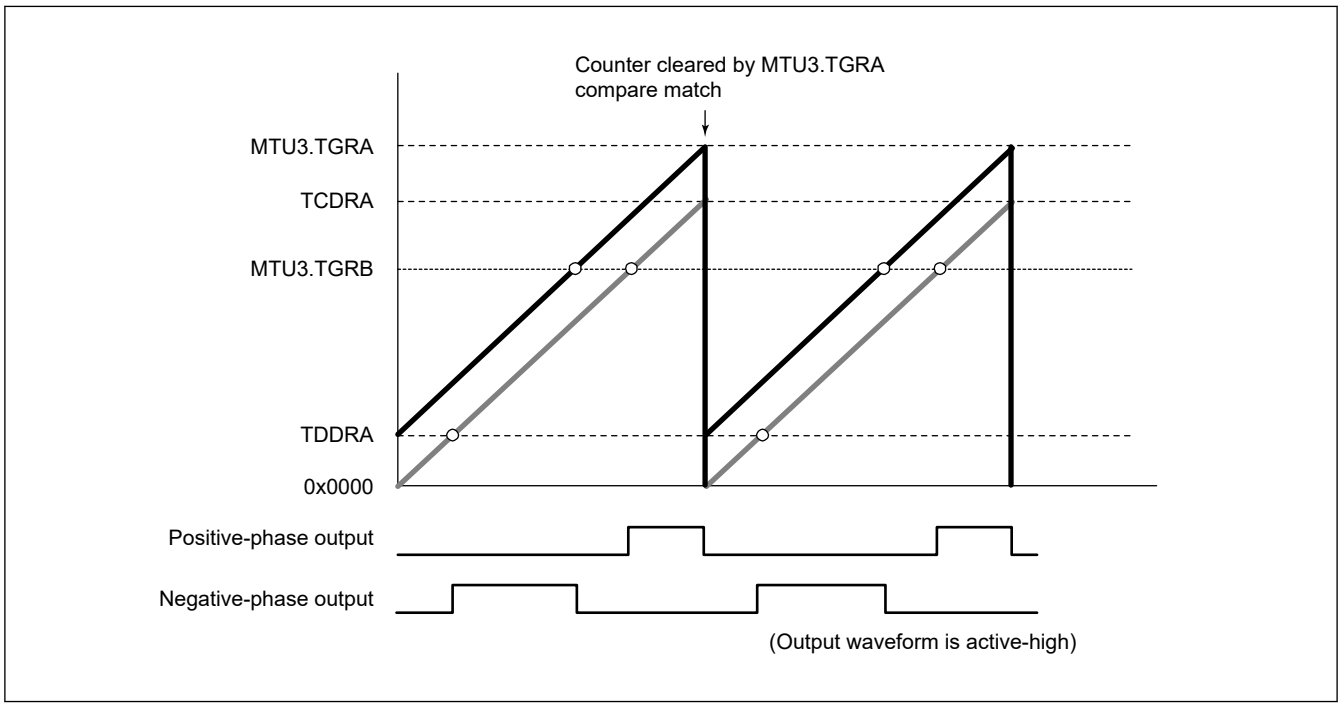


Figure 18.78 Example of counter clearing operation by MTU3.TGRA compare match

(17) Example of Waveform Output for Driving AC Synchronous Motor (Brushless DC Motor)

In complementary PWM mode, a brushless DC motor can easily be controlled using the TGCRA register. Figure 18.79 to Figure 18.82 show examples of brushless DC motor driving waveforms created using TGCRA.

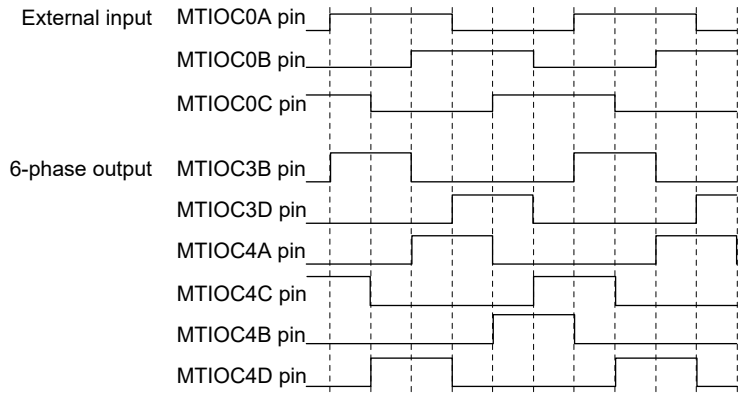
To switch the output phases for a 3-phase brushless DC motor by means of external signals detected with a Hall element, etc., set the TGCRA.FB bit to 0. In this case, the external signals indicating the magnetic pole position should be input to timer input pins MTIOC0A, MTIOC0B, and MTIOC0C in MTU0 (specify appropriate settings with the I/O port registers of the I/O ports). When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.

When the TGCRA.FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCRA is set to 0 or 1.

The driving waveforms are output from the 6-phase output pins for complementary PWM mode.

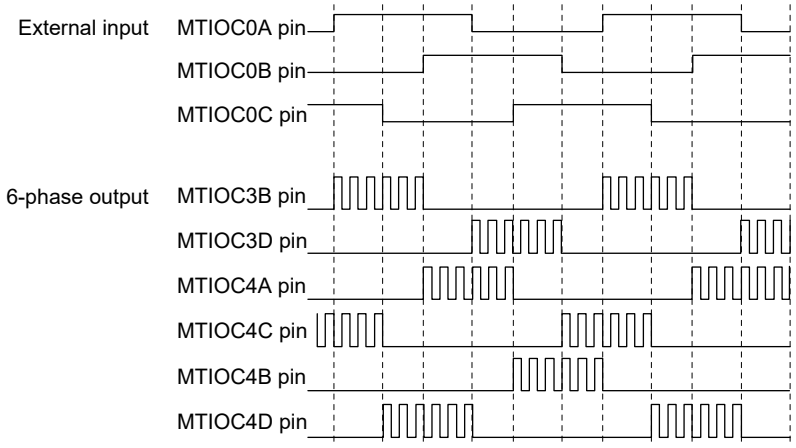
With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the N bit or P bit in TGCRA to 1. When the N bit or P bit is 0, the level output is selected.

The active level of the 6-phase output (on output level) can be set with the TOCR1A.OLSN and TOCR1A.OLSP bits regardless of the setting of the N and P bits.



■ When TGCRA.BDC = 1, TGCRA.N = 0, TGCRA.P = 0, TGCRA.FB = 0, and output active level = high

Figure 18.79 Example of output phase switching by external input (1)



■ When TGCRA.BDC = 1, TGCRA.N = 1, TGCRA.P = 1, TGCRA.FB = 0, and output active level = high

Figure 18.80 Example of output phase switching by external input (2)

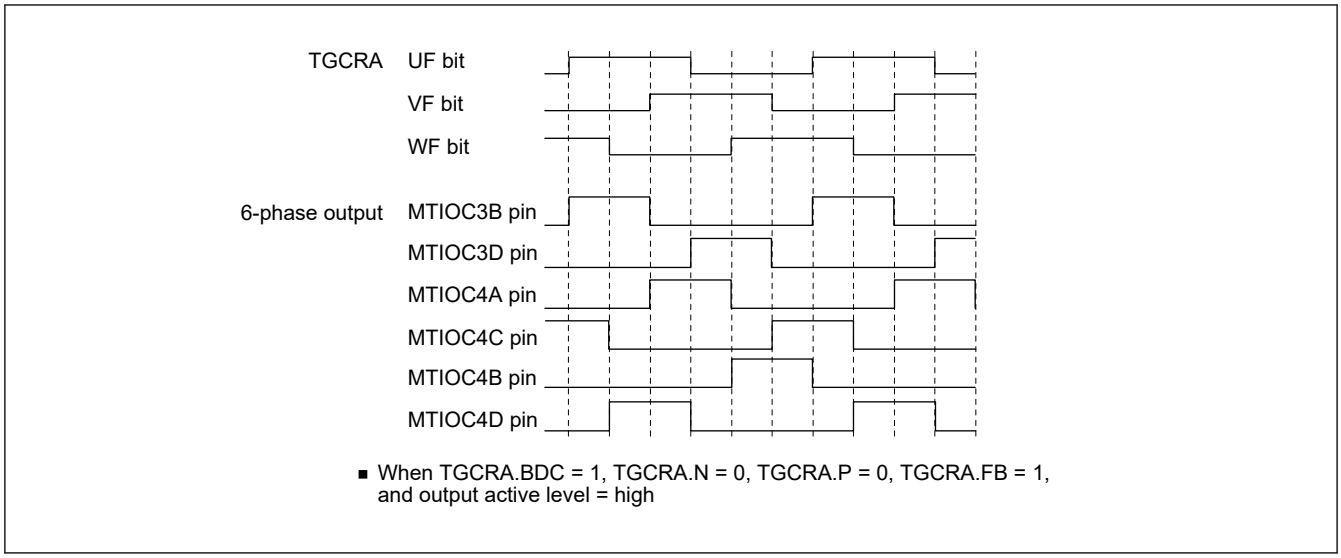


Figure 18.81 Example of output phase switching through UF, VF, and WF bit settings (1)

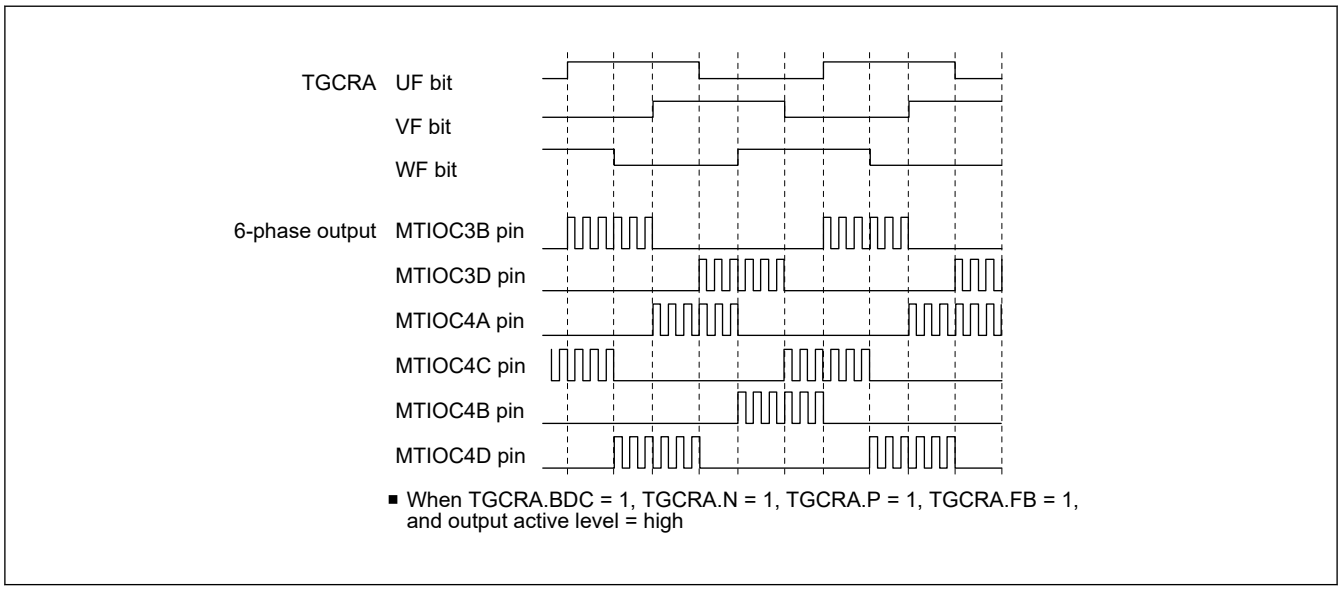


Figure 18.82 Example of output phase switching through UF, VF, and WF bit settings (2)

(18) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using MTU3.TGRA (MTU6.TGRA) compare match, MTU4.TCNT (MTU7.TCNT) underflow (trough), or compare match on a channel other than MTU3 and MTU4 (MTU6 and MTU7).

When start requests using MTU3.TGRA (MTU6.TGRA) compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT (MTU6.TCNT) count.

A/D converter start requests can be specified by setting the TIER.TTGE bit to 1. To issue an A/D converter start request at an MTU4.TCNT (MTU7.TCNT) underflow (trough), set the MTU4.TIER.TTGE2 (MTU7.TIER.TTGE2) bit to 1.

(19) Double Buffer Function in Complementary PWM Mode

In complementary PWM mode 3 (transfer at the crest and trough), the PWM output setting resolution can be improved from ±2 to ±1 by setting the TMDR2A.DRS (TMDR2B.DRS) bit to 1.

When setting buffer registers A (MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD), set also buffer registers B (MTU3.TGRE, MTU4.TGRE, MTU4.TGRF, MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) at the same time. Each buffer register B should be set to the buffer register A value or (buffer register

A value − 1). For details of the setting procedure, see section 18.4.8.1. Example of Complementary PWM Mode Setting Procedure.

Note: When a buffer register B is set to the buffer register A value, symmetric PWM waveforms are output. When a buffer register B is set to (buffer register A value − 1), asymmetric PWM waveforms are output.

Figure 18.83 shows an example of double buffer operation.

Each register data is transferred as follows.

- After MTU4.TGRD or MTU7.TGRD (buffer A) is written to, data is transferred from MTU4.TGRD or MTU7.TGRD (buffer A) to Temp3A or Temp6A (temporary A) and from MTU4.TGRF or MTU7.TGRF (buffer B) to Temp3B or Temp6B (temporary B).
- With timing (1) in the figure, data is transferred from Temp3A or Temp6A (temporary A) to MTU4.TGRB or MTU7.TGRB (compare).
- With timing (2) in the figure, data is transferred from Temp3B or Temp6B (temporary B) to MTU4.TGRB or MTU7.TGRB (compare).

In the crest interval (Tb interval at crest), the compare register and temporary register A are valid; in the trough interval (Tb interval at trough), the compare register and temporary register B are valid.

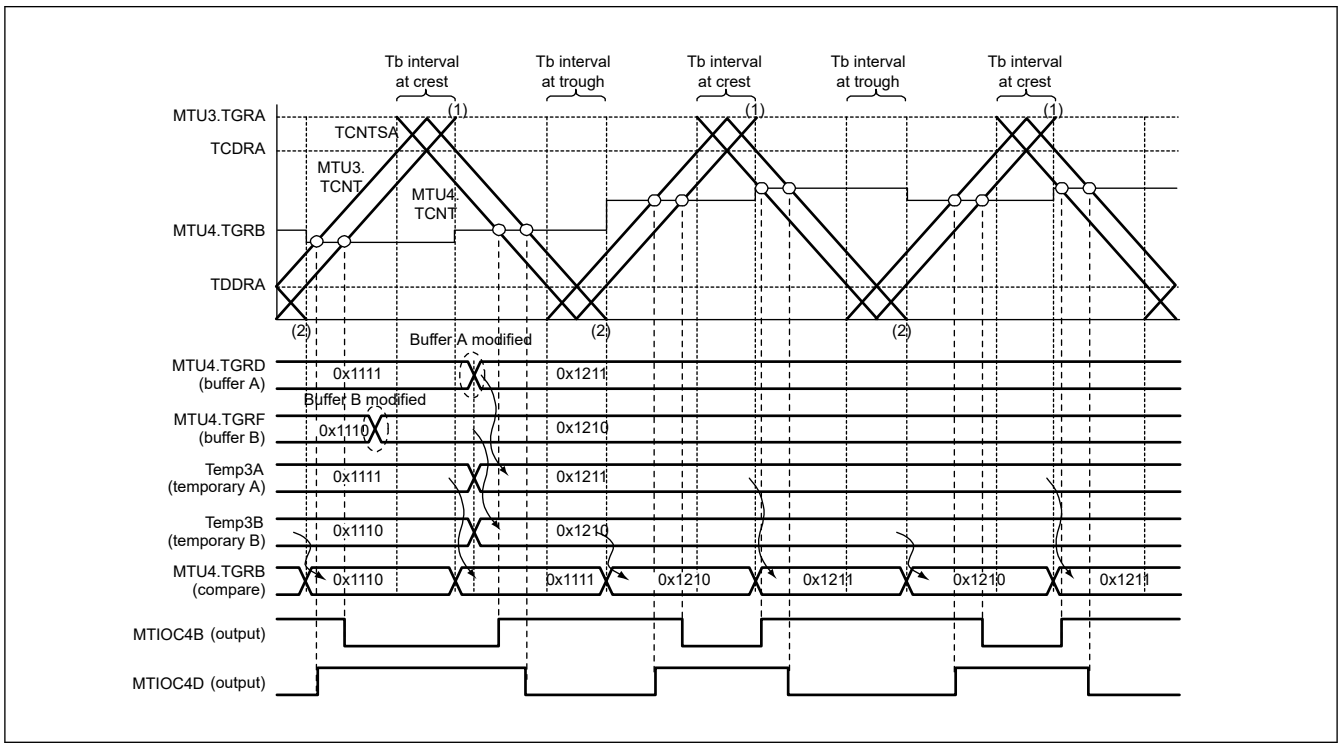


Figure 18.83 Example of double buffer operation

Figure 18.84 shows an example when the buffer write value is smaller than the TDDRA value, and Figure 18.85 shows an example when the write value is greater than TCDRA.

In the crest interval, the output is controlled according to the compare match with the compare register or temporary register A; in the trough interval, the output is controlled according to the compare match with the compare register or temporary register B.

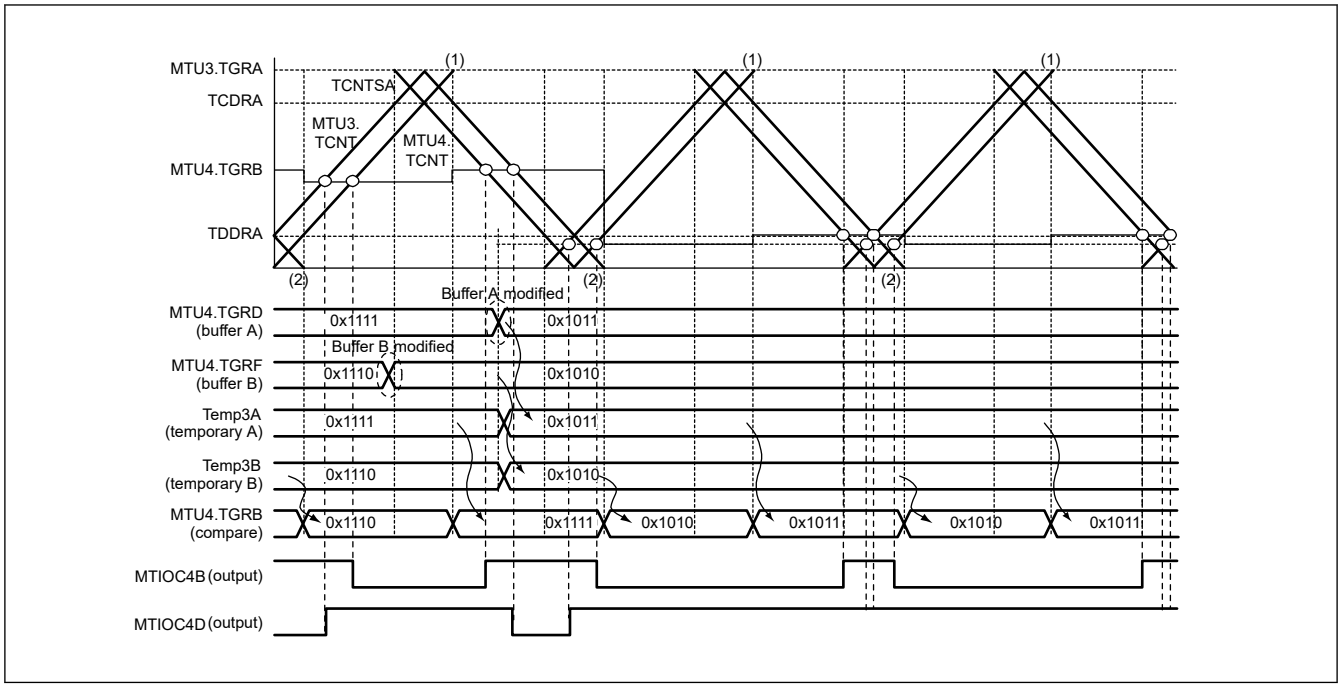


Figure 18.84 Example of double buffer operation (buffer write value is smaller than TDDRA)

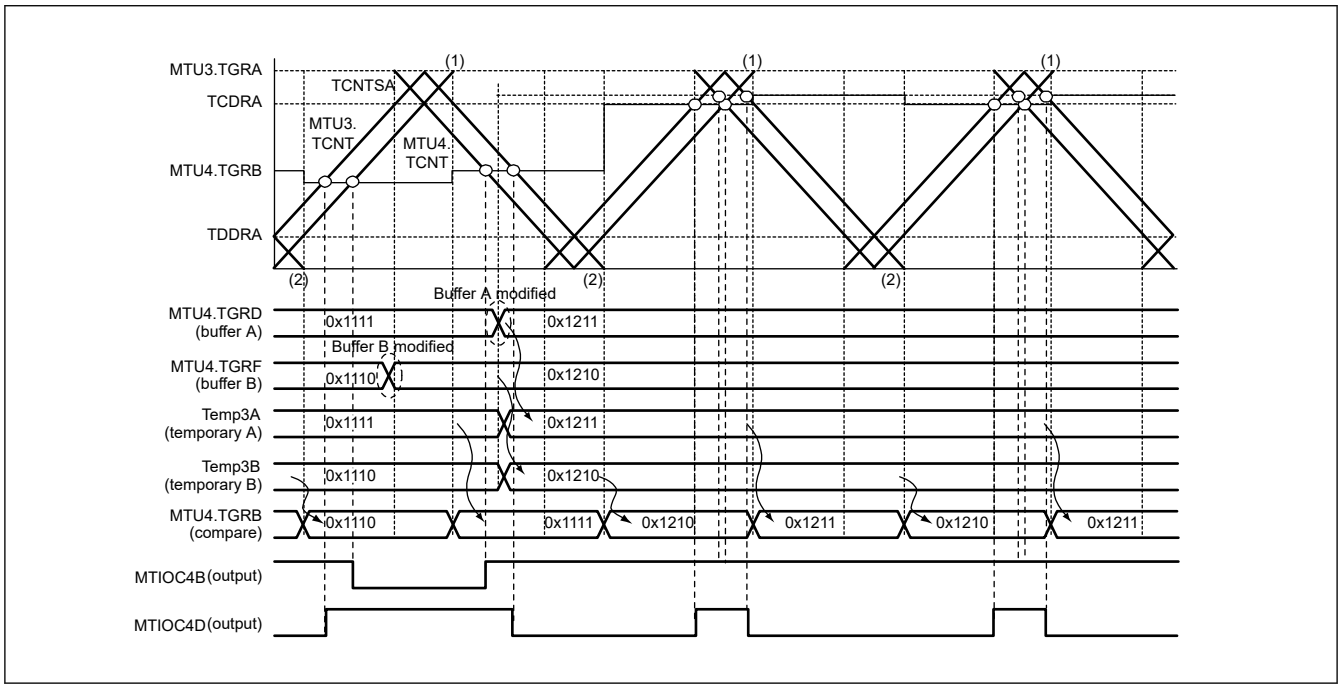


Figure 18.85 Example of double buffer operation (buffer write value is greater than TCDRA)

18.4.8.3 Interrupt Skipping Function 1 in Complementary PWM Mode

Interrupts TGIA3 (TGIA6) (at the crest) and TCIV4 (TCIV7) (at the trough) in MTU3 and MTU4 (MTU6 and MTU7) can be skipped up to seven times by making settings in the TITCR1A (TITCR1B) register.

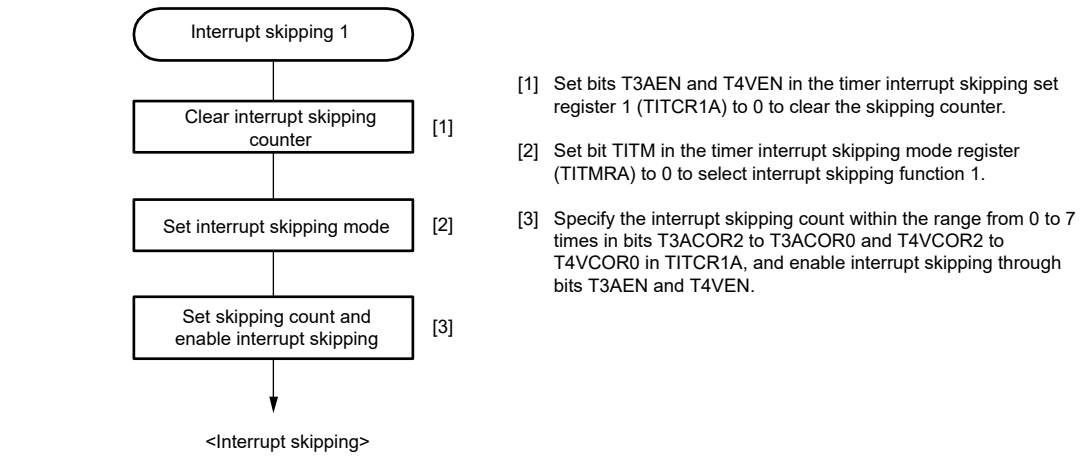
Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the TBTERA (TBTERB) register. For the linkage with buffer registers, see description (3) Buffer Transfer Control Linked with Interrupt Skipping, below.

A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the MTU4.TADCR (MTU7.TADCR) register. For the linkage with the A/D converter start request delaying function, see section 18.4.9. A/D Converter Start Request Delaying Function.

The TITCR1A (TITCR1B) register should be set while interrupt skipping function 1 is selected by setting the TITM bit in the timer interrupt skipping mode register (TITMRA or TITMRB) to 0, TGIA3 (TGIA6) interrupt requests are disabled by setting the MTU3.TIER (MTU6.TIER) register, TCIV4 (TCIV7) interrupt requests are disabled by setting the MTU4.TIER (MTU7.TIER) register, and a compare match is not generated. Before changing the skipping count, be sure to set the T3AEN (T6AEN) and T4VEN (T7VEN) bits to 0 to clear the skipping counter.

(1) Example of Interrupt Skipping Function 1 Setting Procedure

Figure 18.86 shows an example of the interrupt skipping function 1 setting procedure. Figure 18.87 shows the periods during which interrupt skipping count can be changed.



Note: The setting of TITCR1A must be done while the TGIA3 and TCIV4 interrupt requests are disabled by the settings of MTU3.TIER and MTU4.TIER under the conditions in which a compare match is not generated. Before changing the skipping count, be sure to set the T3AEN and T4VEN bits to 0 to clear the skipping counter.

Figure 18.86 Example of interrupt skipping function 1 setting procedure

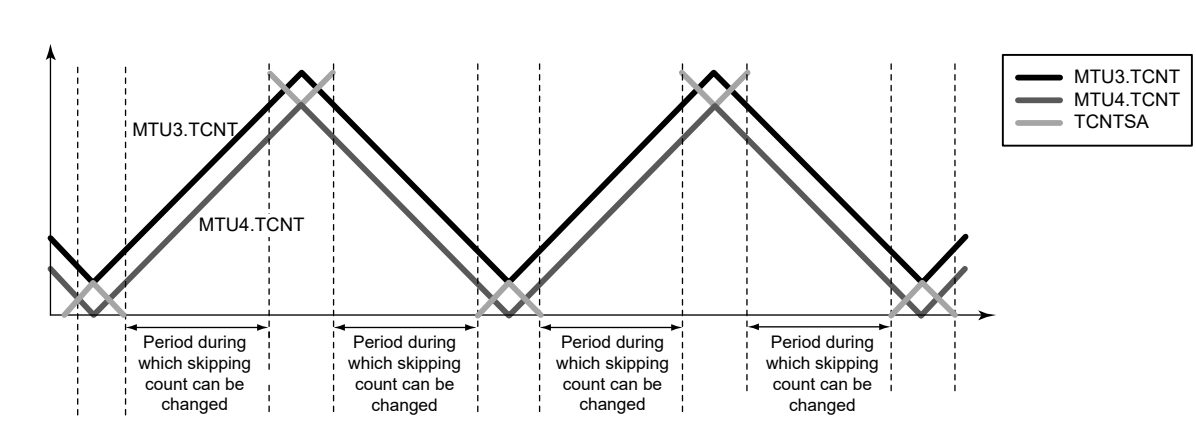


Figure 18.87 Periods during which interrupt skipping count can be changed

(2) Example of Interrupt Skipping Function 1

Figure 18.88 shows an example of TGIA3 (TGIA6) interrupt skipping in which the interrupt skipping count is set to three by the T3ACOR (T6ACOR) bits and the T3AEN (T6AEN) bit is set to 1 in the TITCR1A (TITCR1B) register.

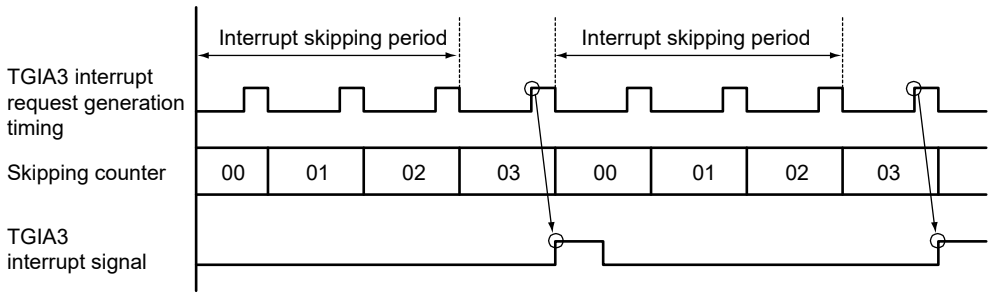


Figure 18.88 Example of interrupt skipping function 1

(3) Buffer Transfer Control Linked with Interrupt Skipping

In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE[1:0] bits in the TBTERA (TBTERB) register.

Figure 18.89 shows an example of operation when buffer transfer is disabled (BTE[1:0] = 01b). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 18.90 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b).

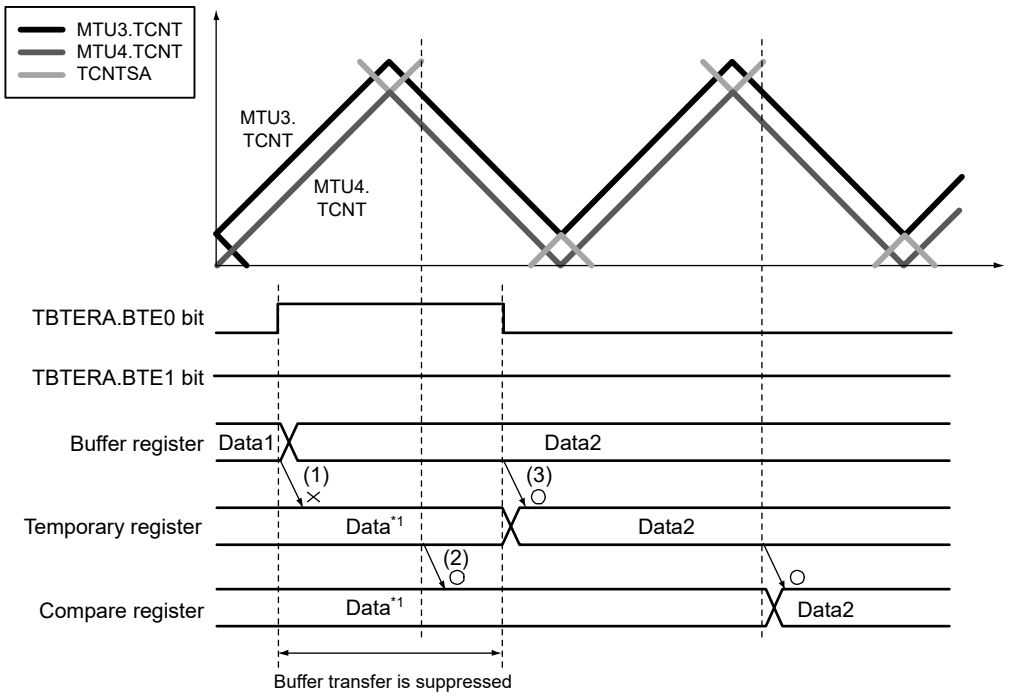
While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period.

Note that the buffer transfer-enabled period depends on the T3AEN (T6AEN) and T4VEN (T7VEN) bit settings in the TITCR1A (TITCR1B) register. Figure 18.91 shows the relationship between the T3AEN (T6AEN) and T4VEN (T7VEN) bit settings in TITCR1A (TITCR1B) and buffer transfer-enabled period.

Note: This function must be used in combination with interrupt skipping function 1.

When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register 1 (TITCR1A or TITCR1B) are set to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), make sure that buffer transfer is not linked with interrupt skipping (set the BTE1 bit in TBTERA or TBTERB to 0).

If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.

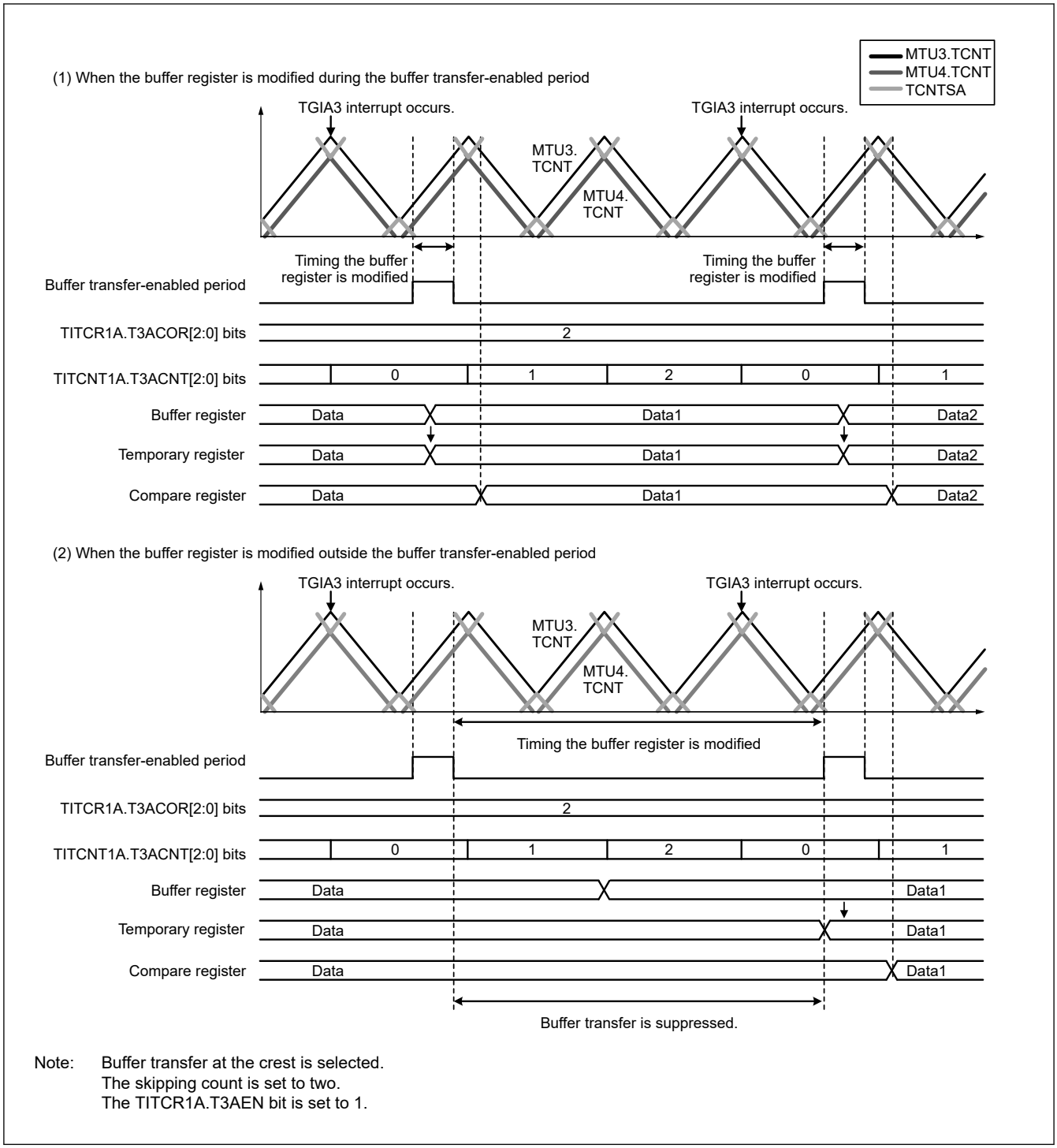


[Legend]

(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE[1:0] in TBTERA are set to 01b).
(2) Data is transferred from the temporary register to the compare register even in the buffer transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

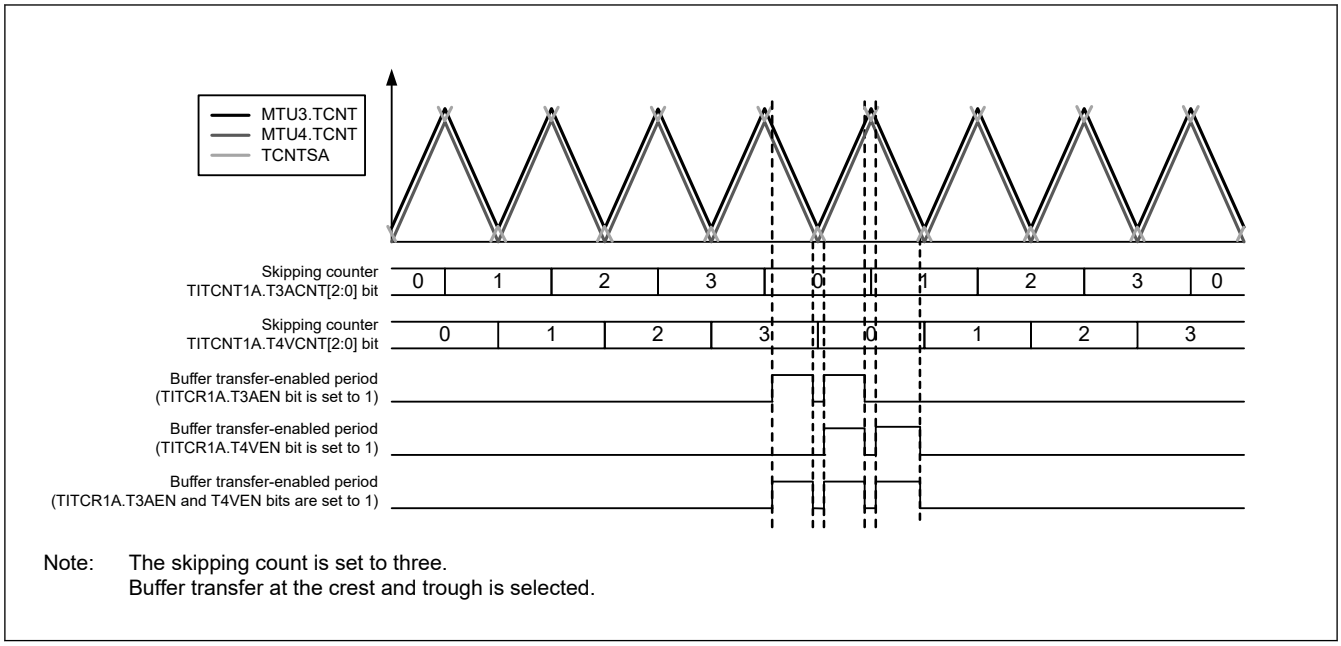
Note 1. When buffer transfer at the crest is selected.

Figure 18.89 Example of operation when buffer transfer is disabled (BTE[1:0] = 01b)



Note: Buffer transfer at the crest is selected.
The skipping count is set to two.
The TITCR1A.T3AEN bit is set to 1.

Figure 18.90 Example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b)



Note: The skipping count is set to three.
Buffer transfer at the crest and trough is selected.

Figure 18.91 Relationship between bits T3AEN and T4VEN in TITCR1A and buffer transfer-enabled period

18.4.8.4 Complementary PWM Mode Output Protection Functions

The MTU provides the following protection functions for complementary PWM mode output.

(1) Register and Counter Miswrite Prevention Function

With the exception of the buffer registers, which can be modified, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of the RWE bit in the timer read/write enable register (TRWERA or TRWERB). The applicable registers are some of the registers in MTU3, MTU4, MTU6, and MTU7 shown below:

47 registers in total

MTU3.TCR, MTU4.TCR, MTU3.TCR2, MTU4.TCR2, MTU3.TMDR1, MTU4.TMDR1, MTU3.TIORH, MTU4.TIORH, MTU3.TIORL, MTU4.TIORL, MTU3.TIER, MTU4.TIER, MTU3.TCNT, MTU4.TCNT, MTU3.TGRA, MTU4.TGRA, MTU3.TGRB, MTU4.TGRB, MTU.TOEERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA, MTU6.TCR, MTU7.TCR, MTU6.TCR2, MTU7.TCR2, MTU6.TMDR1, MTU7.TMDR1, MTU6.TIORH, MTU7.TIORH, MTU6.TIORL, MTU7.TIORL, MTU6.TIER, MTU7.TIER, MTU6.TCNT, MTU7.TCNT, MTU6.TGRA, MTU7.TGRA, MTU6.TGRB, MTU7.TGRB, MTU.TOERB, MTU.TOCR1B, MTU.TOCR2B, MTU.TCDRB, and MTU.TDDRB

This function can disable CPU access to the mode registers, control registers, and counters to prevent miswriting due to CPU runaway. In the access-disabled state, the applicable registers are read as undefined and writing to these registers is ignored.

(2) Halting of PWM Output by External Signal

The 6-phase PWM output pins can be set to the high-impedance state automatically by inputting specified external signals.

For details, see section 19, Port Output Enable 3 (POE3).

18.4.9 A/D Converter Start Request Delaying Function

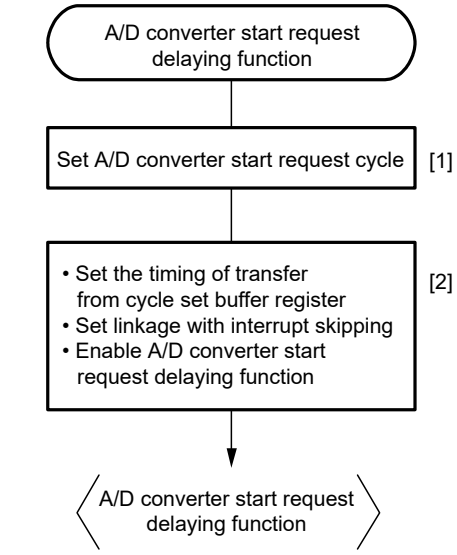
A/D converter start requests can be issued in MTU4 or MTU7 by making settings in the timer A/D converter start request control register (MTU4.TADCR or MTU7.TADCR), timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB), and timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB).

The A/D converter start request delaying function compares MTU4.TCNT with MTU4.TADCORA or MTU4.TADCORB (MTU7.TCNT with MTU7.TADCORA or MTU7.TADCORB), and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN (TRG7AN or TRG7BN)).

A/D converter start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in MTU4.TADCR (the ITA6AE, ITA7VE, ITB6AE, and ITB7VE bits in MTU7.TADCR).

18.4.9.1 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 18.92 shows an example of procedure for specifying the A/D converter start request delaying function.



[1] Set the cycle in the timer A/D converter start request cycle set buffer register (MTU4.TADCOBRA or MTU4.TADCOBRB) and timer A/D converter start request cycle register (MTU4.TADCORA or MTU4.TADCORB).
(When the timing of transfer from the cycle buffer register to cycle set register is set to the trough, or both trough and crest in step [2] below, the same initial value must be specified in the cycle buffer register and cycle register because transfer is performed immediately after counting starts.)

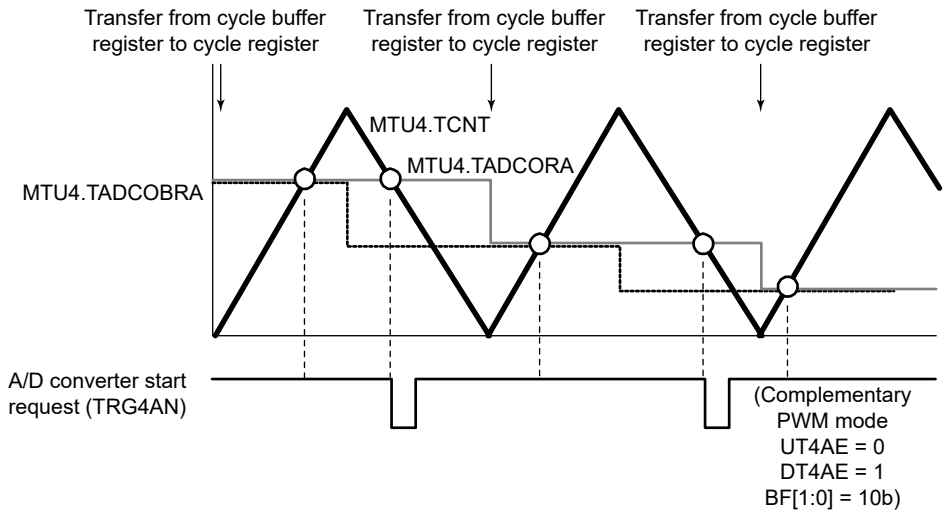
[2] Use the BF[1:0] bits in the timer A/D converter start request control register (MTU4.TADCR) to specify the timing of transfer from the timer A/D converter start request cycle set buffer register to A/D converter start request cycle set register.
- Specify whether to link with interrupt skipping through bits ITA3AE, ITA4VE, ITB3AE, and ITB4VE.
- Use bits UT4AE, DT4AE, UT4BE, and DT4BE to enable A/D converter start requests (TRG4AN or TRG4BN).

Note: Perform MTU4.TADCR setting while MTU4.TCNT is stopped.
Note: Set MTU4.TADCR.BF[1] to 0 when complementary PWM mode is not selected.
Note: Set ITA3AE, ITA4VE, ITB3AE, ITB4VE, DT4AE, and DT4BE to 0 when complementary PWM mode is not selected.

Figure 18.92 Example of procedure for specifying A/D converter start request delaying function (MTU3 and MTU4)

18.4.9.2 Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 18.93 shows a basic example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when the trough of MTU4.TCNT (MTU7.TCNT) is specified for the buffer transfer timing and an A/D converter start request signal is output during MTU4.TCNT (MTU7.TCNT) down-counting.



Note: Period in which A/D converter start requests are enabled (UT4AE = 1): 0 ≤ MTU4.TCNT ≤ TCDR − 1
Period in which A/D converter start requests are enabled (DT4AE = 1): TCDR ≥ MTU4.TCNT ≥ 1
UT4AE, DT4AE, BF[1:0]: Bits in TADCR

Figure 18.93 Basic example of A/D converter start request signal (TRG4AN) operation

18.4.9.3 Buffer Transfer

The data in the timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB) is updated by writing data to the timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF[1:0] bits in the MTU4.TADCR (MTU7.TADCR) register.

In complementary PWM mode, data is also transferred from the timer A/D converter start request cycle set buffer registers to the timer A/D converter start request cycle set registers when MTU4.TGRD (MTU7.TGRD) register is updated.

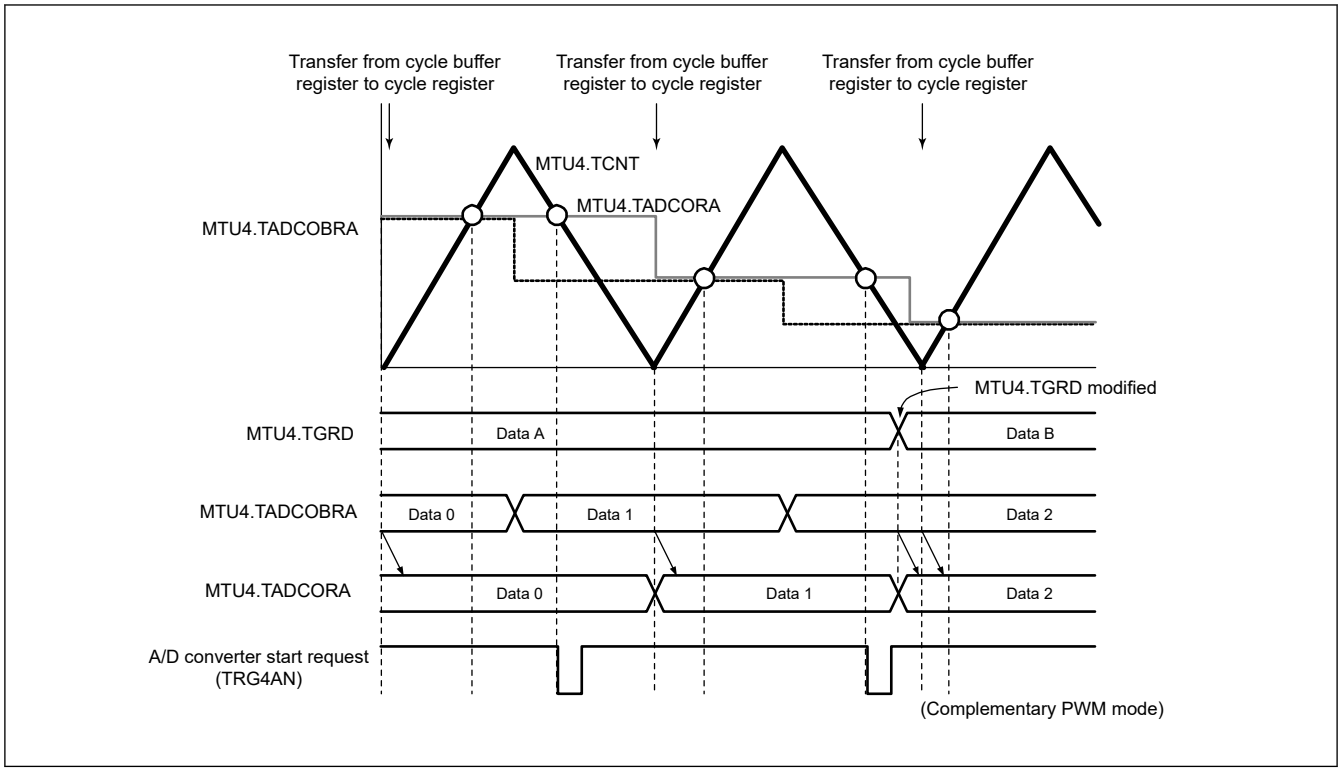


Figure 18.94 Example of A/D converter start request signal (TRG4AN) and buffer transfer operation

18.4.9.4 A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 1

A/D converter start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be issued in coordination with interrupt skipping 1 by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the MTU4.TADCR (MTU7.TADCR) register.

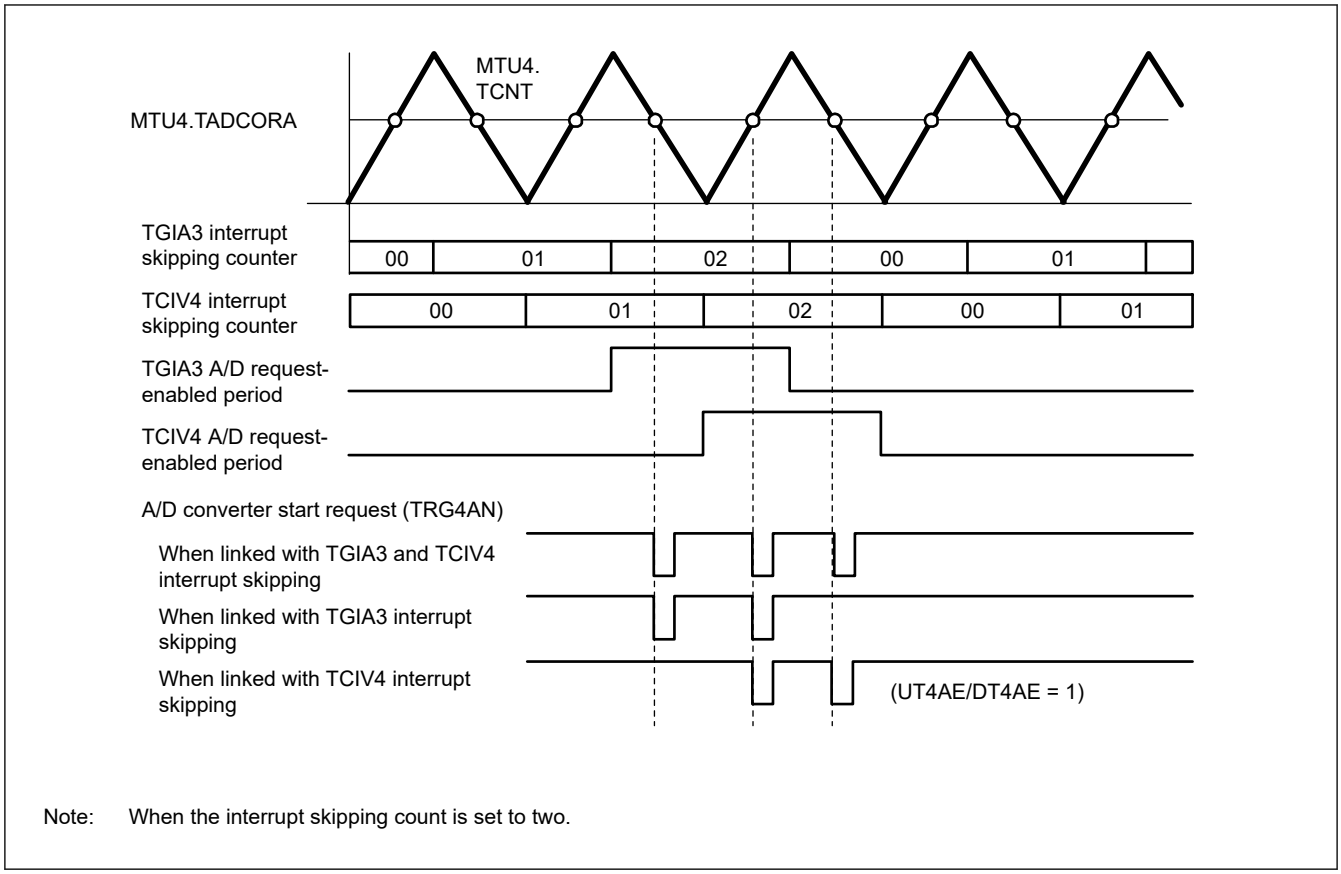
Figure 18.95 shows an example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and down-counting and A/D converter start requests are linked with interrupt skipping 1.

Figure 18.96 shows another example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and A/D converter start requests are linked with interrupt skipping 1.

Note: This function should be used in combination with interrupt skipping 1.

When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register (TITCR1A (TITCR1B)) are set to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), make sure that A/D converter start requests are not linked with interrupt skipping 1 (set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the timer A/D converter start request control register (MTU4.TADCR (MTU7.TADCR)) to 0).

When this function is used, MTU4.TADCORA and MTU4.TADCORB (MTU7.TADCORA and MTU7.TADCORB) should be set with the value ranging 0x0002 to the value set in TCDRA minus 2 (value set in TCDRB minus 2).



Note: When the interrupt skipping count is set to two.

Figure 18.95 Example of A/D converter start request signal (TRG4AN) operation linked with interrupt skipping function 1 (UT4AE and DT4AE = 1)

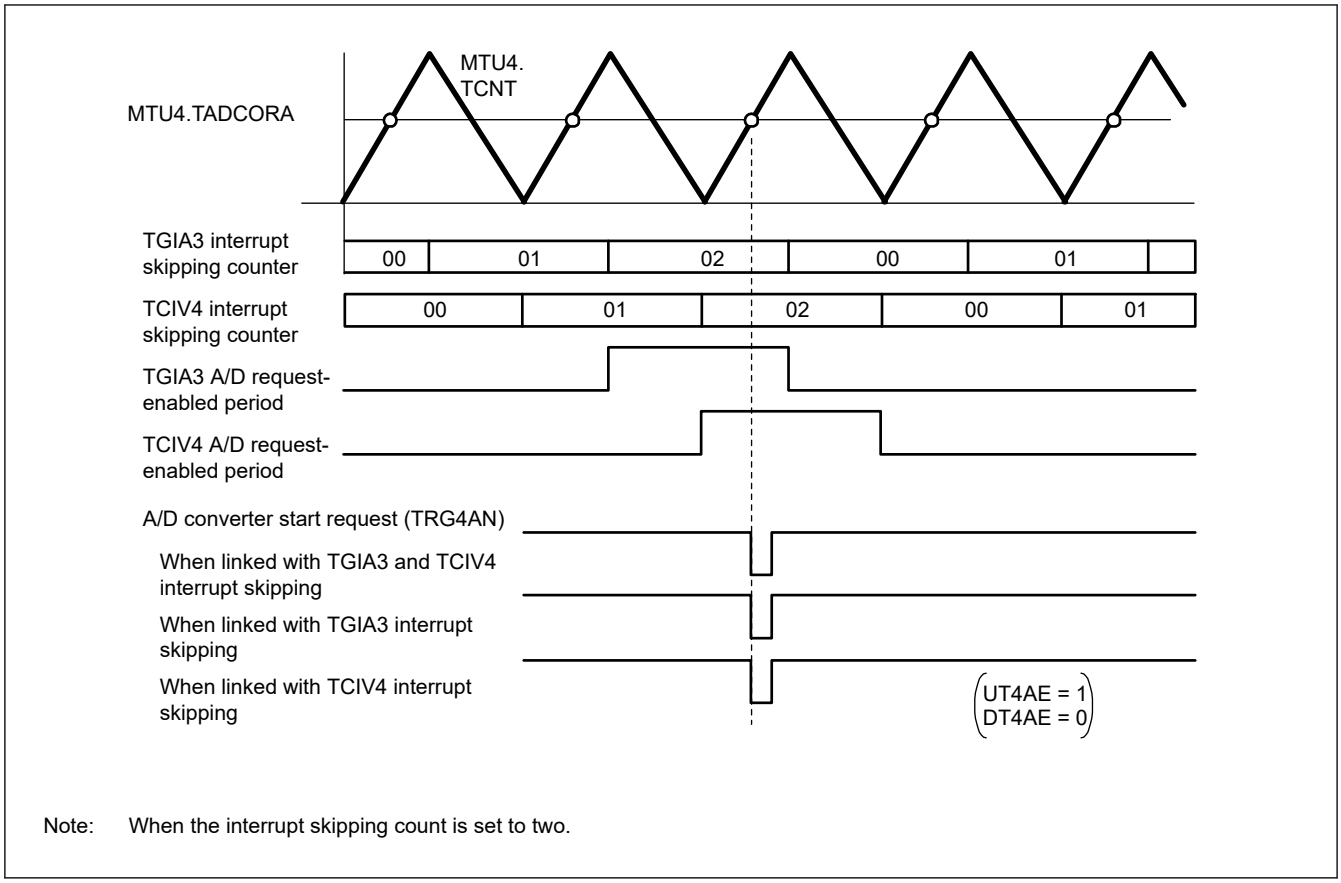


Figure 18.96 Example of A/D converter start request signal (TRG4AN) operation linked with interrupt skipping function 1 (UT4AE = 1, DT4AE = 0)

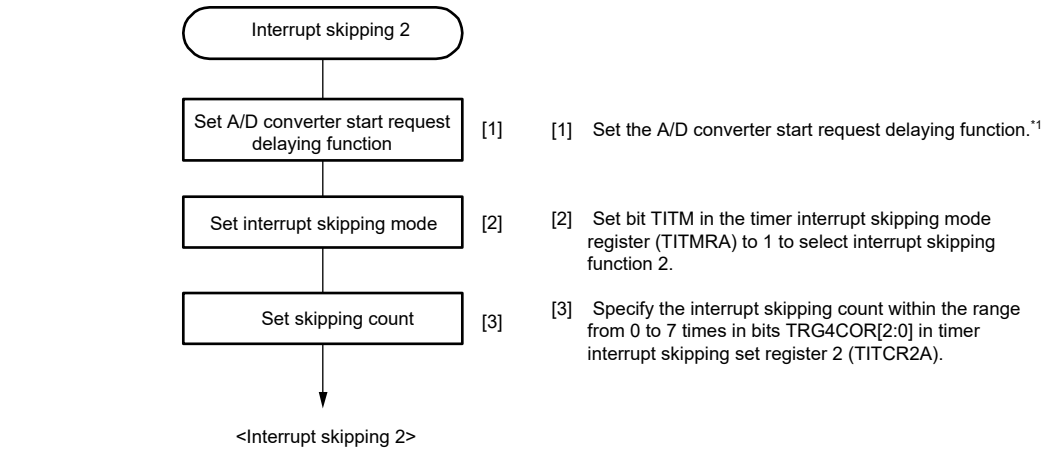
18.4.9.5 A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 2

By setting the TITM bit to 1 in the TITMRA (TITMRB) register, the counter starts down-counting from the value (0 to 7) set in the TRG4COR[2:0] (TRG7COR[2:0]) bits in TITCR2A (TITCR2B) register every time an A/D converter start trigger (TRG4AN or TRG4BN (TRG7AN or TRG7BN)) is generated. When the counter value reaches 0 and is reloaded, the TRG4AN and TRG4BN (TRG7AN and TRG7BN) interrupts become valid and an A/D converter start request signal (TRG4ABN (TRG7ABN)) is output.

This function is valid only when the A/D converter request delaying function is enabled.

(1) Example of Procedure for Setting Interrupt Skipping Function 2

Figure 18.97 shows an example of procedure for setting interrupt skipping function 2.



Note 1. For details, see section 18.4.9.1. Example of Procedure for Specifying A/D Converter Start Request Delaying Function.

Figure 18.97 Example of procedure for setting interrupt skipping function 2

(2) Example of Interrupt Skipping Function 2 Operation

Figure 18.98 shows an example of interrupt skipping function 2 operation.

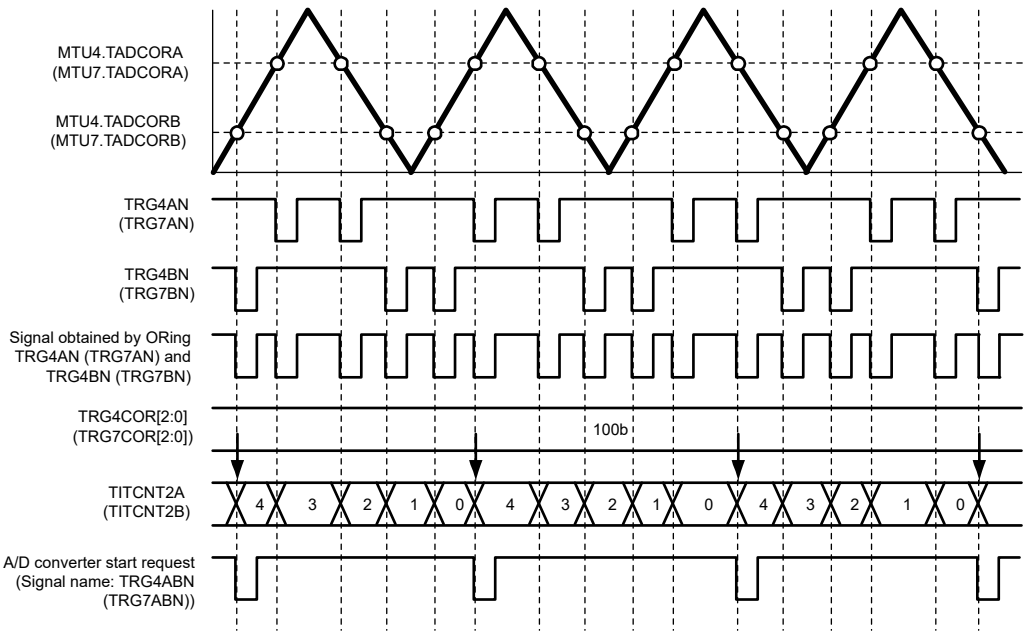


Figure 18.98 Example of interrupt skipping function 2 operation (skipping count is set to four)

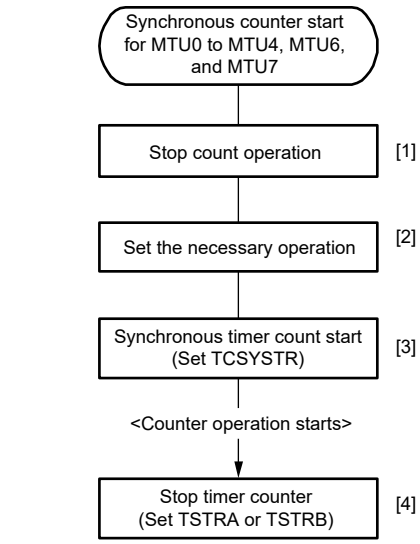
18.4.10 Synchronous Operation of MTU0 to MTU4, MTU6, and MTU7

18.4.10.1 Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

The counters in MTU0 to MTU4, MTU6, and MTU7 can be started synchronously by making the TCSYSTR settings.

(1) Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

Figure 18.99 shows an example of procedure for setting synchronous counter start for MTU0 to MTU4, MTU6, and MTU7.



[1] Set TSTRA and TSTRB to stop the counters used for synchronous start operation.

[2] Specify the necessary operation with appropriate registers such as TCR, TMDR1, and TMDR2.

[3] In TCSYSTR, set the bits corresponding to the counters to be started synchronously to 1 at the same time. When TCSYSTR is set, TSTRA and TSTRB for the target timer counters are automatically set appropriately, the counters start synchronously, and TCSYSTR is automatically cleared.

[4] To stop a timer counter, set the bit in TSTRA or TSTRB that corresponds to the counter to 0.

Note:
- Even if a bit in TCSYSTR corresponding to an operating counter is set to 0, the counter will not stop.
- To start counters synchronously in reset-synchronized PWM mode or complementary PWM mode using MTU3 and MTU4 or MTU6 and MTU7, set the bits in TCSYSTR that correspond to MTU3, MTU4, MTU6, and MTU7 to 1 at the same time.

Figure 18.99 Example of procedure for setting synchronous counter start for MTU0 to MTU4, MTU6, and MTU7

(2) Examples of Synchronous Counter Start Operation

Figure 18.100 shows an examples of synchronous counter start operation for MTU0 to MTU4, MTU6, and MTU7.

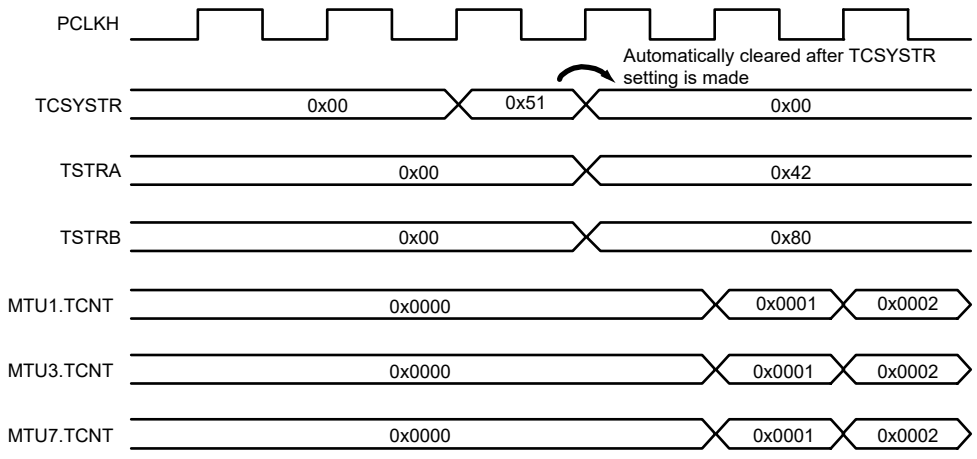


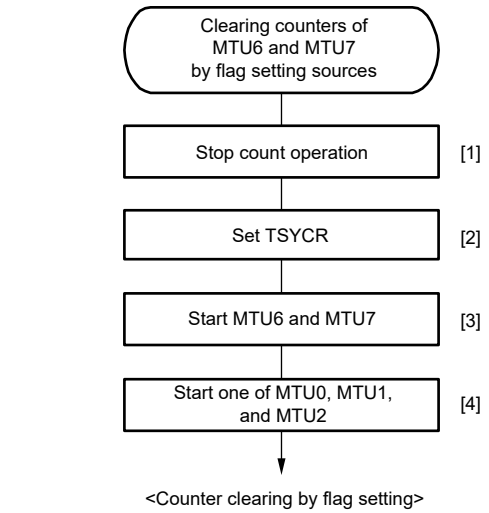
Figure 18.100 Examples of synchronous counter start operation for MTU0 to MTU4, MTU6, and MTU7

18.4.10.2 Synchronous Counter Clearing for MTU6 and MTU7

The counters in MTU6 and MTU7 can be cleared by the TGImn interrupt generation timing (m = A to D; n = 0 to 2) through the TSYCR setting.

(1) Example of Procedure for Specifying Synchronous Counter Clearing for MTU6 and MTU7

Figure 18.101 shows an example of procedure for specifying synchronous counter clearing for MTU6 and MTU7 by flag setting sources.



[1] Set TSTRA and TSTRB to stop the counters used for this function.

[2] Use TSYCR to specify the source to be used for the MTU6.TCNT and MTU7.TCNT counter clearing.

[3] Start MTU6.TCNT or MTU7.TCNT.

[4] Start MTU0.TCNT, MTU1.TCNT, or MTU2.TCNT.

Note: The TSYCR setting is ignored while the counter is stopped. The setting becomes valid after MTU6.TCNT or MTU7.TCNT is started.

Figure 18.101 Example of procedure for specifying synchronous counter clearing for MTU6 and MTU7

(2) Examples of Synchronous Counter Clearing for MTU6 and MTU7

Figure 18.102 and Figure 18.103 show examples of synchronous counter clearing for MTU6 and MTU7 by flag setting sources.

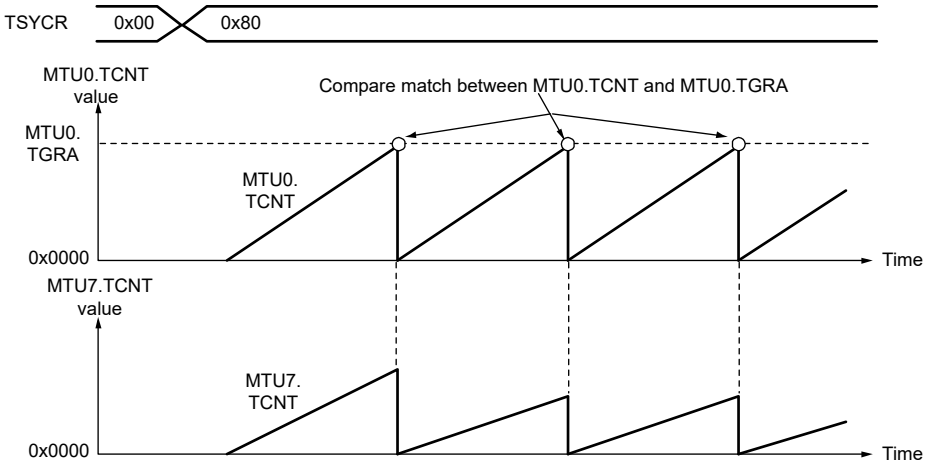


Figure 18.102 Example of synchronous counter clearing for MTU6 and MTU7 (1)

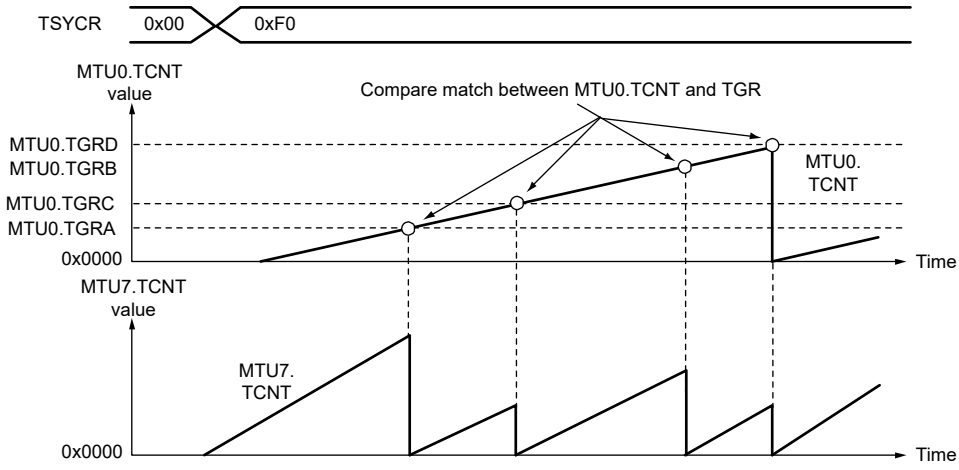


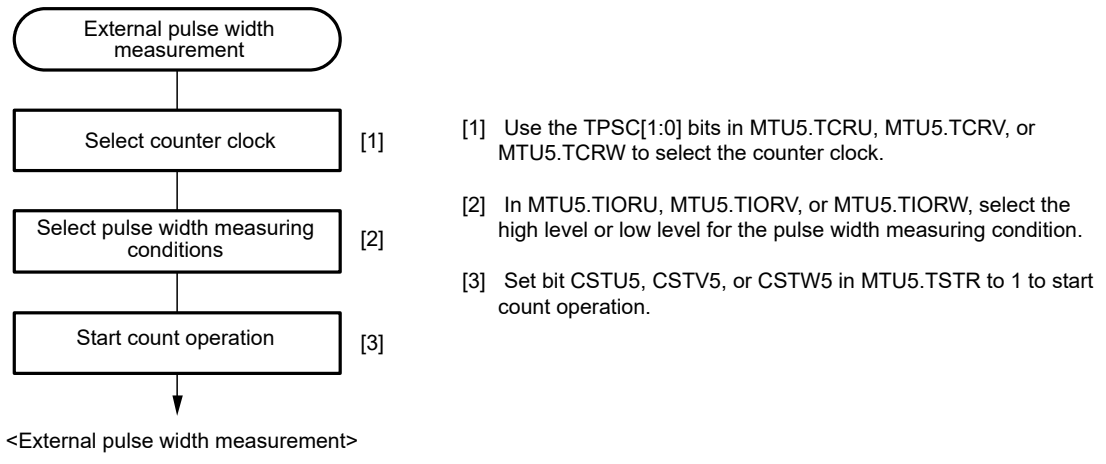
Figure 18.103 Example of synchronous counter clearing for MTU6 and MTU7 (2)

18.4.11 External Pulse Width Measurement

The pulse widths of up to three external input lines can be measured in MTU5.

When the IOC[4:0] bits in MTU5.TIORU, MTU5.TIORV, MTU5.TIORW are set for pulse width measurement, the pulse width of the signal input to the MTIC5U, MTIC5V, and MTIC5W pins are measured. TCNTU, TCNTV, and TCNTW count up while the level specified by the IOC[4:0] bits is input.

Figure 18.104 shows an example of setting external pulse width measurement and Figure 18.105 an example of external pulse width measurement.



Note:
- Set bit CMPCLR5U, CMPCLR5V, and CMPCLR5W in TCNTCMPCLR to 0.
- Set bit TGIE5U, TGIE5V, and TGIE5W in MTU5.TIER to 0.
- The value in TCNT is not captured in TGR.

Figure 18.104 Example of external pulse width measurement setting procedure

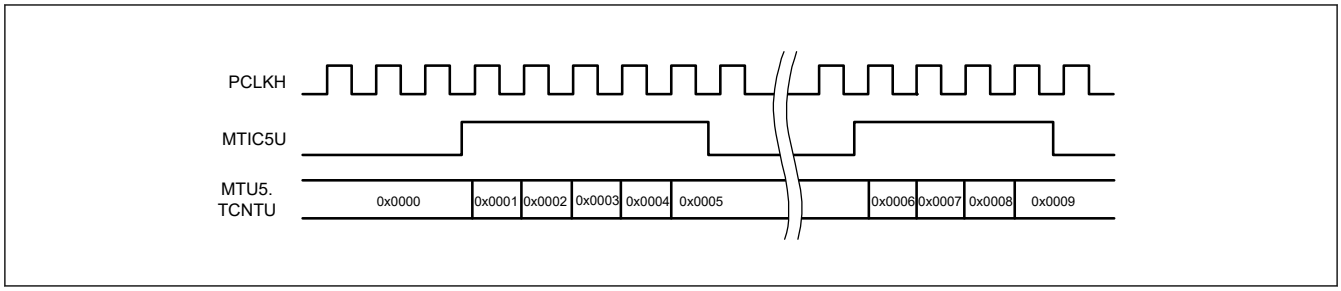


Figure 18.105 Example of external pulse width measurement (measuring high pulse width)

18.4.12 TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode

The MTU5 external pulse width measurement function allows to transfer the value in TCNTU, TCNTV, and TCNTW to TGRU, TGRV, and TGRW at the crest, or trough, or crest and trough when MTU6 and MTU7 operate in complementary PWM mode. The transfer timing is set in TIORU, TIORV, and TIORW. When the CMPCLR5U, CMPCLR5V, and CMPCLR5W bits in the TCNTCMPCLR register are set to 1, TCNTU, TCNTV, and TCNTW become 0x0000 at the transfer timing for TGRU, TGRV, and TGRW.

Figure 18.106 shows an operation example in which MTU5.TCNTU is used as a free-running counter without being cleared, and the value is captured in TGR at the crest and trough in complementary PWM mode.

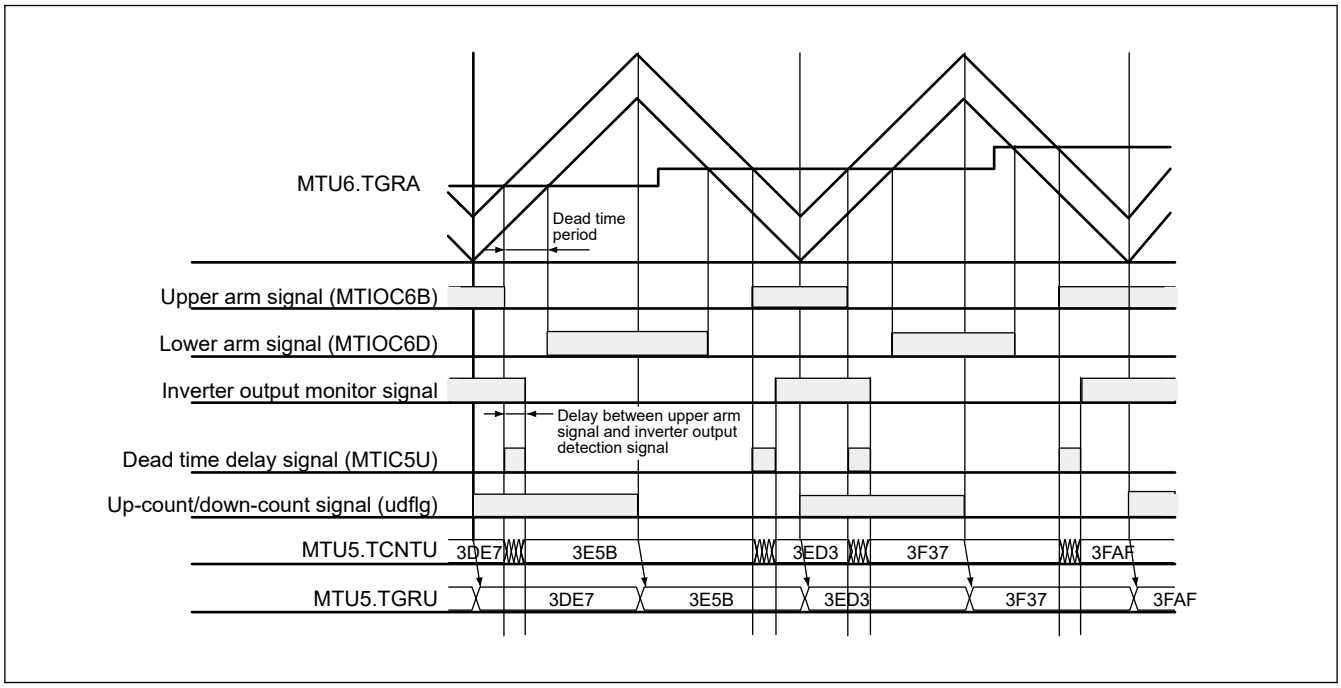


Figure 18.106 TCNTU capture at crest and trough in complementary PWM operation

18.4.13 Noise Filter Function

The input capture input pins and external pulse input pins have a noise filter function.

Set the NFCRn register (n = 0 to 8, C) to enable or disable the noise filter function and set the sampling clock. The noise filter for each pin can be enabled or disabled individually, and the sampling clock can be set for each channel.

Figure 18.107 shows the timing of noise filtering.

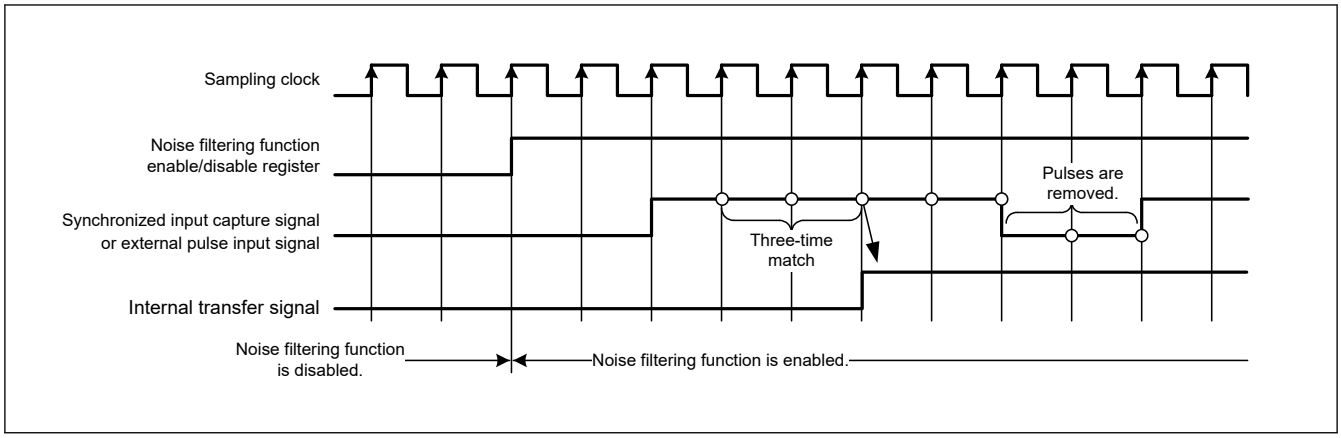


Figure 18.107 Timing of noise filtering

18.5 Interrupt Sources

18.5.1 Interrupt Sources and Priorities

There are three kinds of MTU interrupt sources:

- TGR input capture/compare match
- TCNT overflow
- TCNT underflow

Each interrupt source has its own enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is generated, if the corresponding enable/disable bit in TIER is set to 1, an interrupt is requested.

Relative channel priorities can be changed by the interrupt controller. For details, see section 12, Interrupt Controller (ICU). Table 18.74 lists the MTU interrupt sources.

Table 18.74 MTU interrupt sources

Channel	Name	Interrupt source	DMAC activation	Priority
MTU0	TGIA0	ch0 input capture/compare match A interrupt	Possible	High
	TGIB0	ch0 input capture/compare match B interrupt	Possible	↑
	TGIC0	ch0 input capture/compare match C interrupt	Possible	↑
	TGID0	ch0 input capture/compare match D interrupt	Possible	↑
	TCIV0	ch0 overflow interrupt	Not possible	↑
	TGIE0	ch0 compare match E interrupt	Not possible	↑
	TGIF0	ch0 compare match F interrupt	Not possible	↑
MTU1	TGIA1	ch1 input capture/compare match A interrupt	Possible	↑
	TGIB1	ch1 input capture/compare match B interrupt	Possible	↑
	TCIV1	ch1 overflow interrupt	Not possible	↑
	TCIU1	ch1 underflow interrupt	Not possible	↑
MTU2	TGIA2	ch2 input capture/compare match A interrupt	Possible	↑
	TGIB2	ch2 input capture/compare match B interrupt	Possible	↑
	TCIV2	ch2 overflow interrupt	Not possible	↑
	TCIU2	ch2 underflow interrupt	Not possible	↑
MTU3	TGIA3	ch3 input capture/compare match A interrupt	Possible	↑
	TGIB3	ch3 input capture/compare match B interrupt	Possible	↑
	TGIC3	ch3 input capture/compare match C interrupt	Possible	↑
	TGID3	ch3 input capture/compare match D interrupt	Possible	↑
	TCIV3	ch3 overflow interrupt	Not possible	↑
MTU4	TGIA4	ch4 input capture/compare match A interrupt	Possible	↑
	TGIB4	ch4 input capture/compare match B interrupt	Possible	↑
	TGIC4	ch4 input capture/compare match C interrupt	Possible	↑
	TGID4	ch4 input capture/compare match D interrupt	Possible	↑
	TCIV4	ch4 overflow/underflow interrupt*1	Possible	↑
MTU5	TGIU5	ch5 input capture/compare match U interrupt	Possible	↑
	TGIV5	ch5 input capture/compare match V interrupt	Possible	↑
	TGIW5	ch5 input capture/compare match W interrupt	Possible	↑
MTU6	TGIA6	ch6 compare match A interrupt	Possible	↑
	TGIB6	ch6 compare match B interrupt	Possible	↑
	TGIC6	ch6 compare match C interrupt	Possible	↑
	TGID6	ch6 compare match D interrupt	Possible	↑
	TCIV6	ch6 overflow interrupt	Not possible	↑
MTU7	TGIA7	ch7 compare match A interrupt	Possible	↑
	TGIB7	ch7 compare match B interrupt	Possible	↑
	TGIC7	ch7 compare match C interrupt	Possible	↑
	TGID7	ch7 compare match D interrupt	Possible	↑
	TCIV7	ch7 overflow/underflow interrupt*1	Possible	↑
MTU8	TGIA8	ch8 compare match A interrupt	Possible	↑
	TGIB8	ch8 compare match B interrupt	Possible	↑
	TGIC8	ch8 compare match C interrupt	Possible	↑
	TGID8	ch8 compare match D interrupt	Possible	↑
	TCIV8	ch8 overflow interrupt	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.
Note 1. Only in complementary PWM mode.

(1) Input Capture/Compare Match Interrupt

If the TIER.TGIE bit is set to 1 when a TGR input capture/compare match occurs on a channel, an interrupt is requested. The MTU has 33 input capture/compare match interrupts (six for MTU0, four each for MTU3, MTU4, MTU6, MTU7, and MTU8, two each for MTU1 and MTU2, and three for MTU5).

(2) Overflow Interrupt

If the TIER.TCIEV bit is set to 1 when a TCNT overflow occurs on a channel, an interrupt is requested. The MTU has eight overflow interrupts (one for each channel).

(3) Underflow Interrupt

If the TIER.TCIEU bit is set to 1 when a TCNT underflow occurs on a channel, an interrupt is requested. The MTU has two underflow interrupts (one each for MTU1 and MTU2).

18.5.2 DMAC Trigger Sources

The DMAC can be triggered by the TGR input capture/compare match interrupt in each channel and the overflow interrupt in MTU4 and MTU7. For details, see section 15, DMA Controller (DMAC).

The MTU provides a total of 33 input capture/compare match interrupts and overflow interrupts that can be used as DMAC trigger sources: four each for MTU0, MTU3, MTU6, and MTU8, two each for MTU1 and MTU2, five each for MTU4 and MTU7, and three for MTU5.

18.5.3 A/D Converter Trigger Sources

The A/D converter can be triggered by one of the following three methods in the MTU. Table 18.75 shows the relationship between interrupt sources and A/D converter start request signals.

(1) A/D Conversion Start by MTUn.TGRA (n = 0 to 4, 6, 7) Input Capture/Compare Match or at Trough of MTUm.TCNT (m = 4 or 7) in Complementary PWM Mode

The A/D converter can be triggered by the occurrence of a TGRA input capture/compare match in channels 0 to 4, 6, and 7. In addition, if complementary PWM operation is performed while the TTGE2 bit in MTU4.TIER is set to 1, the A/D converter can be triggered at the trough of MTUm.TCNT count (MTUm.TCNT = 0x0000) (m = 4 or 7).

A/D converter start request signal TRGAnN is issued to the A/D converter under either of the following conditions (n = 0 to 4, 6, 7).

- When a TGRA input capture/compare match occurs on a channel while the TIER.TTGE bit is set to 1
- When the MTUm.TCNT count reaches the trough (MTUm.TCNT = 0x0000) during complementary PWM operation while the TTGE2 bit in MTUm.TIER is set to 1

When either condition is satisfied, if A/D converter start signal TRGAnN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(2) A/D Conversion Start by Compare Match between MTU0.TCNT and MTU0.TGRE

A/D converter start request signal TRG0N is issued to the A/D converter when a compare match occurs between MTU0.TCNT and MTU0.TGRE.

When a compare match occurs between MTU0.TCNT and MTU0.TGRE while the TTGE2 bit in MTU0.TIER2 is set to 1, A/D converter start request TRG0N is issued to the A/D converter. If A/D converter start signal TRG0N from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Conversion Start by A/D Converter Start Request Delaying Function (MTU4 and MTU7)

The A/D converter can be triggered by generating A/D converter start request signal TRG4AN or TRG4BN (TRG7AN or TRG7BN) when the MTU4.TCNT (MTU7.TCNT) count matches the MTU4.TADCORA or MTU4.TADCORB (MTU7.TADCORA or MTU7.TADCORB) value if the UT4AE, DT4AE, UT4BE, or DT4BE (UT7AE, DT7AE, UT7BE, or DT7BE) bit in the A/D converter start request control register (MTU4.TADCR (MTU7.TADCR)) is set to 1. For details, see section 18.4.9. A/D Converter Start Request Delaying Function.

A/D conversion will start when TRG4AN (TRG7AN) is generated if A/D converter start signal TRG4AN (TRG7AN) from the MTU is selected as the trigger in the A/D converter, when TRG4BN (TRG7BN) is generated if TRG4BN (TRG7BN) from the MTU is selected as the trigger in the A/D converter, or when TRG4ABN (TRG7ABN) is generated if TRG4ABN (TRG7ABN) from the MTU is selected as the trigger in the A/D converter.

Table 18.75 Interrupt sources and A/D converter start request signals

Target registers	Interrupt source	A/D converter start request signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGA0N
MTU1.TGRA and MTU1.TCNT		TRGA1N
MTU2.TGRA and MTU2.TCNT		TRGA2N
MTU3.TGRA and MTU3.TCNT		TRGA3N
MTU4.TGRA and MTU4.TCNT*1		TRGA4N
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU6.TGRA and MTU6.TCNT	Input capture/compare match	TRGA6N
MTU7.TGRA and MTU7.TCNT*1		TRGA7N
MTU7.TCNT	MTU7.TCNT trough in complementary PWM mode	TRGA7N
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0N
MTU4.TADCORA and MTU4.TCNT		TRG4AN
MTU4.TADCORB and MTU4.TCNT		TRG4BN
MTU7.TADCORA and MTU7.TCNT		TRG7AN
MTU7.TADCORB and MTU7.TCNT		TRG7BN
MTU4.TADCORA and MTU4.TCNT, MTU4.TADCORB and MTU4.TCNT	Compare match (interrupt skipping function 2)	TRG4ABN
MTU7.TADCORA and MTU7.TCNT, MTU7.TADCORB and MTU7.TCNT		TRG7ABN

Note 1. Since PWM waveforms are generated in complementary PWM mode, MTU4.TGRA (MTU7.TGRA) compare match not only with MTU4.TCNT (MTU7.TCNT) but also with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) is detected. Accordingly, when compare match with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) occurs, TRGA4N (TRGA7N) is also generated. When MTU3 and MTU 4 (MTU 6 and MTU7) are made to operate in complementary PWM mode for generating an A/D converter start request, use the A/D converter start request by compare match between MTU4.TCNT (MTU7.TCNT) and MTU4.TADCORA/TADCORB (MTU7.TADCORA/TADCORB).

18.6 Operation Timing

18.6.1 Input/Output Timing

18.6.1.1 TCNT Count Timing

Figure 18.108 and Figure 18.109 show the TCNT count timing in internal clock operation, Figure 18.110 shows the TCNT count timing in external clock operation (normal mode), and Figure 18.111 shows the TCNT count timing in external clock operation (phase counting mode).

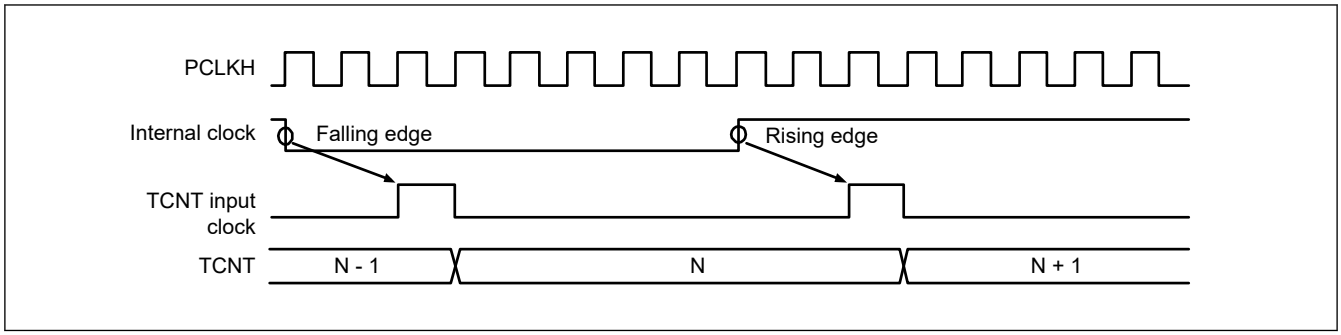


Figure 18.108 Count timing in internal clock operation (MTU0 to MTU4 and MTU6 to MTU8)

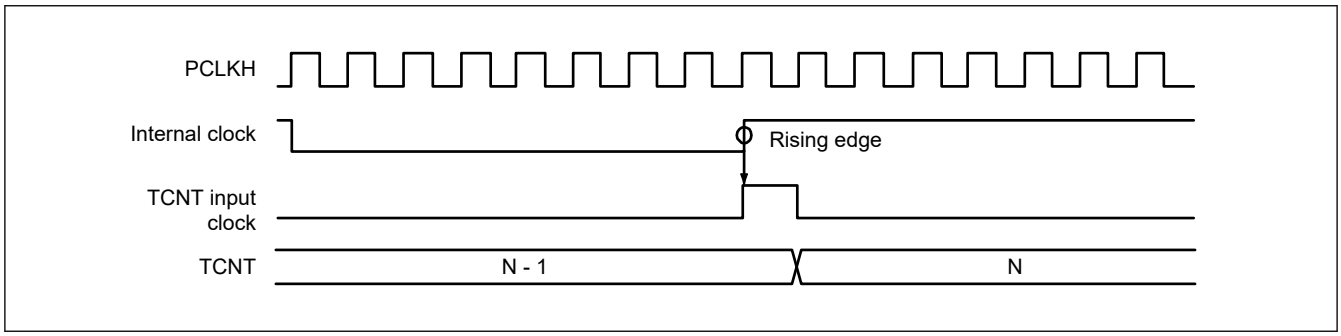


Figure 18.109 Count timing in internal clock operation (MTU5)

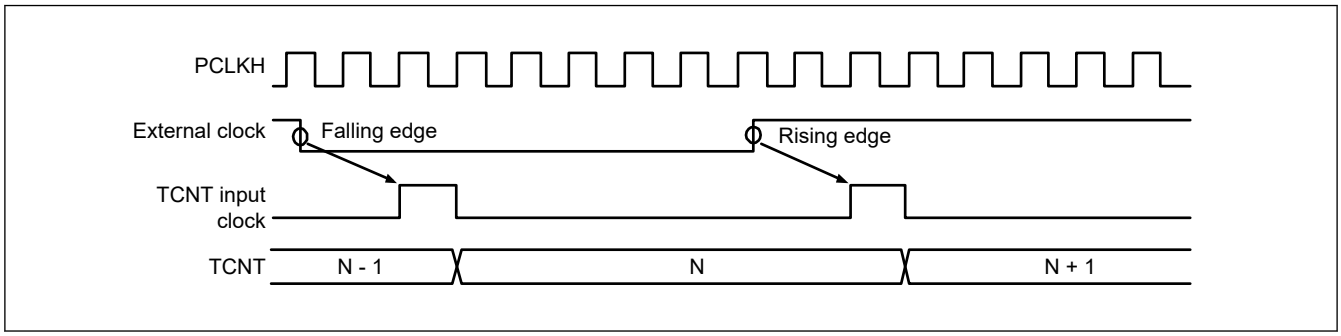


Figure 18.110 Count timing in external clock operation (MTU0 to MTU4 and MTU6 to MTU8)

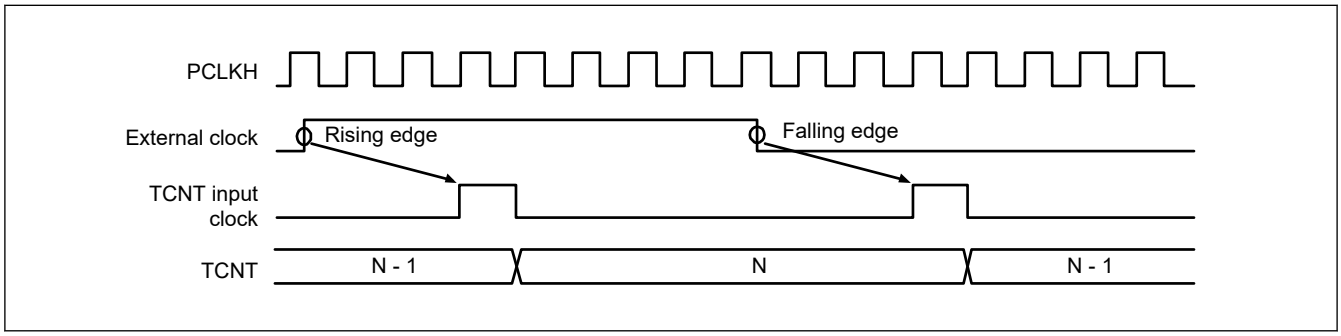


Figure 18.111 Count timing in external clock operation (phase counting mode)

18.6.1.2 Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the value set in TIOR is output from MTIOCnm pin (n = 0 to 4, 6, 7, 8; m = A to D). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 18.112 shows the output compare output timing (normal mode or PWM mode) and Figure 18.113 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

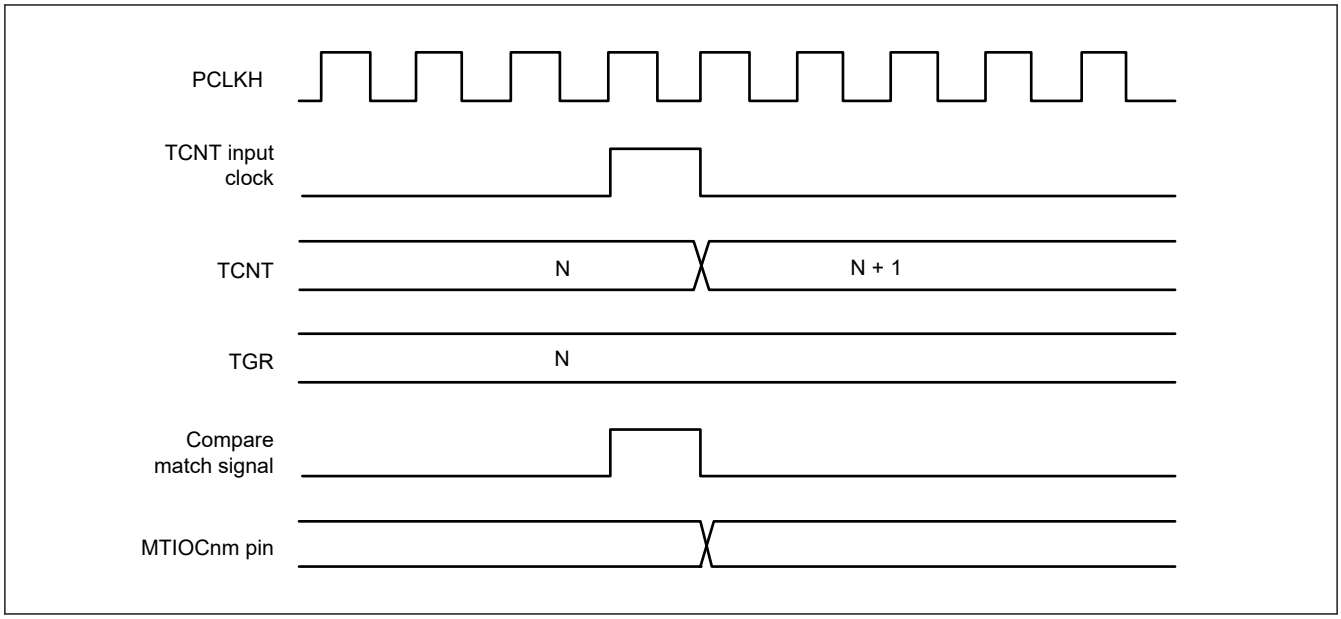


Figure 18.112 Output compare output timing (normal mode or PWM mode) (n = 0 to 4, 6, 7, 8; m = A to D)

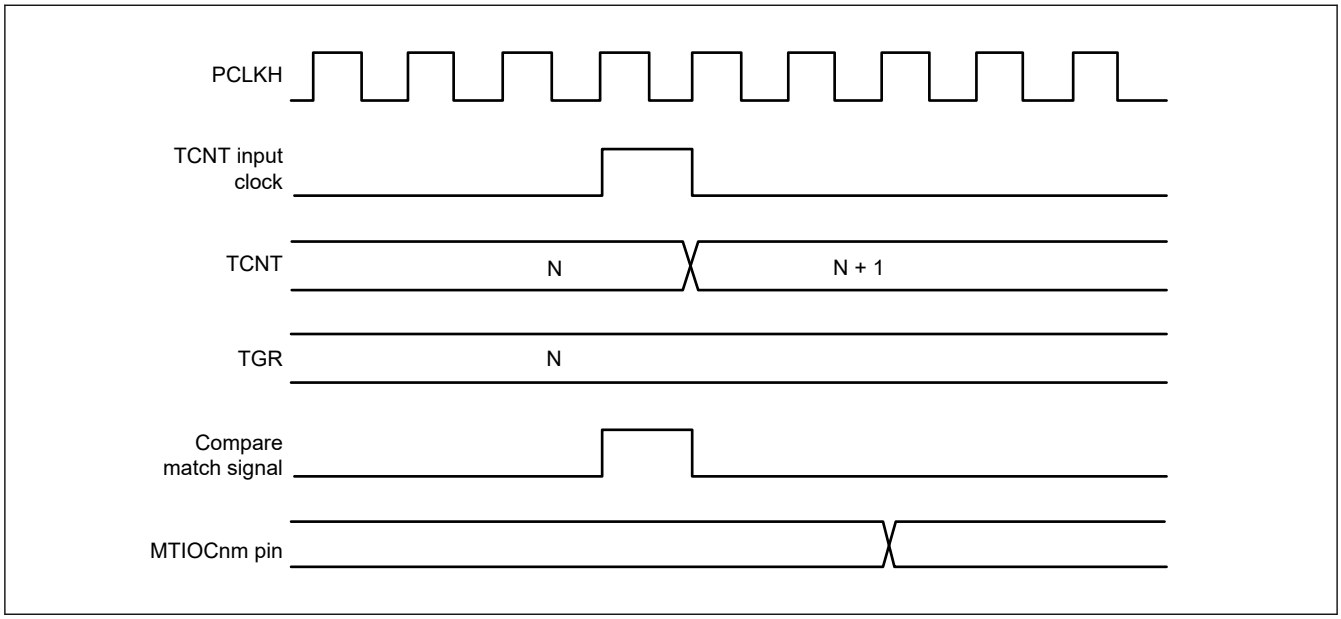


Figure 18.113 Output compare output timing (complementary PWM mode or reset-synchronized PWM mode) (n = 3, 4, 6, 7; m = A to D)

18.6.1.3 Input Capture Signal Timing

Figure 18.114 shows the input capture signal timing.

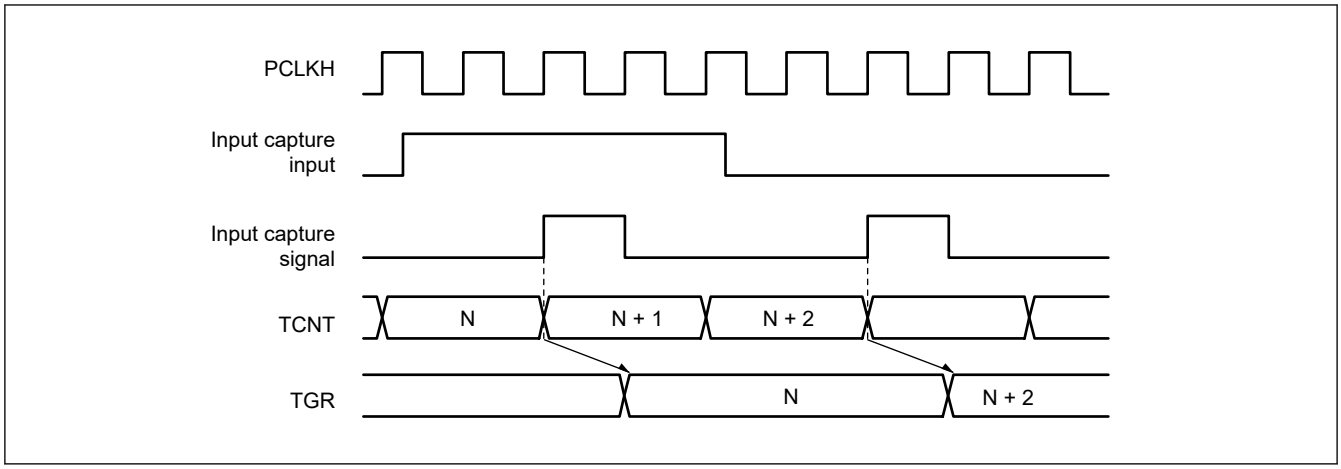


Figure 18.114 Input capture signal timing

18.6.1.4 Timing for Counter Clearing by Compare Match/Input Capture

Figure 18.115 and Figure 18.116 show the timing when counter clearing on compare match is specified, and Figure 18.117 shows the timing when counter clearing on input capture is specified.

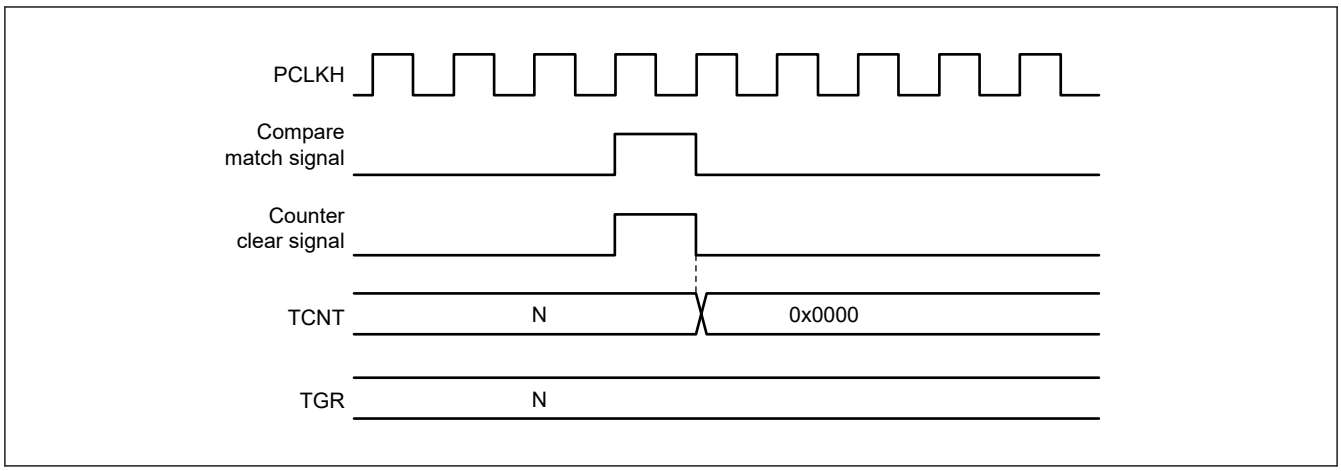


Figure 18.115 Counter clear timing (compare match) (MTU0 to MTU4 and MTU6 to MTU8)

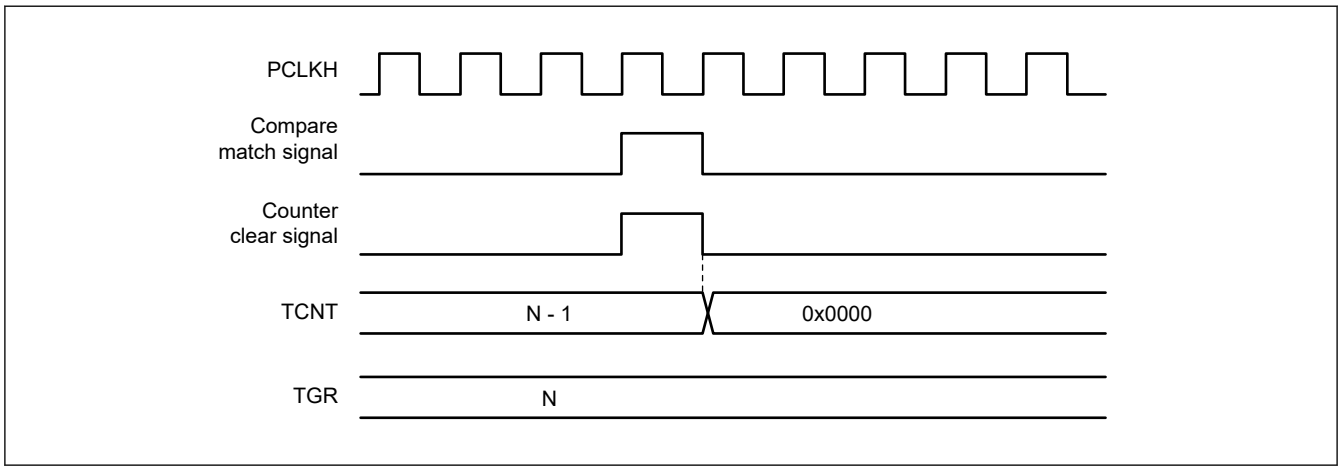


Figure 18.116 Counter clear timing (compare match) (MTU5)

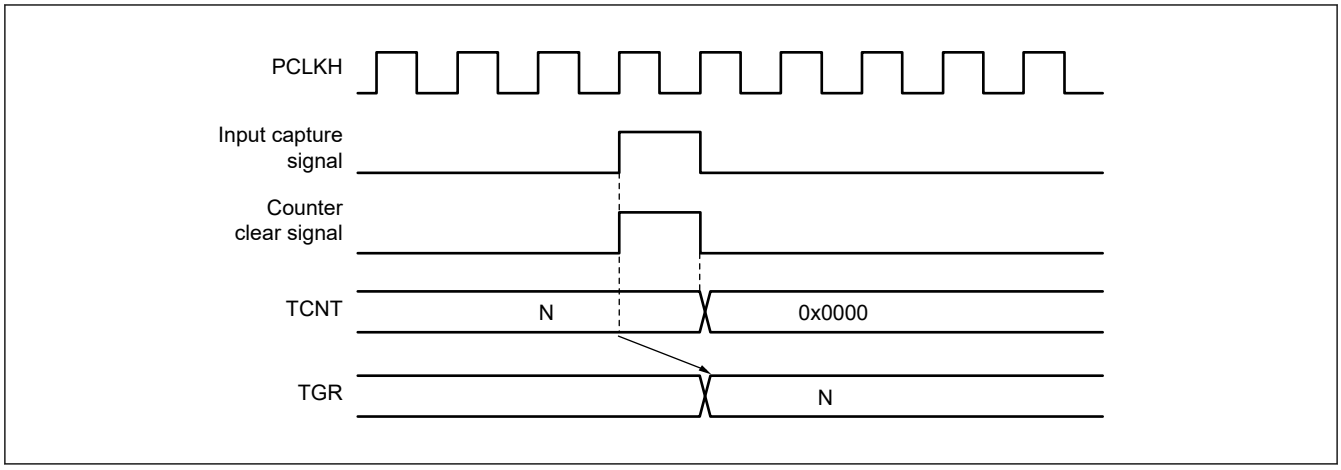


Figure 18.117 Counter clear timing (input capture) (MTU0 to MTU8)

18.6.1.5 Buffer Operation Timing

Figure 18.118 to Figure 18.120 show the timing in buffer operation.

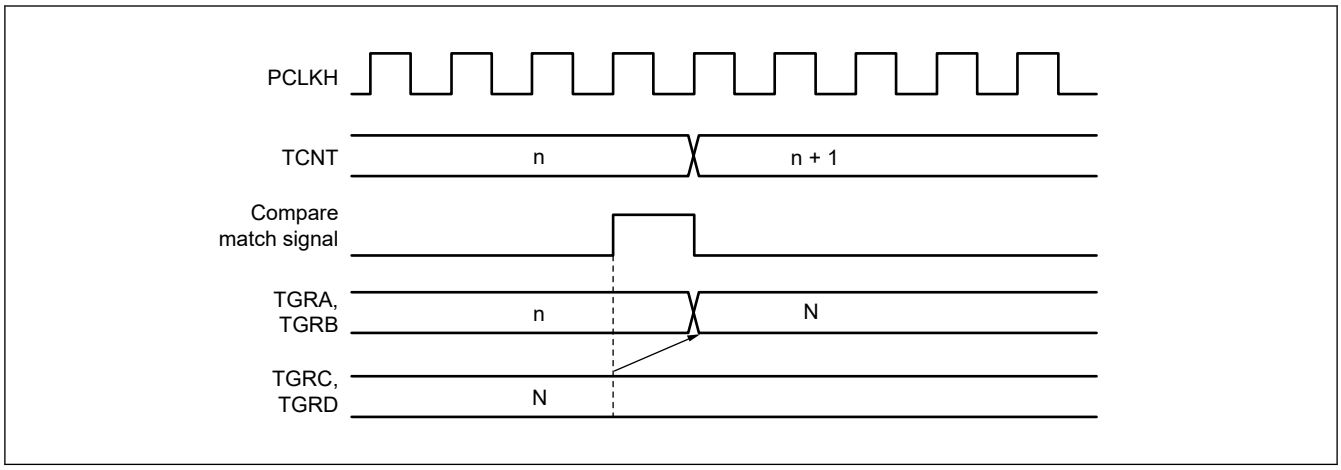


Figure 18.118 Buffer operation timing (compare match)

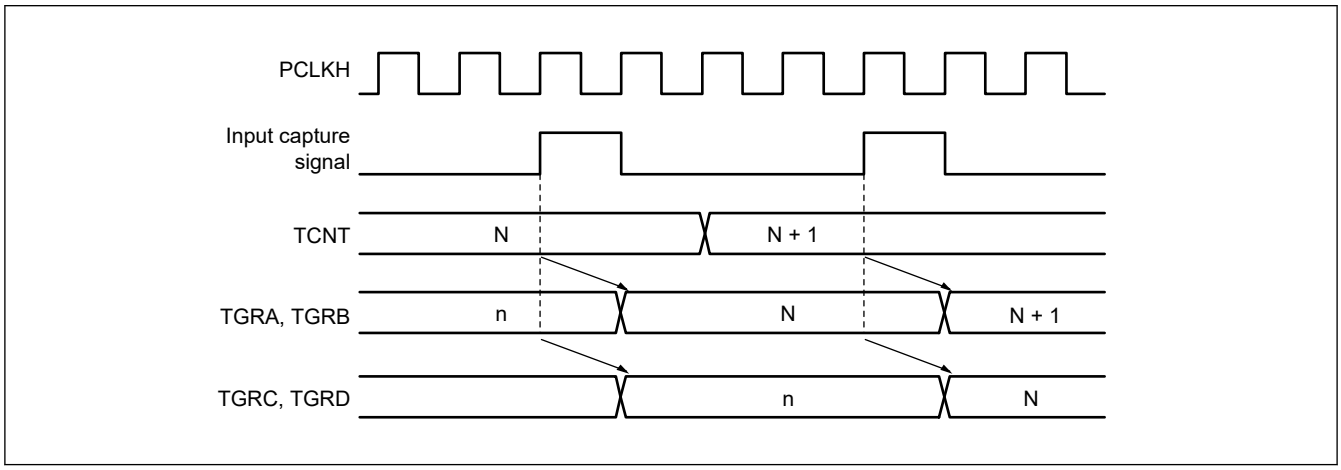


Figure 18.119 Buffer operation timing (input capture)

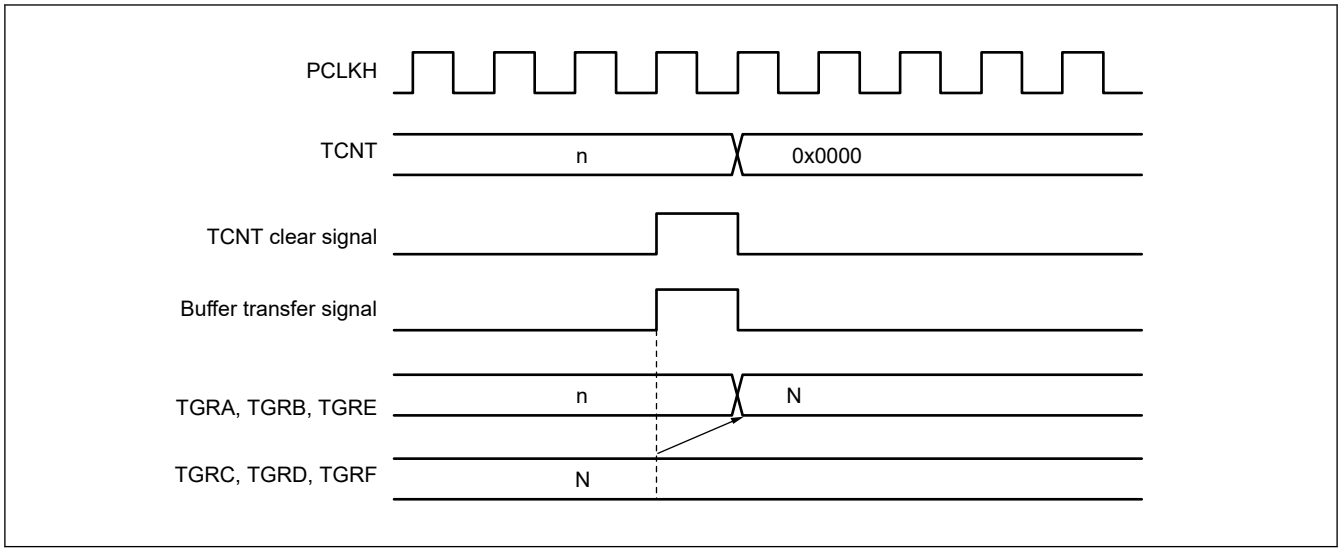


Figure 18.120 Buffer operation timing (when TCNT cleared)

18.6.1.6 Buffer Transfer Timing (Complementary PWM Mode)

Figure 18.121 to Figure 18.123 show the buffer transfer timing in complementary PWM mode.

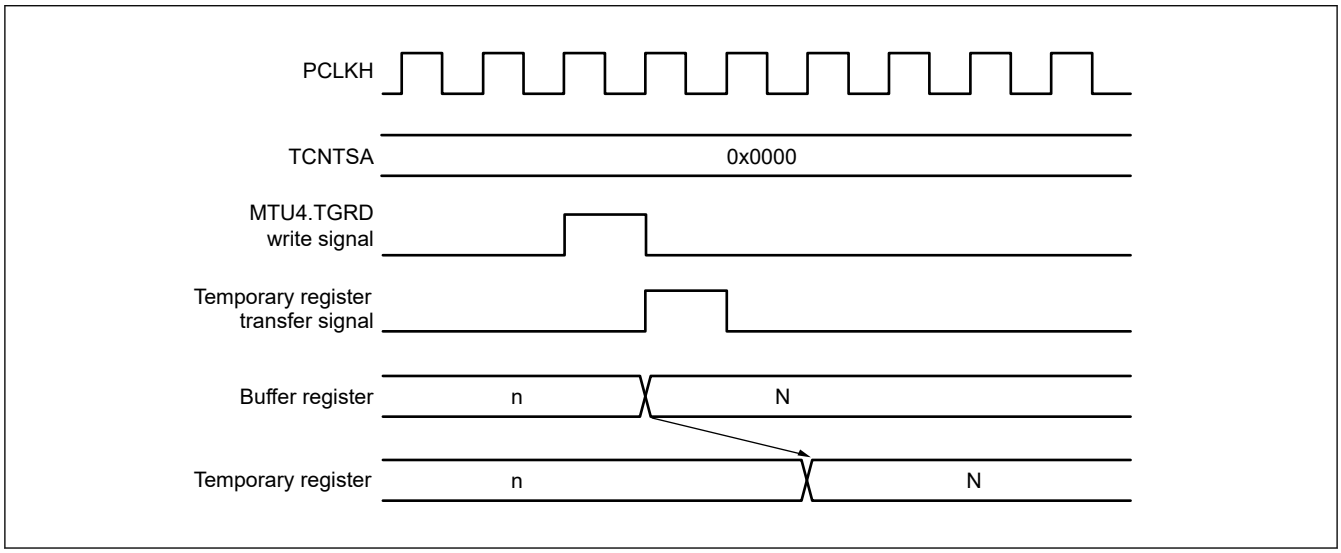


Figure 18.121 Transfer timing from buffer register to temporary register (TCNTSA stopped)

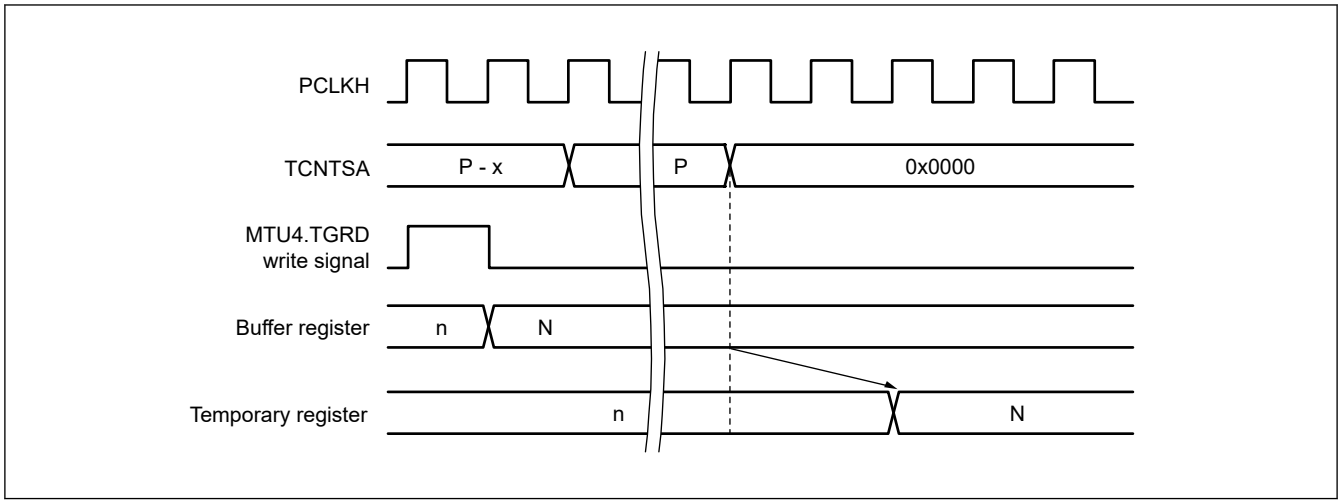


Figure 18.122 Transfer timing from buffer register to temporary register (TCNTSA operating)

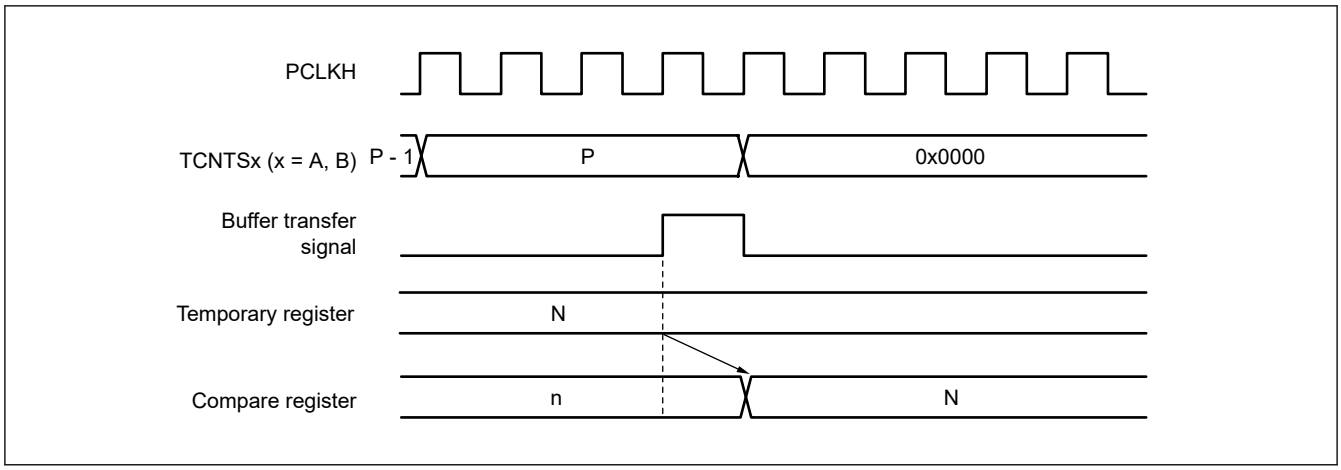


Figure 18.123 Transfer timing from temporary register to compare register

18.6.2 Interrupt Signal Timing

18.6.2.1 TGI Interrupt Timing by Compare Match

Figure 18.124 and Figure 18.125 show the TGI interrupt request signal timing when a compare match occurs.

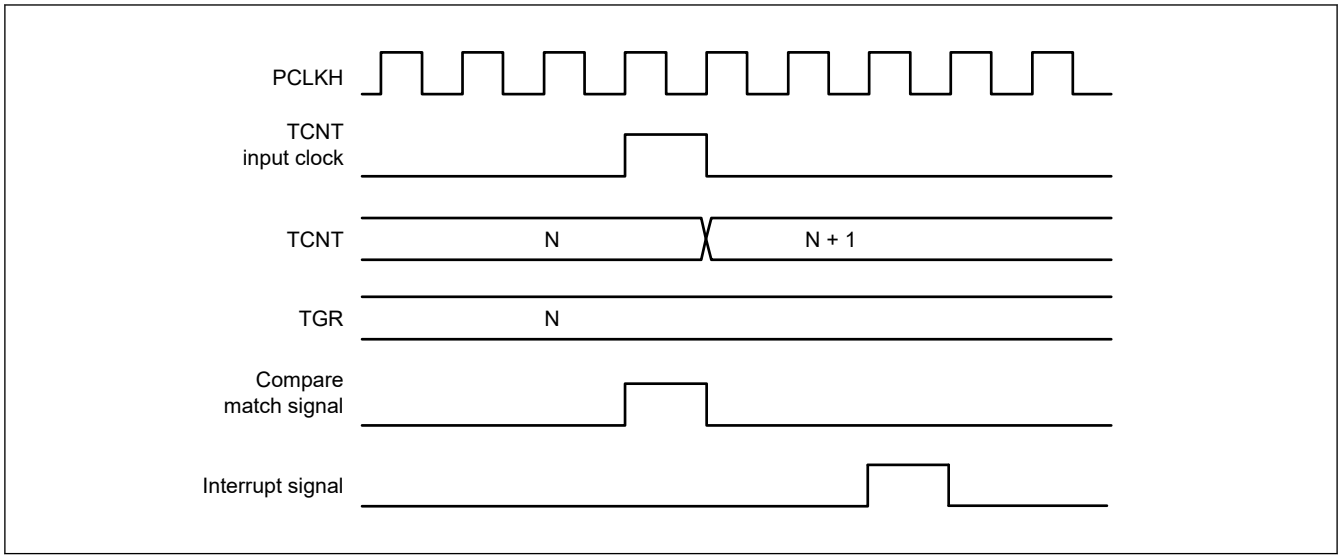


Figure 18.124 TGI interrupt timing (compare match) (MTU0 to MTU4 and MTU6 to MTU8)

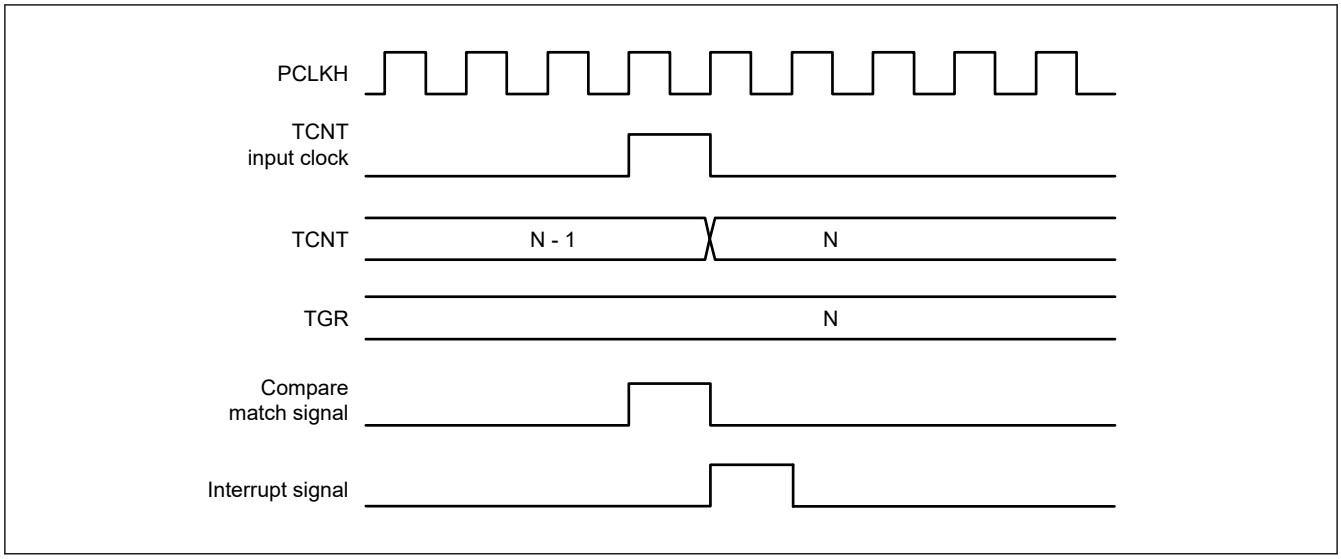


Figure 18.125 TGI interrupt timing (compare match) (MTU5)

18.6.2.2 TGI Interrupt Timing by Input Capture

Figure 18.126 and Figure 18.127 show the TGI interrupt request signal timing when an input capture occurs.

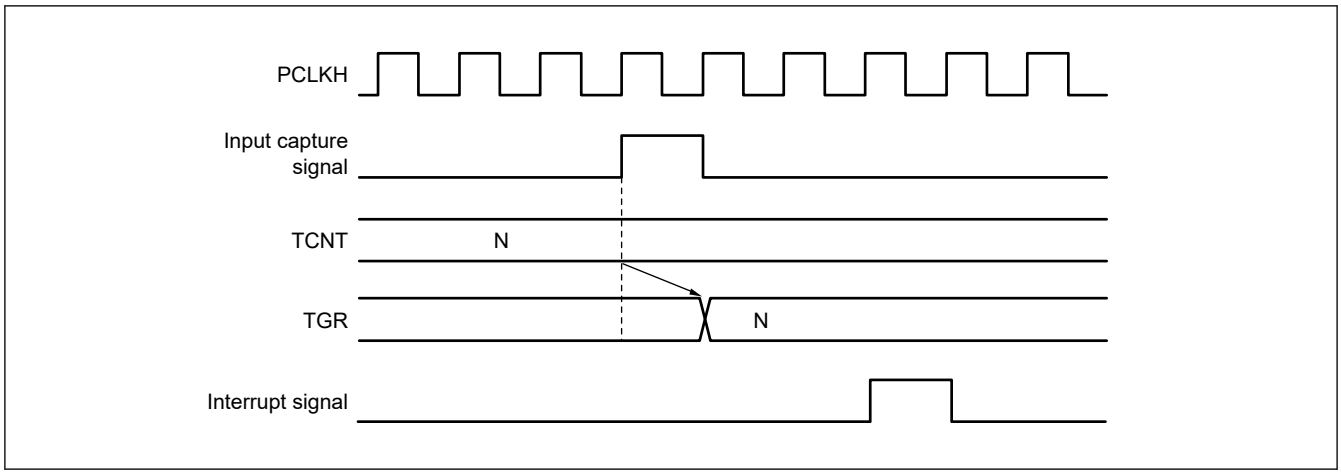


Figure 18.126 TGI interrupt timing (input capture) (MTU0 to MTU4 and MTU6 to MTU8)

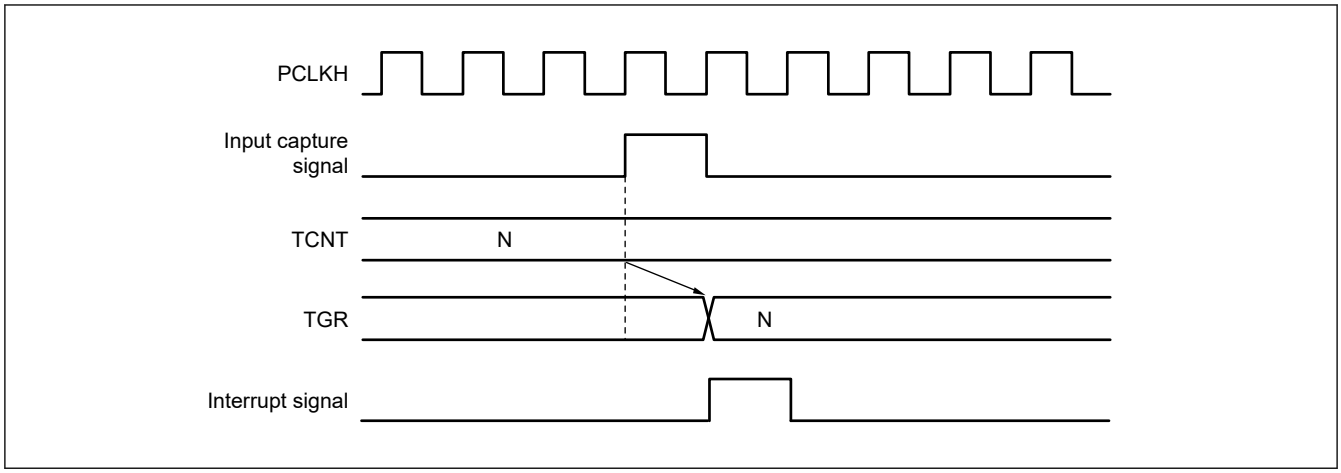


Figure 18.127 TGI interrupt timing (input capture) (MTU5)

18.6.2.3 TCIV and TCIU Interrupt Timing on Overflow/Underflow

Figure 18.128 shows the TCIV interrupt request signal timing when an overflow is generated and Figure 18.129 shows the TCIU interrupt request signal timing when an underflow is generated.

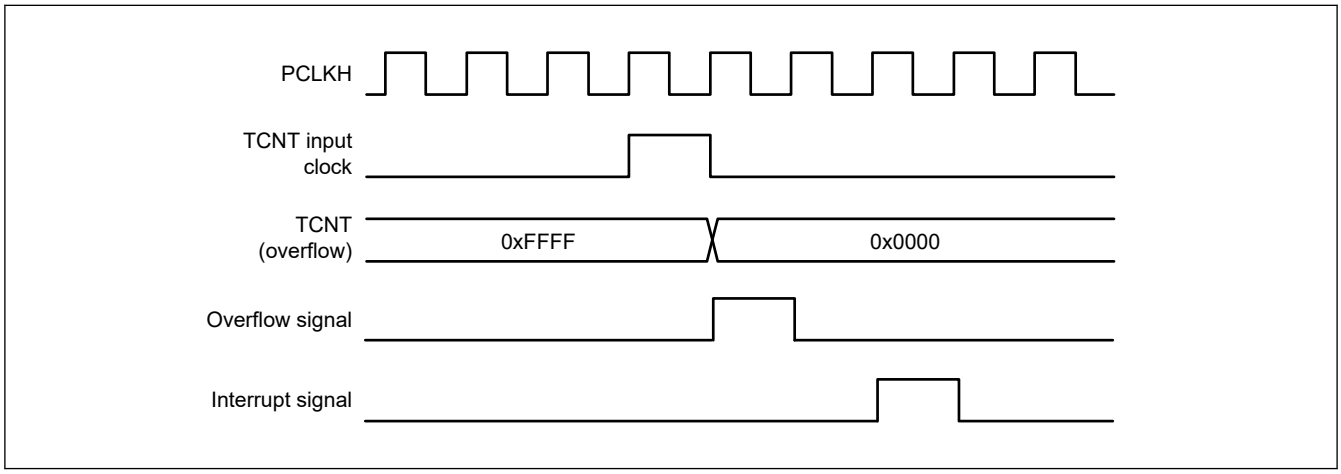


Figure 18.128 TCIV interrupt timing

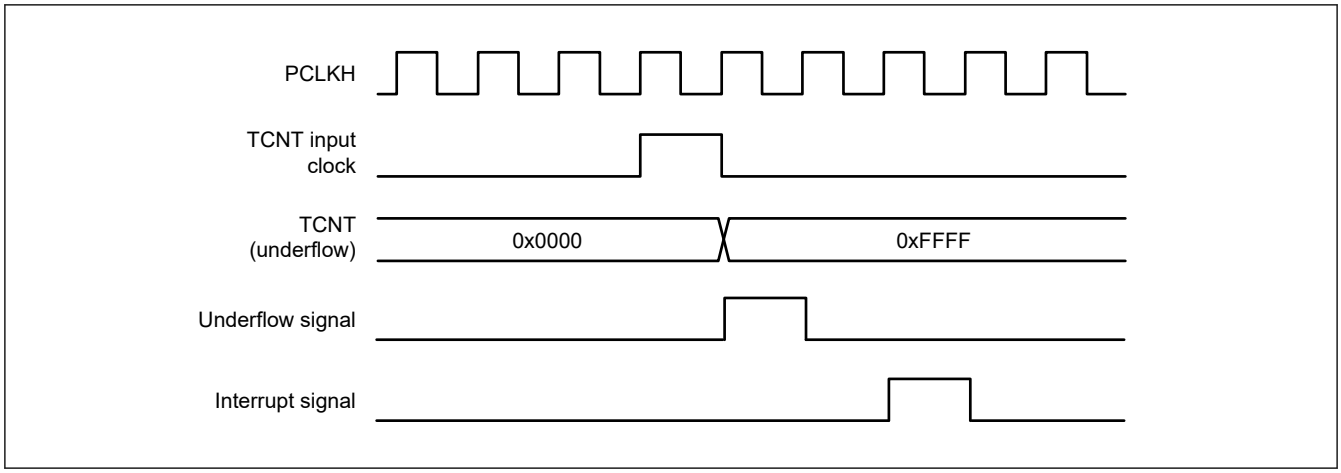


Figure 18.129 TCIU interrupt timing

18.7 Usage Notes

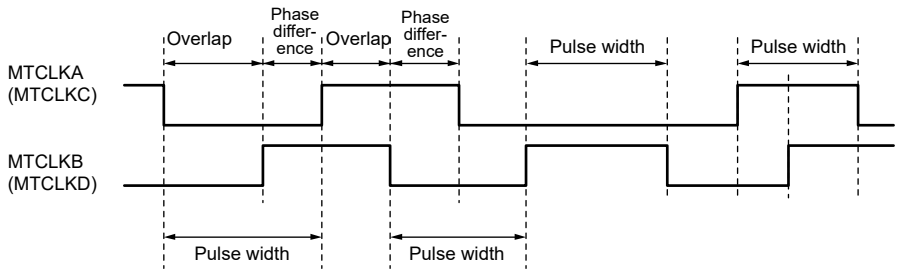
18.7.1 Module Stop Function Setting

MTU operation can be disabled or enabled using the module stop control register. MTU operation is stopped with the initial setting. Register access is enabled by releasing the module clock stop state. For details, see section 9, Low-Power Consumption Function.

18.7.2 Input Clock Restrictions

The input clock pulse width must be at least three PCLKH cycles for single-edge detection, and at least five PCLKH cycles for both-edge detection. The MTU will not operate properly at narrower pulse widths.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least three PCLKH cycles, and the pulse width must be at least five PCLKH cycles. Figure 18.130 shows the input clock conditions in phase counting mode.



Note: Phase difference and overlap: 3 PCLKH cycles or more.
Pulse width: 5 PCLKH cycles or more.

Figure 18.130 Phase difference, overlap, and pulse width in phase counting mode

18.7.3 Note on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which TCNT updates the matched count value). Consequently, the actual counter frequency is given by the following formula:

- MTU0 to MTU4 and MTU6 to MTU8
  $f = \frac{CNTCLK}{N+1}$
- MTU5
  $f = \frac{CNTCLK}{N}$

f: Counter frequency

CNTCLK: The counter clock frequency set by TCR.TPSC[2:0] and TCR2.TPSC2[2:0]

N: TGR setting

18.7.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the TCNT write cycle, TCNT clearing takes precedence and TCNT write operation is not performed.

Figure 18.131 shows the timing in this case.

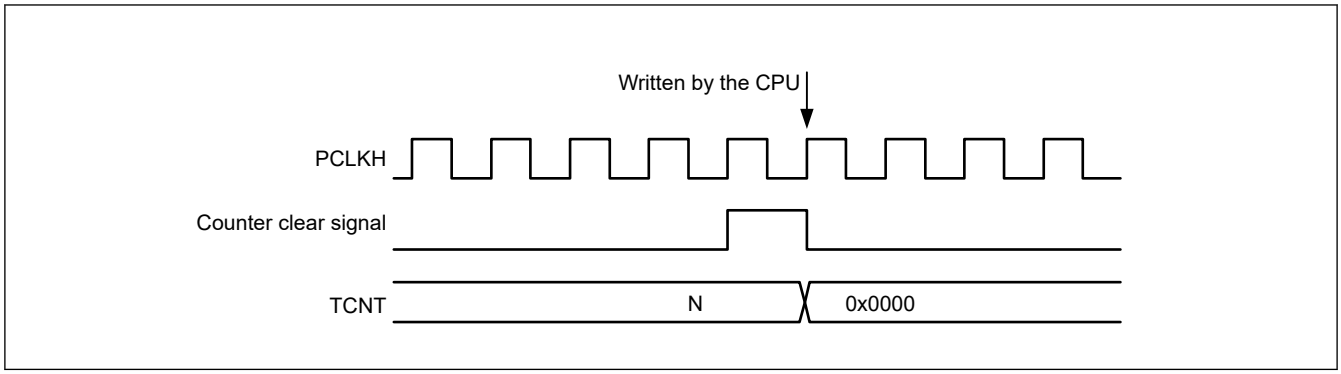


Figure 18.131 Contention between TCNT write and clear operations

18.7.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the TCNT write cycle, TCNT write operation takes precedence and TCNT is not incremented.

Figure 18.132 shows the timing in this case.

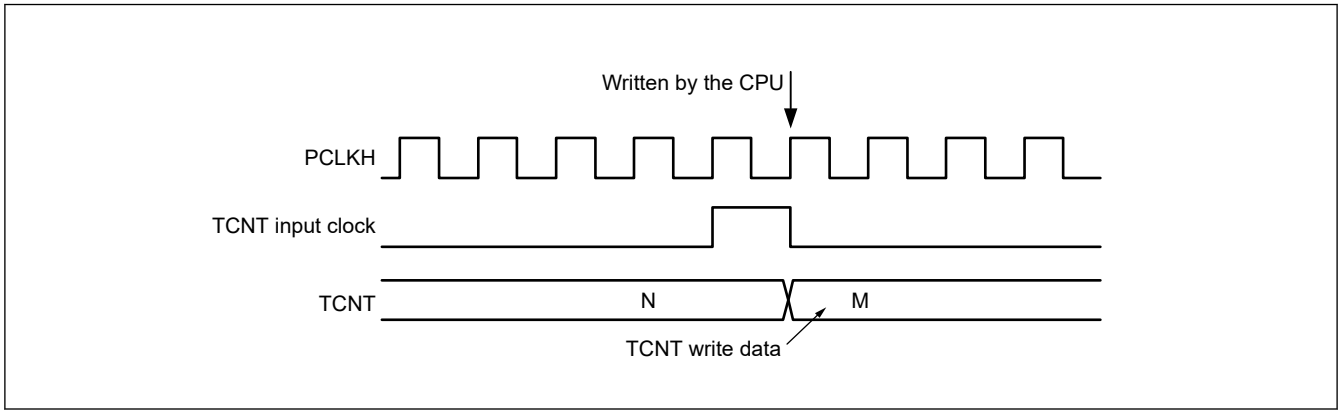


Figure 18.132 Contention between TCNT write and increment operations

18.7.6 Contention between TGR Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, TGR write operation is executed and the compare match signal is also generated.

Figure 18.133 shows the timing in this case.

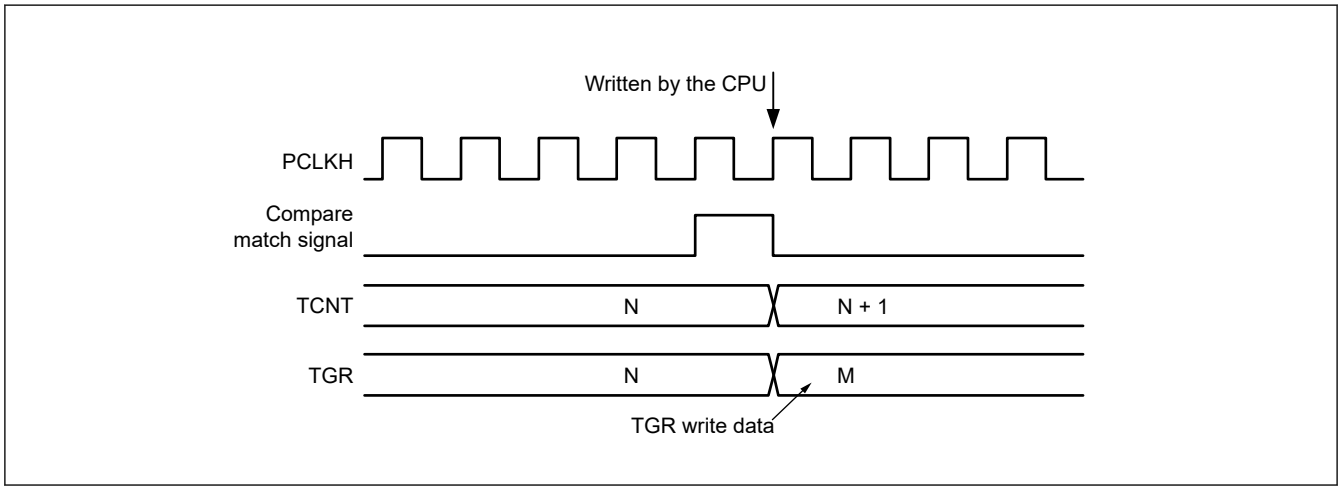


Figure 18.133 Contention between TGR write operation and compare match

18.7.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 18.134 shows the timing in this case.

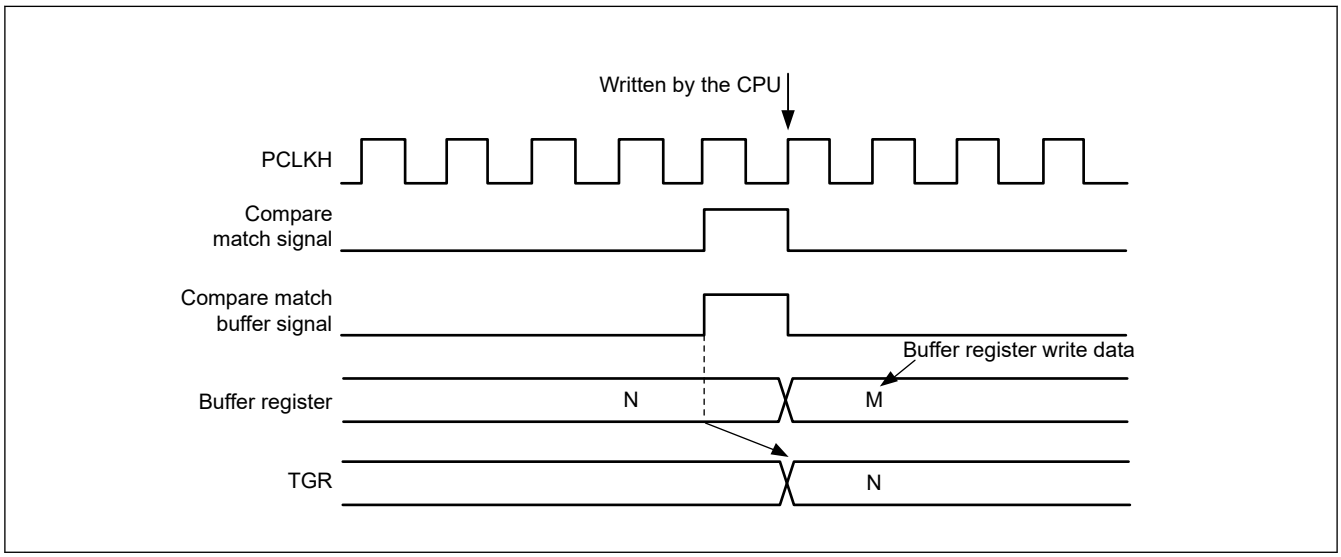


Figure 18.134 Contention between buffer register write operation and compare match

18.7.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the timer buffer operation transfer mode register (TBTM), if TCNT clearing occurs in the TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 18.135 shows the timing in this case.

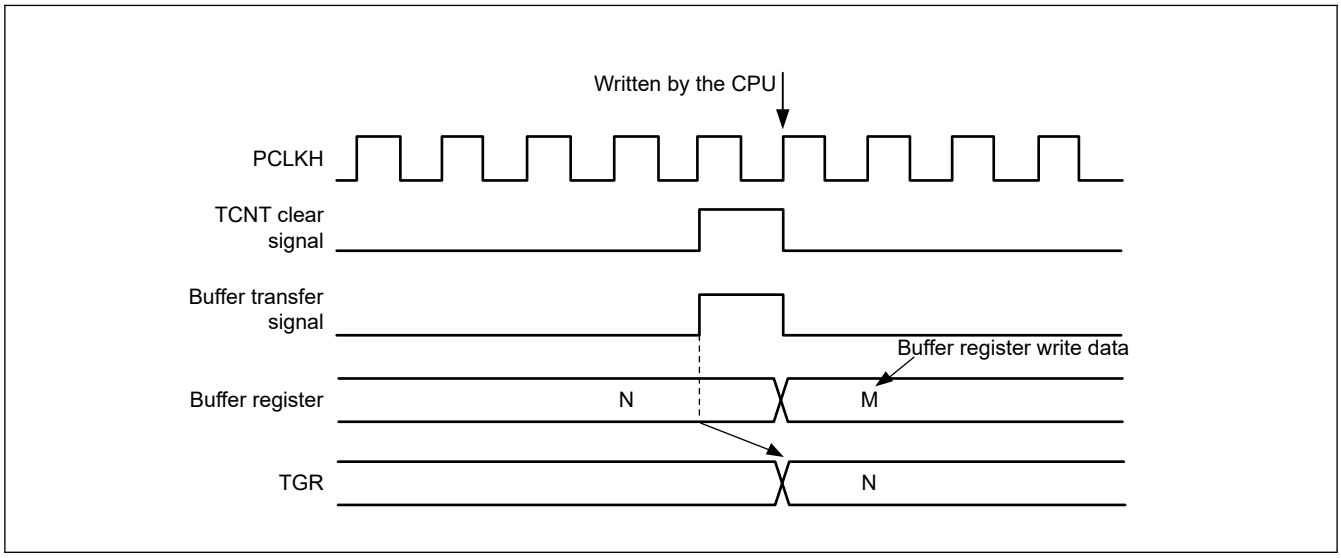


Figure 18.135 Contention between buffer register write and TCNT clear operations

18.7.9 Contention between TGR Read Operation and Input Capture

If an input capture signal is generated in the TGR read cycle, the data before input capture transfer is read.

Figure 18.136 shows the timing in this case.

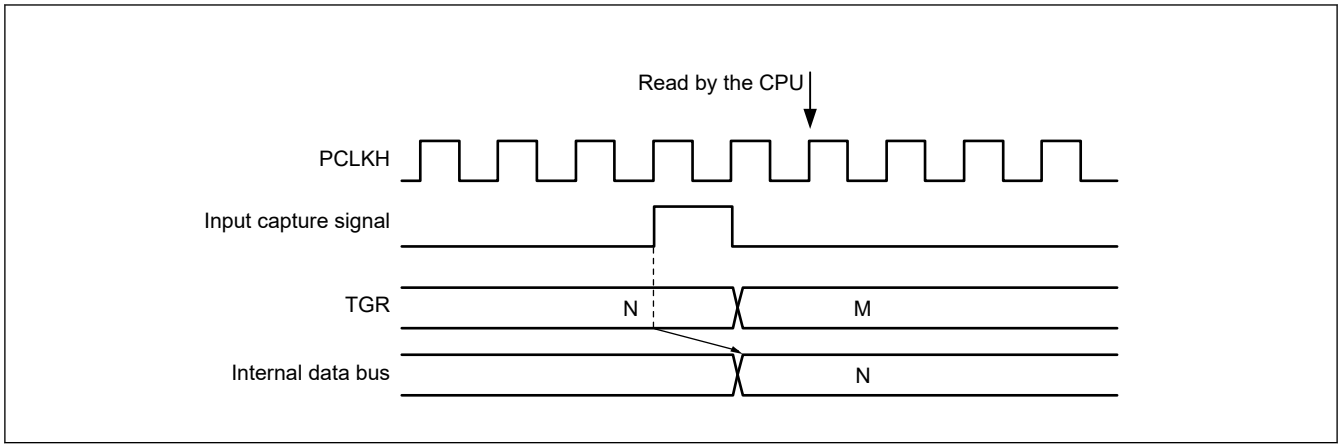


Figure 18.136 Contention between TGR read operation and input capture (MTU0 to MTU8)

18.7.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in a TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4 and MTU6 to MTU8. In MTU5, the TGR write operation is performed and the input capture signal is generated.

Figure 18.137 and Figure 18.138 show the timing in this case.

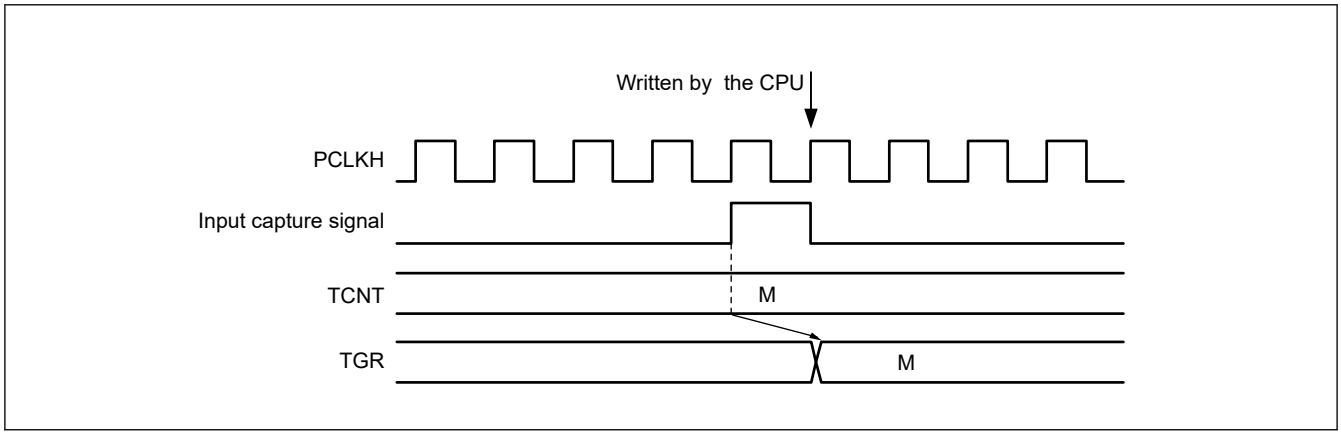


Figure 18.137 Contention between TGR write operation and input capture (MTU0 to MTU4, MTU6 to MTU8)

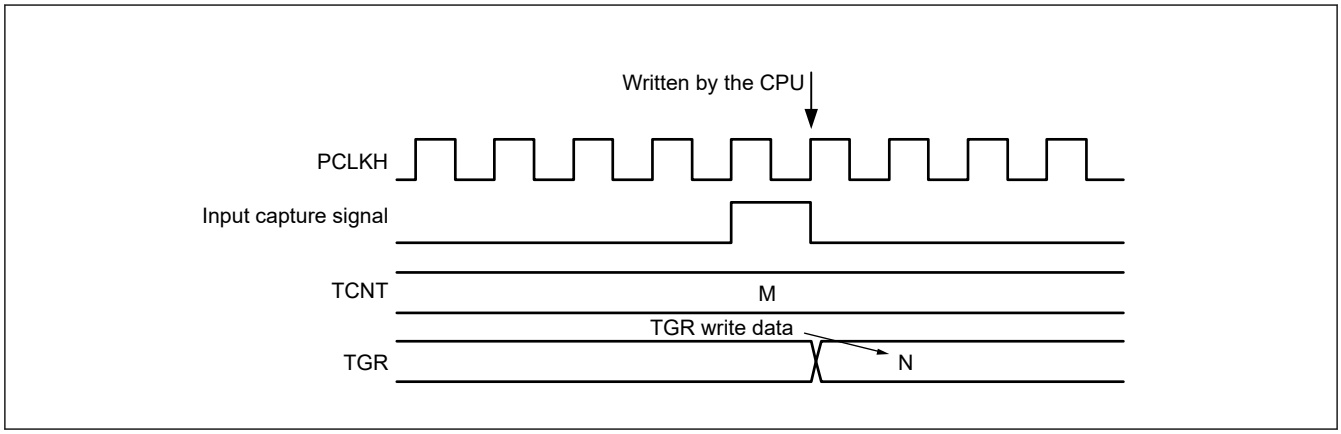


Figure 18.138 Contention between TGR write operation and input capture (MTU5)

18.7.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in a buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.

Figure 18.139 shows the timing in this case.

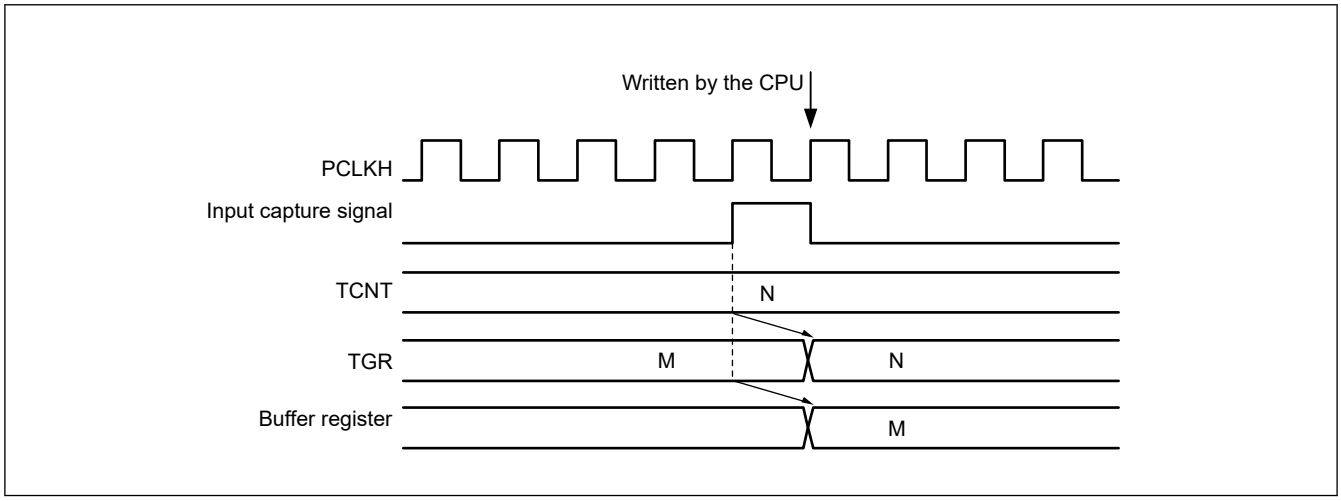


Figure 18.139 Contention between buffer register write operation and input capture

18.7.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

With timer counters MTU1.TCNT and MTU2.TCNT in a cascade, when a contention occurs between MTU1.TCNT counting (an MTU2.TCNT overflow/underflow) and the MTU2.TCNT write operation, the MTU2.TCNT write operation is performed and the MTU1.TCNT count signal is disabled. In this case, if MTU1.TGRA works as a compare match register and there is a match between the MTU1.TGRA and MTU1.TCNT values, a compare match signal is issued.

Furthermore, when the MTU1.TCNT count clock is selected as the input capture source of MTU0, MTU0.TGRA to MTU0.TGRD work in input capture mode. In addition, when the MTU0.TGRC compare match/input capture is selected as the input capture source of MTU1.TGRB, MTU1.TGRB works in input capture mode.

Figure 18.140 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

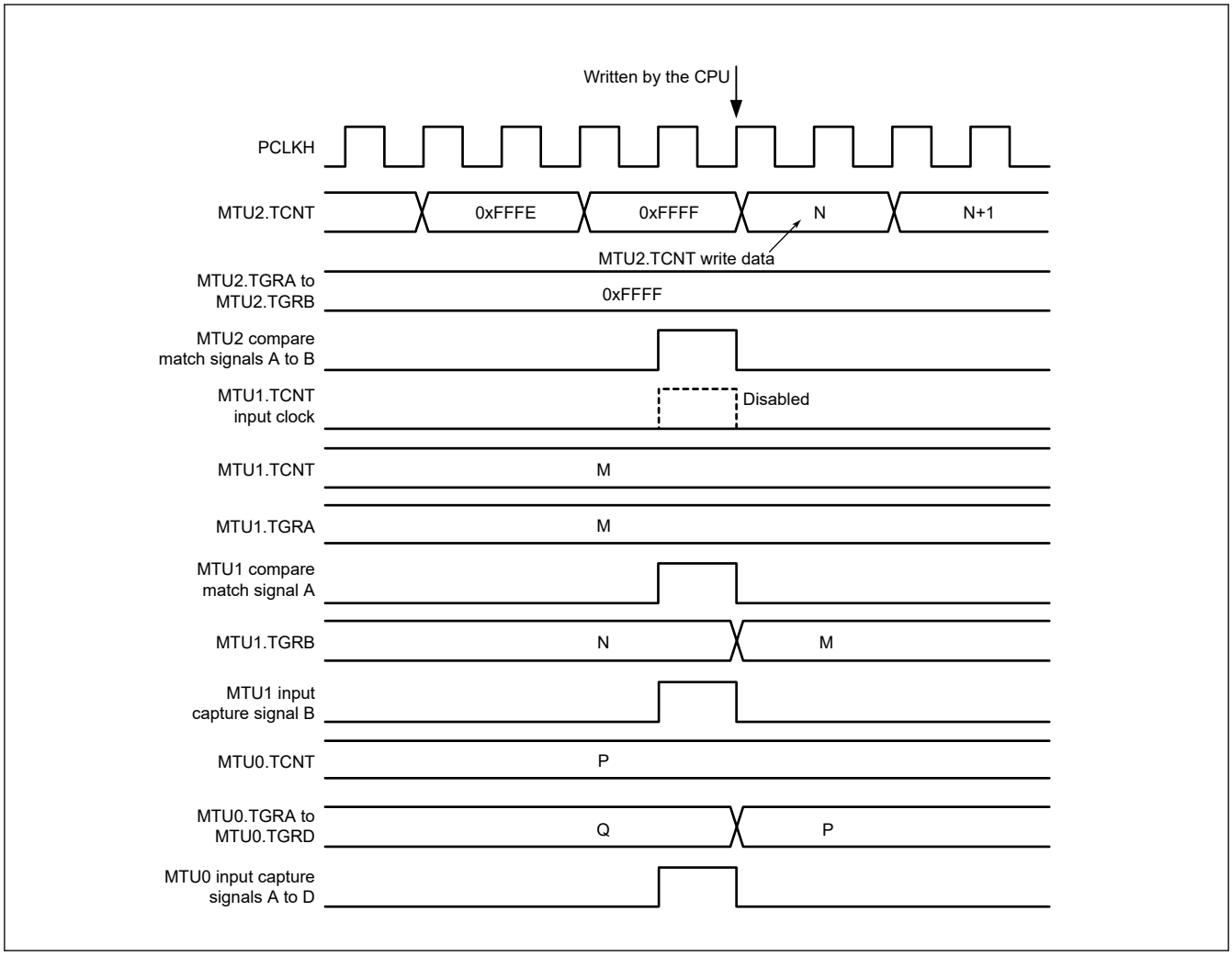


Figure 18.140 Contention between MTU2.TCNT write operation and overflow/underflow in cascaded operation

18.7.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) is stopped in complementary PWM mode, the MTU3.TCNT (MTU6.TCNT) value is set to the timer dead time data register (TDDRA (TDDRB)) value and MTU4.TCNT (MTU7.TCNT) is set to 0x0000.

When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state.

Figure 18.141 shows this operation.

When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT).

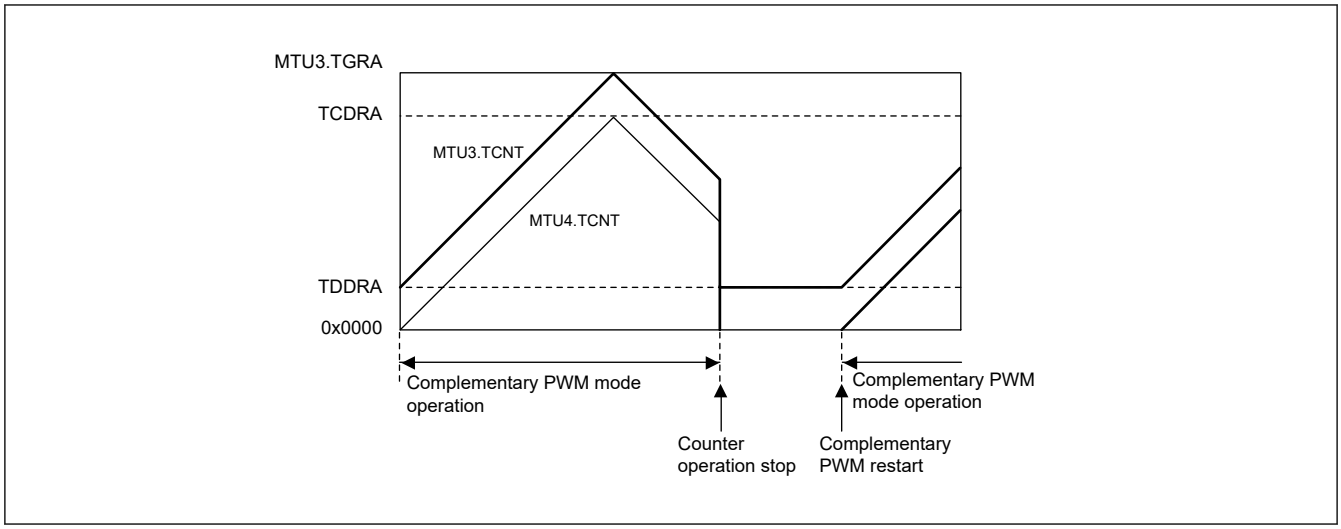


Figure 18.141 Counter value when stopped in complementary PWM mode

18.7.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM period set register (MTU3.TGRA or MTU6.TGRA), timer cycle data register (TCDRA or TCDRB), and duty set registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB)) in complementary PWM mode, be sure to use buffer operation. In addition, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.

In complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in bits BFA and BFB of MTU3.TMDR1 (MTU6.TMDR1). When the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA), and TCBRA (TCBRB) functions as a buffer register for TCDRA (TCDRB).

18.7.15 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in the BFA and BFB bits of MTU3.TMDR1 (MTU6.TMDR1). For example, if the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA).

While the MTU3.TGRC (MTU6.TGRC) and MTU3.TGRD (MTU6.TGRD) are operating as buffer registers, a TGImn interrupt (m = C, D; n = 3, 4 or 6, 7) is not generated.

Figure 18.142 shows an example of MTU3.TGR (MTU6.TGR), MTU4.TGR (MTU7.TGR), MTIOC3 (MTIOC6), and MTIOC4 (MTIOC7) operation with the BFA and BFB bits in MTU3.TMDR1 (MTU6.TMDR1) set to 1 and the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) set to 0.

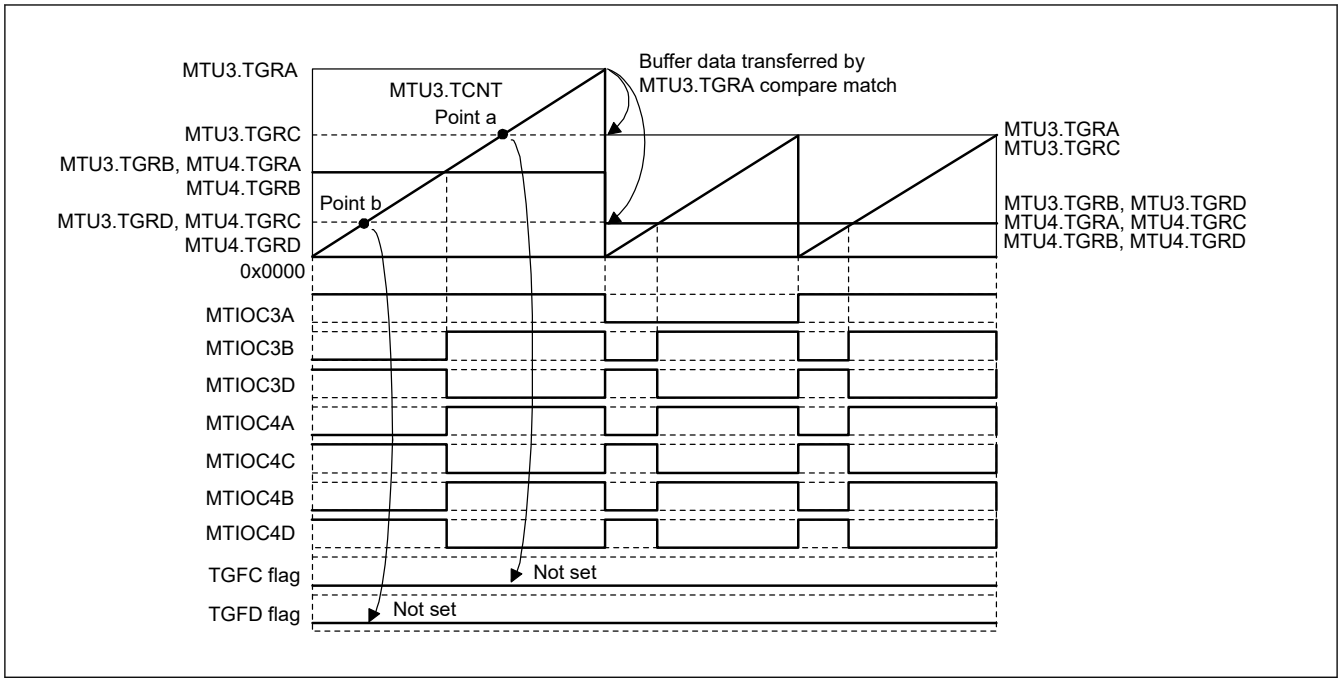


Figure 18.142 Buffer operation and compare match in reset-synchronized PWM mode

18.7.16 Overflow in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) start counting when the CST3 (CST6) bit of TSTRA (TSTRB) is set to 1. In this state, the MTU4.TCNT (MTU7.TCNT) count clock source and count edge are determined by the MTU3.TCR (MTU6.TCR) setting.

In reset-synchronized PWM mode, with period register MTU3.TGRA (MTU6.TGRA) set to 0xFFFF and the MTU3.TGRA (MTU6.TGRA) compare match selected as the counter clearing source, MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) count up to 0xFFFF, then a compare match occurs with MTU3.TGRA (MTU6.TGRA), and MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) are both cleared. In this case, a TCIVn interrupt (n = 3, 4 or 6, 7) is not generated.

Figure 18.143 shows an example of operation in reset-synchronized PWM mode with period register MTU3.TGRA (MTU6.TGRA) set to 0xFFFF and the MTU3.TGRA (MTU6.TGRA) compare match specified for the counter clearing source.

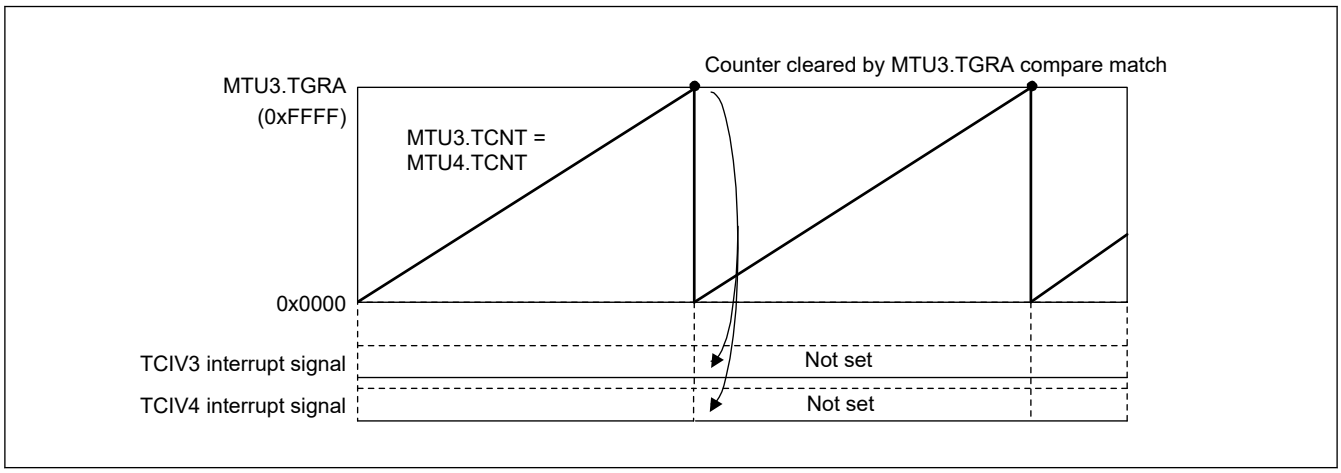


Figure 18.143 Overflow in reset-synchronized PWM mode

18.7.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, neither a TCIVn interrupt (n = 0 to 4, 6 to 8) nor a TCIUn interrupt (n = 1, 2) is generated and TCNT clearing takes precedence.

Figure 18.144 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to 0xFFFF.

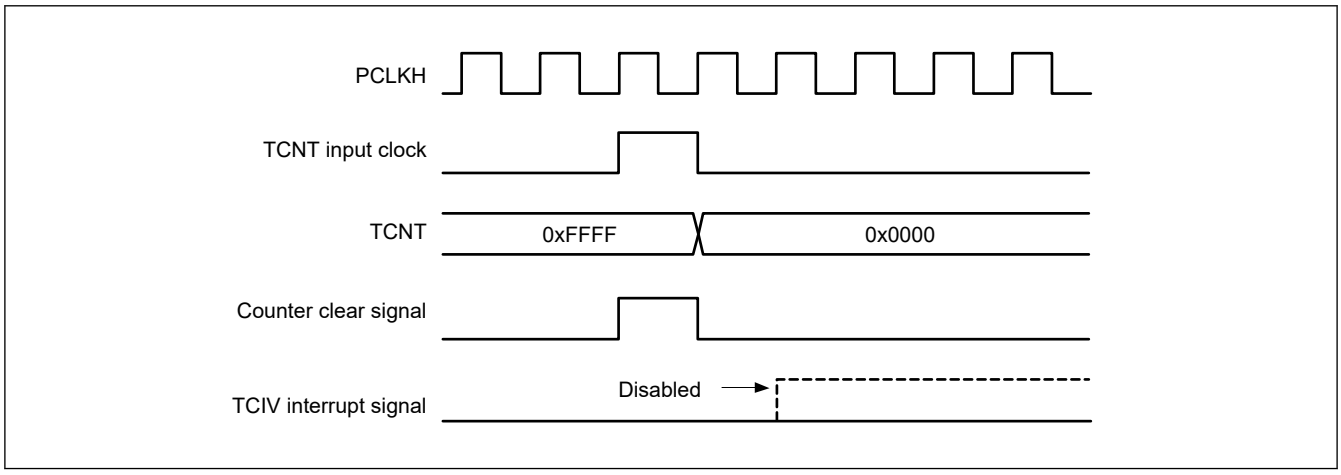


Figure 18.144 Contention between overflow and counter clearing

18.7.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT counts up or down in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. A TCIVn interrupt (n = 0 to 4, 6 to 8) nor a TCIUn interrupt (n = 1, 2) is not generated.

Figure 18.145 shows the operation timing when there is contention between TCNT write operation and overflow.

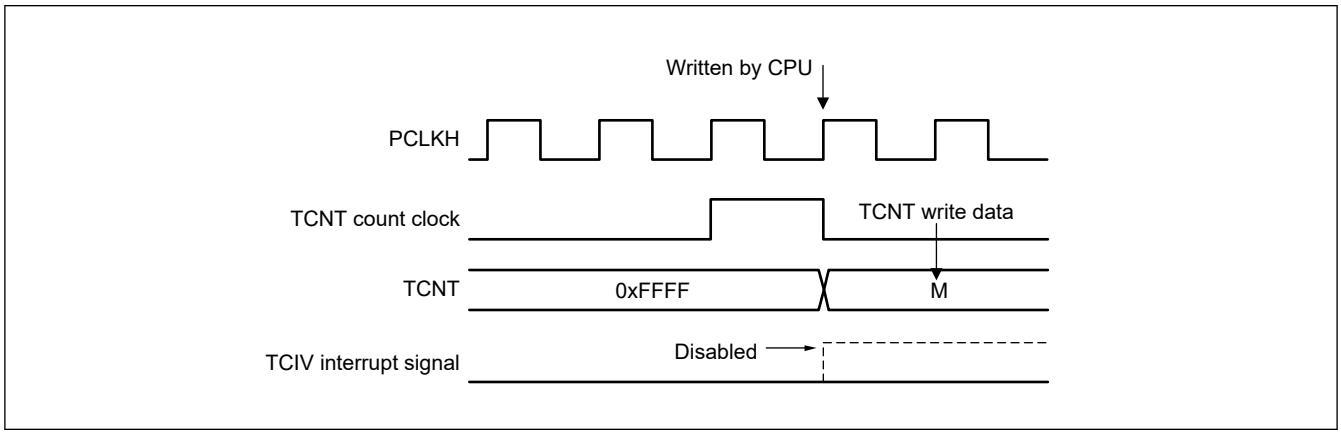


Figure 18.145 Contention between TCNT write operation and overflow

18.7.19 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal operation or PWM mode 1 to reset-synchronized PWM mode in MTU3 and MTU4 (or MTU6 and MTU7), if the counter is stopped while the output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D) are held at a high level and then operation is started after a transition to reset-synchronized PWM mode, the initial pin output will not be correct.

When making a transition from normal mode to reset-synchronized PWM mode, write 0x11 to MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL (MTU6.TIORH, MTU6.TIORL, MTU7.TIORH, and MTU7.TIORL) to initialize the output pin state to a low level, then set the registers to the initial value (0x00) before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronized PWM mode, switch to normal mode, initialize the output pin state to a low level, and then set the registers to the initial value (0x00) before making the transition to reset-synchronized PWM mode.

18.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When MTU3 and MTU4 (or MTU6 and MTU7) are in complementary PWM mode or reset-synchronized PWM mode, the PWM waveform output level is determined by the OLSP and OLSN bits in the timer output control register (TOCR1A or TOCR1B). In complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to 0x00.

The output level in negative phase when the TDERA.TDER (TDERB.TDER) bit is set to 0 in complementary PWM mode (the dead time is not generated) does not depend on the setting of the TOCR1A.OLSN (TOCR1B.OLSN) bit. It is equivalent to the inverted level of positive phase output based on the setting of the TOCR1A.OLSP (TOCR1B.OLSP) bit.

18.7.21 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection

When timer counters 1 and 2 (MTU1.TCNT and MTU2.TCNT) operate as a 32-bit counter in cascade connection, the cascaded counter value cannot be captured successfully in some cases even if input-capture input is simultaneously done to MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B. This is because the input timing of MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B may not be the same when external input-capture signals input into MTU1.TCNT and MTU2.TCNT are taken in synchronization with the internal clock.

For example, MTU1.TCNT (the counter for upper 16 bits) does not capture the count-up value by an overflow from MTU2.TCNT (the counter for lower 16 bits) but captures the count value before the up-counting. In this case, the values of MTU1.TCNT = 0xFFF1 and MTU2.TCNT = 0x0000 should be transferred to MTU1.TGRA and MTU2.TGRA or to MTU1.TGRB and MTU2.TGRB, but the values of MTU1.TCNT = 0xFFF0 and MTU2.TCNT = 0x0000 are erroneously transferred.

The MTU has a new function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input capture input as the trigger. This function can be used to read the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, see section 18.3.12. TICCR : Timer Input Capture Control Register.

18.7.22 Interrupt Skipping Function 2

When interrupt skipping function 2 is in use and the difference between the values in MTU4.TADCORA and MTU4.TADCORB is small, correct counting of the number skipped may not be possible, in which case requests for A/D conversion will not be generated with the expected timing. The conditions listed below thus apply to these settings.

For MTU6 and MTU7, the same conditions apply to the settings of MTU7.TADCORA and MTU7.TADCORB.

1. When the number skipped is zero for skipping function 2
   - The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least four.
   - The interval of comparison for MTU4.TADCORA must be at least four cycles of PCLKH (the updated value of MTU4.TADCORA is set to the previous value plus or minus at least four).
   - The interval of comparison for MTU4.TADCORB must be at least four cycles of PCLKH (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least four).
2. When the number skipped is one or more for skipping function 2
   - The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least two.
   - The interval of comparison for MTU4.TADCORB must be at least two cycles of PCLKH (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least two).

18.7.23 Notes When Complementary PWM Mode Output Protection Function is Not Used

The complementary PWM mode output protection function is initially enabled. If it is not used, write 0x00 in the POE3.POECR2 register.

18.7.24 Notes Regarding Timer Counter (MTU5.TCNT) and Timer General Register (MTU5.TGR)

Do not set an MTU5.TGRj (j = U, V, W) bit to the value of the corresponding MTU5.TCNTj (j = U, V, W) plus one while counting by the MTU5.TCNTj (j = U, V, W) register is stopped. If an MTU5.TGRj (j = U, V, W) bit is set to the value of

the corresponding MTU5.TCNTj (j = U, V, W) plus one while counting by the MTU5.TCNTj (j = U, V, W) is stopped, a compare-match will be generated even though counting is stopped.

In this case, if the compare match enable bit (MTU5.TIER.TGIE5j (j = U, V, W) bit is set to 1 (enabling interrupts), a compare-match interrupt will also be generated. If the value of the timer compare match clear register is 1 (enabled), the timer is automatically cleared to 0x0000 when the compare-match is generated, regardless of whether interrupts from the MTU5.TCNTj (j = U, V, W) are enabled or disabled.

18.7.25 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCRA.WRE bit = 1 or TWCRB.WRE bit = 1) at the time of synchronous counter clearing in complementary PWM mode, satisfaction of either condition 1 or 2 below has the following effects.

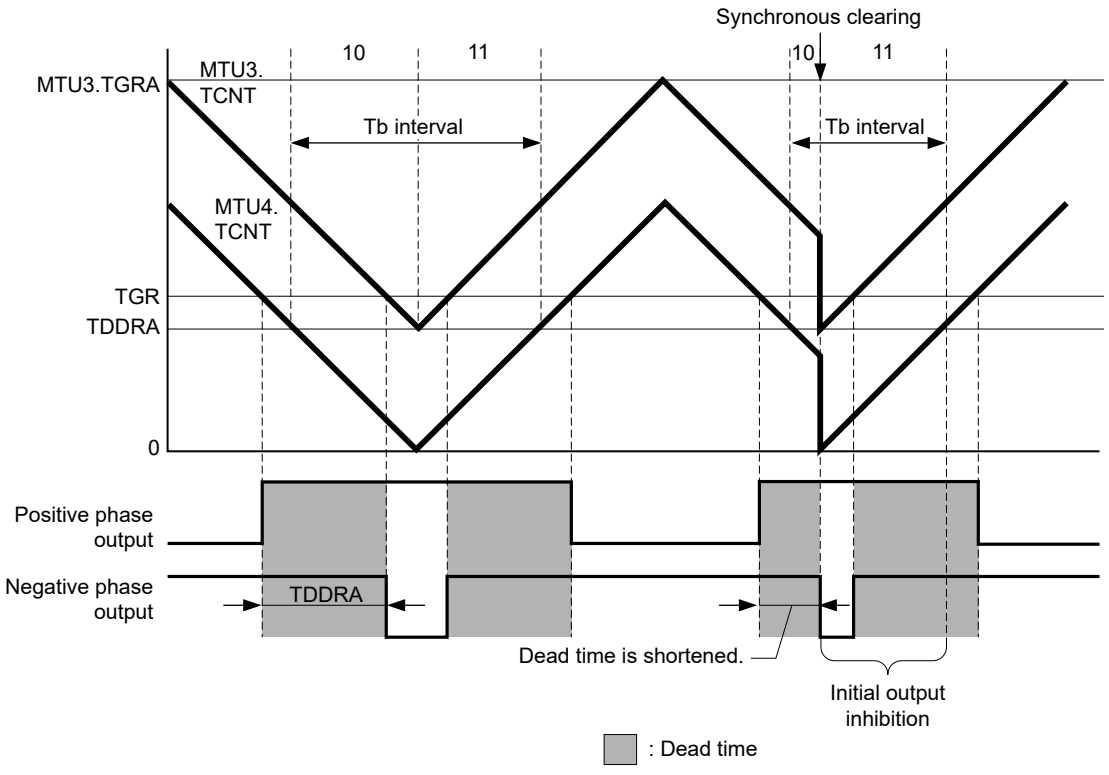
- Dead time on the PWM output pins is shortened (or disappears).
- The active level is output on the negative phase PWM output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 18.146, synchronous clearing occurs within the dead-time period for PWM output.

Condition 2: In portions (10) and (11) of the initial output inhibition period in Figure 18.147, synchronous clearing occurs when any condition from among MTU3.TGRB (MTU6.TGRB) ≤ TDDRA, MTU4.TGRA (MTU7.TGRA) ≤ TDDRA, or MTU4.TGRB (MTU7.TGRB) ≤ TDDRA is satisfied.

The following method avoids the above phenomena.

Ensure that synchronous clearing proceeds with the value of each comparison register (MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB)) set to at least double the value of the TDDRA register (TDDRB register).



Note: PWM output is active-low.

Figure 18.146 Example of synchronous clearing (when condition 1 applies)

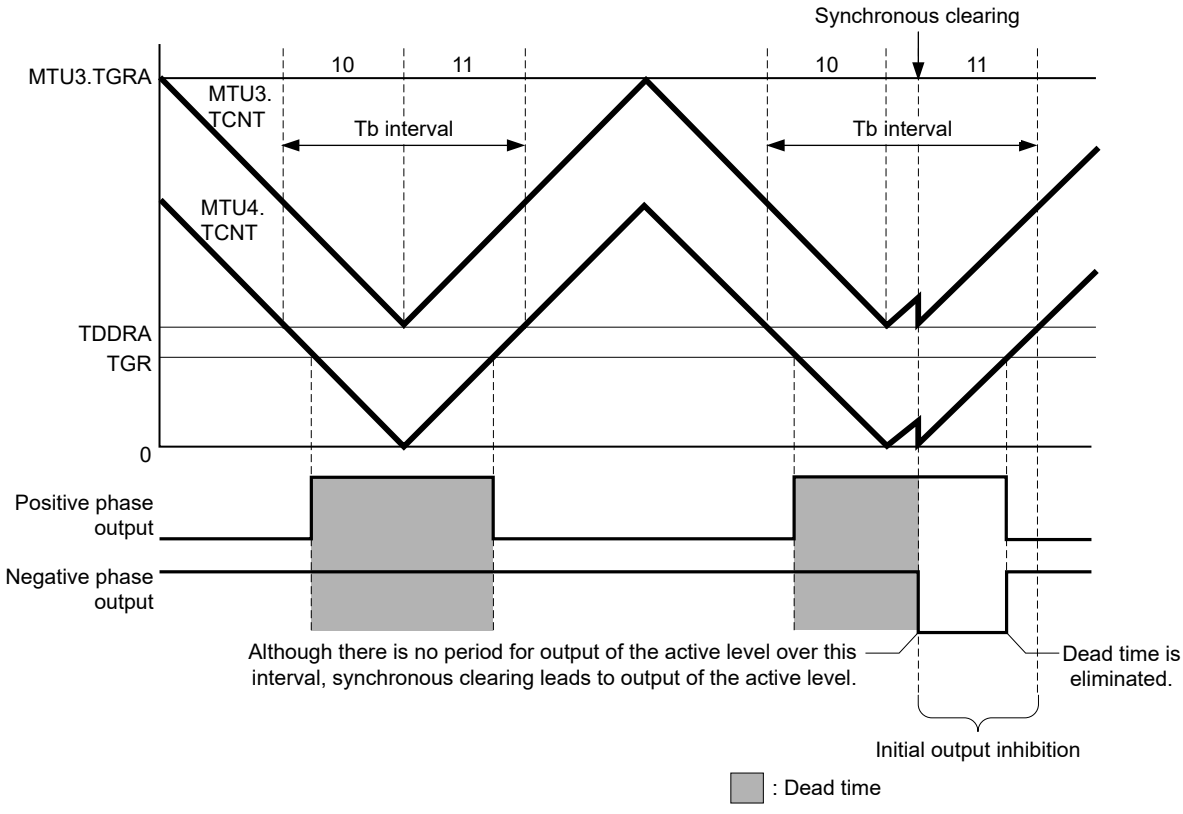


Figure 18.147 Example of synchronous clearing (when condition 2 Applies)

18.7.26 Notes on Timer Mode Register Setting for ELC Event Input

When MTU is used in ELC operation, set the timer mode register (TMDR) of the relevant channel to its initial value (0x00).

18.7.27 Continuous Output of Interrupt Signal in Response to a Compare Match

When the TGR register is set to 0x0000, the PCLKH/1 clock is set as the counter clock, and compare match is set as the trigger for clearing of the counter clock, the value of the TCNT counter remains 0x0000, and the interrupt signal is output continuously (i.e. its level is flat) rather than output over a single cycle. Consequently, interrupts are not detected in response to second and subsequent compare matches.

Figure 18.148 shows the timing for continuous output of the interrupt signal in response to a compare match.

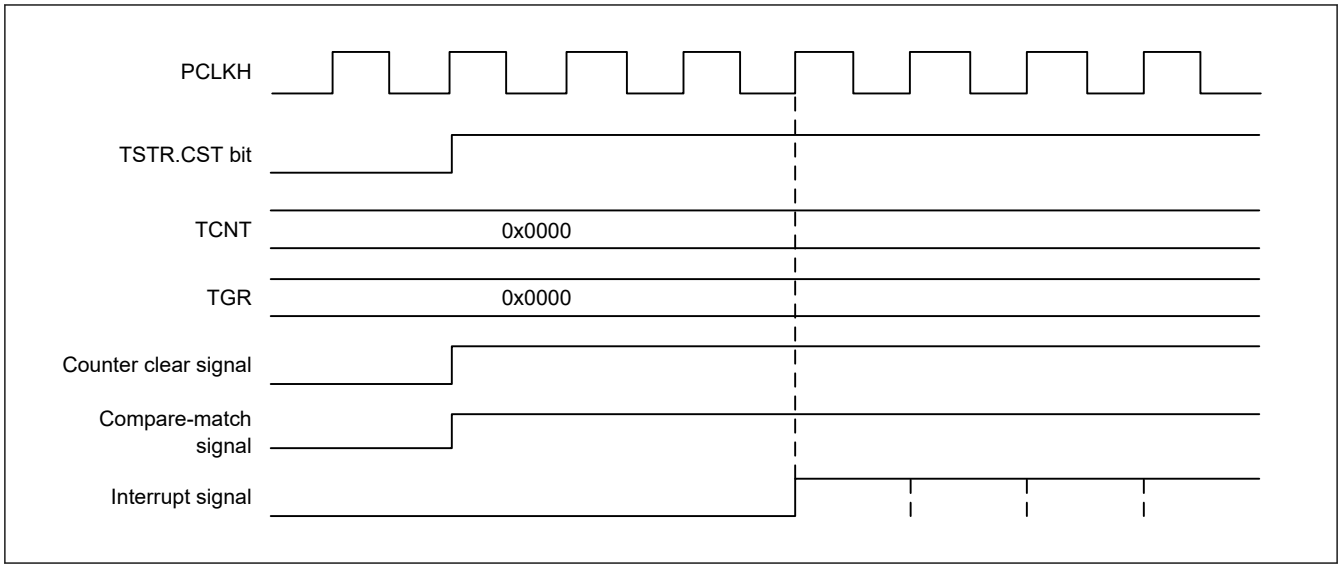


Figure 18.148 Continuous output of interrupt signal in response to a compare match

18.8 MTU Output Pin Initialization

18.8.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (MTU0 to MTU4 and MTU6 to MTU8)
- PWM mode 1 (MTU0 to MTU4, MTU6, and MTU7)
- PWM mode 2 (MTU0 to MTU2)
- Phase counting modes 1 to 5 (MTU1 and MTU2)
- Complementary PWM mode (MTU3, MTU4, MTU6, and MTU7)
- Reset-synchronized PWM mode (MTU3, MTU4, MTU6, and MTU7)

This section describes how to initialize the MTU output pins in each of these modes.

18.8.2 Operation in Case of Re-Setting Due to Error during Operation

If an error occurs during MTU operation, MTU output should be cut off by the system. The output can be cut off by allowing non-active level output from the pins by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports. MTU output can be disabled through TIOR settings. Complementary PWM output (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D) should be specified through TOERA and TOERB settings. For PWM output pins, output can also be cut by hardware, using port output enable 3 (POE3). The pin initialization procedures for re-setting due to an error during operation and the procedures for restarting in a different mode after re-setting are described below.

The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Available mode transition combinations are shown in Table 18.76.

Table 18.76 Mode transition combinations (1 of 2)

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	Not available	Not available
PCM	(17)	(18)	(19)	(20)	Not available	Not available

Table 18.76 Mode transition combinations (2 of 2)

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
CPWM	(21)	(22)	Not available	Not available	(23) (24)	(25)
RPWM	(26)	(27)	Not available	Not available	(28)	(29)

Note: Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 5
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

18.8.3 Overview of Initialization Procedures and Mode Transitions in Case of Error During Operation

- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of TIOR setting.
- In PWM mode 1, waveforms are not output to the MTIOCnB and MTIOCnD (n = 3, 4, 6, 7) pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.
- In PWM mode 2, waveforms are not output to the cycle register pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, waveforms are not output to the corresponding pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, waveforms are not output to the corresponding pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the Timer Output Control register (TOCR1A, TOCR2A, TOCR1B, or TOCR2B) setting, temporarily disable output in MTU3 and MTU4 (or MTU6 and MTU7) with the Timer Output Master Enable register (TOERA or TOERB). At this time, if no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports. Switch to normal mode, perform initialization with TIOR, restore TIOR to its initial value, then operate the MTU in accordance with the mode setting procedure (TOCR1A setting, TOCR2A setting, TMDR1 setting, and TOERA setting (TOCR1B setting, TOCR2B setting, TMDR1 setting, and TOERB setting)).

Note: Channel number is substituted for "n" indicated in this section, unless otherwise stated.

Pin initialization procedures are described below for the numbered combinations in Table 18.76. The active level is assumed to be low.

18.8.3.1 Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 18.149 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after resetting.

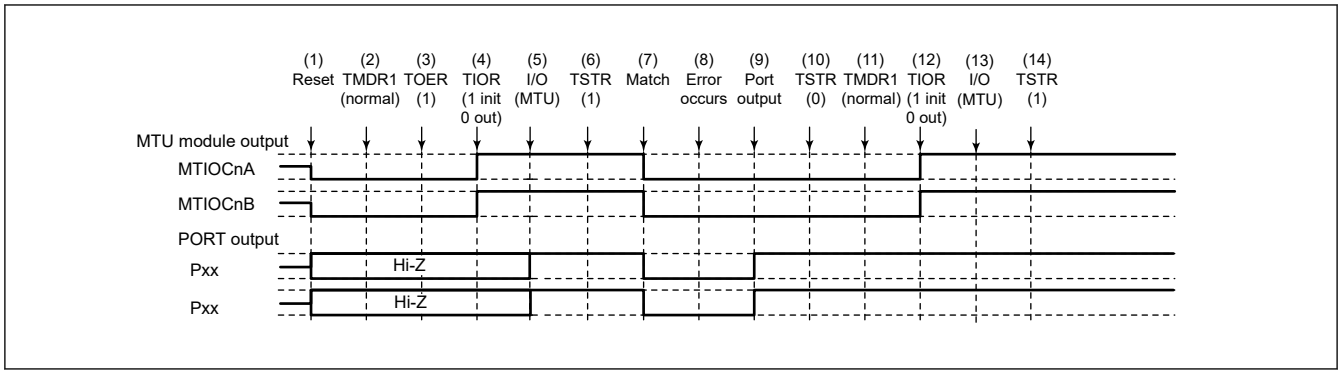


Figure 18.149 Error occurrence in normal mode, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) After a reset, the TMDR1 setting is for normal mode.

(3) For MTU3 and MTU4 (MTU6 and MTU7), enable output with the TOERA (TOERB) register before initializing the pins with the TIOR register.

(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)

(5) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(6) Start count operation by setting the TSTRA (TSTRB) register.

(7) Output goes low on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(10) Stop count operation by setting the TSTRA (TSTRB) register.

(11) This step is not necessary when restarting in normal mode.

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.2 Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 18.150 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

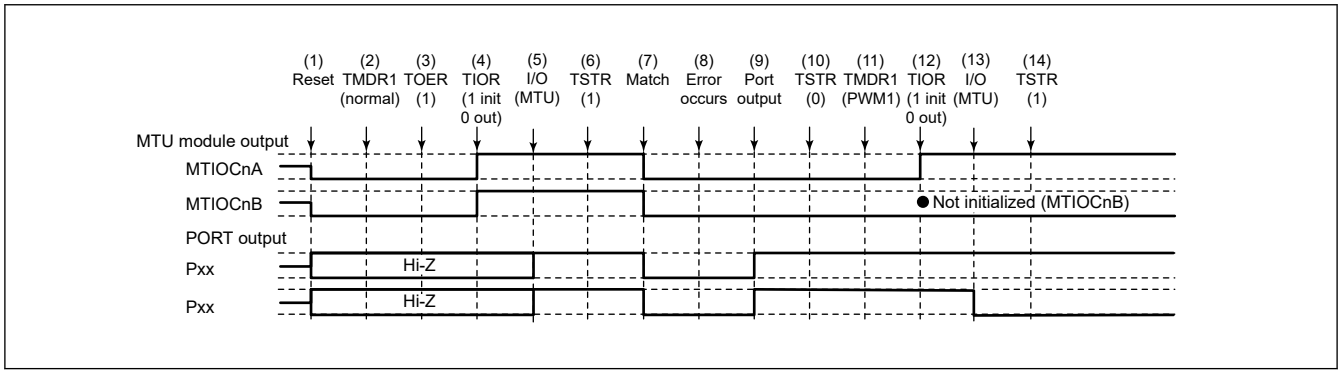


Figure 18.150 Error occurrence in normal mode, recovery in PWM mode 1

(1) to (10) are the same as in Figure 18.149.

(11) Set PWM mode 1.

(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.3 Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM mode 2

Figure 18.151 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

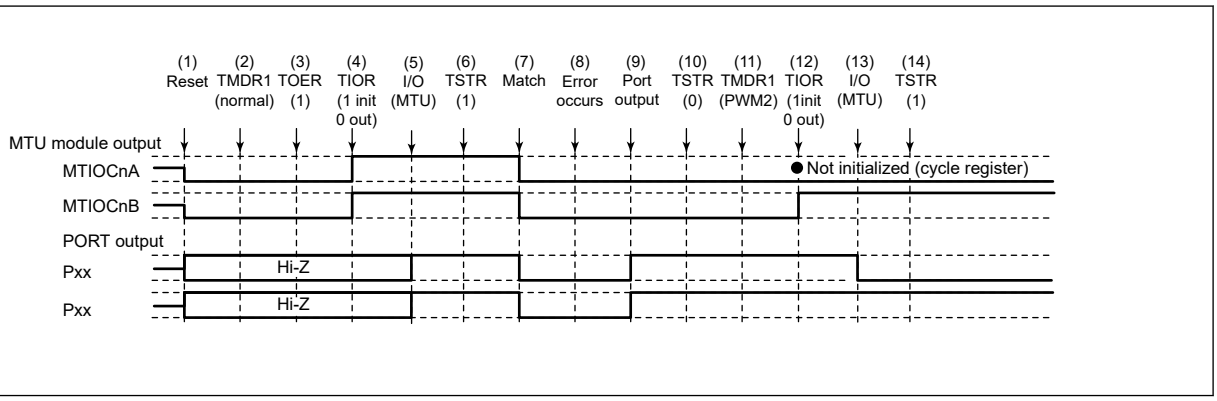


Figure 18.151 Error occurrence in normal mode, recovery in PWM mode 2

(1) to (10) are the same as in Figure 18.149.

(11) Set PWM mode 2.

(12) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore the TOERA register setting is not necessary.

18.8.3.4 Operation When Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 18.152 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

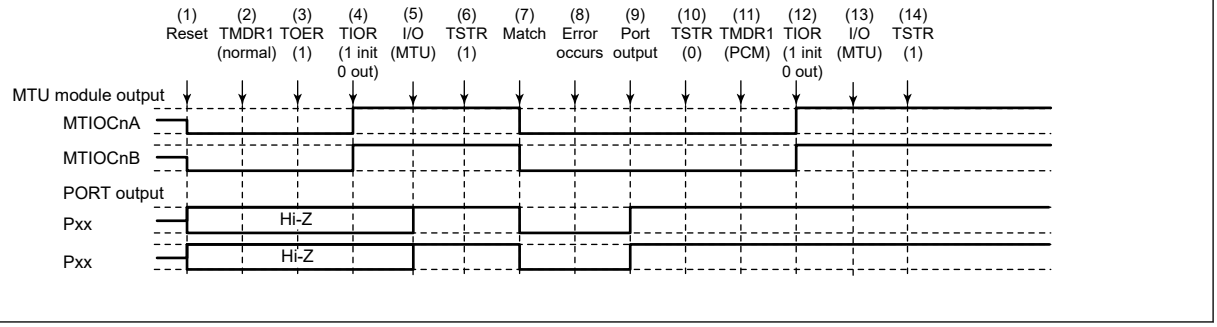


Figure 18.152 Error occurrence in normal mode, recovery in phase counting mode

(1) to (10) are the same as in Figure 18.149.

(11) Set the phase counting mode.

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOERA register setting is not necessary.

18.8.3.5 Operation When Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 18.153 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

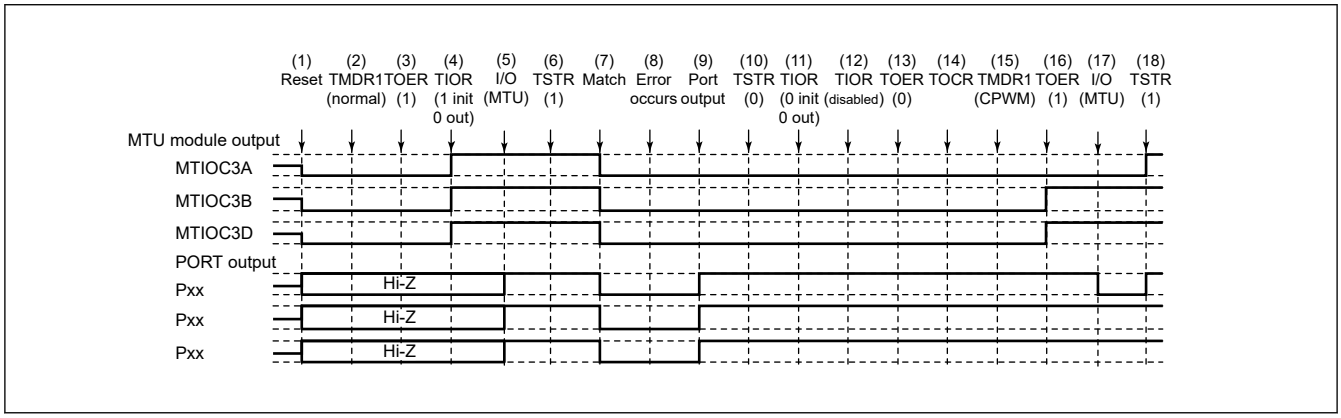


Figure 18.153 Error occurrence in normal mode, recovery in complementary PWM mode

(1) to (10) are the same as in Figure 18.149.

(11) Initialize the normal mode waveform generation block with the TIOR register.

(12) Disable operation of the normal mode waveform generation block with the TIOR register.

(13) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(14) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(15) Set complementary PWM mode.

(16) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(17) Set MTU output using the I/O port registers corresponding to the I/O ports.

(18) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.6 Operation When Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 18.154 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

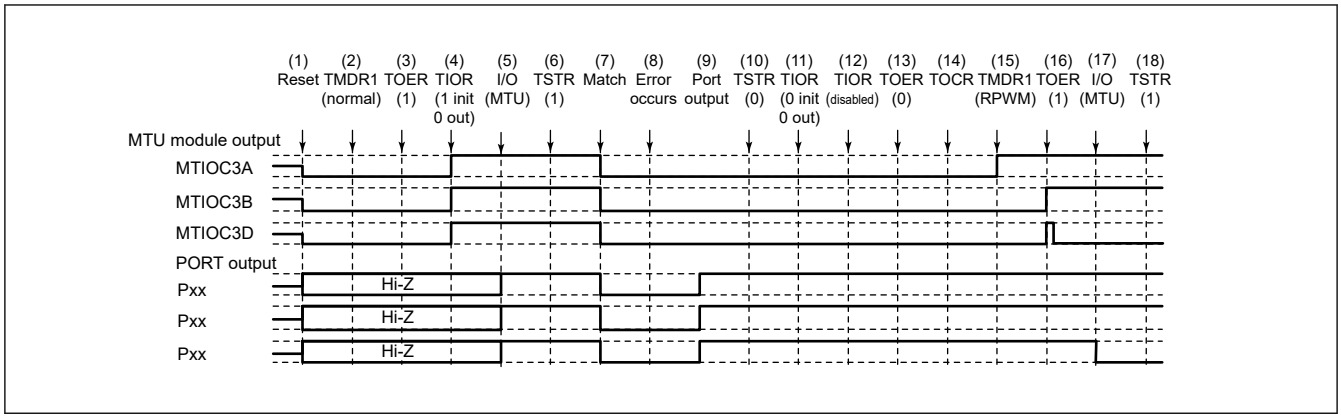


Figure 18.154 Error occurrence in normal mode, recovery in reset-synchronized PWM mode

(1) to (13) are the same as in Figure 18.153.

(14) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(15) Set reset-synchronized PWM mode.

(16) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(17) Set MTU output using the I/O port registers corresponding to the I/O ports.

(18) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.7 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 18.155 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

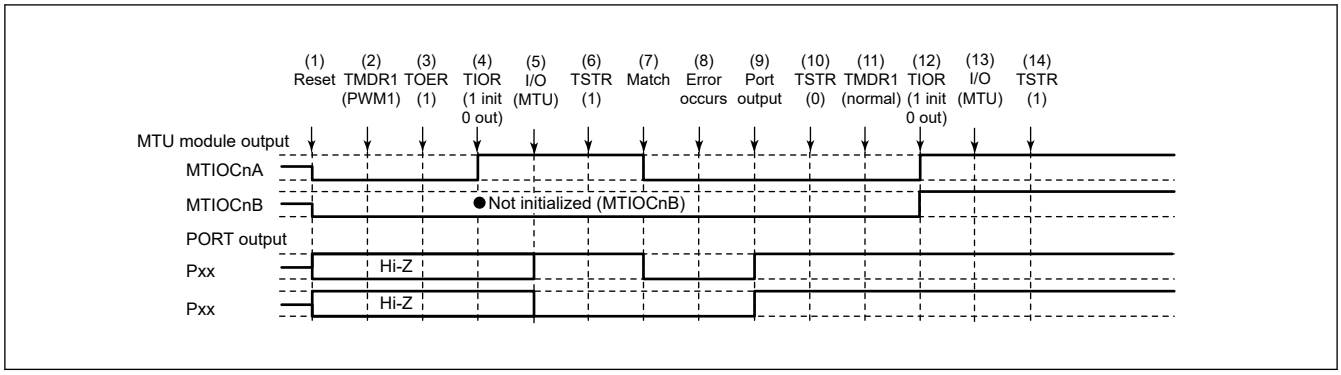


Figure 18.155 Error occurrence in PWM mode 1, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Set PWM mode 1.

(3) For MTU3 and MTU4 (MTU6 and MTU7), enable output with the TOERA (TOERB) register before initializing the pins with the TIOR register.

(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 1, the MTIOCnB side is not initialized.)

(5) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(6) Start count operation by setting the TSTRA (TSTRB) register.

(7) Output goes low on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(10) Stop count operation by setting the TSTRA (TSTRB) register.

(11) Set normal mode.

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.8 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 18.156 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

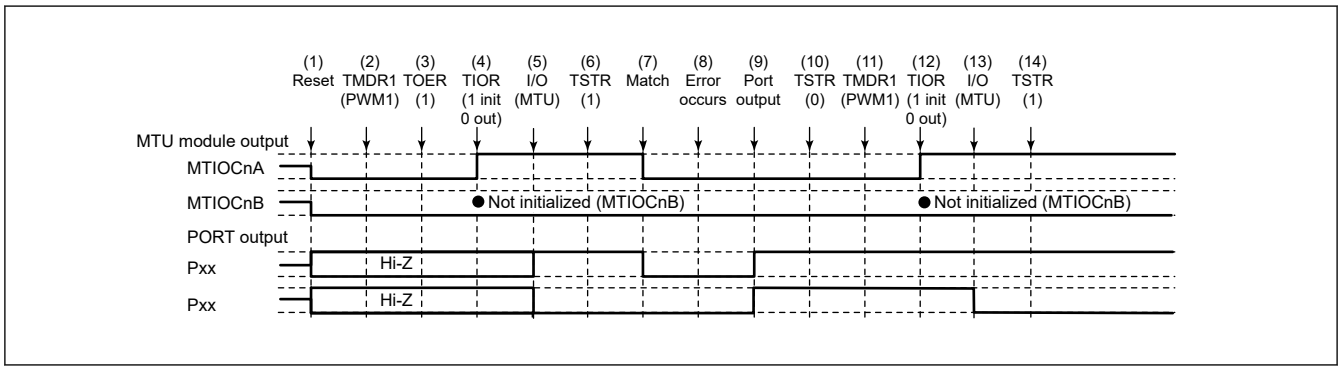


Figure 18.156 Error occurrence in PWM mode 1, recovery in PWM mode 1

(1) to (10) are the same as in Figure 18.155.

(11) This step is not necessary when restarting in PWM mode 1.

(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.9 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 18.157 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

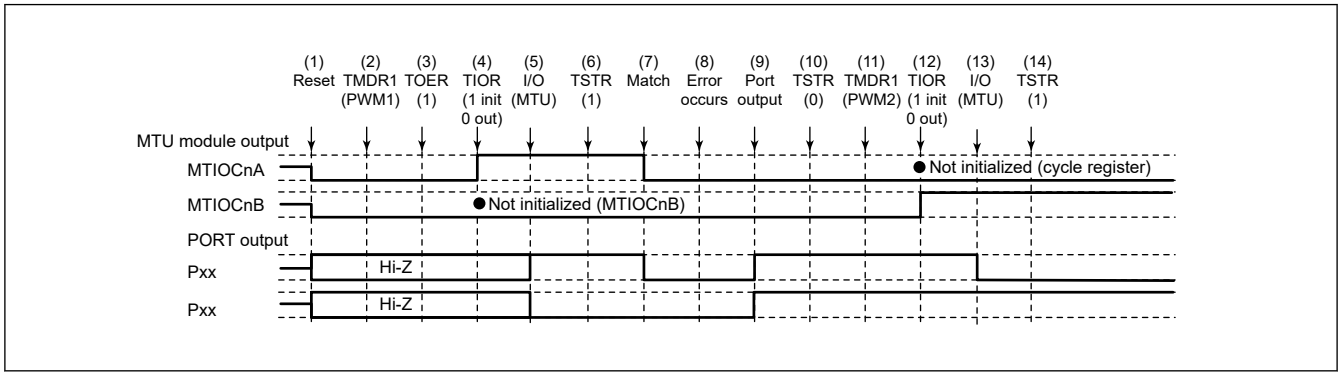


Figure 18.157 Error occurrence in PWM mode 1, recovery in PWM mode 2

(1) to (10) are the same as in Figure 18.155.

(11) Set PWM mode 2.

(12) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore the TOERA register setting is not necessary.

18.8.3.10 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 18.158 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

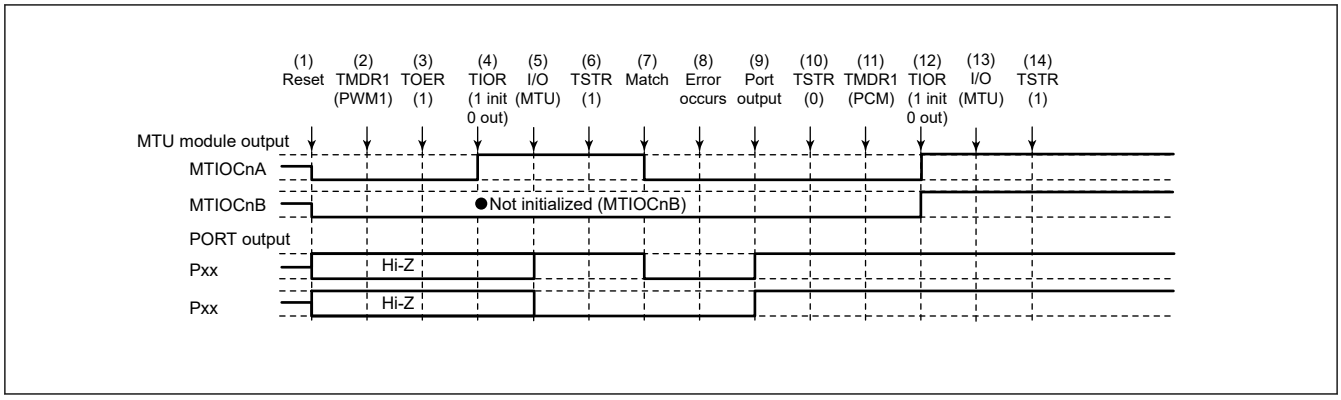


Figure 18.158 Error occurrence in PWM mode 1, recovery in phase counting mode

(1) to (10) are the same as in Figure 18.155.

(11) Set the phase counting mode.

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOERA register setting is not necessary.

18.8.3.11 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 18.159 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

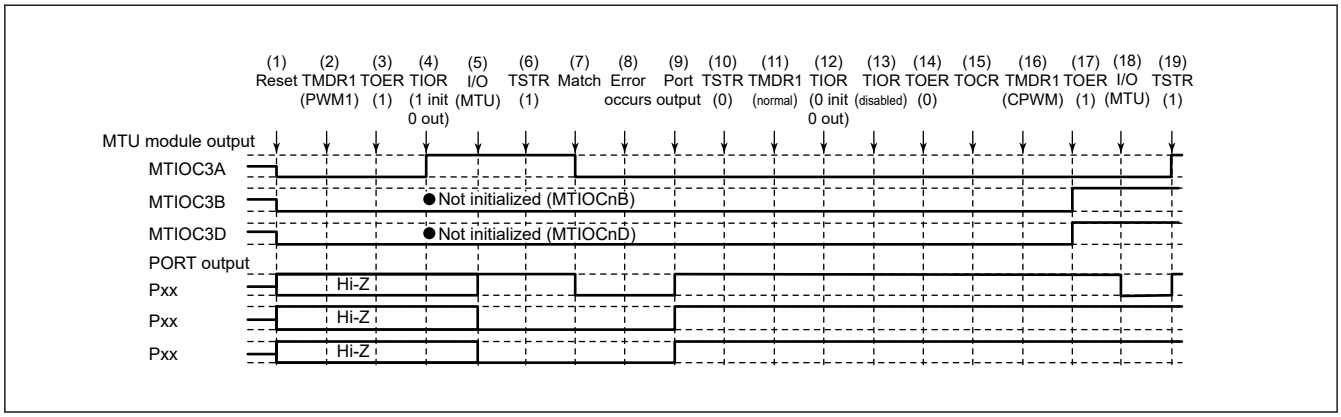


Figure 18.159 Error occurrence in PWM mode 1, recovery in complementary PWM mode

(1) to (10) are the same as in Figure 18.155.

(11) Set normal mode to initialize the normal mode waveform generation block.

(12) Initialize the PWM mode 1 waveform generation block with the TIOR register.

(13) Disable operation of the PWM mode 1 waveform generation block with the TIOR register.

(14) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(15) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TSTRB).

(16) Set complementary PWM mode.

(17) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(18) Set MTU output using the I/O port registers corresponding to the I/O ports.

(19) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.12 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 18.160 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

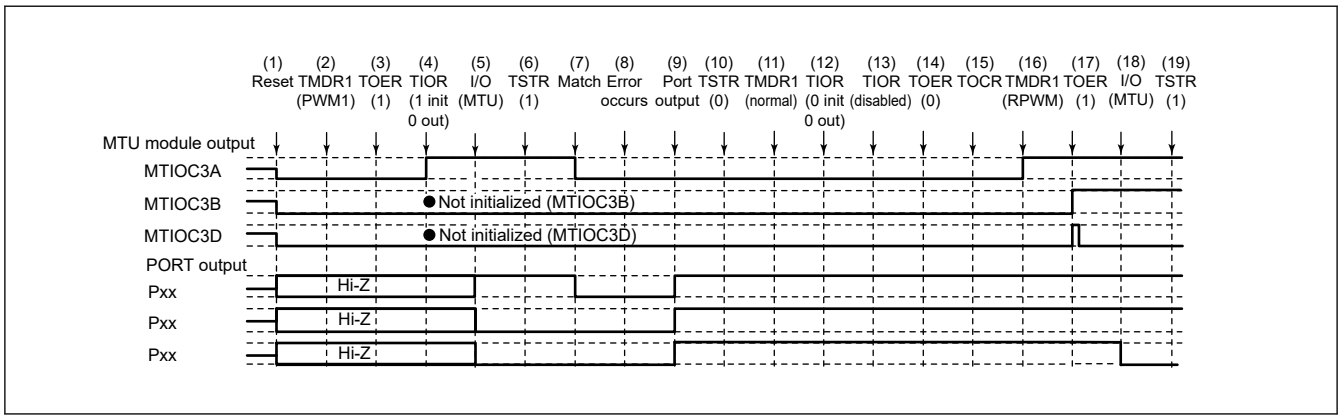


Figure 18.160 Error occurrence in PWM mode 1, recovery in reset-synchronized PWM mode

(1) to (14) are the same as in Figure 18.159.

(15) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(16) Set reset-synchronized PWM mode.

(17) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(18) Set MTU output using the I/O port registers corresponding to the I/O ports.

(19) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.13 Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 18.161 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

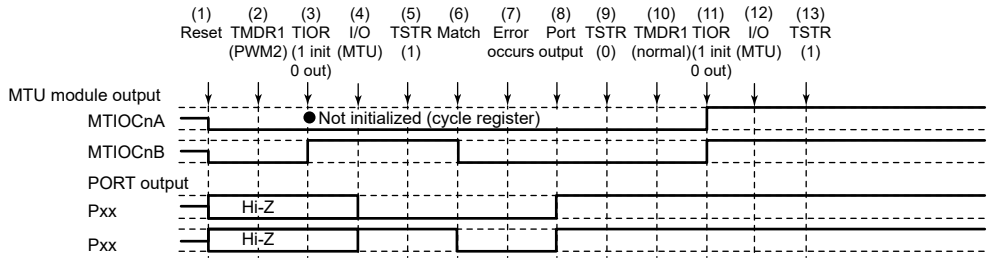


Figure 18.161 Error occurrence in PWM mode 2, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Set PWM mode 2.

(3) Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, MTIOCnA is the cycle register.)

(4) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(5) Start count operation by setting the TSTRA (TSTRB) register.

(6) Output goes low on compare match occurrence.

(7) An error occurs.

(8) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(9) Stop count operation by setting the TSTRA (TSTRB) register.

(10) Set normal mode.

(11) Initialize the pins with the TIOR register.

(12) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.14 Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 18.162 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

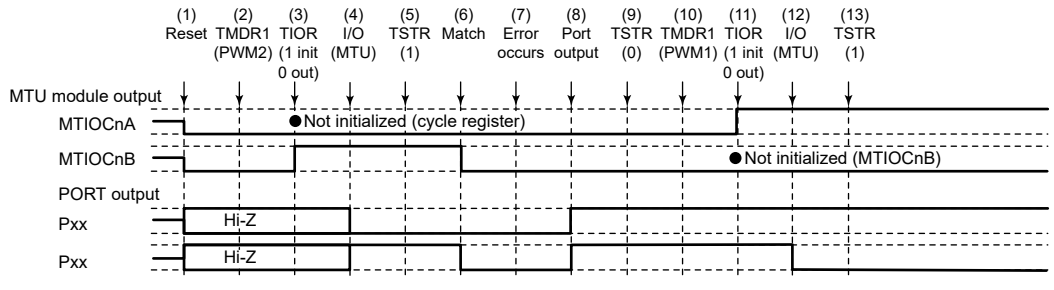


Figure 18.162 Error occurrence in PWM mode 2, recovery in PWM mode 1

(1) to (9) are the same as in Figure 18.161.

(10) Set PWM mode 1.

(11) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(12) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.15 Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 18.163 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

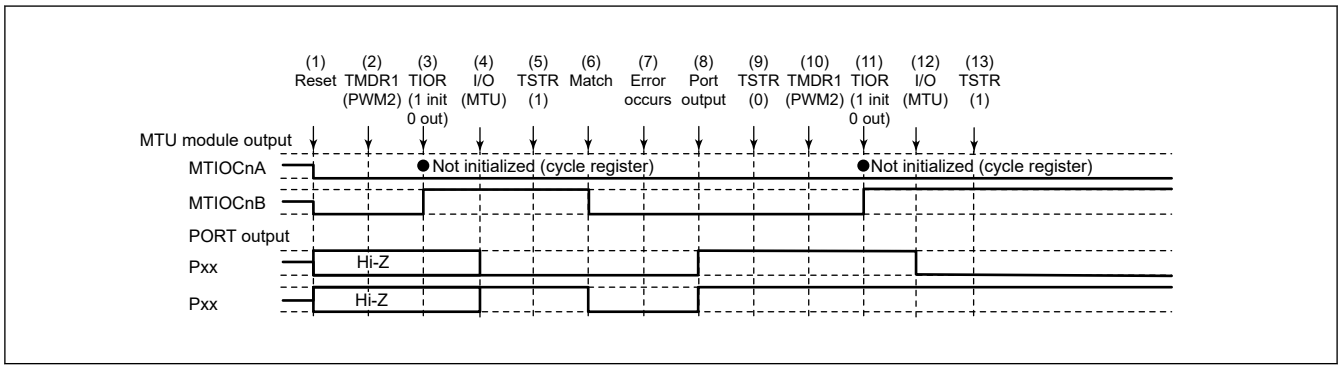


Figure 18.163 Error occurrence in PWM mode 2, recovery in PWM mode 2

(1) to (9) are the same as in Figure 18.161.

(10) This step is not necessary when restarting in PWM mode 2.

(11) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(12) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA register.

18.8.3.16 Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 18.164 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

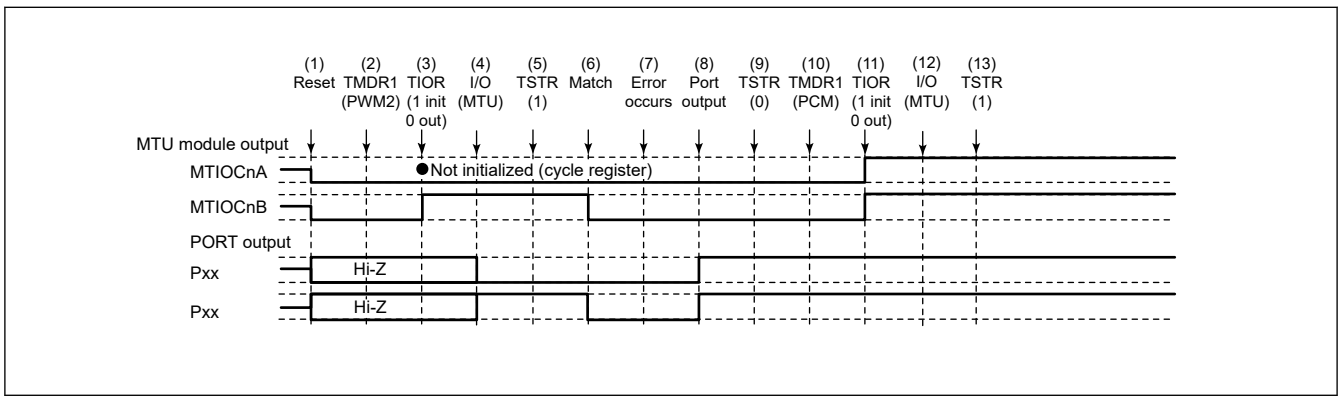


Figure 18.164 Error occurrence in PWM mode 2, recovery in phase counting mode

(1) to (9) are the same as in Figure 18.161.

(10) Set the phase counting mode.

(11) Initialize the pins with the TIOR register.

(12) Set MTU output using the I/O port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA register.

18.8.3.17 Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 18.165 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

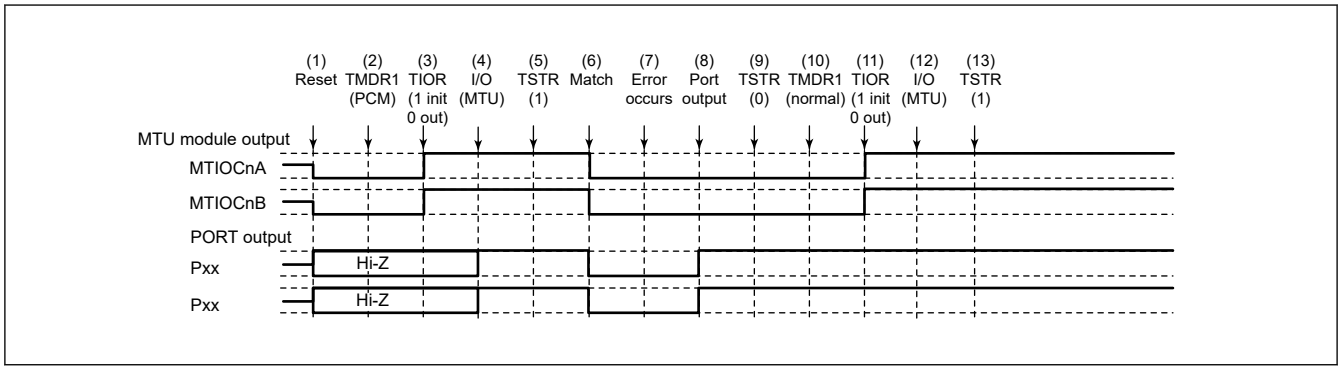


Figure 18.165 Error occurrence in phase counting mode, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Set phase counting mode.

(3) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)

(4) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(5) Start count operation by setting the TSTRA (TSTRB) register.

(6) Output goes low on compare match occurrence.

(7) An error occurs.

(8) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(9) Stop count operation by setting the TSTRA (TSTRB) register.

(10) Set normal mode.

(11) Initialize the pins with the TIOR register.

(12) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.18 Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 18.166 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

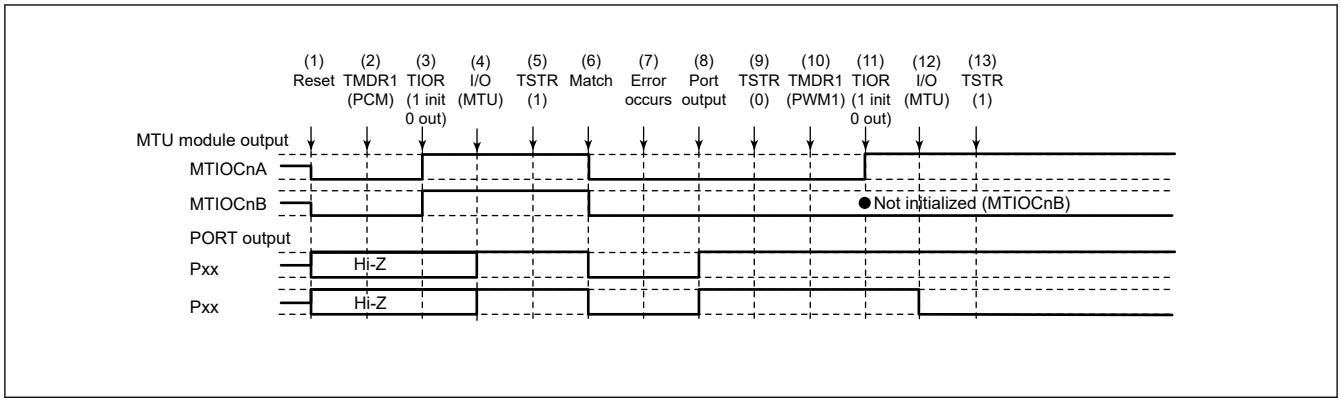


Figure 18.166 Error occurrence in phase counting mode, recovery in PWM mode 1

(1) to (9) are the same as in Figure 18.165.

(10) Set PWM mode 1.

(11) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(12) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.19 Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 18.167 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

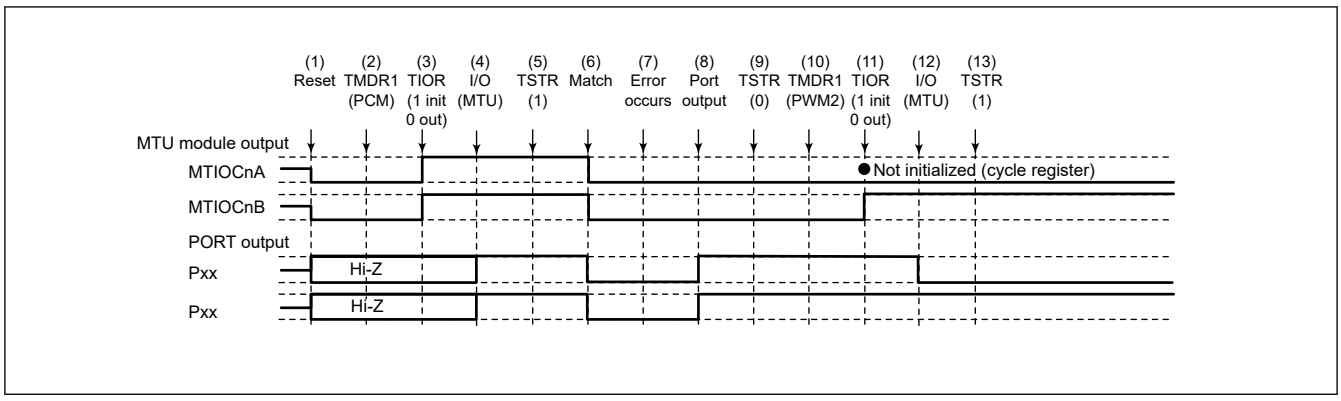


Figure 18.167 Error occurrence in phase counting mode, recovery in PWM mode 2

(1) to (9) are the same as in Figure 18.165.

(10) Set PWM mode 2.

(11) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(12) Set MTU output using the I/O port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA register.

18.8.3.20 Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 18.168 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

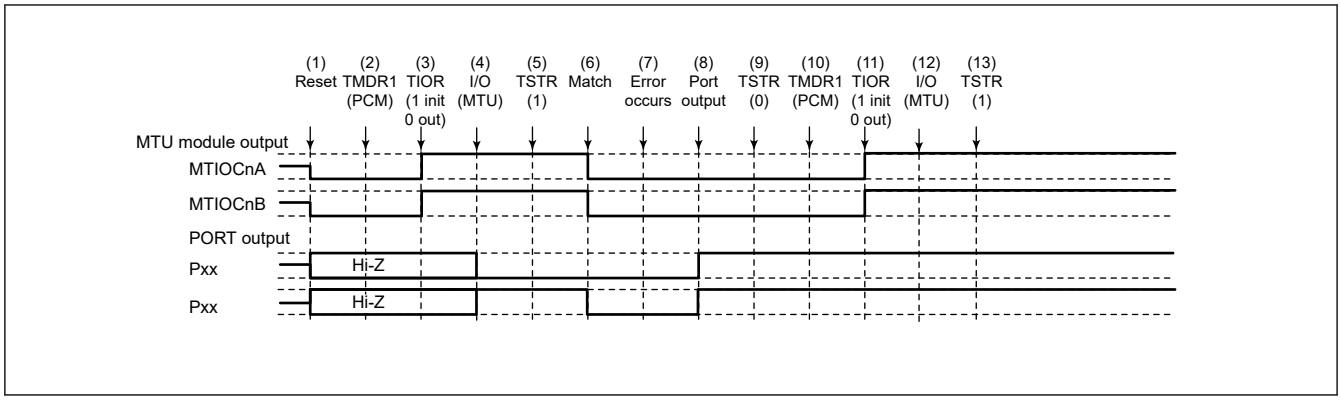


Figure 18.168 Error occurrence in phase counting mode, recovery in phase counting mode

(1) to (9) are the same as in Figure 18.165.

(10) This step is not necessary when restarting in phase counting mode.

(11) Initialize the pins with the TIOR register.

(12) Set MTU output using the I/O port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA register.

18.8.3.21 Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 18.169 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

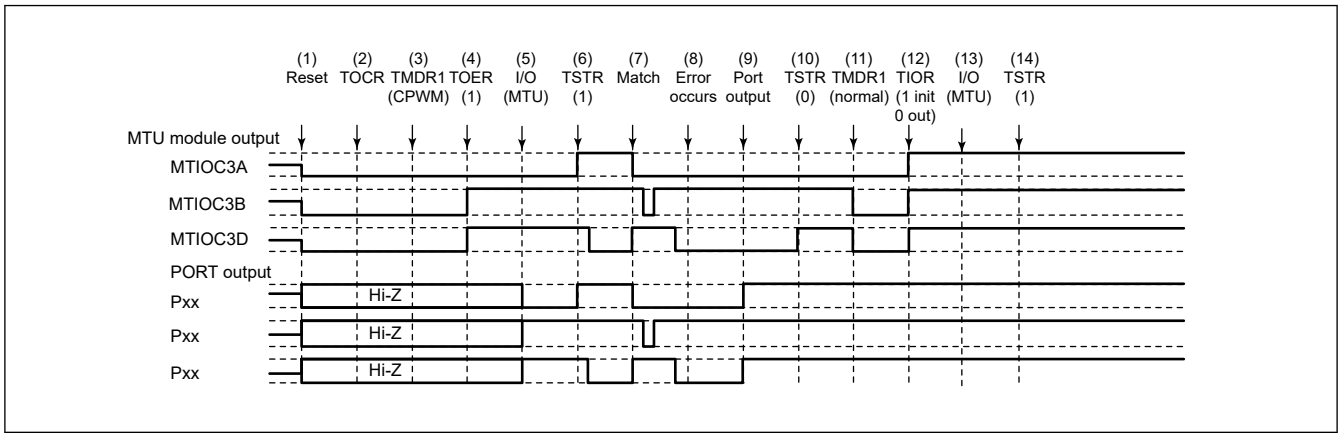


Figure 18.169 Error occurrence in complementary PWM mode, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(3) Set complementary PWM mode.

(4) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(5) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(6) Start count operation by setting the TSTRA (TSTRB) register.

(7) The complementary PWM waveform is output on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(10) Stop count operation by setting the TSTRA (TSTRB) register. (MTU output becomes the initial complementary PWM output value).

(11) Set normal mode (MTU output goes low).

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.22 Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 18.170 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

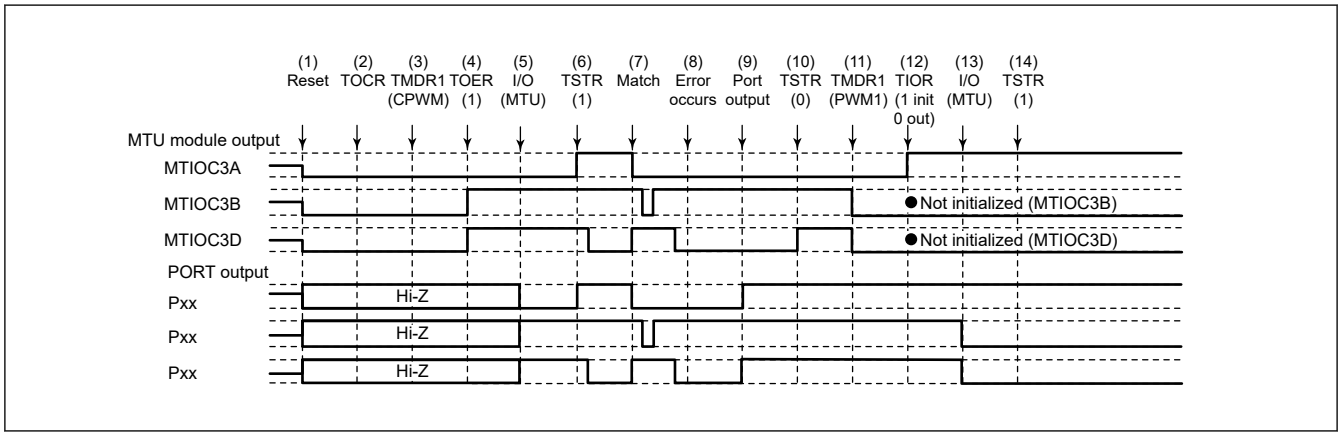


Figure 18.170 Error occurrence in complementary PWM mode, recovery in PWM mode 1

(1) to (10) are the same as in Figure 18.169.

(11) Set PWM mode 1 (MTU output goes low).

(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.23 Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 18.171 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the period and duty settings at the time of stopping the counter).

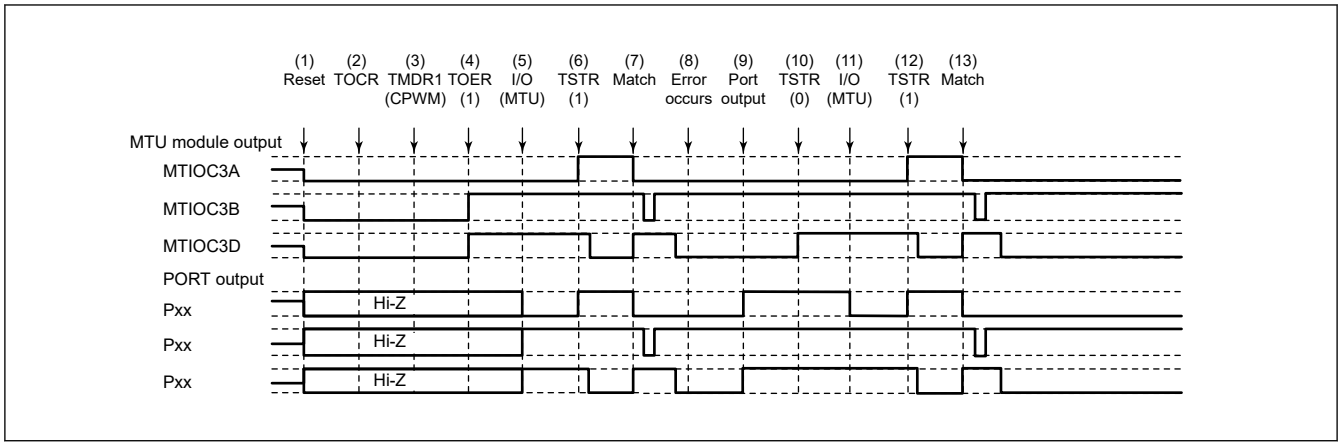


Figure 18.171 Error occurrence in complementary PWM mode, recovery in complementary PWM mode (1)

(1) to (10) are the same as in Figure 18.169.

(11) Set MTU output using the I/O port registers corresponding to the I/O ports.

(12) Restart operation by setting the TSTRA (TSTRB) register.

(13) The complementary PWM waveform is output on compare match occurrence.

18.8.3.24 Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode with New Settings

Figure 18.172 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (operation is restarted using new period and duty ratio settings).

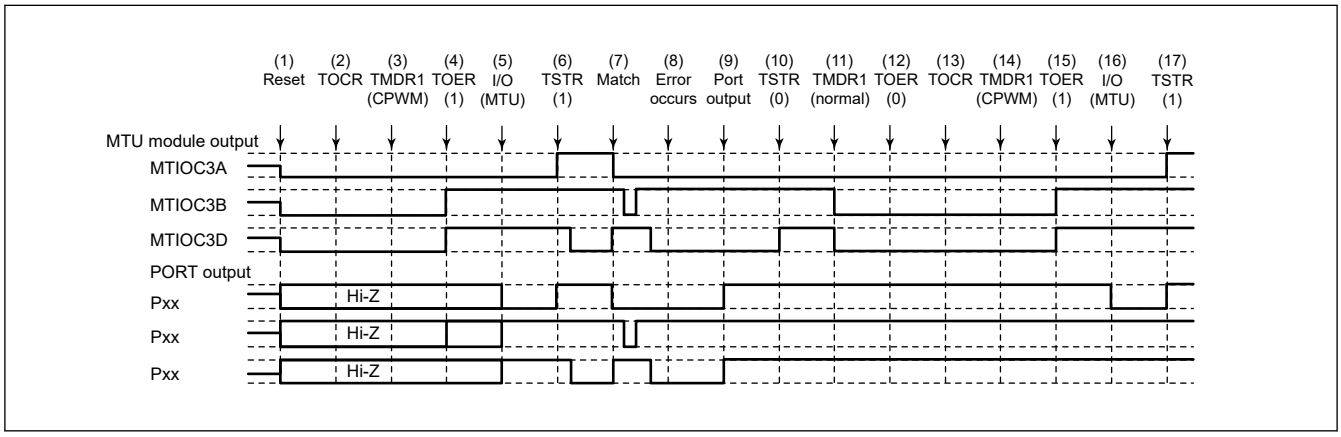


Figure 18.172 Error occurrence in complementary PWM mode, recovery in complementary PWM mode (2)

(1) to (10) are the same as in Figure 18.169.

(11) Set normal mode and make new settings (MTU output goes low).

(12) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(13) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(14) Set complementary PWM mode.

(15) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(16) Set MTU output using the I/O port registers corresponding to the I/O ports.

(17) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.25 Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 18.173 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

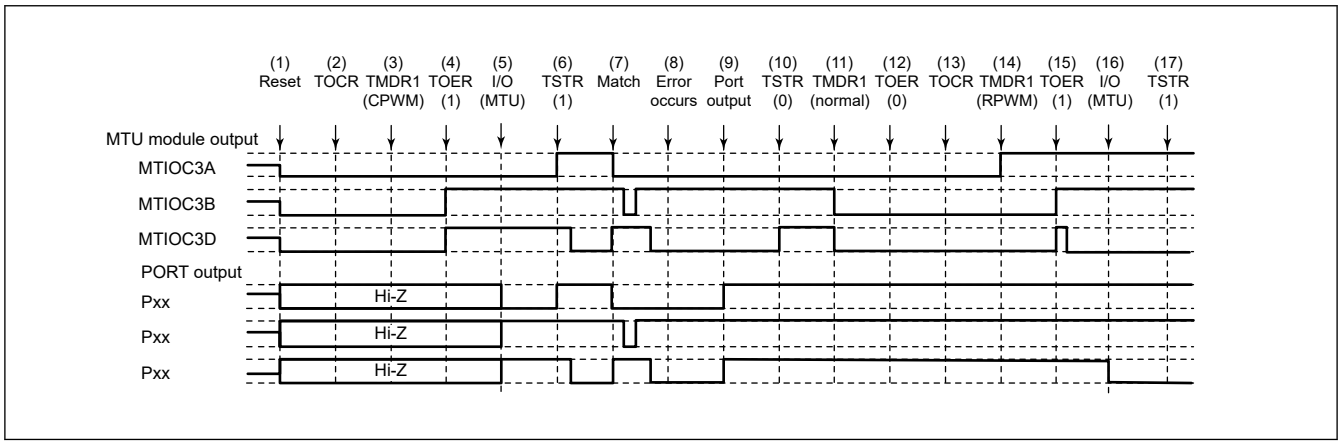


Figure 18.173 Error occurrence in complementary PWM mode, recovery in reset-synchronized PWM mode

(1) to (10) are the same as in Figure 18.169.

(11) Set normal mode (MTU output goes low).

(12) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(13) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(14) Set reset-synchronized PWM mode.

(15) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(16) Set MTU output using the I/O port registers corresponding to the I/O ports.

(17) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.26 Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 18.174 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

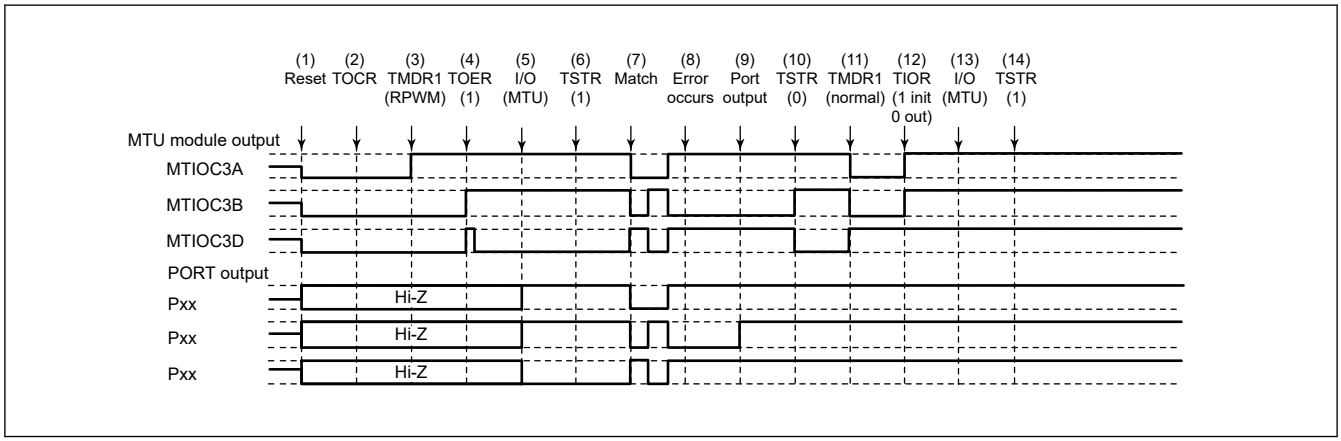


Figure 18.174 Error occurrence in reset-synchronized PWM mode, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(3) Set reset-synchronized PWM mode.

(4) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(5) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(6) Start count operation by setting the TSTRA (TSTRB) register.

(7) The reset-synchronized PWM waveform is output on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(10) Stop count operation by setting the TSTRA (TSTRB) register. (MTU output becomes the initial reset-synchronized PWM output value.)

(11) Set normal mode (positive-phase MTU output goes low, and negative-phase output goes high).

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.27 Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 18.175 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

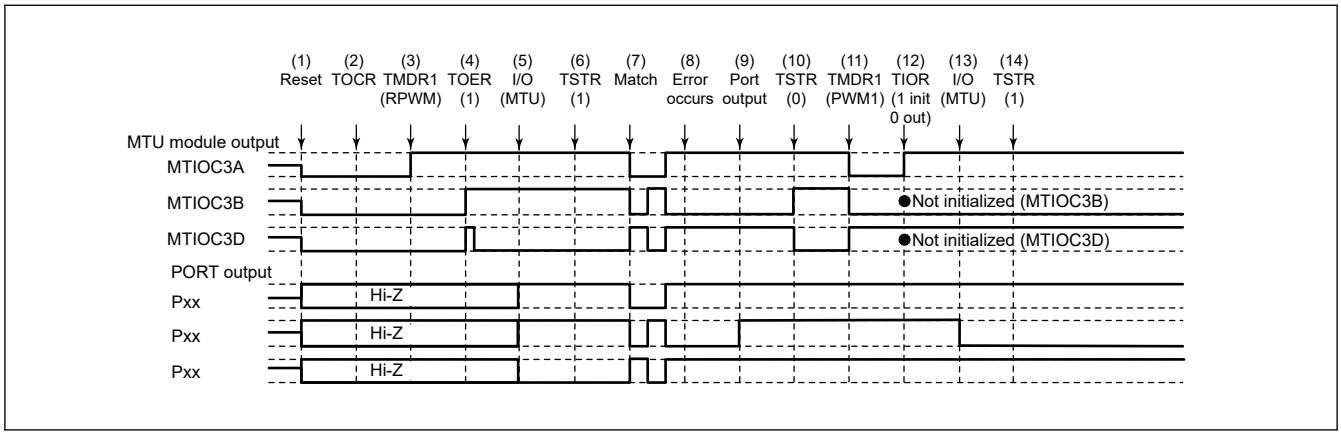


Figure 18.175 Error occurrence in reset-synchronized PWM mode, recovery in PWM mode 1

(1) to (10) are the same as in Figure 18.174.

(11) Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).

(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.28 Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 18.176 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

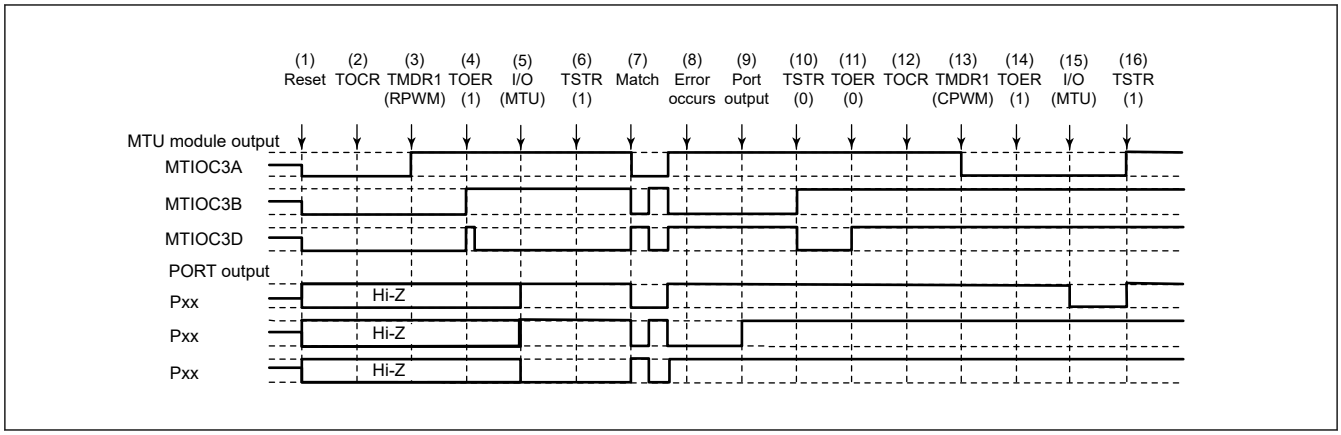


Figure 18.176 Error occurrence in reset-synchronized PWM mode, recovery in complementary PWM mode

(1) to (10) are the same as in Figure 18.174.

(11) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(12) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(13) Set complementary PWM mode (MTU cyclic output pin goes low).

(14) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(15) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(16) Restart operation by setting the TSTRA (TSTRB) register.

18.8.3.29 Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 18.177 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

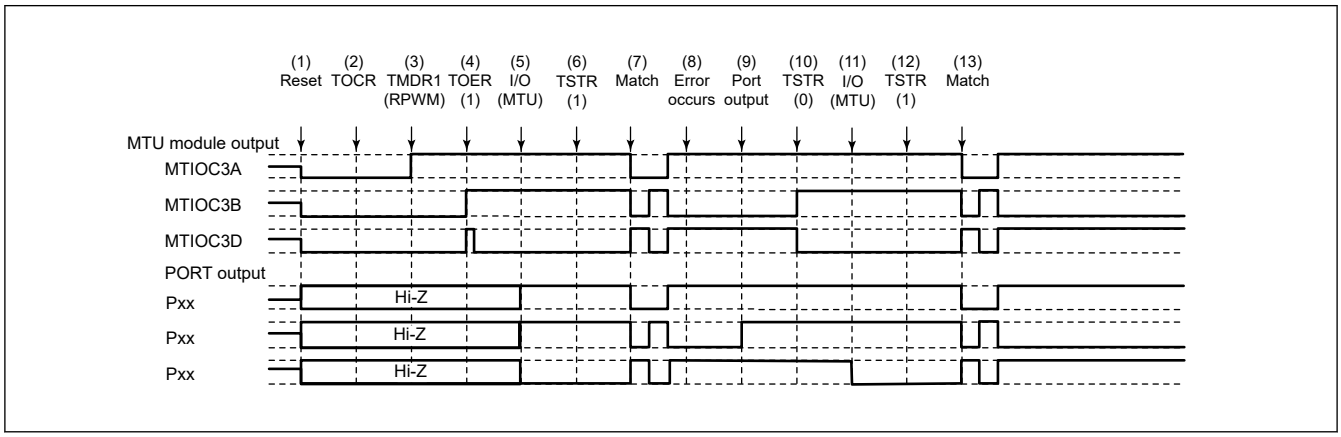


Figure 18.177 Error occurrence in reset-synchronized PWM mode, recovery in reset-synchronized PWM mode

(1) to (10) are the same as in Figure 18.174.

(11) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(12) Restart operation by setting the TSTRA (TSTRB) register.

(13) The reset-synchronized PWM waveform is output on compare match occurrence.

18.9 Operations Linked by the ELC

18.9.1 Event Signal Output to the ELC

The MTU is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).

The MTU outputs the event signal regardless of the setting of the corresponding interrupt request enable bit.

18.9.2 MTU Operations in Response to Receiving Event Signals from the ELC

The MTU can perform the following operations in response to the event set in advance in the ELC_SSEL0 register of the event link controller (ELC).

(1) Start Counting Operation

Select "counting is started" as the operation of the MTU by setting the ELOPA or ELOPB register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3 and the ELOPB register controls the operation of MTU4. When the event specified in the ELC_SSEL0 register occurs, the CSTn bit in the TSTRA register shown in Table 18.77 is set to 1, and the MTU counter starts.

However, when the specified event is generated while the CSTn bit in the TSTRA register has already been set to 1, the event has no effect. Table 18.77 lists the TSTRA register bits used for each channel.

Table 18.77 Counter start bit set by the ELC

Channel no.	Counter start bit
MTU0	TSTRA.CST0 bit
MTU3	TSTRA.CST3 bit
MTU4	TSTRA.CST4 bit

(2) Input Capture Operation

Select "input capture" as the operation of the MTU by setting the ELOPA or ELOPB register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3 and the ELOPB register controls the operation of MTU4. When the event specified in the ELC_SSEL0 register occurs, the value of the TCNT register is captured in the TGR register. When using input capture in response to an event, the corresponding bit of the TIOR register in the MTU should be set for input capture and the CSTn bit of TSTRA register should be set to 1 to start counting by the counter.

In this case, the TIOCnA pin (input capture pin) input has no effect.

Table 18.78 lists the timer general register and I/O control bit used for each channel in input capture operations in response to the ELC.

Table 18.78 Timer general register and I/O control bit used in the Input capture operation

Channel no.	Timer general register	I/O control bit
MTU0	MTU0.TGRA	MTU0.TIORH.IOA[3:0] bits
MTU3	MTU3.TGRA	MTU3.TIORH.IOA[3:0] bits
MTU4	MTU4.TGRA	MTU4.TIORH.IOA[3:0] bits

(3) Restart Counting (Clear Counter) Operation

Select "counting is restarted" as the operation of the MTU by setting the ELOPA or ELOPB register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3 and the ELOPB register controls the operation of MTU4. When the event specified in the ELC_SSEL0 register occurs, the TCNT register is cleared. If the corresponding CSTn bit in the TSTRA register is set to 1, counting continues. For the CSTn bits in the TSTRA register, see Table 18.77.

18.9.3 Usage Notes on MTU Operation by Event Signal Reception from the ELC

The following notes on usage apply when the MTU is used in event link operation.

(1) Start Counting

If the event specified in the ELC_SSEL0 register occurs during a cycle of writing to a CSTn bit in the TSTRA register, writing to the CSTn bit in the TSTRA register does not proceed because setting of the bit to 1 due to the event takes priority.

(2) Restart Counting (Clear Counter)

If the event specified in the ELC_SSEL0 register occurs during a cycle of writing to the TCNT counter, writing to the TCNT counter does not proceed because clearing of the counter due to the event takes priority.

In addition, for MTU3 and MTU4 in complementary PWM mode, do not use the counter restarting by the ELC.

19. Port Output Enable 3 (POE3)

19.1 Overview

The LSI incorporates a port output enable 3 (POE3) module that can be used to place output pins for the MTU3 in the high-impedance state under various conditions. Every output signal is put in the high-impedance state when the output is disabled.

Table 19.1 lists the specifications of the POE3 and Figure 19.1 shows a block diagram of the POE3.

Table 19.1 POE3 specifications

Parameter	Description
Target pins to be placed in the high-impedance state	• MTU3 output pins MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) MTU3 pins (MTIOC3B, MTIOC3D) MTU4 pins (MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) MTU6 pins (MTIOC6B, MTIOC6D) MTU7 pins (MTIOC7A, MTIOC7B, MTIOC7C, MTIOC7D)
Conditions for the high-impedance state	• Setting pins as inputs: setting the POE0#, POE4#, POE8#, POE10#, and POE11# pins as inputs (falling edge or low-level sampling) • Short-circuits between output pins: a match (short circuit) between the output signal levels at the active level over one or more cycle on the following combination of pins: – MTU complementary PWM output pins 1. MTIOC3B and MTIOC3D 2. MTIOC4A and MTIOC4C 3. MTIOC4B and MTIOC4D 4. MTIOC6B and MTIOC6D 5. MTIOC7A and MTIOC7C 6. MTIOC7B and MTIOC7D • SPOER register setting being made • Detection that the clock generation circuit had stopped oscillating or the oscillation is abnormal (CLMA0 to CLMA4: PLL0 to PLL4, CLMA6: main clock) • DSMIF error interrupt
Function	• Each of the POE0#, POE4#, POE8#, POE10#, and POE11# input pins can be set for falling edge, PCLKH/8 × 16, PCLKH/16 × 16, or PCLKH/128 × 16 low-level sampling. • The outputs of the target pins can be in the high-impedance state by falling-edge or low-level sampling of the POE0#, POE4#, POE8#, POE10#, or POE11# pin. • The outputs of the target pins can be in the high-impedance state when oscillation stop is detected by the oscillation stop detection function of the clock generator. • The outputs of the target pins can be in the high-impedance state when DSMIF error occurs. • The MTU complementary PWM outputs can be in the high-impedance state when output levels of the MTU complementary PWM output pins are compared and simultaneous active-level output continues for one cycle or more. • The outputs of the target pins can be in the high-impedance state by modifying the settings of the POE3 registers. • Interrupts can be generated by input-level sampling or output-level comparison results.

The POE3 has input-level detection circuits, pin selection circuits, output-level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in Figure 19.1.

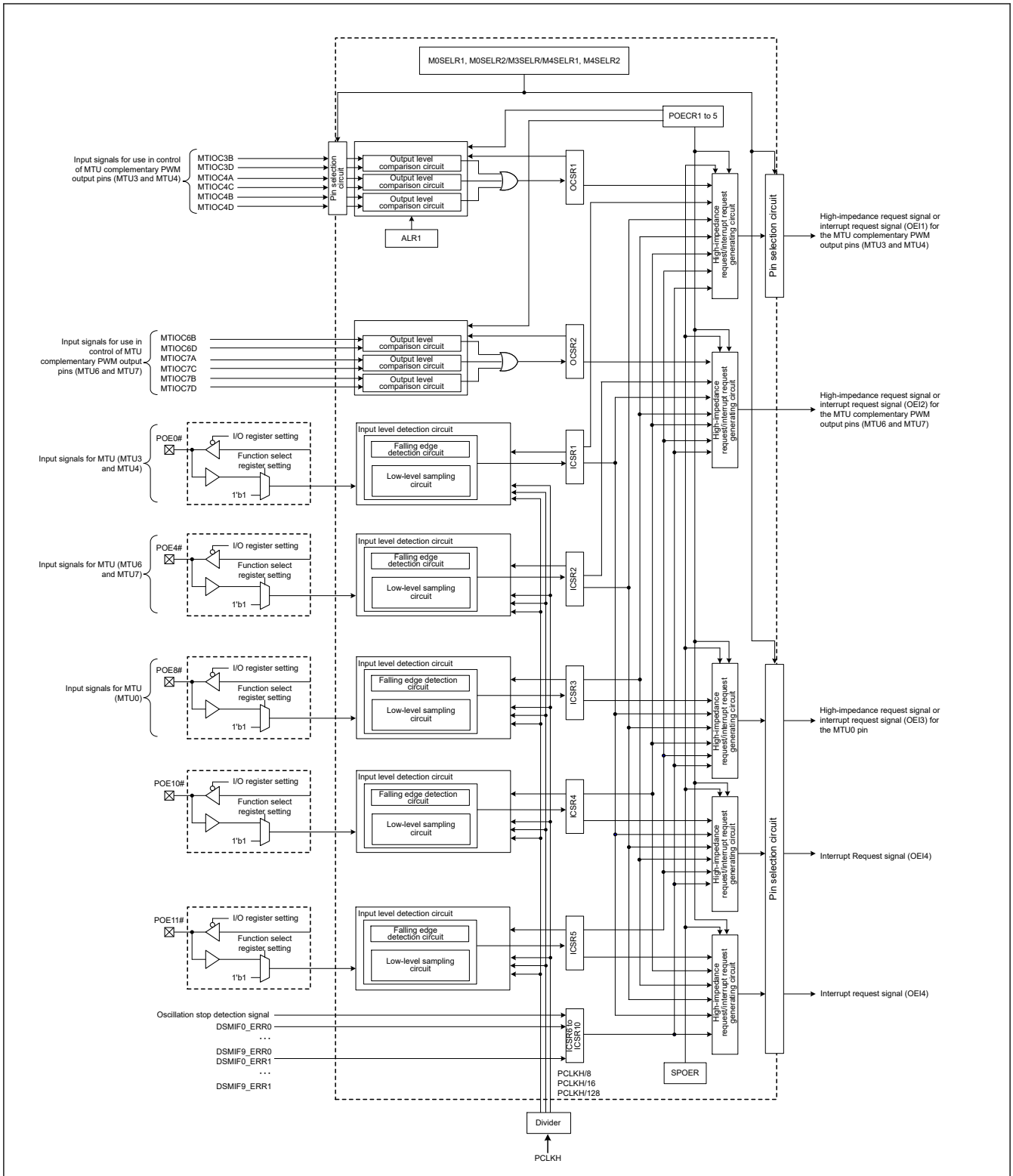


Figure 19.1 POE3 block diagram

Table 19.2 shows I/O pins to be used by the POE3.

Table 19.2 POE3 I/O pins

Pin name	I/O	Description
POE0#	Input	Input pin for the request signal to place the MTU3 and MTU4 pins for MTU complementary PWM output in high-impedance state. In accord with register settings, this pin is also capable of placing the MTU0, MTU6, and MTU7 pins in the high-impedance state.
POE4#	Input	Input pin for the request signal to place the MTU6 and MTU7 pins for MTU complementary PWM output in high-impedance state. In accord with register settings, this pin is also capable of placing the MTU0, MTU3, and MTU4 pins in the high-impedance state.
POE8#	Input	Input pin for the request signal to place the pins for MTU0 in high-impedance state. In accord with register settings, this pin is also capable of placing the MTU3 and MTU4 pins or MTU6 and MTU7 pins for MTU complementary PWM output.
POE10#	Input	In accord with register settings, this pin is capable of placing the any target pins in high-impedance state.
POE11#	Input	In accord with register settings, this pin is capable of placing the any target pins in high-impedance state.

Table 19.3 shows output-level comparisons with pin combinations.

Table 19.3 Pin combinations

Pin combination	I/O	Description
MTIOC3B and MTIOC3D	Output	The MTU3 and MTU4 pins for MTU complementary PWM output are placed in high-impedance state when both pins of a pair simultaneously output the active level*1 for one or more cycles of the peripheral clock (PCLKH). Pin combinations for output comparison and high-impedance control can be selected by registers of POE3.
MTIOC4A and MTIOC4C	Output	
MTIOC4B and MTIOC4D	Output	
MTIOC6B and MTIOC6D	Output	The MTU6 and MTU7 pins for MTU complementary PWM output are placed in high-impedance state when both pins of a pair simultaneously output the active level*2 for one or more cycles of the peripheral clock (PCLKH). Pin combinations for output comparison and high-impedance control can be selected by registers of POE3.
MTIOC7A and MTIOC7C	Output	
MTIOC7B and MTIOC7D	Output	

Note 1. The low level is output when the OLSP bit in TOCR1A of MTUn is 0 with the TOCS bit in TOCR1A of MTUn cleared to 0, or the high level is output when the OLSP bit is 1. Otherwise, the low level is output when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits in TOCR2A of MTUn are 0 with the TOCS bit in TOCR1A of MTUn set to 1, or the high level is output when these bits are 1.

Note 2. The low level is output when the OLSP bit in TOCR1B of MTUn is 0 with the TOCS bit in TOCR1B of MTUn cleared to 0, or the high level is output when the OLSP bit is 1. Otherwise, the low level is output when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits in TOCR2B of MTUn are 0 with the TOCS bit in TOCR1B of MTUn set to 1, or the high level is output when these bits are 1.

19.2 Register Descriptions

The POE3 registers are initialized by a reset.

19.2.1 ICSR1 : Input Level Control/Status Register 1

Base address: POE3 = 0x9001_8000

Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	POE0F	—	—	—	PIE1	—	—	—	—	—	—	POE0M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	POE0M[1:0]	POE0 Mode Select 0 0: Accepts a request on the falling edge of POE0# pin input 0 1: Accepts a request when POE0# pin input has been sampled 16 times at PCLKH/8 clock pulses and all are low level 1 0: Accepts a request when POE0# pin input has been sampled 16 times at PCLKH/16 clock pulses and all are low level 1 1: Accepts a request when POE0# pin input has been sampled 16 times at PCLKH/128 clock pulses and all are low level	R/W*1

Bit	Symbol	Function	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	PIE1	Port Interrupt Enable 1 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	POE0F	POE0 Flag 0: Indicates that a high-impedance request has not been input to the POE0# pin 1: Indicates that a high-impedance request has been input to the POE0# pin	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR1 register selects the input mode for the POE0# pin, controls the enable/disable of interrupts, and indicates status.

POE0M[1:0] bits (POE0 Mode Select)

These bits select the input mode of the POE0# pin.

PIE1 bit (Port Interrupt Enable 1)

This bit enables or disables interrupt requests when the POE0F flag in ICSR1 is set to 1.

POE0F flag (POE0 Flag)

This flag indicates that a high-impedance request has been input to the POE0# pin.

[Setting condition]

- When the input set by the POE0M[1:0] bits occurs at the POE0# pin

[Clearing condition]

- By writing 0 to the POE0F flag after reading POE0F = 1

19.2.2 ICSR2 : Input Level Control/Status Register 2

Base address: POE3 = 0x9001_8000

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	POE4F	—	—	—	PIE2	—	—	—	—	—	—	POE4M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	POE4M[1:0]	POE4 Mode Select 0 0: Accepts a request on the falling edge of POE4# pin input 0 1: Accepts a request when POE4# pin input has been sampled 16 times at PCLKH/8 clock pulses and all are low level 1 0: Accepts a request when POE4# pin input has been sampled 16 times at PCLKH/16 clock pulses and all are low level 1 1: Accepts a request when POE4# pin input has been sampled 16 times at PCLKH/128 clock pulses and all are low level	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	PIE2	Port Interrupt Enable 2 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	POE4F	POE4 Flag 0: Indicates that a high-impedance request has not been input to the POE4# pin 1: Indicates that a high-impedance request has been input to the POE4# pin	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR2 register selects the input mode for the POE4# pin, controls the enable/disable of interrupts, and indicates status.

POE4M[1:0] bits (POE4 Mode Select)

These bits select the input mode of the POE4# pin.

PIE2 bit (Port Interrupt Enable 2)

This bit enables or disables interrupt requests when the POE4F flag in ICSR2 is set to 1.

POE4F flag (POE4 Flag)

This flag indicates that a high-impedance request has been input to the POE4# pin.

[Setting condition]

- When the input set by the POE4M[1:0] bits occurs at the POE4# pin

[Clearing condition]

- By writing 0 to the POE4F flag after reading POE4F = 1

19.2.3 ICSR3 : Input Level Control/Status Register 3

Base address: POE3 = 0x9001_8000

Offset address: 0x08

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	POE8F	—	—	POE8E	PIE3	—	—	—	—	—	—	POE8M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	POE8M[1:0]	POE8 Mode Select 0 0: Accepts a request on the falling edge of POE8# pin input 0 1: Accepts a request when POE8# pin input has been sampled 16 times at PCLKH/8 clock pulses and all are low level 1 0: Accepts a request when POE8# pin input has been sampled 16 times at PCLKH/16 clock pulses and all are low level 1 1: Accepts a request when POE8# pin input has been sampled 16 times at PCLKH/128 clock pulses and all are low level	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	PIE3	Port Interrupt Enable 3 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
9	POE8E	POE8 High-Impedance Enable 0: Does not place the pin in the high-impedance state 1: Places the pin in the high-impedance state	R/W*1
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	POE8F	POE8 Flag 0: Indicates that a high-impedance request has not been input to the POE8# pin. 1: Indicates that a high-impedance request has been input to the POE8# pin.	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR3 register selects the input mode for the POE8# pin, controls the enable/disable of interrupts, and indicates status.

POE8M[1:0] bits (POE8 Mode Select)

These bits select the input mode of the POE8# pin.

PIE3 bit (Port Interrupt Enable 3)

This bit enables or disables interrupt requests when the POE8F flag in ICSR3 is set to 1.

POE8E bit (POE8 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in the high-impedance state when the POE8F flag is set to 1.

POE8F flag (POE8 Flag)

This flag indicates that a high-impedance request has been input to the POE8# pin.

[Setting condition]

- When the input set by the POE8M[1:0] bits occurs at the POE8# pin

[Clearing condition]

- By writing 0 to the POE8F flag after reading POE8F = 1

19.2.4 ICSR4 : Input Level Control/Status Register 4

Base address: POE3 = 0x9001_8000

Offset address: 0x16

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	POE10F	—	—	POE10E	PIE4	—	—	—	—	—	—	POE10M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	POE10M[1:0]	POE10 Mode Select 0 0: Accepts a request on the falling edge of POE10# pin input 0 1: Accepts a request when POE10# pin input has been sampled 16 times at PCLKH/8 clock pulses and all are low level 1 0: Accepts a request when POE10# pin input has been sampled 16 times at PCLKH/16 clock pulses and all are low level 1 1: Accepts a request when POE10# pin input has been sampled 16 times at PCLKH/128 clock pulses and all are low level	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	PIE4	Port Interrupt Enable 4 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
9	POE10E	POE10 High-Impedance Enable 0: Does not place the pin in the high-impedance state 1: Places the pin in the high-impedance state	R/W*1
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	POE10F	POE10 Flag 0: Indicates that a high-impedance request has not been input to the POE10# pin 1: Indicates that a high-impedance request has been input to the POE10# pin	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR4 register selects the input mode for the POE10# pin, controls the enable/disable of interrupts, and indicates status.

POE10M[1:0] bits (POE10 Mode Select)

These bits select the input mode of the POE10# pin.

PIE4 bit (Port Interrupt Enable 4)

This bit enables or disables interrupt requests when the POE10F flag in ICSR4 is set to 1.

POE10E bit (POE10 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in the high-impedance state when the POE10F flag is set to 1.

POE10F flag (POE10 Flag)

This flag indicates that a high-impedance request has been input to the POE10# pin.

[Setting condition]

- When the input set by the POE10M[1:0] bits occurs at the POE10# pin

[Clearing condition]

- By writing 0 to the POE10F flag after reading POE10F = 1

19.2.5 ICSR5 : Input Level Control/Status Register 5

Base address: POE3 = 0x9001_8000

Offset address: 0x18

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	POE11F	—	—	POE11E	PIE5	—	—	—	—	—	—	POE11M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	POE11M[1:0]	POE11 Mode Select 0 0: Accepts a request on the falling edge of POE11# pin input 0 1: Accepts a request when POE11# pin input has been sampled 16 times at PCLKH/8 clock pulses and all are low level 1 0: Accepts a request when POE11# pin input has been sampled 16 times at PCLKH/16 clock pulses and all are low level 1 1: Accepts a request when POE11# pin input has been sampled 16 times at PCLKH/128 clock pulses and all are low level	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	PIE5	Port Interrupt Enable 5 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
9	POE11E	POE11 High-Impedance Enable 0: Does not place the pin in the high-impedance state 1: Places the pin in the high-impedance state	R/W*1
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	POE11F	POE11 Flag 0: Indicates that a high-impedance request has not been input to the POE11# pin 1: Indicates that a high-impedance request has been input to the POE11# pin	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: For the ICSR5 register, set the values the same as those in the ICSR4 register.
Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR5 register selects the pin input mode of the POE11#, controls the enable/disable of interrupts, and indicates status.

POE11M[1:0] bits (POE11 Mode Select)

These bits select the input mode of the POE11# pin.

PIE5 bit (Port Interrupt Enable 5)

This bit enables or disables interrupt requests when the POE11F flag in ICSR5 is set to 1.

POE11E bit (POE11 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in the high-impedance state when the POE11F flag is set to 1. The interrupt output is multiplexed with OEI4.

POE11F flag (POE11 Flag)

This flag indicates that a high-impedance request has been input to the POE11# pin.

[Setting condition]

- When the input set by the POE11M[1:0] bits occurs at the POE11# pin

[Clearing condition]

- By writing 0 to the POE11F flag after reading POE11F = 1

19.2.6 ICSR6 : Input Level Control/Status Register 6

Base address: POE3 = 0x9001_8000

Offset address: 0x1C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OSTSTF	—	—	OSTSTE	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	—	These bits are read as 0. The write value should be 0.	R/W
9	OSTSTE	Oscillation Stop High-Impedance Enable 0: Does not place the MTU complementary PWM output pins and MTU0 pins in high-impedance state 1: Places the MTU complementary PWM output pins and MTU0 pins in high-impedance state	R/W*1
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	OSTSTF	Oscillation Stop High-Impedance Flag 0: Indicates that a high-impedance request or oscillation abnormality has not been generated. 1: Indicates that a high-impedance request by oscillation stop or oscillation abnormality has been generated.	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR6 register controls the high-impedance for stopped oscillation or oscillation abnormality and indicates the status.

OSTSTE bit (Oscillation Stop High-Impedance Enable)

This bit enables or disables the MTU complementary PWM output pins and MTU0 pins to be placed in the high-impedance state when stopped oscillation or oscillation abnormality of the PLL is detected.

OSTSTF flag (Oscillation Stop High-Impedance Flag)

This flag indicates that a stopped oscillation high-impedance request has been generated.

When stopped oscillation is detected, this flag is set to 1. When clearing this flag, write 0 while the detection signal for stopped oscillation or oscillation abnormality of the PLL is negated. Writing 0 to this flag while the detection signal for

stopped oscillation or oscillation abnormality is asserted does not clear this flag to 0. After clearing this flag, confirm that the flag has actually been modified to 0.

[Setting condition]

- When stopped oscillation is detected

[Clearing condition]

- By writing 0 to the OSTSTF flag after reading OSTSTF = 1

19.2.7 ICSR7 : Input Level Control/Status Register 7

Base address: POE3 = 0x9001_8000

Offset address: 0x1E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9ERR0ST	D8ERR0ST	D7ERR0ST	D6ERR0ST	D5ERR0ST	D4ERR0ST	D3ERR0ST	D2ERR0ST	D1ERR0ST	D0ERR0ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0ERR0ST	DSMIF0 Error 0 Status 0: Indicates that DSMIF0 error 0 has not been generated 1: Indicates that DSMIF0 error 0 has been generated	R
1	D1ERR0ST	DSMIF1 Error 0 Status 0: Indicates that DSMIF1 error 0 has not been generated 1: Indicates that DSMIF1 error 0 has been generated	R
2	D2ERR0ST	DSMIF2 Error 0 Status 0: Indicates that DSMIF2 error 0 has not been generated 1: Indicates that DSMIF2 error 0 has been generated	R
3	D3ERR0ST	DSMIF3 Error 0 Status 0: Indicates that DSMIF3 error 0 has not been generated 1: Indicates that DSMIF3 error 0 has been generated	R
4	D4ERR0ST	DSMIF4 Error 0 Status 0: Indicates that DSMIF4 error 0 has not been generated 1: Indicates that DSMIF4 error 0 has been generated	R
5	D5ERR0ST	DSMIF5 Error 0 Status 0: Indicates that DSMIF5 error 0 has not been generated 1: Indicates that DSMIF5 error 0 has been generated	R
6	D6ERR0ST	DSMIF6 Error 0 Status 0: Indicates that DSMIF6 error 0 has not been generated 1: Indicates that DSMIF6 error 0 has been generated	R
7	D7ERR0ST	DSMIF7 Error 0 Status 0: Indicates that DSMIF7 error 0 has not been generated 1: Indicates that DSMIF7 error 0 has been generated	R
8	D8ERR0ST	DSMIF8 Error 0 Status 0: Indicates that DSMIF8 error 0 has not been generated 1: Indicates that DSMIF8 error 0 has been generated	R
9	D9ERR0ST	DSMIF9 Error 0 Status 0: Indicates that DSMIF9 error 0 has not been generated 1: Indicates that DSMIF9 error 0 has been generated	R
15:10	—	These bits are read as 0.	R

The ICSR7 register indicates DSMIF error 0 status.

D0ERR0ST bit (DSMIF0 Error 0 Status)

This bit indicates that a DSMIF0 error 0 interrupt request has been generated.

When DSMIF0 error 0 is detected, this bit is set to 1.

D1ERR0ST bit (DSMIF1 Error 0 Status)

This bit indicates that a DSMIF1 error 0 interrupt request has been generated.

When DSMIF1 error 0 is detected, this bit is set to 1.

D2ERR0ST bit (DSMIF2 Error 0 Status)

This bit indicates that a DSMIF2 error 0 interrupt request has been generated.

When DSMIF2 error 0 is detected, this bit is set to 1.

D3ERR0ST bit (DSMIF3 Error 0 Status)

This bit indicates that a DSMIF3 error 0 interrupt request has been generated.

When DSMIF3 error 0 is detected, this bit is set to 1.

D4ERR0ST bit (DSMIF4 Error 0 Status)

This bit indicates that a DSMIF4 error 0 interrupt request has been generated.

When DSMIF4 error 0 is detected, this bit is set to 1.

D5ERR0ST bit (DSMIF5 Error 0 Status)

This bit indicates that a DSMIF5 error 0 interrupt request has been generated.

When DSMIF5 error 0 is detected, this bit is set to 1.

D6ERR0ST bit (DSMIF6 Error 0 Status)

This bit indicates that a DSMIF6 error 0 interrupt request has been generated.

When DSMIF6 error 0 is detected, this bit is set to 1.

D7ERR0ST bit (DSMIF7 Error 0 Status)

This bit indicates that a DSMIF7 error 0 interrupt request has been generated.

When DSMIF7 error 0 is detected, this bit is set to 1.

D8ERR0ST bit (DSMIF8 Error 0 Status)

This bit indicates that a DSMIF8 error 0 interrupt request has been generated.

When DSMIF8 error 0 is detected, this bit is set to 1.

D9ERR0ST bit (DSMIF9 Error 0 Status)

This bit indicates that a DSMIF9 error 0 interrupt request has been generated.

When DSMIF9 error 0 is detected, this bit is set to 1.

19.2.8 ICSR8 : Input Level Control/Status Register 8

Base address: POE3 = 0x9001_8000

Offset address: 0x30

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9ERR1ST	D8ERR1ST	D7ERR1ST	D6ERR1ST	D5ERR1ST	D4ERR1ST	D3ERR1ST	D2ERR1ST	D1ERR1ST	D0ERR1ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0ERR1ST	DSMIF0 Error 1 Status 0: Indicates that DSMIF0 error 1 has not been generated 1: Indicates that DSMIF0 error 1 has been generated	R

Bit	Symbol	Function	R/W
1	D1ERR1ST	DSMIF1 Error 1 Status 0: Indicates that DSMIF1 error 1 has not been generated 1: Indicates that DSMIF1 error 1 has been generated	R
2	D2ERR1ST	DSMIF2 Error 1 Status 0: Indicates that DSMIF2 error 1 has not been generated 1: Indicates that DSMIF2 error 1 has been generated	R
3	D3ERR1ST	DSMIF3 Error 1 Status 0: Indicates that DSMIF3 error 1 has not been generated 1: Indicates that DSMIF3 error 1 has been generated	R
4	D4ERR1ST	DSMIF4 Error 1 Status 0: Indicates that DSMIF4 error 1 has not been generated 1: Indicates that DSMIF4 error 1 has been generated	R
5	D5ERR1ST	DSMIF5 Error 1 Status 0: Indicates that DSMIF5 error 1 has not been generated 1: Indicates that DSMIF5 error 1 has been generated	R
6	D6ERR1ST	DSMIF6 Error 1 Status 0: Indicates that DSMIF6 error 1 has not been generated 1: Indicates that DSMIF6 error 1 has been generated	R
7	D7ERR1ST	DSMIF7 Error 1 Status 0: Indicates that DSMIF7 error 1 has not been generated 1: Indicates that DSMIF7 error 1 has been generated	R
8	D8ERR1ST	DSMIF8 Error 1 Status 0: Indicates that DSMIF8 error 1 has not been generated 1: Indicates that DSMIF8 error 1 has been generated	R
9	D9ERR1ST	DSMIF9 Error 1 Status 0: Indicates that DSMIF9 error 1 has not been generated 1: Indicates that DSMIF9 error 1 has been generated	R
15:10	—	These bits are read as 0.	R

The ICSR8 register indicates DSMIF error 1 status.

D0ERR1ST bit (DSMIF0 Error 1 Status)

This bit indicates that a DSMIF0 error 1 interrupt request has been generated.

When DSMIF0 error 1 is detected, this bit is set to 1.

D1ERR1ST bit (DSMIF1 Error 1 Status)

This bit indicates that a DSMIF1 error 1 interrupt request has been generated.

When DSMIF1 error 1 is detected, this bit is set to 1.

D2ERR1ST bit (DSMIF2 Error 1 Status)

This bit indicates that a DSMIF2 error 1 interrupt request has been generated.

When DSMIF2 error 1 is detected, this bit is set to 1.

D3ERR1ST bit (DSMIF3 Error 1 Status)

This bit indicates that a DSMIF3 error 1 interrupt request has been generated.

When DSMIF3 error 1 is detected, this bit is set to 1.

D4ERR1ST bit (DSMIF4 Error 1 Status)

This bit indicates that a DSMIF4 error 1 interrupt request has been generated.

When DSMIF4 error 1 is detected, this bit is set to 1.

D5ERR1ST bit (DSMIF5 Error 1 Status)

This bit indicates that a DSMIF5 error 1 interrupt request has been generated.

When DSMIF5 error 1 is detected, this bit is set to 1.

D6ERR1ST bit (DSMIF6 Error 1 Status)

This bit indicates that a DSMIF6 error 1 interrupt request has been generated.

When DSMIF6 error 1 is detected, this bit is set to 1.

D7ERR1ST bit (DSMIF7 Error 1 Status)

This bit indicates that a DSMIF7 error 1 interrupt request has been generated.

When DSMIF7 error 1 is detected, this bit is set to 1.

D8ERR1ST bit (DSMIF8 Error 1 Status)

This bit indicates that a DSMIF8 error 1 interrupt request has been generated.

When DSMIF8 error 1 is detected, this bit is set to 1.

D9ERR1ST bit (DSMIF9 Error 1 Status)

This bit indicates that a DSMIF9 error 1 interrupt request has been generated.

When DSMIF9 error 1 is detected, this bit is set to 1.

19.2.9 ICSR9 : Input Level Control/Status Register 9

Base address: POE3 = 0x9001_8000

Offset address: 0x32

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9ER R0IE	D8ER R0IE	D7ER R0IE	D6ER R0IE	D5ER R0IE	D4ER R0IE	D3ER R0IE	D2ER R0IE	D1ER R0IE	D0ER R0IE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0ERR0IE	DSMIF0 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
1	D1ERR0IE	DSMIF1 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
2	D2ERR0IE	DSMIF2 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
3	D3ERR0IE	DSMIF3 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
4	D4ERR0IE	DSMIF4 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
5	D5ERR0IE	DSMIF5 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
6	D6ERR0IE	DSMIF6 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
7	D7ERR0IE	DSMIF7 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W

Bit	Symbol	Function	R/W
8	D8ERR0IE	DSMIF8 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
9	D9ERR0IE	DSMIF9 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

The ICSR9 register controls the high-impedance for DSMIF error 0.

D0ERR0IE bit (DSMIF0 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF0 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

D1ERR0IE bit (DSMIF1 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF1 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

D2ERR0IE bit (DSMIF2 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF2 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

D3ERR0IE bit (DSMIF3 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF3 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

D4ERR0IE bit (DSMIF4 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF4 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

D5ERR0IE bit (DSMIF5 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF5 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

D6ERR0IE bit (DSMIF6 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF6 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

D7ERR0IE bit (DSMIF7 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF7 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

D8ERR0IE bit (DSMIF8 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF8 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

D9ERR0IE bit (DSMIF9 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF9 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

19.2.10 ICSR10 : Input Level Control/Status Register 10

Base address: POE3 = 0x9001_8000

Offset address: 0x34

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9ERR1IE	D8ERR1IE	D7ERR1IE	D6ERR1IE	D5ERR1IE	D4ERR1IE	D3ERR1IE	D2ERR1IE	D1ERR1IE	D0ERR1IE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0ERR1IE	DSMIF0 Error 1 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
1	D1ERR1IE	DSMIF1 Error 1 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
2	D2ERR1IE	DSMIF2 Error 1 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
3	D3ERR1IE	DSMIF3 Error 1 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
4	D4ERR1IE	DSMIF4 Error 1 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
5	D5ERR1IE	DSMIF5 Error 1 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
6	D6ERR1IE	DSMIF6 Error 1 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
7	D7ERR1IE	DSMIF7 Error 1 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
8	D8ERR1IE	DSMIF8 Error 1 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
9	D9ERR1IE	DSMIF9 Error 1 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

The ICSR10 register controls the high-impedance for DSMIF error 1.

D0ERR1IE bit (DSMIF0 Error 1 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF0 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

D1ERR1IE bit (DSMIF1 Error 1 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF1 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

D2ERR1IE bit (DSMIF2 Error 1 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF2 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

D3ERR1IE bit (DSMIF3 Error 1 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF3 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

D4ERR1IE bit (DSMIF4 Error 1 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF4 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

D5ERR1IE bit (DSMIF5 Error 1 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF5 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

D6ERR1IE bit (DSMIF6 Error 1 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF6 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

D7ERR1IE bit (DSMIF7 Error 1 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF7 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

D8ERR1IE bit (DSMIF8 Error 1 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF8 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

D9ERR1IE bit (DSMIF9 Error 1 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF9 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

19.2.11 OCSR1 : Output Level Control/Status Register 1

Base address: POE3 = 0x9001_8000

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OSF1	—	—	—	—	—	OCE1	OIE1	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	OIE1	Output Short Circuit Interrupt Enable 1 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
9	OCE1	Output Short Circuit High-Impedance Enable 1 0: Does not place the pins in the high-impedance state 1: Places the pins in the high-impedance state	R/W*1
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	OSF1	Output Short Circuit Flag 1 0: Indicates that outputs have not simultaneously become an active level 1: Indicates that outputs have simultaneously become an active level	R/W*2

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The OCSR1 register controls the enable or disable of output-level comparison and interrupts and indicates status.

OIE1 bit (Output Short Circuit Interrupt Enable 1)

This bit enables or disables interrupt requests when the OSF1 flag is set to 1.

OCE1 bit (Output Short Circuit High-Impedance Enable 1)

This bit specifies whether to place the pins in high-impedance state when the OSF1 bit in OCSR1 is set to 1.

OSF1 flag (Output Short Circuit Flag 1)

This flag indicates that any one of the three pairs of 2-phase MTU3 and MTU4 pins for MTU complementary PWM output to be compared has simultaneously become an active level.

[Setting condition]

- When any one of the three pairs of 2-phase outputs has simultaneously become an active level

[Clearing condition]

- By writing 0 to the OSF1 flag after reading OSF1 = 1

19.2.12 OCSR2 : Output Level Control/Status Register 2

Base address: POE3 = 0x9001_8000

Offset address: 0x06

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OSF2	—	—	—	—	—	OCE2	OIE2	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	OIE2	Output Short Circuit Interrupt Enable 2 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
9	OCE2	Output Short Circuit High-Impedance Enable 2 0: Does not place the pins in the high-impedance state 1: Places the pins in the high-impedance state	R/W*1
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	OSF2	Output Short Circuit Flag 2 0: Indicates that outputs have not simultaneously become an active level 1: Indicates that outputs have simultaneously become an active level	R/W*2

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The OCSR2 register controls the enable or disable of output-level comparison and interrupts and indicates status.

OIE2 bit (Output Short Circuit Interrupt Enable 2)

This bit enables or disables interrupt requests when the OSF2 flag in OCSR2 is set to 1.

OCE2 bit (Output Short Circuit High-Impedance Enable 2)

This bit specifies whether to place the pins in the high-impedance state when the OSF2 flag in OCSR2 is set to 1.

OSF2 flag (Output Short Circuit Flag 2)

This flag indicates that any one of the three pairs of 2-phase MTU6 and MTU7 pins for MTU complementary PWM output to be compared has simultaneously become an active level.

[Setting condition]

- When any one of the three pairs of 2-phase outputs has simultaneously become an active level

[Clearing condition]

- By writing 0 to the OSF2 flag after reading OSF2 = 1

19.2.13 ALR1 : Active Level Setting Register 1

Base address: POE3 = 0x9001_8000

Offset address: 0x1A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	OLSEN	—	OLSG2B	OLSG2A	OLSG1B	OLSG1A	OLSG0B	OLSG0A
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OLSG0A	MTIOC3B Pin Active Level Setting 0: Active-low 1: Active-high	R/W*1
1	OLSG0B	MTIOC3D Pin Active Level Setting 0: Active-low 1: Active-high	R/W*1
2	OLSG1A	MTIOC4A Pin Active Level Setting 0: Active-low 1: Active-high	R/W*1
3	OLSG1B	MTIOC4C Pin Active Level Setting 0: Active-low 1: Active-high	R/W*1
4	OLSG2A	MTIOC4B Pin Active Level Setting 0: Active-low 1: Active-high	R/W*1
5	OLSG2B	MTIOC4D Pin Active Level Setting 0: Active low 1: Active high	R/W*1
6	—	This bit is read as 0. The write value should be 0.	R/W
7	OLSEN	Active Level Setting Enable 0: Disabled 1: Enabled	R/W*1
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The ALR1 register specifies the active levels of the MTU outputs selected by the I/O Ports, as the active levels for detection of short circuits of those outputs reflected in OCSR1.

OLSG0A bit (MTIOC3B Pin Active Level Setting)

This bit sets the active level of the MTIOC3B output. Specifically, setting the OLSG0A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG0B bit (MTIOC3D Pin Active Level Setting)

This bit sets the active level of the MTIOC3D output. Specifically, setting the OLSG0B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG1A bit (MTIOC4A Pin Active Level Setting)

This bit sets the active level of the MTIOC4A output. Specifically, setting the OLSG1A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG1B bit (MTIOC4C Pin Active Level Setting)

This bit sets the active level of the MTIOC4C output. Specifically, setting the OLSG1B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG2A bit (MTIOC4B Pin Active Level Setting)

This bit sets the active level of the MTIOC4B output. Specifically, setting the OLSG2A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG2B bit (MTIOC4D Pin Active Level Setting)

This bit sets the active level of the MTIOC4D output. Specifically, setting the OLSG2B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSEN bit (Active Level Setting Enable)

This bit enables or disables the active-level settings in the OLSGnm bits (n = 0 to 2; m = A, B). Clearing the OLSEN bit to 0 disables the OLSGnm bits, in which case the active levels of the MTU output are determined by the MTU.TOCR1A and MTU.TOCR2A registers. Setting the OLSEN bit to 1 enables the OLSGnm bits, in which case the active levels of the MTU output are as selected by the OLSGnm bits in this register.

19.2.14 SPOER : Software Port Output Enable Register

Base address: POE3 = 0x9001_8000

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MTUCH0HIZ	MTUCH67HIZ	MTUCH34HIZ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MTUCH34HIZ	MTU3 and MTU4 Output High-Impedance Enable 0: Does not place the pins in the high-impedance state 1: Places the pins in the high-impedance state	R/W
1	MTUCH67HIZ	MTU6 and MTU7 Output High-Impedance Enable 0: Does not place the pins in the high-impedance state 1: Places the pins in the high-impedance state	R/W
2	MTUCH0HIZ	MTU0 Pin High-Impedance Enable 0: Does not place the pins in the high-impedance state 1: Places the pins in the high-impedance state	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The SPOER register controls the high-impedance state of the pins.

MTUCH34HIZ bit (MTU3 and MTU4 Output High-Impedance Enable)

This bit specifies whether to place the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D) in high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH34HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the MTUCH34HIZ bit after reading MTUCH34HIZ = 1.
  To write 0 to this bit, be sure to read 1 first and then write 0.

MTUCH67HIZ bit (MTU6 and MTU7 Output High-Impedance Enable)

This bit specifies whether to place the MTU complementary PWM output pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D) in the high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH67HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the MTUCH67HIZ bit after reading MTUCH67HIZ = 1.
  To write 0 to this bit, be sure to read 1 first and then write 0.

MTUCH0HIZ bit (MTU0 Pin High-Impedance Enable)

This bit specifies whether to place the MTU0 pins in the high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH0HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the MTUCH0HIZ bit after reading MTUCH0HIZ = 1.
  To write 0 to this bit, be sure to read 1 first and then write 0.

19.2.15 POECR1 : Port Output Enable Control Register 1

Base address: POE3 = 0x9001_8000

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MTU0 DZE	MTU0 CZE	MTU0 BZE	MTU0 AZE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MTU0AZE	MTIOC0A High-Impedance Enable 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state	R/W*1
1	MTU0BZE	MTIOC0B High-Impedance Enable 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state	R/W*1
2	MTU0CZE	MTIOC0C High-Impedance Enable 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state	R/W*1
3	MTU0DZE	MTIOC0D High-Impedance Enable 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state	R/W*1
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR1 register controls high-impedance state of the MTU0 pins.

MTU0AZE bit (MTIOC0A High-Impedance Enable)

This bit specifies whether to place the MTIOC0A output for the MTU0 pin in the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), the ICSRn.POEmF flag

(n = 1, 2, 4, 5; m = 0, 4, 10, 11) specified in the POECR5 register, ICSR7.DmERR0ST (m = 0 to 9) specified in the POECR7 register, or ICSR8.DmERR1ST (m = 0 to 9) specified in the POECR8 register, is set to 1.

MTU0BZE bit (MTIOC0B High-Impedance Enable)

This bit specifies whether to place the MTIOC0B output for the MTU0 pin in the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11) specified in the POECR5 register, ICSR7.DmERR0ST (m = 0 to 9) specified in the POECR7 register, or ICSR8.DmERR1ST (m = 0 to 9) specified in the POECR8 register, is set to 1.

MTU0CZE bit (MTIOC0C High-Impedance Enable)

This bit specifies whether to place the MTIOC0C output for the MTU0 pin in the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11) specified in the POECR5 register, ICSR7.DmERR0ST (m = 0 to 9) specified in the POECR7 register, or ICSR8.DmERR1ST (m = 0 to 9) specified in the POECR8 register, is set to 1.

MTU0DZE bit (MTIOC0D High-Impedance Enable)

This bit specifies whether to place the MTIOC0D output for the MTU0 pin in the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11) specified in the POECR5 register, ICSR7.DmERR0ST (m = 0 to 9) specified in the POECR7 register, or ICSR8.DmERR1ST (m = 0 to 9) specified in the POECR8 register, is set to 1.

19.2.16 POECR2 : Port Output Enable Control Register 2

Base address: POE3 = 0x9001_8000

Offset address: 0x0C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MTU3 BDZE	MTU4 ACZE	MTU4 BDZE	—	—	—	—	—	MTU6 BDZE	MTU7 ACZE	MTU7 BDZE
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	MTU7BDZE	MTIOC7B/MTIOC7D High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
1	MTU7ACZE	MTIOC7A/MTIOC7C High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
2	MTU6BDZE	MTIOC6B/MTIOC6D High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	MTU4BDZE	MTIOC4B/MTIOC4D High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
9	MTU4ACZE	MTIOC4A/MTIOC4C High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
10	MTU3BDZE	MTIOC3B/MTIOC3D High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR2 register controls high-impedance state of the MTU complementary PWM output pins (MTU3, MTU4, MTU6, and MTU7 pins).

MTU7BDZE bit (MTIOC7B/MTIOC7D High-Impedance Enable)

This bit specifies whether to place the MTIOC7B output and MTIOC7D output for the MTU7 pin in the high-impedance state when any one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11) specified in the POECR4 register, ICSR7.DmERR0ST (m = 0 to 9) specified in the POECR11 register, or ICSR8.DmERR1ST (m = 0 to 9) specified in the POECR12 register, is set to 1.

MTU7ACZE bit (MTIOC7A/MTIOC7C High-Impedance Enable)

This bit specifies whether to place the MTIOC7A output and MTIOC7C output for the MTU7 pin in the high-impedance state when any one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11) specified in the POECR4 register, ICSR7.DmERR0ST (m = 0 to 9) specified in the POECR11 register, or ICSR8.DmERR1ST (m = 0 to 9) specified in the POECR12 register, is set to 1.

MTU6BDZE bit (MTIOC6B/MTIOC6D High-Impedance Enable)

This bit specifies whether to place the MTIOC6B output and MTIOC6D output for the MTU6 pin in the high-impedance state when any one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11) specified in the POECR4 register, ICSR7.DmERR0ST (m = 0 to 9) specified in the POECR11 register, or ICSR8.DmERR1ST (m = 0 to 9) specified in the POECR12 register, is set to 1.

MTU4BDZE bit (MTIOC4B/MTIOC4D High-Impedance Enable)

This bit specifies whether to place the MTIOC4B output and MTIOC4D output for the MTU4 pin in the high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11) specified in the POECR4 register, ICSR7.DmERR0ST (m = 0 to 9) specified in the POECR9 register, or ICSR8.DmERR1ST (m = 0 to 9) specified in the POECR10 register, is set to 1.

MTU4ACZE bit (MTIOC4A/MTIOC4C High-Impedance Enable)

This bit specifies whether to place the MTIOC4A output and MTIOC4C output for the MTU4 pin in the high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11) specified in the POECR4 register, ICSR7.DmERR0ST (m = 0 to 9) specified in the POECR9 register, or ICSR8.DmERR1ST (m = 0 to 9) specified in the POECR10 register, is set to 1.

MTU3BDZE bit (MTIOC3B/MTIOC3D High-Impedance Enable)

This bit specifies whether to place the MTIOC3B output and MTIOC3D output for the MTU3 pin in the high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11) specified in the POECR4 register, ICSR7.DmERR0ST (m = 0 to 9) specified in the POECR9 register, or ICSR8.DmERR1ST (m = 0 to 9) specified in the POECR10 register, is set to 1.

19.2.17 POECR4 : Port Output Enable Control Register 4

Base address: POE3 = 0x9001_8000

Offset address: 0x10

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IC5ADDMT67ZE	IC4ADDMT67ZE	IC3ADDMT67ZE	—	IC1ADDMT67ZE	—	—	—	IC5ADDMT34ZE	IC4ADDMT34ZE	IC3ADDMT34ZE	IC2ADDMT34ZE	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
2	IC2ADDMT34ZE	MTU3 and MTU4 High-Impedance POE4F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
3	IC3ADDMT34ZE	MTU3 and MTU4 High-Impedance POE8F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
4	IC4ADDMT34ZE	MTU3 and MTU4 High-Impedance POE10F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
5	IC5ADDMT34ZE	MTU3 and MTU4 High-Impedance POE11F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	IC1ADDMT67ZE	MTU6 and MTU7 High-Impedance POE0F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
10	—	This bit is read as 0. The write value should be 0.	R/W
11	IC3ADDMT67ZE	MTU6 and MTU7 High-Impedance POE8F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
12	IC4ADDMT67ZE	MTU6 and MTU7 High-Impedance POE10F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
13	IC5ADDMT67ZE	MTU6 and MTU7 High-Impedance POE11F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR4 register is used to extend the control conditions of the high-impedance state for the MTU complementary PWM output pins (MTU3, MTU4, MTU6, and MTU7).

For details about the targets and conditions of high-impedance control, see Figure 19.2.

IC2ADDMT34ZE bit (MTU3 and MTU4 High-Impedance POE4F Add)

Adds the ICSR2.POE4F flag (POE4#) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC3ADDMT34ZE bit (MTU3 and MTU4 High-Impedance POE8F Add)

Adds the ICSR3.POE8F flag (POE8#) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC4ADDMT34ZE bit (MTU3 and MTU4 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag (POE10#) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC5ADDMT34ZE bit (MTU3 and MTU4 High-Impedance POE11F Add)

Adds the ICSR5.POE11F flag (POE11#) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance POE0F Add)

Adds the ICSR1.POE0F flag (POE0#) to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC3ADDMT67ZE bit (MTU6 and MTU7 High-Impedance POE8F Add)

Adds the ICSR3.POE8F flag (POE8#) to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC4ADDMT67ZE bit (MTU6 and MTU7 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag (POE10#) to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC5ADDMT67ZE bit (MTU6 and MTU7 High-Impedance POE11F Add)

Adds the ICSR5.POE11F flag (POE11#) to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

19.2.18 POECR5 : Port Output Enable Control Register 5

Base address: POE3 = 0x9001_8000

Offset address: 0x12

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	IC5ADDMT0ZE	IC4ADDMT0ZE	—	IC2ADDMT0ZE	IC1ADDMT0ZE	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	IC1ADDMT0ZE	MTU0 High-Impedance POE0F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
2	IC2ADDMT0ZE	MTU0 High-Impedance POE4F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
3	—	This bit is read as 0. The write value should be 0.	R/W
4	IC4ADDMT0ZE	MTU0 High-Impedance POE10F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
5	IC5ADDMT0ZE	MTU0 High-Impedance POE11F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
15:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR5 register is used to extend the control conditions to place the output of the MTU0 pins in the high-impedance state.

IC1ADDMT0ZE bit (MTU0 High-Impedance POE0F Add)

Adds the ICSR1.POE0F flag (POE0#) to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC2ADDMT0ZE bit (MTU0 High-Impedance POE4F Add)

Adds the ICSR2.POE4F flag (POE4#) to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC4ADDMT0ZE bit (MTU0 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag (POE10#) to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC5ADDMT0ZE bit (MTU0 High-Impedance POE11F Add)

Adds the ICSR5.POE11F flag (POE11#) to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

19.2.19 POECR7 : Port Output Enable Control Register 7

Base address: POE3 = 0x9001_8000

Offset address: 0x36

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9E0A DDMT 0ZE	D8E0A DDMT 0ZE	D7E0A DDMT 0ZE	D6E0A DDMT 0ZE	D5E0A DDMT 0ZE	D4E0A DDMT 0ZE	D3E0A DDMT 0ZE	D2E0A DDMT 0ZE	D1E0A DDMT 0ZE	D0E0A DDMT 0ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0E0ADDMT0ZE	MTU0 High-Impedance D0ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
1	D1E0ADDMT0ZE	MTU0 High-Impedance D1ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
2	D2E0ADDMT0ZE	MTU0 High-Impedance D2ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
3	D3E0ADDMT0ZE	MTU0 High-Impedance D3ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
4	D4E0ADDMT0ZE	MTU0 High-Impedance D4ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
5	D5E0ADDMT0ZE	MTU0 High-Impedance D5ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
6	D6E0ADDMT0ZE	MTU0 High-Impedance D6ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
7	D7E0ADDMT0ZE	MTU0 High-Impedance D7ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
8	D8E0ADDMT0ZE	MTU0 High-Impedance D8ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
9	D9E0ADDMT0ZE	MTU0 High-Impedance D9ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR7 register is used to extend the control conditions to place the output of the MTU0 pins in the high-impedance state.

D0E0ADDMT0ZE bit (MTU0 High-Impedance D0ERR0ST Add)

Adds the ICSR7.D0ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D1E0ADDMT0ZE bit (MTU0 High-Impedance D1ERR0ST Add)

Adds the ICSR7.D1ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D2E0ADDMT0ZE bit (MTU0 High-Impedance D2ERR0ST Add)

Adds the ICSR7.D2ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D3E0ADDMT0ZE bit (MTU0 High-Impedance D3ERR0ST Add)

Adds the ICSR7.D3ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D4E0ADDMT0ZE bit (MTU0 High-Impedance D4ERR0ST Add)

Adds the ICSR7.D4ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D5E0ADDMT0ZE bit (MTU0 High-Impedance D5ERR0ST Add)

Adds the ICSR7.D5ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D6E0ADDMT0ZE bit (MTU0 High-Impedance D6ERR0ST Add)

Adds the ICSR7.D6ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D7E0ADDMT0ZE bit (MTU0 High-Impedance D7ERR0ST Add)

Adds the ICSR7.D7ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D8E0ADDMT0ZE bit (MTU0 High-Impedance D8ERR0ST Add)

Adds the ICSR7.D8ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D9E0ADDMT0ZE bit (MTU0 High-Impedance D9ERR0ST Add)

Adds the ICSR7.D9ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

19.2.20 POECR8 : Port Output Enable Control Register 8

Base address: POE3 = 0x9001_8000

Offset address: 0x38

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9E1ADDMT0ZE	D8E1ADDMT0ZE	D7E1ADDMT0ZE	D6E1ADDMT0ZE	D5E1ADDMT0ZE	D4E1ADDMT0ZE	D3E1ADDMT0ZE	D2E1ADDMT0ZE	D1E1ADDMT0ZE	D0E1ADDMT0ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0E1ADDMT0ZE	MTU0 High-Impedance D0ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
1	D1E1ADDMT0ZE	MTU0 High-Impedance D1ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1

Bit	Symbol	Function	R/W
2	D2E1ADDMT0ZE	MTU0 High-Impedance D2ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
3	D3E1ADDMT0ZE	MTU0 High-Impedance D3ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
4	D4E1ADDMT0ZE	MTU0 High-Impedance D4ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
5	D5E1ADDMT0ZE	MTU0 High-Impedance D5ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
6	D6E1ADDMT0ZE	MTU0 High-Impedance D6ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
7	D7E1ADDMT0ZE	MTU0 High-Impedance D7ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
8	D8E1ADDMT0ZE	MTU0 High-Impedance D8ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
9	D9E1ADDMT0ZE	MTU0 High-Impedance D9ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR8 register is used to extend the control conditions to place the output of the MTU0 pins in the high-impedance state.

D0E1ADDMT0ZE bit (MTU0 High-Impedance D0ERR1ST Add)

Adds the ICSR8.D0ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D1E1ADDMT0ZE bit (MTU0 High-Impedance D1ERR1ST Add)

Adds the ICSR8.D1ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D2E1ADDMT0ZE bit (MTU0 High-Impedance D2ERR1ST Add)

Adds the ICSR8.D2ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D3E1ADDMT0ZE bit (MTU0 High-Impedance D3ERR1ST Add)

Adds the ICSR8.D3ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D4E1ADDMT0ZE bit (MTU0 High-Impedance D4ERR1ST Add)

Adds the ICSR8.D4ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D5E1ADDMT0ZE bit (MTU0 High-Impedance D5ERR1ST Add)

Adds the ICSR8.D5ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D6E1ADDMT0ZE bit (MTU0 High-Impedance D6ERR1ST Add)

Adds the ICSR8.D6ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D7E1ADDMT0ZE bit (MTU0 High-Impedance D7ERR1ST Add)

Adds the ICSR8.D7ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D8E1ADDMT0ZE bit (MTU0 High-Impedance D8ERR1ST Add)

Adds the ICSR8.D8ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D9E1ADDMT0ZE bit (MTU0 High-Impedance D9ERR1ST Add)

Adds the ICSR8.D9ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

19.2.21 POECR9 : Port Output Enable Control Register 9

Base address: POE3 = 0x9001_8000

Offset address: 0x3A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9E0ADDMT34ZE	D8E0ADDMT34ZE	D7E0ADDMT34ZE	D6E0ADDMT34ZE	D5E0ADDMT34ZE	D4E0ADDMT34ZE	D3E0ADDMT34ZE	D2E0ADDMT34ZE	D1E0ADDMT34ZE	D0E0ADDMT34ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D0ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
1	D1E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D1ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
2	D2E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D2ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
3	D3E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D3ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
4	D4E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D4ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
5	D5E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D5ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
6	D6E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D6ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
7	D7E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D7ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
8	D8E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D8ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1

Bit	Symbol	Function	R/W
9	D9E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D9ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR9 register is used to extend the control conditions of the high-impedance state for the MTU complementary PWM output pins (MTU3 and MTU4).

D0E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D0ERR0ST Add)

Adds the ICSR7.D0ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D1E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D1ERR0ST Add)

Adds the ICSR7.D1ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D2E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D2ERR0ST Add)

Adds the ICSR7.D2ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D3E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D3ERR0ST Add)

Adds the ICSR7.D3ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D4E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D4ERR0ST Add)

Adds the ICSR7.D4ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D5E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D5ERR0ST Add)

Adds the ICSR7.D5ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D6E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D6ERR0ST Add)

Adds the ICSR7.D6ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D7E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D7ERR0ST Add)

Adds the ICSR7.D7ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D8E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D8ERR0ST Add)

Adds the ICSR7.D8ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D9E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D9ERR0ST Add)

Adds the ICSR7.D9ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

19.2.22 POECR10 : Port Output Enable Control Register 10

Base address: POE3 = 0x9001_8000

Offset address: 0x3C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9E1ADDMT34ZE	D8E1ADDMT34ZE	D7E1ADDMT34ZE	D6E1ADDMT34ZE	D5E1ADDMT34ZE	D4E1ADDMT34ZE	D3E1ADDMT34ZE	D2E1ADDMT34ZE	D1E1ADDMT34ZE	D0E1ADDMT34ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D0ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
1	D1E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D1ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
2	D2E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D2ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
3	D3E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D3ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
4	D4E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D4ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
5	D5E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D5ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
6	D6E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D6ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
7	D7E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D7ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
8	D8E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D8ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
9	D9E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D9ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR10 register is used to extend the control conditions of the high-impedance state for the MTU complementary PWM output pins (MTU3 and MTU4).

D0E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D0ERR1ST Add)

Adds the ICSR8.D0ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D1E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D1ERR1ST Add)

Adds the ICSR8.D1ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D2E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D2ERR1ST Add)

Adds the ICSR8.D2ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D3E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D3ERR1ST Add)

Adds the ICSR8.D3ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D4E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D4ERR1ST Add)

Adds the ICSR8.D4ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D5E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D5ERR1ST Add)

Adds the ICSR8.D5ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D6E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D6ERR1ST Add)

Adds the ICSR8.D6ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D7E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D7ERR1ST Add)

Adds the ICSR8.D7ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D8E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D8ERR1ST Add)

Adds the ICSR8.D8ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D9E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D9ERR1ST Add)

Adds the ICSR8.D9ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

19.2.23 POECR11 : Port Output Enable Control Register 11

Base address: POE3 = 0x9001_8000

Offset address: 0x3E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9E0ADDMT67ZE	D8E0ADDMT67ZE	D7E0ADDMT67ZE	D6E0ADDMT67ZE	D5E0ADDMT67ZE	D4E0ADDMT67ZE	D3E0ADDMT67ZE	D2E0ADDMT67ZE	D1E0ADDMT67ZE	D0E0ADDMT67ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D0ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
1	D1E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D1ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
2	D2E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D2ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
3	D3E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D3ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1

Bit	Symbol	Function	R/W
4	D4E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D4ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
5	D5E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D5ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
6	D6E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D6ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
7	D7E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D7ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
8	D8E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D8ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
9	D9E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D9ERR0ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR11 register is used to extend the control conditions of the high-impedance state for the MTU complementary PWM output pins (MTU6 and MTU7).

D0E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D0ERR0ST Add)

Adds the ICSR7.D0ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D1E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D1ERR0ST Add)

Adds the ICSR7.D1ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D2E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D2ERR0ST Add)

Adds the ICSR7.D2ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D3E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D3ERR0ST Add)

Adds the ICSR7.D3ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D4E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D4ERR0ST Add)

Adds the ICSR7.D4ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D5E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D5ERR0ST Add)

Adds the ICSR7.D5ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D6E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D6ERR0ST Add)

Adds the ICSR7.D6ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D7E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D7ERR0ST Add)

Adds the ICSR7.D7ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D8E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D8ERR0ST Add)

Adds the ICSR7.D8ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D9E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D9ERR0ST Add)

Adds the ICSR7.D9ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

19.2.24 POECR12 : Port Output Enable Control Register 12

Base address: POE3 = 0x9001_8000

Offset address: 0x40

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9E1ADDMT67ZE	D8E1ADDMT67ZE	D7E1ADDMT67ZE	D6E1ADDMT67ZE	D5E1ADDMT67ZE	D4E1ADDMT67ZE	D3E1ADDMT67ZE	D2E1ADDMT67ZE	D1E1ADDMT67ZE	D0E1ADDMT67ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D0ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
1	D1E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D1ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
2	D2E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D2ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
3	D3E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D3ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
4	D4E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D4ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
5	D5E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D5ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
6	D6E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D6ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
7	D7E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D7ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
8	D8E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D8ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
9	D9E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D9ERR1ST Add 0: Does not add the status to the high-impedance control conditions 1: Adds the status to the high-impedance control conditions	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR12 register is used to extend the control conditions of the high-impedance state for the MTU complementary PWM output pins (MTU6 and MTU7).

D0E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D0ERR1ST Add)

Adds the ICSR8.D0ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D1E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D1ERR1ST Add)

Adds the ICSR8.D1ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D2E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D2ERR1ST Add)

Adds the ICSR8.D2ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D3E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D3ERR1ST Add)

Adds the ICSR8.D3ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D4E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D4ERR1ST Add)

Adds the ICSR8.D4ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D5E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D5ERR1ST Add)

Adds the ICSR8.D5ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D6E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D6ERR1ST Add)

Adds the ICSR8.D6ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D7E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D7ERR1ST Add)

Adds the ICSR8.D7ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D8E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D8ERR1ST Add)

Adds the ICSR8.D8ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D9E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D9ERR1ST Add)

Adds the ICSR8.D9ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

19.2.25 M0SELR1 : MTU0 Pin Select Register 1

Base address: POE3 = 0x9001_8000

Offset address: 0x24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M0BSEL[3:0]				M0ASEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M0ASEL[3:0]	MTU0-A (MTIOC0A) Pin Select 0x0: Controls the high-impedance state of P01_6 when it is in use as the MTIOC0A pin 0x1: Controls the high-impedance state of P04_7 when it is in use as the MTIOC0A pin 0x2: Controls the high-impedance state of P10_5 when it is in use as the MTIOC0A pin Others: Setting prohibited.	R/W*1

Bit	Symbol	Function	R/W
7:4	M0BSEL[3:0]	MTU0-B (MTIOC0B) Pin Select 0x0: Controls the high-impedance state of P01_7 when it is in use as the MTIOC0B pin 0x1: Controls the high-impedance state of P05_0 when it is in use as the MTIOC0B pin 0x2: Controls the high-impedance state of P10_6 when it is in use as the MTIOC0B pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M0SELR1 register is an 8-bit readable/writable register that selects the MTU0-A/B pins as targets for high-impedance control.

M0ASEL[3:0] bits (MTU0-A (MTIOC0A) Pin Select)

These bits select the target MTIOC0A pin for high-impedance control.

M0BSEL[3:0] bits (MTU0-B (MTIOC0B) Pin Select)

These bits select the target MTIOC0B pin for high-impedance control.

19.2.26 M0SELR2 : MTU0 Pin Select Register 2

Base address: POE3 = 0x9001_8000

Offset address: 0x25

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M0DSEL[3:0]				M0CSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M0CSEL[3:0]	MTU0-C (MTIOC0C) Pin Select 0x0: Controls the high-impedance state of P02_0 when it is in use as the MTIOC0C pin 0x1: Controls the high-impedance state of P04_5 when it is in use as the MTIOC0C pin Others: Setting prohibited.	R/W*1
7:4	M0DSEL[3:0]	MTU0-D (MTIOC0D) Pin Select 0x0: Controls the high-impedance state of P02_1 when it is in use as the MTIOC0D pin 0x1: Controls the high-impedance state of P04_6 when it is in use as the MTIOC0D pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M0SELR2 register is an 8-bit readable/writable register that selects the MTU0-C/D pins as targets for high-impedance control.

M0CSEL[3:0] bits (MTU0-C (MTIOC0C) Pin Select)

These bits select the target MTIOC0C pin for high-impedance control.

M0DSEL[3:0] bits (MTU0-D (MTIOC0D) Pin Select)

These bits select the target MTIOC0D pin for high-impedance control.

19.2.27 M3SELR : MTU3 Pin Select Register

Base address: POE3 = 0x9001_8000

Offset address: 0x26

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M3DSEL[3:0]				M3BSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M3BSEL[3:0]	MTU3-B (MTIOC3B) Pin Select 0x0: Controls the high-impedance state of P00_0 when it is in use as the MTIOC3B pin 0x1: Controls the high-impedance state of P02_5 when it is in use as the MTIOC3B pin Others: Setting prohibited.	R/W*1
7:4	M3DSEL[3:0]	MTU3-D (MTIOC3D) Pin Select 0x0: Controls the high-impedance state of P00_1 when it is in use as the MTIOC3D pin 0x1: Controls the high-impedance state of P02_6 when it is in use as the MTIOC3D pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M3SELR register is an 8-bit readable/writable register that selects the MTU3-B/D pins as targets for high-impedance control.

M3BSEL[3:0] bits (MTU3-B (MTIOC3B) Pin Select)

These bits select the target MTIOC3B pin for high-impedance control.

M3DSEL[3:0] bits (MTU3-D (MTIOC3D) Pin Select)

These bits select the target MTIOC3D pin for high-impedance control.

19.2.28 M4SELR1 : MTU4 Pin Select Register 1

Base address: POE3 = 0x9001_8000

Offset address: 0x27

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M4CSEL[3:0]				M4ASEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M4ASEL[3:0]	MTU4-A (MTIOC4A) Pin Select 0x0: Controls the high-impedance state of P00_2 when it is in use as the MTIOC4A pin 0x1: Controls the high-impedance state of P02_7 when it is in use as the MTIOC4A pin Others: Setting prohibited.	R/W*1
7:4	M4CSEL[3:0]	MTU4-C (MTIOC4C) Pin Select 0x0: Controls the high-impedance state of P00_3 when it is in use as the MTIOC4C pin 0x1: Controls the high-impedance state of P03_0 when it is in use as the MTIOC4C pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M4SELR1 register is an 8-bit readable/writable register that selects the MTU4-A/C pins as targets for high-impedance control.

M4ASEL[3:0] bits (MTU4-A (MTIOC4A) Pin Select)

These bits select the target MTIOC4A pin for high-impedance control.

M4CSEL[3:0] bits (MTU4-C (MTIOC4C) Pin Select)

These bits select the target MTIOC4C pin for high-impedance control.

19.2.29 M4SELR2 : MTU4 Pin Select Register 2

Base address: POE3 = 0x9001_8000

Offset address: 0x28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M4DSEL[3:0]				M4BSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M4BSEL[3:0]	MTU4-B (MTIOC4B) Pin Select 0x0: Controls the high-impedance state of P00_4 when it is in use as the MTIOC4B pin 0x1: Controls the high-impedance state of P03_1 when it is in use as the MTIOC4B pin Others: Setting prohibited.	R/W*1
7:4	M4DSEL[3:0]	MTU4-D (MTIOC4D) Pin Select 0x0: Controls the high-impedance state of P00_5 when it is in use as the MTIOC4D pin 0x1: Controls the high-impedance state of P03_2 when it is in use as the MTIOC4D pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M4SELR2 register is an 8-bit readable/writable register that selects the MTU4-B/D pins as targets for high-impedance control.

M4BSEL[3:0] bits (MTU4-B (MTIOC4B) Pin Select)

These bits select the target MTIOC4B pin for high-impedance control.

M4DSEL[3:0] bits (MTU4-D (MTIOC4D) Pin Select)

These bits select the target MTIOC4D pin for high-impedance control.

19.2.30 M6SELR : MTU6 Pin Select Register

Base address: POE3 = 0x9001_8000

Offset address: 0x2A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M6DSEL[3:0]				M6BSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M6BSEL[3:0]	MTU6-B (MTIOC6B) Pin Select 0x0: Controls the high-impedance state of P01_2 when it is in use as the MTIOC6B pin 0x1: Controls the high-impedance state of P03_7 when it is in use as the MTIOC6B pin 0x2: Controls the high-impedance state of P09_4 when it is in use as the MTIOC6B pin Others: Setting prohibited.	R/W*1
7:4	M6DSEL[3:0]	MTU6-D (MTIOC6D) Pin Select 0x0: Controls the high-impedance state of P01_3 when it is in use as the MTIOC6D pin 0x1: Controls the high-impedance state of P04_0 when it is in use as the MTIOC6D pin 0x2: Controls the high-impedance state of P09_5 when it is in use as the MTIOC6D pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M6SELR register is an 8-bit readable/writable register that selects the MTU6-B/D pins as targets for high-impedance control.

M6BSEL[3:0] bits (MTU6-B (MTIOC6B) Pin Select)

These bits select the target MTIOC6B pin for high-impedance control.

M6DSEL[3:0] bits (MTU6-D (MTIOC6D) Pin Select)

These bits select the target MTIOC6D pin for high-impedance control.

19.2.31 M7SELR1 : MTU7 Pin Select Register 1

Base address: POE3 = 0x9001_8000

Offset address: 0x2B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M7CSEL[3:0]				M7ASEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M7ASEL[3:0]	MTU7-A (MTIOC7A) Pin Select 0x0: Controls the high-impedance state of P01_4 when it is in use as the MTIOC7A pin 0x1: Controls the high-impedance state of P04_1 when it is in use as the MTIOC7A pin 0x2: Controls the high-impedance state of P09_6 when it is in use as the MTIOC7A pin Others: Setting prohibited.	R/W*1
7:4	M7CSEL[3:0]	MTU7-C (MTIOC7C) Pin Select 0x0: Controls the high-impedance state of P01_5 when it is in use as the MTIOC7C pin 0x1: Controls the high-impedance state of P04_2 when it is in use as the MTIOC7C pin 0x2: Controls the high-impedance state of P09_7 when it is in use as the MTIOC7C pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M7SELR1 register is an 8-bit readable/writable register that selects the MTU7-A/C pins as targets for high-impedance control.

M7ASEL[3:0] bits (MTU7-A (MTIOC7A) Pin Select)

The M7ASEL[3:0] bits select the target MTIOC7A pin for high-impedance control.

M7CSEL[3:0] bits (MTU7-C (MTIOC7C) Pin Select)

The M7CSEL[3:0] bits select the target MTIOC7C pin for high-impedance control.

19.2.32 M7SELR2 : MTU7 Pin Select Register 2

Base address: POE3 = 0x9001_8000

Offset address: 0x2C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M7DSEL[3:0]				M7BSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M7BSEL[3:0]	MTU7-B (MTIOC7B) Pin Select 0x0: Controls the high-impedance state of P01_6 when it is in use as the MTIOC7B pin 0x1: Controls the high-impedance state of P04_3 when it is in use as the MTIOC7B pin 0x2: Controls the high-impedance state of P10_0 when it is in use as the MTIOC7B pin Others: Setting prohibited.	R/W*1

Bit	Symbol	Function	R/W
7:4	M7DSEL[3:0]	MTU7-D (MTIOC7D) Pin Select 0x0: Controls the high-impedance state of P01_7 when it is in use as the MTIOC7D pin 0x1: Controls the high-impedance state of P04_4 when it is in use as the MTIOC7D pin 0x2: Controls the high-impedance state of P10_1 when it is in use as the MTIOC7D pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M7SELR2 register is an 8-bit readable/writable register that selects the MTU7-B/D pins as targets for high-impedance control.

M7BSEL[3:0] bits (MTU7-B (MTIOC7B) Pin Select)

The M7BSEL[3:0] bits select the target MTIOC7B pin for high-impedance control.

M7DSEL[3:0] bits (MTU7-D (MTIOC7D) Pin Select)

The M7DSEL[3:0] bits select the target MTIOC7D pin for high-impedance control.

19.3 Operation

Table 19.4 shows the target pins for high-impedance control and conditions to place the pins in high-impedance state.

Table 19.4 Target pins and conditions for high-impedance control (1 of 6)

Pins	Conditions	Detailed conditions
MTU3 pins (MTIOC3B and MTIOC3D)	• Operation for detection of the POE0# input level • Operation for comparison of the output levels on the MTIOC3B and MTIOC3D pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt specified in the POECR9 and POECR10	MTU3BDZE • ((POE0F) + (OSF1 • OCE1) + (MTUCH34HIZ) + (IC2ADDMT34ZE • POE4F) + (IC3ADDMT34ZE • POE8E • POE8F) + (IC4ADDMT34ZE • POE10E • POE10F) + (IC5ADDMT34ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E0ADDMT34ZE • D0ERR0ST) + (D0E1ADDMT34ZE • D0ERR1ST) + (D1E0ADDMT34ZE • D1ERR0ST) + (D1E1ADDMT34ZE • D1ERR1ST) + (D2E0ADDMT34ZE • D2ERR0ST) + (D2E1ADDMT34ZE • D2ERR1ST) + (D3E0ADDMT34ZE • D3ERR0ST) + (D3E1ADDMT34ZE • D3ERR1ST) + (D4E0ADDMT34ZE • D4ERR0ST) + (D4E1ADDMT34ZE • D4ERR1ST) + (D5E0ADDMT34ZE • D5ERR0ST) + (D5E1ADDMT34ZE • D5ERR1ST) + (D6E0ADDMT34ZE • D6ERR0ST) + (D6E1ADDMT34ZE • D6ERR1ST) + (D7E0ADDMT34ZE • D7ERR0ST) + (D7E1ADDMT34ZE • D7ERR1ST) + (D8E0ADDMT34ZE • D8ERR0ST) + (D8E1ADDMT34ZE • D8ERR1ST) + (D9E0ADDMT34ZE • D9ERR0ST) + (D9E1ADDMT34ZE • D9ERR1ST))

Table 19.4 Target pins and conditions for high-impedance control (2 of 6)

Pins	Conditions	Detailed conditions
MTU4 pins (MTIOC4A and MTIOC4C)	• Operation for detection of the POE0# input level • Operation for comparison of the output levels on the MTIOC4A and MTIOC4C pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt specified in the POECR9 and POECR10	MTU4ACZE • ((POE0F) + (OSF1 • OCE1) + (MTUCH34HIZ) + (IC2ADDMT34ZE • POE4F) + (IC3ADDMT34ZE • POE8E • POE8F) + (IC4ADDMT34ZE • POE10E • POE10F) + (IC5ADDMT34ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E0ADDMT34ZE • D0ERR0ST) + (D0E1ADDMT34ZE • D0ERR1ST) + (D1E0ADDMT34ZE • D1ERR0ST) + (D1E1ADDMT34ZE • D1ERR1ST) + (D2E0ADDMT34ZE • D2ERR0ST) + (D2E1ADDMT34ZE • D2ERR1ST) + (D3E0ADDMT34ZE • D3ERR0ST) + (D3E1ADDMT34ZE • D3ERR1ST) + (D4E0ADDMT34ZE • D4ERR0ST) + (D4E1ADDMT34ZE • D4ERR1ST) + (D5E0ADDMT34ZE • D5ERR0ST) + (D5E1ADDMT34ZE • D5ERR1ST) + (D6E0ADDMT34ZE • D6ERR0ST) + (D6E1ADDMT34ZE • D6ERR1ST) + (D7E0ADDMT34ZE • D7ERR0ST) + (D7E1ADDMT34ZE • D7ERR1ST) + (D8E0ADDMT34ZE • D8ERR0ST) + (D8E1ADDMT34ZE • D8ERR1ST) + (D9E0ADDMT34ZE • D9ERR0ST) + (D9E1ADDMT34ZE • D9ERR1ST))
MTU4 pins (MTIOC4B and MTIOC4D)	• Operation for detection of the POE0# input level • Operation for comparison of the output levels on the MTIOC4B and MTIOC4D pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt specified in the POECR9 and POECR10	MTU4BDZE • ((POE0F) + (OSF1 • OCE1) + (MTUCH34HIZ) + (IC2ADDMT34ZE • POE4F) + (IC3ADDMT34ZE • POE8E • POE8F) + (IC4ADDMT34ZE • POE10E • POE10F) + (IC5ADDMT34ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E0ADDMT34ZE • D0ERR0ST) + (D0E1ADDMT34ZE • D0ERR1ST) + (D1E0ADDMT34ZE • D1ERR0ST) + (D1E1ADDMT34ZE • D1ERR1ST) + (D2E0ADDMT34ZE • D2ERR0ST) + (D2E1ADDMT34ZE • D2ERR1ST) + (D3E0ADDMT34ZE • D3ERR0ST) + (D3E1ADDMT34ZE • D3ERR1ST) + (D4E0ADDMT34ZE • D4ERR0ST) + (D4E1ADDMT34ZE • D4ERR1ST) + (D5E0ADDMT34ZE • D5ERR0ST) + (D5E1ADDMT34ZE • D5ERR1ST) + (D6E0ADDMT34ZE • D6ERR0ST) + (D6E1ADDMT34ZE • D6ERR1ST) + (D7E0ADDMT34ZE • D7ERR0ST) + (D7E1ADDMT34ZE • D7ERR1ST) + (D8E0ADDMT34ZE • D8ERR0ST) + (D8E1ADDMT34ZE • D8ERR1ST) + (D9E0ADDMT34ZE • D9ERR0ST) + (D9E1ADDMT34ZE • D9ERR1ST))

Table 19.4 Target pins and conditions for high-impedance control (3 of 6)

Pins	Conditions	Detailed conditions
MTU6 pins (MTIOC6B and MTIOC6D)	• Operation for detection of the POE4# input level • Operation for comparison of the output levels on the MTIOC6B and MTIOC6D pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt specified in the POECR11 and POECR12	MTU6BDZE • ((POE4F) + (OSF2 • OCE2) + (MTUCH67HIZ) + (IC1ADDMT67ZE • POE0F) + (IC3ADDMT67ZE • POE8E • POE8F) + (IC4ADDMT67ZE • POE10E • POE10F) + (IC5ADDMT67ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E0ADDMT67ZE • D0ERR0ST) + (D0E1ADDMT67ZE • D0ERR1ST) + (D1E0ADDMT67ZE • D1ERR0ST) + (D1E1ADDMT67ZE • D1ERR1ST) + (D2E0ADDMT67ZE • D2ERR0ST) + (D2E1ADDMT67ZE • D2ERR1ST) + (D3E0ADDMT67ZE • D3ERR0ST) + (D3E1ADDMT67ZE • D3ERR1ST) + (D4E0ADDMT67ZE • D4ERR0ST) + (D4E1ADDMT67ZE • D4ERR1ST) + (D5E0ADDMT67ZE • D5ERR0ST) + (D5E1ADDMT67ZE • D5ERR1ST) + (D6E0ADDMT67ZE • D6ERR0ST) + (D6E1ADDMT67ZE • D6ERR1ST) + (D7E0ADDMT67ZE • D7ERR0ST) + (D7E1ADDMT67ZE • D7ERR1ST) + (D8E0ADDMT67ZE • D8ERR0ST) + (D8E1ADDMT67ZE • D8ERR1ST) + (D9E0ADDMT67ZE • D9ERR0ST) + (D9E1ADDMT67ZE • D9ERR1ST))
MTU7 pins (MTIOC7A and MTIOC7C)	• Operation for detection of the POE4# input level • Operation for comparison of the output levels on the MTIOC7A and MTIOC7C pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt specified in the POECR11 and POECR12	MTU7ACZE • ((POE4F) + (OSF2 • OCE2) + (MTUCH67HIZ) + (IC1ADDMT67ZE • POE0F) + (IC3ADDMT67ZE • POE8E • POE8F) + (IC4ADDMT67ZE • POE10E • POE10F) + (IC5ADDMT67ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E0ADDMT67ZE • D0ERR0ST) + (D0E1ADDMT67ZE • D0ERR1ST) + (D1E0ADDMT67ZE • D1ERR0ST) + (D1E1ADDMT67ZE • D1ERR1ST) + (D2E0ADDMT67ZE • D2ERR0ST) + (D2E1ADDMT67ZE • D2ERR1ST) + (D3E0ADDMT67ZE • D3ERR0ST) + (D3E1ADDMT67ZE • D3ERR1ST) + (D4E0ADDMT67ZE • D4ERR0ST) + (D4E1ADDMT67ZE • D4ERR1ST) + (D5E0ADDMT67ZE • D5ERR0ST) + (D5E1ADDMT67ZE • D5ERR1ST) + (D6E0ADDMT67ZE • D6ERR0ST) + (D6E1ADDMT67ZE • D6ERR1ST) + (D7E0ADDMT67ZE • D7ERR0ST) + (D7E1ADDMT67ZE • D7ERR1ST) + (D8E0ADDMT67ZE • D8ERR0ST) + (D8E1ADDMT67ZE • D8ERR1ST) + (D9E0ADDMT67ZE • D9ERR0ST) + (D9E1ADDMT67ZE • D9ERR1ST))

Table 19.4 Target pins and conditions for high-impedance control (4 of 6)

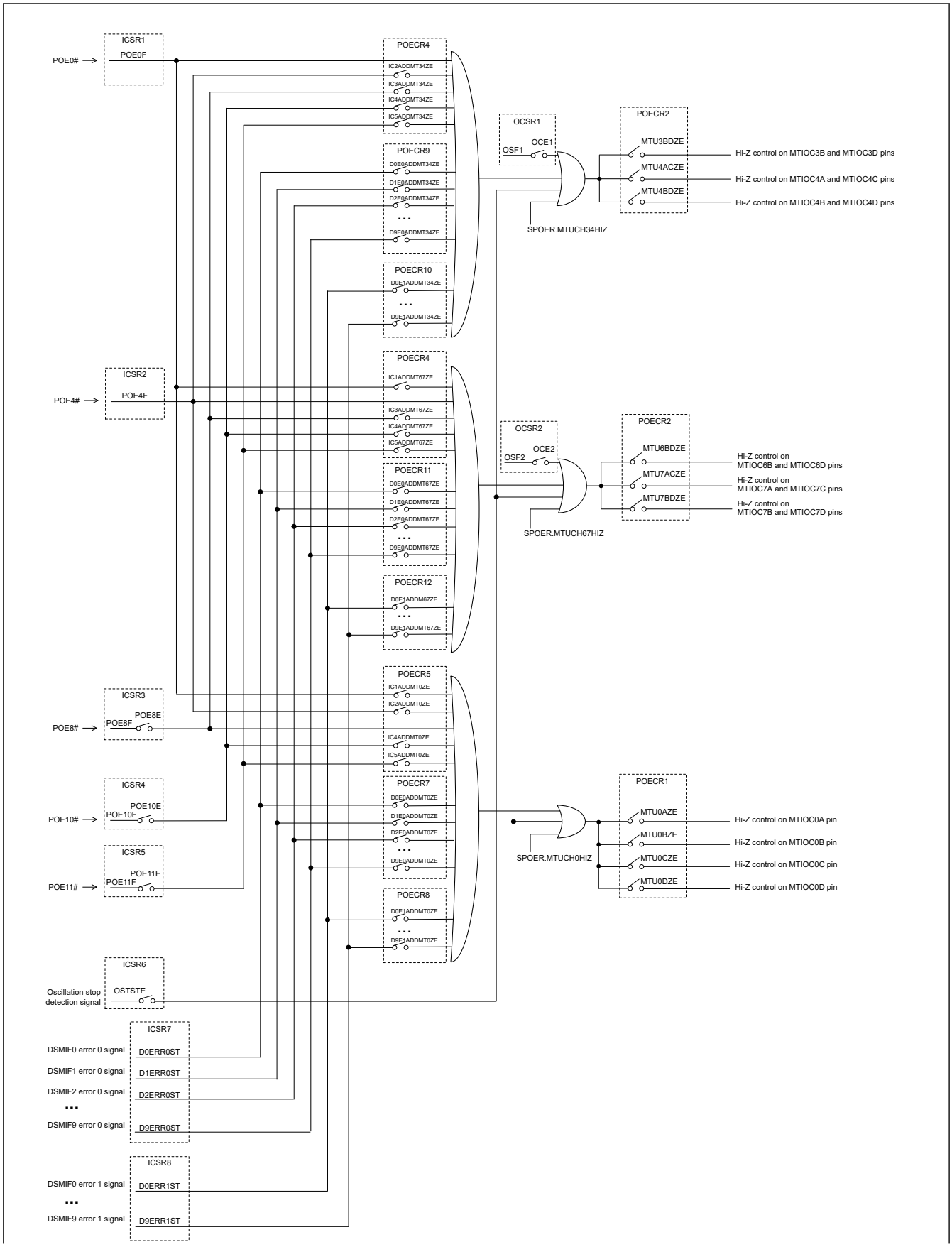
Pins	Conditions	Detailed conditions
MTU7 pins (MTIOC7B and MTIOC7D)	• Operation for detection of the POE4# input level • Operation for comparison of the output levels on the MTIOC7B and MTIOC7D pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt specified in the POECR11 and POECR12	MTU7BDZE • ((POE4F) + (OSF2 • OCE2) + (MTUCH67HIZ) + (IC1ADDMT67ZE • POE4F) + (IC3ADDMT67ZE • POE8E • POE8F) + (IC4ADDMT67ZE • POE10E • POE10F) + (IC5ADDMT67ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E0ADDMT67ZE • D0ERR0ST) + (D0E1ADDMT67ZE • D0ERR1ST) + (D1E0ADDMT67ZE • D1ERR0ST) + (D1E1ADDMT67ZE • D1ERR1ST) + (D2E0ADDMT67ZE • D2ERR0ST) + (D2E1ADDMT67ZE • D2ERR1ST) + (D3E0ADDMT67ZE • D3ERR0ST) + (D3E1ADDMT67ZE • D3ERR1ST) + (D4E0ADDMT67ZE • D4ERR0ST) + (D4E1ADDMT67ZE • D4ERR1ST) + (D5E0ADDMT67ZE • D5ERR0ST) + (D5E1ADDMT67ZE • D5ERR1ST) + (D6E0ADDMT67ZE • D6ERR0ST) + (D6E1ADDMT67ZE • D6ERR1ST) + (D7E0ADDMT67ZE • D7ERR0ST) + (D7E1ADDMT67ZE • D7ERR1ST) + (D8E0ADDMT67ZE • D8ERR0ST) + (D8E1ADDMT67ZE • D8ERR1ST) + (D9E0ADDMT67ZE • D9ERR0ST) + (D9E1ADDMT67ZE • D9ERR1ST))
MTU0 pin (MTIOC0A)	• Operation for detection of the POE8# input level • SPOER setting • Additional conditions of the POECR5 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt specified in the POECR7 and POECR8	MTU0AZE • ((POE8F • POE8E) + (MTUCH0HIZ) + (IC1ADDMT0ZE • POE0F) + (IC2ADDMT0ZE • POE4F) + (IC4ADDMT0ZE • POE10E • POE10F) + (IC5ADDMT0ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E0ADDMT0ZE • D0ERR0ST) + (D0E1ADDMT0ZE • D0ERR1ST) + (D1E0ADDMT0ZE • D1ERR0ST) + (D1E1ADDMT0ZE • D1ERR1ST) + (D2E0ADDMT0ZE • D2ERR0ST) + (D2E1ADDMT0ZE • D2ERR1ST) + (D3E0ADDMT0ZE • D3ERR0ST) + (D3E1ADDMT0ZE • D3ERR1ST) + (D4E0ADDMT0ZE • D4ERR0ST) + (D4E1ADDMT0ZE • D4ERR1ST) + (D5E0ADDMT0ZE • D5ERR0ST) + (D5E1ADDMT0ZE • D5ERR1ST) + (D6E0ADDMT0ZE • D6ERR0ST) + (D6E1ADDMT0ZE • D6ERR1ST) + (D7E0ADDMT0ZE • D7ERR0ST) + (D7E1ADDMT0ZE • D7ERR1ST) + (D8E0ADDMT0ZE • D8ERR0ST) + (D8E1ADDMT0ZE • D8ERR1ST) + (D9E0ADDMT0ZE • D9ERR0ST) + (D9E1ADDMT0ZE • D9ERR1ST))

Table 19.4 Target pins and conditions for high-impedance control (5 of 6)

Pins	Conditions	Detailed conditions
MTU0 pin (MTIOC0B)	• Operation for detection of the POE8# input level • SPOER setting • Additional conditions of the POECR5 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt specified in the POECR7 and POECR8	MTU0BZE • ((POE8F • POE8E) + (MTUCH0HIZ) + (IC1ADDMT0ZE • POE0F) + (IC2ADDMT0ZE • POE4F) + (IC4ADDMT0ZE • POE10E • POE10F) + (IC5ADDMT0ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E0ADDMT0ZE • D0ERR0ST) + (D0E1ADDMT0ZE • D0ERR1ST) + (D1E0ADDMT0ZE • D1ERR0ST) + (D1E1ADDMT0ZE • D1ERR1ST) + (D2E0ADDMT0ZE • D2ERR0ST) + (D2E1ADDMT0ZE • D2ERR1ST) + (D3E0ADDMT0ZE • D3ERR0ST) + (D3E1ADDMT0ZE • D3ERR1ST) + (D4E0ADDMT0ZE • D4ERR0ST) + (D4E1ADDMT0ZE • D4ERR1ST) + (D5E0ADDMT0ZE • D5ERR0ST) + (D5E1ADDMT0ZE • D5ERR1ST) + (D6E0ADDMT0ZE • D6ERR0ST) + (D6E1ADDMT0ZE • D6ERR1ST) + (D7E0ADDMT0ZE • D7ERR0ST) + (D7E1ADDMT0ZE • D7ERR1ST) + (D8E0ADDMT0ZE • D8ERR0ST) + (D8E1ADDMT0ZE • D8ERR1ST) + (D9E0ADDMT0ZE • D9ERR0ST) + (D9E1ADDMT0ZE • D9ERR1ST))
MTU0 pin (MTIOC0C)	• Operation for detection of the POE8# input level • SPOER setting • Additional conditions of the POECR5 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt specified in the POECR7 and POECR8	MTU0CZE • ((POE8F • POE8E) + (MTUCH0HIZ) + (IC1ADDMT0ZE • POE0F) + (IC2ADDMT0ZE • POE4F) + (IC4ADDMT0ZE • POE10E • POE10F) + (IC5ADDMT0ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E0ADDMT0ZE • D0ERR0ST) + (D0E1ADDMT0ZE • D0ERR1ST) + (D1E0ADDMT0ZE • D1ERR0ST) + (D1E1ADDMT0ZE • D1ERR1ST) + (D2E0ADDMT0ZE • D2ERR0ST) + (D2E1ADDMT0ZE • D2ERR1ST) + (D3E0ADDMT0ZE • D3ERR0ST) + (D3E1ADDMT0ZE • D3ERR1ST) + (D4E0ADDMT0ZE • D4ERR0ST) + (D4E1ADDMT0ZE • D4ERR1ST) + (D5E0ADDMT0ZE • D5ERR0ST) + (D5E1ADDMT0ZE • D5ERR1ST) + (D6E0ADDMT0ZE • D6ERR0ST) + (D6E1ADDMT0ZE • D6ERR1ST) + (D7E0ADDMT0ZE • D7ERR0ST) + (D7E1ADDMT0ZE • D7ERR1ST) + (D8E0ADDMT0ZE • D8ERR0ST) + (D8E1ADDMT0ZE • D8ERR1ST) + (D9E0ADDMT0ZE • D9ERR0ST) + (D9E1ADDMT0ZE • D9ERR1ST))

Table 19.4 Target pins and conditions for high-impedance control (6 of 6)

Pins	Conditions	Detailed conditions
MTU0 pin (MTIOC0D)	• Operation for detection of the POE8# input level • SPOER setting • Additional conditions of the POECR5 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt specified in the POECR7 and POECR8	MTU0DZE • ((POE8F • POE8E) + (MTUCH0HIZ) + (IC1ADDMT0ZE • POE0F) + (IC2ADDMT0ZE • POE4F) + (IC4ADDMT0ZE • POE10E • POE10F) + (IC5ADDMT0ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E0ADDMT0ZE • D0ERR0ST) + (D0E1ADDMT0ZE • D0ERR1ST) + (D1E0ADDMT0ZE • D1ERR0ST) + (D1E1ADDMT0ZE • D1ERR1ST) + (D2E0ADDMT0ZE • D2ERR0ST) + (D2E1ADDMT0ZE • D2ERR1ST) + (D3E0ADDMT0ZE • D3ERR0ST) + (D3E1ADDMT0ZE • D3ERR1ST) + (D4E0ADDMT0ZE • D4ERR0ST) + (D4E1ADDMT0ZE • D4ERR1ST) + (D5E0ADDMT0ZE • D5ERR0ST) + (D5E1ADDMT0ZE • D5ERR1ST) + (D6E0ADDMT0ZE • D6ERR0ST) + (D6E1ADDMT0ZE • D6ERR1ST) + (D7E0ADDMT0ZE • D7ERR0ST) + (D7E1ADDMT0ZE • D7ERR1ST) + (D8E0ADDMT0ZE • D8ERR0ST) + (D8E1ADDMT0ZE • D8ERR1ST) + (D9E0ADDMT0ZE • D9ERR0ST) + (D9E1ADDMT0ZE • D9ERR1ST))


ICSR1
POE0F
POE0#
POECR4
IC2ADDMT34ZE
IC3ADDMT34ZE
IC4ADDMT34ZE
IC5ADDMT34ZE
POECR9
D0E0ADDMT34ZE
D1E0ADDMT34ZE
D2E0ADDMT34ZE
D9E0ADDMT34ZE
POECR10
D0E1ADDMT34ZE
D9E1ADDMT34ZE
OCSR1
OCE1
OSF1
SPOER.MTUCH34HIZ
POECR2
MTU3BDZE
MTU4ACZE
MTU4BDZE
Hi-Z control on MTIOC3B and MTIOC3D pins
Hi-Z control on MTIOC4A and MTIOC4C pins
Hi-Z control on MTIOC4B and MTIOC4D pins
ICSR2
POE4F
POE4#
POECR4
IC1ADDMT67ZE
IC3ADDMT67ZE
IC4ADDMT67ZE
IC5ADDMT67ZE
POECR11
D0E0ADDMT67ZE
D1E0ADDMT67ZE
D2E0ADDMT67ZE
D9E0ADDMT67ZE
POECR12
D0E1ADDM67ZE
D9E1ADDMT67ZE
OCSR2
OCE2
OSF2
SPOER.MTUCH67HIZ
POECR2
MTU6BDZE
MTU7ACZE
MTU7BDZE
Hi-Z control on MTIOC6B and MTIOC6D pins
Hi-Z control on MTIOC7A and MTIOC7C pins
Hi-Z control on MTIOC7B and MTIOC7D pins
ICSR3
POE8E
POE8F
POE8#
POECR5
IC1ADDMT0ZE
IC2ADDMT0ZE
IC4ADDMT0ZE
IC5ADDMT0ZE
ICSR4
POE10E
POE10F
POE10#
POECR7
D0E0ADDMT0ZE
D1E0ADDMT0ZE
D2E0ADDMT0ZE
D9E0ADDMT0ZE
ICSR5
POE11E
POE11F
POE11#
POECR8
D0E1ADDMT0ZE
D9E1ADDMT0ZE
SPOER.MTUCH0HIZ
POECR1
MTU0AZE
MTU0BZE
MTU0CZE
MTU0DZE
Hi-Z control on MTIOC0A pin
Hi-Z control on MTIOC0B pin
Hi-Z control on MTIOC0C pin
Hi-Z control on MTIOC0D pin
ICSR6
OSTSTE
Oscillation stop detection signal
ICSR7
D0ERR0ST
D1ERR0ST
D2ERR0ST
D9ERR0ST
DSMIF0 error 0 signal
DSMIF1 error 0 signal
DSMIF2 error 0 signal
DSMIF9 error 0 signal
ICSR8
D0ERR1ST
D9ERR1ST
DSMIF0 error 1 signal
DSMIF9 error 1 signal

Note: The switch symbol is ON when the corresponding register bit is 1 and OFF when the corresponding register bit is 0.

Figure 19.2 Target pins and conditions for high-impedance control

High-impedance requests to individual pins can be controlled by the settings in the POECR1 to POECR2 registers.

The following input pins can be added to high-impedance control conditions by the settings in the POECR4 to POECR5 registers:

- Input pins other than the POE0# pin for the MTU3 and MTU4 pins
- Input pins other than the POE4# pin for the MTU6 and MTU7 pins
- Input pins other than the POE8# pin for the MTU0 pins

For example, setting the IC2ADDMT34ZE bit in POECR4 to 1 outputs high-impedance requests to the MTU3 and MTU4 pins even when the POE4# input is detected.

High-impedance requests due to the detection of POE8# input, POE10# input, POE11# input, stopped oscillation, oscillation abnormality, or DSMIF error can be controlled by the settings in the ICSR3 to ICSR10 registers. The ICSR1 and ICSR2 registers do not control enabling or disabling of output of high-impedance requests.

High-impedance requests to the MTU3, MTU4, MTU6, and MTU7 pins as the results of output level comparison can be controlled by the settings in the OCSR1 and OCSR2 registers.

19.3.1 MTU3 Pin Selection

In this LSI, the pin functions for MTU are respectively assigned to multiple sets of port pins. The target pins for high-impedance control can be selected by the pin select register in POE3 (M0SELR1, M0SELR2, M3SELR, M4SELR1, M4SELR2, M6SELR, M7SELR1, or M7SELR7 register). Table 19.5 shows the correspondence between MTU pins and select registers.

Note: Settings for pins to be used as MTU must be separately made in the registers of the I/O Ports. Take care so that there are no differences between the pins selected in the POE3 registers and the pins selected in the I/O port registers. For details of I/O port registers, see section 17, I/O Ports.

Table 19.5 Correspondence between MTU

MTU pin functions	Corresponding ports	Select registers
MTIOC0A	P01_6, P04_7, P10_5	M0SELR1
MTIOC0B	P01_7, P05_0, P10_6	
MTIOC0C	P02_0, P04_5	M0SELR2
MTIOC0D	P02_1, P04_6	
MTIOC3B	P00_0, P02_5	M3SELR
MTIOC3D	P00_1, P02_6	
MTIOC4A	P00_2, P02_7	M4SELR1
MTIOC4C	P00_3, P03_0	
MTIOC4B	P00_4, P03_1	M4SELR2
MTIOC4D	P00_5, P03_2	
MTIOC6B	P01_2, P03_7, P09_4	M6SELR
MTIOC6D	P01_3, P04_0, P09_5	
MTIOC7A	P01_4, P04_1, P09_6	M7SELR1
MTIOC7C	P01_5, P04_2, P09_7	
MTIOC7B	P01_6, P04_3, P10_0	M7SELR2
MTIOC7D	P01_7, P04_4, P10_1	

19.3.2 Input-Level Detection Operation

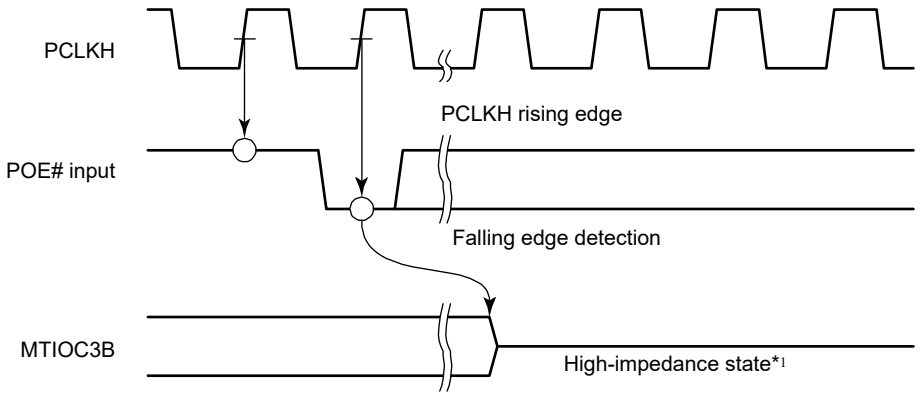
If the input conditions set by ICSR1 to ICSR5 occur on the POE0#, POE4#, POE8#, POE10#, and POE11# pins, the MTU3 and MTU4 pins for the MTU complementary PWM output, and MTU0 pins are placed in high-impedance state.

Note: These pins are still placed in the high-impedance state even when the MTU3 function is not selected for the pins on which they are multiplexed.

(1) Falling Edge Detection

When a change from a high to low level is input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins, the pin functions which are multiplexed with the pins for the MTU complementary PWM output, and MTU0 pins are placed in high-impedance state.

Figure 19.3 shows a sample timing after the level changes in input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins until the respective pins enter the high-impedance state.



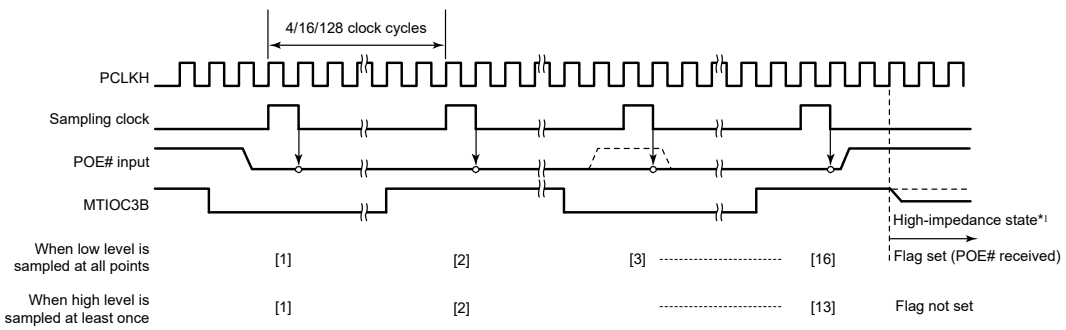
Note 1. Other pins also enter the high-impedance state at the same timing.

Figure 19.3 Falling edge detection

(2) Low-Level Detection

Figure 19.4 shows the low-level detection operation. When the POE# input has been sampled 16 times at the clock pulses selected by ICSR1 to ICSR5 and all are low level, the request is accepted. If even one high level is detected during this interval, the request is not accepted.

The timing when pins for the MTU complementary PWM output, and MTU0 pins enter the high-impedance state after the sampling clock is input is the same in both falling-edge detection and in low-level detection.



Note 1. Other pins also enter the high-impedance state at the same timing.

Figure 19.4 Low-level detection operation

19.3.3 Output-Level Compare Operation

Figure 19.5 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D. The operation is the same for the other pin combinations.

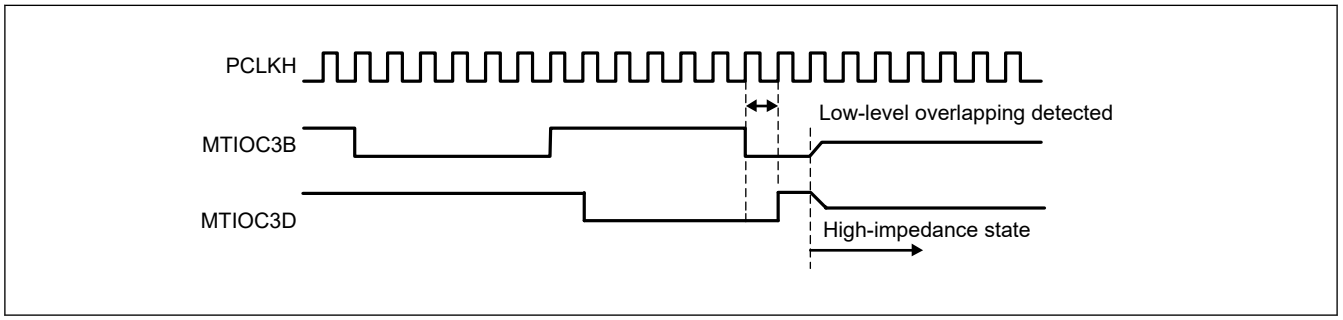


Figure 19.5 Output-level compare operation

19.3.4 High-Impedance Control Using Registers

The high-impedance state of the MTU pins (MTU0, MTU3, MTU4, MTU6, and MTU7) can be directly controlled by using the Software Port Output Enable Register (SPOER).

For instance, setting the MTUCH34HIZ bit in SPOER to 1 places the MTU3 and MTU4 pins specified by the Port Output Enable Control Register 2 (POECR2) in the high-impedance state.

The high-impedance state of other pins can also be controlled by setting the appropriate bits in SPOER.

19.3.5 High-Impedance Control through Detection of Stopped Oscillation and Oscillation Abnormality

When the clock monitor circuit detects the oscillation abnormality for PLLs and main clock, with the OSTSTE bit in ICSR6 set, the MTU3 and MTU4 or MTU6 and MTU7 pins if specified in the Port Output Enable Control Register 2 (POECR2), the MTU0 pins if specified in the Port Output Enable Control Register 1 (POECR1) can be placed in the high-impedance state.

19.3.6 Additional Functions for High-Impedance Control

Settings in the Port Output Enable Control Registers 4 to 5 (POECR4 to POECR5) can add additional high-impedance control conditions for the MTU complementary PWM output, MTU0 pins.

For instance, the following settings can be added as high-impedance control conditions for the MTU3 and MTU4 pins:

- Setting the POECR4.IC2ADDMT34ZE bit to 1 adds the input-level detection by the POE4# pin
- Setting the POECR4.IC3ADDMT34ZE bit to 1 and adds the input-level detection by the POE8# pin
- Setting the POECR4.IC4ADDMT34ZE bit to 1 and adds the input-level detection by the POE10# pin
- Setting the POECR4.IC5ADDMT34ZE bit to 1 and adds the input-level detection by the POE11# pin

The high-impedance control of other pins can also be controlled by setting the appropriate bits in the POECR4 and POECR5 registers.

19.3.7 High-Impedance Control through Detection of DSMIF error

When DSMIFn (n = 0 to 9) detect error events, the MTU3 and MTU4 pins if specified in the POECR9 and POECR10, the MTU6 and MTU7 pins if specified in the POECR1 and POECR12, or the MTU0 pins if specified in the POECR7 and POECR8, can be placed in the high-impedance state.

19.3.8 Recover from High-Impedance State

The outputs which have been in the high-impedance state due to input-level detection can be recovered from the state either by returning them to their initial state with a reset, or by clearing all of the ICSR1.POE0F, ICSR2.POE4F, ICSR3.POE8F, ICSR4.POE10F, and ICSR5.POE11F flags. However, when low-level sampling is selected with the ICSR1.POE0M[1:0],

ICSR2.POE4M[1:0], ICSR3.POE8M[1:0], ICSR4.POE10M[1:0], and ICSR5.POE11M[1:0] bits, just writing 0 to a flag is ignored (the flag is not set to 0). Flag can be cleared by writing 0 to it only after a high level is input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins and is detected.

The outputs which have been in the high-impedance state due to output-level detection can be recovered from the state either by returning them to their initial state with a reset, or by setting the OCSR1.OSF1 flag or the OCSR2.OSF2 flag to 0. However, just writing 0 to a flag is ignored (the flag is not set to 0). Flag can be cleared by writing 0 to it only after setting the inactive level to be output from the pin. In the MTU, the inactive level (initial output level) can be output by stopping the count operation.

The outputs which have been in the high-impedance state due to SPOER register setting can be recovered from the state either by returning them to their initial state with reset, or by setting the SPOER.MTUCH34HIZ bit, SPOER.MTUCH67HIZ bit or SPOER.MTUCH0HIZ bit to 0.

The outputs which have been in the high-impedance state due to oscillation stop detection for PLLs and main clock can be recovered from the state by returning them to their initial state with a reset.

The outputs which have been in the high-impedance state due to DSMIF error detection can be recovered from the state either by returning them to their initial state with reset, or by clearing DSMIF error status flags.

19.4 POE3 Setting Procedure

The procedure for setting POE3 is described in the section that follows. It shows an example of high-impedance control in response to comparison of the output levels on the MTU3 pins (MTIOC3B/MTIOC3D). In the procedure, P00_0 is selected as the MTIOC3B pin and P00_1 is selected as the MTIOC3D pin.

Set the MTU3 pins (P00_0 and P00_1) for the peripheral function by the PMCm and PFCm registers of the I/O Ports before making the following settings:

1. Set PMC00.PMC0 and PMC00.PMC1 bit to 1.
   Set the PMC00 of the I/O Ports to select P00_0 and P00_1 for the peripheral function pins.
2. Set PFC00.PFC0[5:0] bits to 0x6 and PFC00.PFC1[5:0] bits to 0x6.
   Set the PFC00 register of the I/O Ports to select P00_0 for the MTIOC3B pin and P00_1 for the MTIOC3D pin.
3. Set the M3SELR.M3BSEL[3:0] bits to 0x0 and M3SELR.M3DSEL[3:0] bits to 0x0.
   Set the M3SELR register to control the high-impedance state of P00_0 for MTIOC3B and P00_1 for MTIOC3D.
4. Set the ALR1.OLSG0A and OLSG0B bits to 0 and the OLSEN bit to 1 to set MTIOC3B and MTIOC3D as active-low.
5. Set the POECR2.MTU3BDZE bit to 1 to enable high-impedance control on the MTIOC3B and MTIOC3D pins.
6. Specify the setting to operate MTU3.

After the above settings, if MTIOC3B (P00_0) and MTIOC3D (P00_1) are simultaneously at the low level, then the pins are placed in high-impedance state and the OCSR1.OSF1 bit is set.

For the method of releasing high-impedance, see section 19.3.8. Recover from High-Impedance State.

19.5 Interrupts

The POE3 issues a request to generate an interrupt when the specified condition is satisfied during input-level detection or output-level comparison. Table 19.6 shows the interrupt sources and their conditions.

Table 19.6 Interrupt sources and conditions (1 of 2)

Name	Interrupt source	Interrupt flag	Condition
OEI1	Output enable interrupt 1	POE0F and OSF1	PIE1 • POE0F + OIE1 • OSF1
OEI2	Output enable interrupt 2	POE4F and OSF2	PIE2 • POE4F + OIE2 • OSF2
OEI3	Output enable interrupt 3	POE8F	PIE3 • POE8F

Table 19.6 Interrupt sources and conditions (2 of 2)

Name	Interrupt source	Interrupt flag	Condition
OEI4	Output enable interrupt 4	POE10F, POE11F, DnERR0ST, and DnERR1ST (n = 0 to 9)	PIE4 • POE10F + PIE5 • POE11F + D0ERR0IE • D0ERR0ST + D0ERR1IE • D0ERR1ST + D1ERR1IE • D1ERR0ST + D1ERR1IE • D1ERR1ST + D2ERR0IE • D2ERR0ST + D2ERR1IE • D2ERR1ST + D3ERR1IE • D3ERR0ST + D3ERR1IE • D3ERR1ST + D4ERR0IE • D4ERR0ST + D4ERR1IE • D4ERR1ST + D5ERR1IE • D5ERR0ST + D5ERR1IE • D5ERR1ST + D6ERR0IE • D6ERR0ST + D6ERR1IE • D6ERR1ST + D7ERR1IE • D7ERR0ST + D7ERR1IE • D7ERR1ST + D8ERR0IE • D8ERR0ST + D8ERR1IE • D8ERR1ST + D9ERR1IE • D9ERR0ST + D9ERR1IE • D9ERR1ST

19.6 Usage Notes

19.6.1 High-Impedance Control when the MTU is not Selected

If high-impedance control for a pin having a multiplexed MTU pin function on a register POECR1 to POECR2 is enabled and the high-impedance condition is satisfied, the pin is placed in the high-impedance state even if the MTU function is not selected for the pin on which it is multiplexed.

To avoid unintended high-impedance states, ensure that there are no differences between the settings for MTU pin selection in the PMCm and PFCm registers of the I/O Ports and for MTU pin selection in the pin select register of the POE.

20. General PWM Timer (GPT)

20.1 Overview

The LSI has a General PWM Timer (GPT) that is a 32-bit timer with 56 channels. These 56 channels are divided into four groups (LLPP (LLPP0 and LLPP1), NONSAFETY, and SAFETY). LLPP0 consists of 6 units, LLPP1 consists of 3 units, NONSAFETY and SAFETY consist of 1 unit respectively. Also, each unit is independent of each other. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. The GPT can also be used as a general-purpose timer.

Table 20.1 lists the specifications for the GPT and Figure 20.1 shows a block diagram of the GPT. Table 20.2 lists the I/O pins and Table 20.3 lists the GPT interrupt sources.

Table 20.1 GPT specifications (1 of 2)

Parameter	Description
Number of units	11 units (unit 00 to 05 in LLPP0, unit 06 to 08 in LLPP1, unit 09 in NONSAFETY and unit 10 in SAFETY)
Number of channels	• Total 56 channels • 30 channels (6 units x 5 channels) in LLPP0 (GPT00_0 to GPT05_4) • 15 channels (3 units x 5 channels) in LLPP1 (GPT06_0 to GPT08_4) • 7 channels (1 unit x 7 channels) in NONSAFETY (GPT09_0 to GPT_09_6) • 4 channels (1 unit x 4 channels) in SAFETY (GPT10_0 to GPT10_3)
Functions	• 32 bits • Up-counting or down-counting (saw waves) or up/down-counting (triangle waves) for each counter • Clock sources independently selectable for each channel • 2 I/O pins per channel • 2 output compare/input capture registers per channel • For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use • In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetric PWM waveforms • Registers for setting up frame cycles in each channel with capability for generating interrupts at overflow or underflow • Generation of dead times in PWM operation • Count direction switching is available in the triangle-wave PWM modes • Synchronous starting, stopping, clearing, and switching counters for arbitrary channels within group (LLPP, NONSAFETY, SAFETY) • Starting, stopping, clearing, and switching up/down counters in response to a maximum of six ELC events for each unit in LLPP and eight for NONSAFETY • Starting, stopping, and clearing up/down counters in response to input level comparison • Starting, stopping, clearing, and switching up/down counters in response to a maximum of four external triggers • Output pin disable function by dead time error and detected short circuits between output pins • A/D converter start trigger can be generated • Compare match A to F event and overflow/underflow event can be output to the ELC for channels in LLPP and NONSAFETY • Enables the noise filter for input capture • Bus clock: PCLKH in LLPP, PCLKM in NONSAFETY, and PCLKM in SAFETY • Core clock: PCLKGPTL in LLPP, PCLKM in NONSAFETY, and PCLKM in SAFETY • Ch 4 input are selectable from ch3 and ch4 input signals for units in LLPP. It enables to realize phase counting and pulse width measurement at same time
Prescaler	• 9 dividing clocks are output • Selectable from any of core clock, core clock/2, core clock/4, core clock/8, core clock/16, core clock/32, core clock/64, core clock/256, core clock/1024
Interrupts	9 sources: • GTCCRA compare match/input capture (GPTm_n_CCMPA) • GTCCRB compare match/input capture (GPTm_n_CCMPB) • GTCCRC compare match (GPTm_n_CMPC) • GTCCRD compare match (GPTm_n_CMPD) • GTCCRE compare match (GPTm_n_CMPE) • GTCCRF compare match (GPTm_n_CMPF) • GTADTRA compare match (GPTm_n_ADTRGA) → only ELC for channels in LLPP and NONSAFETY • GTADTRB compare match (GPTm_n_ADTRGB) → only ELC for channels in LLPP and NONSAFETY • GTCNT overflow (GTPR compare match) (GPTm_n_OVF) • GTCNT underflow (GPTm_n_UDF) • Dead time error (GPTm_n_DTE) → only for channels in LLPP and NONSAFETY

Table 20.1 GPT specifications (2 of 2)

Parameter	Description
Module-stop function	Module-stop state can be set to reduce power consumption

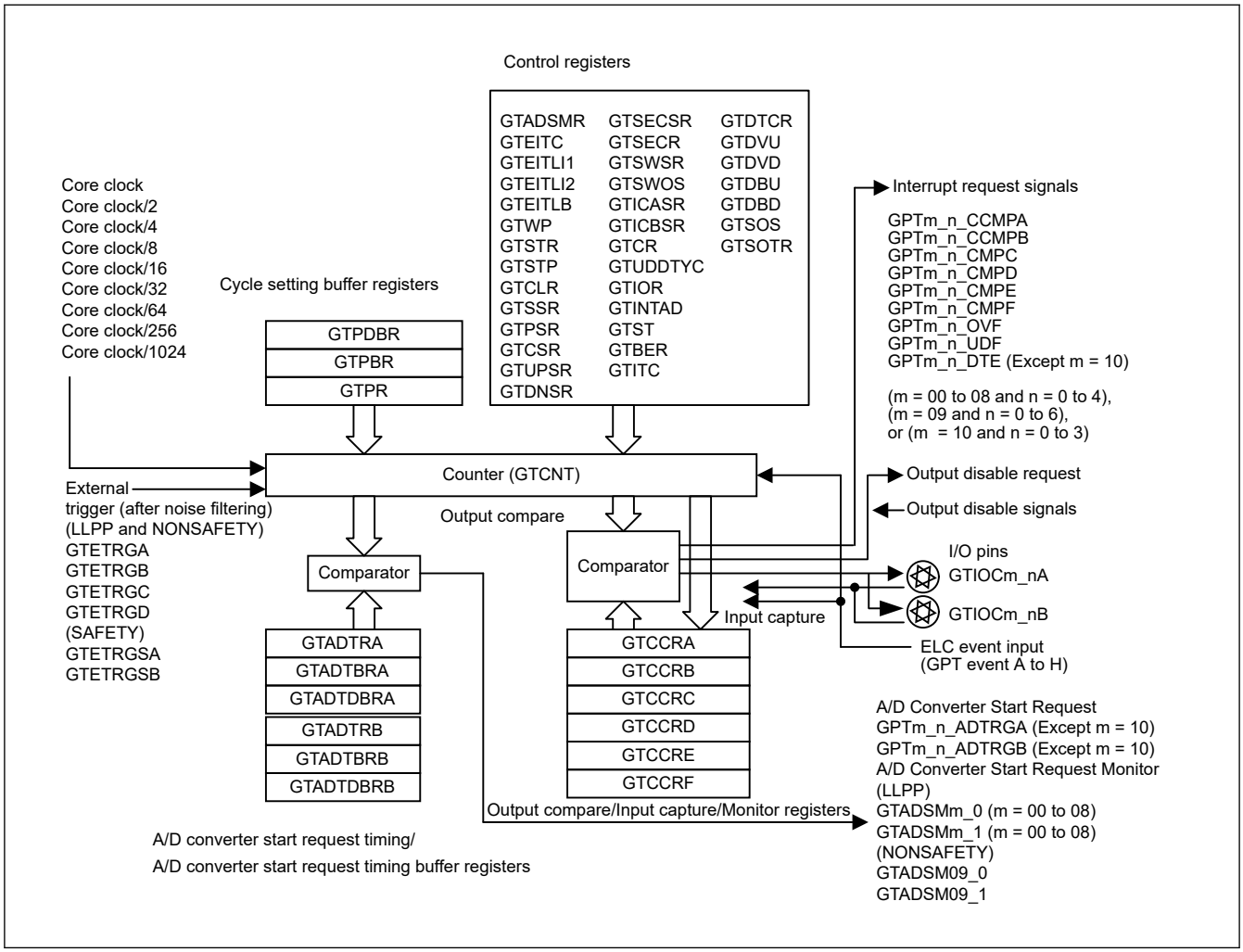


Figure 20.1 GPT block diagram (1 channel)

Table 20.2 GPT I/O pins (1 of 2)

Channel	Pin name	I/O	Function
Shared (LLPP and NONSAFETY)	GTETRGA	Input	External trigger input pin A (after noise filtering) for channels in LLPP and NONSAFETY
	GTETRGB	Input	External trigger input pin B (after noise filtering) for channels in LLPP and NONSAFETY
	GTETRGC	Input	External trigger input pin C (after noise filtering) for channels in LLPP and NONSAFETY
	GTETRGD	Input	External trigger input pin D (after noise filtering) for channels in LLPP and NONSAFETY
Shared (SAFETY)	GTETRGSA	Input	External trigger input pin A (after noise filtering) for channels in SAFETY
	GTETRGSB	Input	External trigger input pin B (after noise filtering) for channels in SAFETY

Table 20.2 GPT I/O pins (2 of 2)

Channel	Pin name	I/O	Function
Shared (LLPP (m = 00 to 08))	GTADSMm_0	Output	A/D conversion start request monitor 0 output pin for unit m in LLPP
	GTADSMm_1	Output	A/D conversion start request monitor 1 output pin for unit m in LLPP
Shared (NONSAFETY)	GTADSM09_0	Output	A/D conversion start request monitor 0 output pin for unit 09 in NONSAFETY
	GTADSM09_1	Output	A/D conversion start request monitor 1 output pin for unit 09 in NONSAFETY
GPTm_n (m = 00 to 08, n = 0 to 4) (LLPP)	GTIOCm_nA	I/O	GPTm_n GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCm_nB	I/O	GPTm_n GTCCRB register input capture input/output compare output/PWM output pin
GPT09_n (n = 0 to 6) (NONSAFETY)	GTIOC09_nA	I/O	GPT09_n GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC09_nB	I/O	GPT09_n GTCCRB register input capture input/output compare output/PWM output pin
GPT10_n (n = 0 to 3) (SAFETY)	GTIOC10_nA	I/O	GPT10_n GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC10_nB	I/O	GPT10_n GTCCRB register input capture input/output compare output/PWM output pin

Note: External triggers are available through POEG.

Table 20.3 GPT interrupt sources

Name	Interrupt sources
GPTm_n_CCMPA	GTCCRA input capture / compare match
GPTm_n_CCMPB	GTCCRB input capture / compare match
GPTm_n_CMPC	GTCCRC compare match
GPTm_n_CMPD	GTCCRD compare match
GPTm_n_CMPE	GTCCRE compare match
GPTm_n_CMPF	GTCCRF compare match
GPTm_n_OVF	GTCNT overflow (GTPR compare match)
GPTm_n_UDF	GTCNT underflow
GPTm_n_DTE*2	Dead time error
GPTm_n_ADTRGA*1 *2	GTADTRA compare match (A/D Converter Start Request A)
GPTm_n_ADTRGB*1 *2	GTADTRB compare match (A/D Converter Start Request B)

Note: (m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3).
Note 1. A/D converter start requests are only used for ELC.
Note 2. Except m = 10 (Unit 10, SAFETY)

20.2 Register Map

Table 20.4 GPT register map (1 of 5)

Address	Register symbol	Register name	Write protection
0x9000_2000 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0000 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0000 + 0x0100 × n (n = 0 to 6) 0x8100_0000 + 0x0100 × n (n = 0 to 3)	GTWP	General PWM Timer Write-Protection Register	—
0x9000_2004 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0004 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0004 + 0x0100 × n (n = 0 to 6) 0x8100_0004 + 0x0100 × n (n = 0 to 3)	GTSTR	General PWM Timer Software Start Register	GTWP.STRWP

Table 20.4 GPT register map (2 of 5)

Address	Register symbol	Register name	Write protection
0x9000_2008 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0008 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0008 + 0x0100 × n (n = 0 to 6) 0x8100_0008 + 0x0100 × n (n = 0 to 3)	GTSTP	General PWM Timer Software Stop Register	GTWP.STPWP
0x9000_200C + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_000C + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_000C + 0x0100 × n (n = 0 to 6) 0x8100_000C + 0x0100 × n (n = 0 to 3)	GTCLR	General PWM Timer Software Clear Register	GTWP.CLRWP
0x9000_2010 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0010 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0010 + 0x0100 × n (n = 0 to 6) 0x8100_0010 + 0x0100 × n (n = 0 to 3)	GTSSR	General PWM Timer Start Source Select Register	GTWP.WP
0x9000_2014 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0014 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0014 + 0x0100 × n (n = 0 to 6) 0x8100_0014 + 0x0100 × n (n = 0 to 3)	GTPSR	General PWM Timer Stop Source Select Register	GTWP.WP
0x9000_2018 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0018 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0018 + 0x0100 × n (n = 0 to 6) 0x8100_0018 + 0x0100 × n (n = 0 to 3)	GTCSR	General PWM Timer Clear Source Select Register	GTWP.WP
0x9000_201C + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_001C + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_001C + 0x0100 × n (n = 0 to 6) 0x8100_001C + 0x0100 × n (n = 0 to 3)	GTUPSR	General PWM Timer Count-Up Source Select Register	GTWP.WP
0x9000_2020 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0020 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0020 + 0x0100 × n (n = 0 to 6) 0x8100_0020 + 0x0100 × n (n = 0 to 3)	GTDNSR	General PWM Timer Count-Down Source Select Register	GTDNSR
0x9000_2024 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0024 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0024 + 0x0100 × n (n = 0 to 6) 0x8100_0024 + 0x0100 × n (n = 0 to 3)	GTICASR	General PWM Timer Input Capture Source Select Register A	GTICASR
0x9000_2028 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0028 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0028 + 0x0100 × n (n = 0 to 6) 0x8100_0028 + 0x0100 × n (n = 0 to 3)	GTICBSR	General PWM Timer Input Capture Source Select Register B	GTICBSR
0x9000_202C + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_002C + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_002C + 0x0100 × n (n = 0 to 6) 0x8100_002C + 0x0100 × n (n = 0 to 3)	GTCR	General PWM Timer Control Register	GTCR
0x9000_2030 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0030 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0030 + 0x0100 × n (n = 0 to 6) 0x8100_0030 + 0x0100 × n (n = 0 to 3)	GTUDDTYC	General PWM Timer Count Direction and Duty Setting Register	GTWP.WP
0x9000_2034 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0034 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0034 + 0x0100 × n (n = 0 to 6) 0x8100_0034 + 0x0100 × n (n = 0 to 3)	GTIOR	General PWM Timer I/O Control Register	GTWP.WP
0x9000_2038 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0038 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0038 + 0x0100 × n (n = 0 to 6) 0x8100_0038 + 0x0100 × n (n = 0 to 3)	GTINTAD	General PWM Timer Interrupt Output Setting Register	GTWP.WP
0x9000_203C + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_003C + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_003C + 0x0100 × n (n = 0 to 6) 0x8100_003C + 0x0100 × n (n = 0 to 3)	GTST	General PWM Timer Status Register	GTWP.WP
0x9000_2040 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0040 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0040 + 0x0100 × n (n = 0 to 6) 0x8100_0040 + 0x0100 × n (n = 0 to 3)	GTBER	General PWM Timer Buffer Enable Register	GTWP.WP
0x9000_2044 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0044 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0044 + 0x0100 × n (n = 0 to 6) 0x8100_0044 + 0x0100 × n (n = 0 to 3)	GTITC	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register	GTWP.WP

Table 20.4 GPT register map (3 of 5)

Address	Register symbol	Register name	Write protection
0x9000_2048 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0048 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0048 + 0x0100 × n (n = 0 to 6) 0x8100_0048 + 0x0100 × n (n = 0 to 3)	GTCNT	General PWM Timer Counter	GTWP.WP
0x9000_204C + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_004C + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_004C + 0x0100 × n (n = 0 to 6) 0x8100_004C + 0x0100 × n (n = 0 to 3)	GTCCRA	General PWM Timer Compare Capture Register A	GTWP.WP
0x9000_2050 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0050 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0050 + 0x0100 × n (n = 0 to 6) 0x8100_0050 + 0x0100 × n (n = 0 to 3)	GTCCRB	General PWM Timer Compare Capture Register B	GTWP.WP
0x9000_2054 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0054 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0054 + 0x0100 × n (n = 0 to 6) 0x8100_0054 + 0x0100 × n (n = 0 to 3)	GTCCRC	General PWM Timer Compare Capture Register C	GTWP.WP
0x9000_2058 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0058 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0058 + 0x0100 × n (n = 0 to 6) 0x8100_0058 + 0x0100 × n (n = 0 to 3)	GTCCRE	General PWM Timer Compare Capture Register E	GTWP.WP
0x9000_205C + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_005C + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_005C + 0x0100 × n (n = 0 to 6) 0x8100_005C + 0x0100 × n (n = 0 to 3)	GTCCRD	General PWM Timer Compare Capture Register D	GTWP.WP
0x9000_2060 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0060 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0060 + 0x0100 × n (n = 0 to 6) 0x8100_0060 + 0x0100 × n (n = 0 to 3)	GTCCRF	General PWM Timer Compare Capture Register F	GTWP.WP
0x9000_2064 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0064 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0064 + 0x0100 × n (n = 0 to 6) 0x8100_0064 + 0x0100 × n (n = 0 to 3)	GTPR	General PWM Timer Cycle Setting Register	GTWP.WP
0x9000_2068 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0068 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0068 + 0x0100 × n (n = 0 to 6) 0x8100_0068 + 0x0100 × n (n = 0 to 3)	GTPBR	General PWM Timer Cycle Setting Buffer Register	GTWP.WP
0x9000_206C + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_006C + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_006C + 0x0100 × n (n = 0 to 6) 0x8100_006C + 0x0100 × n (n = 0 to 3)	GTPDBR	General PWM Timer Cycle Setting Double-Buffer Register	GTWP.WP
0x9000_2070 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0070 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0070 + 0x0100 × n (n = 0 to 6) 0x8100_0070 + 0x0100 × n (n = 0 to 3)	GTADTRA	A/D Converter Start Request Timing Register A (m = A, B)	GTWP.WP
0x9000_2074 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0074 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0074 + 0x0100 × n (n = 0 to 6) 0x8100_0074 + 0x0100 × n (n = 0 to 3)	GTADTBRA	A/D Converter Start Request Timing Buffer Register A	GTWP.WP
0x9000_2078 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0078 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0078 + 0x0100 × n (n = 0 to 6) 0x8100_0078 + 0x0100 × n (n = 0 to 3)	GTADTDBRA	A/D Converter Start Request Timing Double-Buffer Register A	GTWP.WP
0x9000_207C + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_007C + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_007C + 0x0100 × n (n = 0 to 6) 0x8100_007C + 0x0100 × n (n = 0 to 3)	GTADTRB	A/D Converter Start Request Timing Register B	GTWP.WP
0x9000_2080 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0080 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0080 + 0x0100 × n (n = 0 to 6) 0x8100_0080 + 0x0100 × n (n = 0 to 3)	GTADTBRB	A/D Converter Start Request Timing Buffer Register B	GTWP.WP
0x9000_2084 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0084 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0084 + 0x0100 × n (n = 0 to 6) 0x8100_0084 + 0x0100 × n (n = 0 to 3)	GTADTDBRB	A/D Converter Start Request Timing Double-Buffer Register B	GTWP.WP

Table 20.4 GPT register map (4 of 5)

Address	Register symbol	Register name	Write protection
0x9000_2088 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0088 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0088 + 0x0100 × n (n = 0 to 6) 0x8100_0088 + 0x0100 × n (n = 0 to 3)	GTDTCR	General PWM Timer Dead Time Control Register	GTWP.WP
0x9000_208C + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_008C + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_008C + 0x0100 × n (n = 0 to 6) 0x8100_008C + 0x0100 × n (n = 0 to 3)	GTDVU	General PWM Timer Dead Time Value Register U	GTWP.WP
0x9000_2090 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0090 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0090 + 0x0100 × n (n = 0 to 6) 0x8100_0090 + 0x0100 × n (n = 0 to 3)	GTDVD	General PWM Timer Dead Time Value Register D	GTWP.WP
0x9000_2094 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0094 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0094 + 0x0100 × n (n = 0 to 6) 0x8100_0094 + 0x0100 × n (n = 0 to 3)	GTDBU	General PWM Timer Dead Time Value Buffer Register U	GTWP.WP
0x9000_2098 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_0098 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_0098 + 0x0100 × n (n = 0 to 6) 0x8100_0098 + 0x0100 × n (n = 0 to 3)	GTDBD	General PWM Timer Dead Time Value Buffer Register D	GTWP.WP
0x9000_209C + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_009C + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_009C + 0x0100 × n (n = 0 to 6) 0x8100_009C + 0x0100 × n (n = 0 to 3)	GTSOS	General PWM Timer Output Protection Function Status Register	GTWP.WP
0x9000_20A0 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_00A0 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_00A0 + 0x0100 × n (n = 0 to 6) 0x8100_00A0 + 0x0100 × n (n = 0 to 3)	GTSOTR	General PWM Timer Output Protection Function Temporary Release Register	GTWP.WP
0x9000_20A4 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_00A4 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_00A4 + 0x0100 × n (n = 0 to 6) 0x8100_00A4 + 0x0100 × n (n = 0 to 3)	GTADSMR	General PWM Timer A/D Conversion Start Request Signal Monitoring Register	GTWP.WP
0x9000_20A8 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_00A8 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_00A8 + 0x0100 × n (n = 0 to 6) 0x8100_00A8 + 0x0100 × n (n = 0 to 3)	GTEITC	General PWM Timer Extended Interrupt Skipping Counter Control Register	GTWP.WP
0x9000_20AC + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_00AC + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_00AC + 0x0100 × n (n = 0 to 6) 0x8100_00AC + 0x0100 × n (n = 0 to 3)	GTEITLI1	General PWM Timer Extended Interrupt Skipping Setting Register 1	GTWP.WP
0x9000_20B0 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_00B0 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_00B0 + 0x0100 × n (n = 0 to 6) 0x8100_00B0 + 0x0100 × n (n = 0 to 3)	GTEITLI2	General PWM Timer Extended Interrupt Skipping Setting Register 2	GTWP.WP
0x9000_20B4 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_00B4 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_00B4 + 0x0100 × n (n = 0 to 6) 0x8100_00B4 + 0x0100 × n (n = 0 to 3)	GTEITLB	General PWM Timer Extended Buffer Transfer Skipping Setting Register	GTWP.WP
0x9000_20D0 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_00D0 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_00D0 + 0x0100 × n (n = 0 to 6) 0x8100_00D0 + 0x0100 × n (n = 0 to 3)	GTSECSR	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTWP.CMNWP
0x9000_20D4 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_00D4 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_00D4 + 0x0100 × n (n = 0 to 6) 0x8100_00D4 + 0x0100 × n (n = 0 to 3)	GTSECR	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTWP.CMNWP
0x9000_20D8 + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_00D8 + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_00D8 + 0x0100 × n (n = 0 to 6) 0x8100_00D8 + 0x0100 × n (n = 0 to 3)	GTSWSR	General PWM Timer Switch Source Select Register	GTWP.WP
0x9000_20DC + 0x1000 × m + 0x0100 × n (m = 0 to 5, n = 0 to 4) 0x9010_00DC + 0x1000 × (m - 6) + 0x0100 × n (m = 6 to 8, n = 0 to 4) 0x8000_00DC + 0x0100 × n (n = 0 to 6) 0x8100_00DC + 0x0100 × n (n = 0 to 3)	GTSWOS	General PWM Timer Switch Offset Setting Register	GTWP.WP

Table 20.4 GPT register map (5 of 5)

Address	Register symbol	Register name	Write protection
0x8029_0008	GTIOCSEL	GPT Input Capture Signal Select Register	—

Table 20.5 GPT related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
00 (LLPP0)	—	MSTPCRC.MSTPCRC01	SLVACCCTL7.LLPP_SL*1
01 (LLPP0)	—	MSTPCRC.MSTPCRC02	SLVACCCTL7.LLPP_SL*1
02 (LLPP0)	—	MSTPCRC.MSTPCRC16	SLVACCCTL7.LLPP_SL*1
03 (LLPP0)	—	MSTPCRC.MSTPCRC17	SLVACCCTL7.LLPP_SL*1
04 (LLPP0)	—	MSTPCRC.MSTPCRC18	SLVACCCTL7.LLPP_SL*1
05 (LLPP0)	—	MSTPCRC.MSTPCRC19	SLVACCCTL7.LLPP_SL*1
06 (LLPP1)	—	MSTPCRC.MSTPCRC20	SLVACCCTL7.LLPP_SL*1
07 (LLPP1)	—	MSTPCRC.MSTPCRC21	SLVACCCTL7.LLPP_SL*1
08 (LLPP1)	—	MSTPCRC.MSTPCRC22	SLVACCCTL7.LLPP_SL*1
09 (NONSAFETY)	—	MSTPCRC.MSTPCRC23	SLVACCCTL0.GPT09_SL
10 (SAFETY)	—	MSTPCRG.MSTPCRG03	SLVACCCTL2.GPT10_SL
GTIOCSEL register	—	—	SLVACCCTL7.LLPP_SL*1

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

20.3 Register Descriptions

20.3.1 GTWP : General PWM Timer Write-Protection Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	CMNWP	CLRWP	STPWP	STRWP	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WP	Register Write Disabled 0: Write to the register is enabled 1: Write to the register is disabled	R/W
1	STRWP	GTSTR.CSTRT Bit Write Disabled 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
2	STPWP	GTSTP.CSTOP Bit Write Disabled 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W

Bit	Symbol	Function	R/W
3	CLRWP	GTCLR.CCLR Bit Write Disabled 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
4	CMNWP	Common Register Write Disabled 0: Write to the register is enabled 1: Write to the register is disabled	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	GTWP Key Code To modify the WP, STRWP, STPWP, CLRWP, and CMNWP bits, write 0xA5.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

For detail, see section 20.9.1. Write-Protection for Registers.

20.3.2 GTSTR : General PWM Timer Software Start Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CSTRT6	CSTRT5	CSTRT4	CSTRT3	CSTRT2	CSTRT1	CSTRT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CSTRT0	Channel 0 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
1	CSTRT1	Channel 1 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
2	CSTRT2	Channel 2 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
3	CSTRT3	Channel 3 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
4	CSTRT4*1	Channel 4 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W

Bit	Symbol	Function	R/W
5	CSTRT5*1 *2	Channel 5 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
6	CSTRT6*1 *2	Channel 6 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to 08 (LLPP0 and LLPP1)

The GTSTR register starts the GTCNT counter operation of each channel in the same unit.

The GTSTR register of each channel is a common register, and writing 1 to the GTSTR register in any channel and updating it can start operation of the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.

20.3.3 GTSTP : General PWM Timer Software Stop Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CSTOP6	CSTOP5	CSTOP4	CSTOP3	CSTOP2	CSTOP1	CSTOP0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	CSTOP0	Channel 0 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
1	CSTOP1	Channel 1 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
2	CSTOP2	Channel 2 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
3	CSTOP3	Channel 3 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W

Bit	Symbol	Function	R/W
4	CSTOP4*1	Channel 4 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
5	CSTOP5*1 *2	Channel 5 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
6	CSTOP6*1 *2	Channel 6 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
31:7	—	These bits are read as 1. The write value should be 1.	R/W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to 08 (LLPP0 and LLPP1)

The GTSTP register stops the GTCNT counter operation of each channel in the same unit.

The GTSTP register of each channel is a common register, and writing 1 to the GTSTP register in any channel and updating it can stop operation of the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.

20.3.4 GTCLR : General PWM Timer Software Clear Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CCLR6	CCLR5	CCLR4	CCLR3	CCLR2	CCLR1	CCLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CCLR0	Channel 0 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
1	CCLR1	Channel 1 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
2	CCLR2	Channel 2 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
3	CCLR3	Channel 3 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
4	CCLR4*1	Channel 4 Count Clear 0: The counter is not cleared 1: The counter is cleared	W

Bit	Symbol	Function	R/W
5	CCLR5*1 *2	Channel 5 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
6	CCLR6*1 *2	Channel 6 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
31:7	—	The write value should be 0.	W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to 08 (LLPP0 and LLPP1)

The GTCLR register is a write-only register that sets the clearing of the GTCNT counter of each channel in the same unit. The GTCLR register of each channel is a common register, and writing 1 to the GTCLR register in any channel and updating it can clear the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.

20.3.5 GTSSR : General PWM Timer Start Source Select Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTRT	—	—	—	—	—	—	—	SSELCH	SSELCG	SSELCF	SSELCE	SSELCD	SSELCC	SSELCB	SSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SSCBFAHL[1:0]		SSCBRAHL[1:0]		SSCAFBHL[1:0]		SSCARBHL[1:0]		SSGTRGDFR[1:0]		SSGTRGCFR[1:0]		SSGTRGBFR[1:0]		SSGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SSGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTETRGA signal 1 0: The counter starts at a falling edge of the GTETRGA signal 1 1: The counter starts at both edges of the GTETRGA signal	R/W
3:2	SSGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTETRGB signal 1 0: The counter starts at a falling edge of the GTETRGB signal 1 1: The counter starts at both edges of the GTETRGB signal	R/W
5:4	SSGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTIOC10_0A output Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTETRGC signal 1 0: The counter starts at a falling edge of the GTETRGC signal 1 1: The counter starts at both edges of the GTETRGC signal	R/W
7:6	SSGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTIOC10_1A output Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTETRGD signal 1 0: The counter starts at a falling edge of the GTETRGD signal 1 1: The counter starts at both edges of the GTETRGD signal	R/W

Bit	Symbol	Function	R/W
9:8	SSCARBHL[1:0]	GTIOCm_nA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nA signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter starts at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter starts at a rising edge of the GTIOCm_nA signal	R/W
11:10	SSCAFBHL[1:0]	GTIOCm_nA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nA signal is not used as a trigger to start counting 0 1: The counter starts at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter starts at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter starts at a falling edge of the GTIOCm_nA signal	R/W
13:12	SSCBRAHL[1:0]	GTIOCm_nB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nB signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter starts at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter starts at a rising edge of the GTIOCm_nB signal	R/W
15:14	SSCBFAHL[1:0]	GTIOCm_nB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nB signal is not used as a trigger to start counting 0 1: The counter starts at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter starts at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter starts at a falling edge of the GTIOCm_nB signal	R/W
16	SSELCA*1	Event Source Count Start Enable 0: Disables count start by the event A input 1: Enables count start by the event A input	R/W
17	SSELCB*1	Event Source Count Start Enable 0: Disables count start by the event B input 1: Enables count start by the event B input	R/W
18	SSELCC*1	Event Source Count Start Enable 0: Disables count start by the event C input 1: Enables count start by the event C input	R/W
19	SSELCD*1	Event Source Count Start Enable 0: Disables count start by the event D input 1: Enables count start by the event D input	R/W
20	SSELCE*1	Event Source Count Start Enable 0: Disables count start by the event E input 1: Enables count start by the event E input	R/W
21	SSELCF*1	Event Source Count Start Enable 0: Disables count start by the event F input 1: Enables count start by the event F input	R/W
22	SSELCG*1 *2	Event Source Count Start Enable 0: Disables count start by the event G input 1: Enables count start by the event G input	R/W
23	SSELCH*1 *2	Event Source Count Start Enable 0: Disables count start by the event H input 1: Enables count start by the event H input	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTRT	Software Source Count Start Enable 0: Disables count start by the GTSTR register 1: Enables count start by the GTSTR register	R/W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to unit 08 (LLPP0 and LLPP1)

20.3.6 GTPSR : General PWM Timer Stop Source Select Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTOP	—	—	—	—	—	—	—	PSELCH	PSELCG	PSELCF	PSELCE	PSELCD	PSELCC	PSELCB	PSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSCBFAHL[1:0]		PSCBRAHL[1:0]		PSCAFBHL[1:0]		PSCARBHL[1:0]		PSGTRGDFR[1:0]		PSGTRGCFR[1:0]		PSGTRGBFR[1:0]		PSGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PSGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTETRGA signal 1 0: The counter stops at a falling edge of the GTETRGA signal 1 1: The counter stops at both edges of the GTETRGA signal	R/W
3:2	PSGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTETRGB signal 1 0: The counter stops at a falling edge of the GTETRGB signal 1 1: The counter stops at both edges of the GTETRGB signal	R/W
5:4	PSGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTIOC10_0A output Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTETRGC signal 1 0: The counter stops at a falling edge of the GTETRGC signal 1 1: The counter stops at both edges of the GTETRGC signal	R/W
7:6	PSGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTIOC10_1A output Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTETRGD signal 1 0: The counter stops at a falling edge of the GTETRGD signal 1 1: The counter stops at both edges of the GTETRGD signal	R/W
9:8	PSCARBHL[1:0]	GTIOCm_nA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nA signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter stops at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter stops at a rising edge of the GTIOCm_nA signal	R/W
11:10	PSCAFBHL[1:0]	GTIOCm_nA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nA signal is not used as a trigger to stop counting 0 1: The counter stops at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter stops at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter stops at a falling edge of the GTIOCm_nA signal	R/W
13:12	PSCBRAHL[1:0]	GTIOCm_nB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nB signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter stops at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter stops at a rising edge of the GTIOCm_nB signal	R/W

Bit	Symbol	Function	R/W
15:14	PSCBFAHL[1:0]	GTIOCm_nB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nB signal is not used as a trigger to stop counting 0 1: The counter stops at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter stops at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter stops at a falling edge of the GTIOCm_nB signal	R/W
16	PSELCA*1	Event Source Count Stop Enable 0: Disables count stop by the event A input 1: Enables count stop by the event A input	R/W
17	PSELCB*1	Event Source Count Stop Enable 0: Disables count stop by the event B input 1: Enables count stop by the event B input	R/W
18	PSELCC*1	Event Source Count Stop Enable 0: Disables count stop by the event C input 1: Enables count stop by the event C input	R/W
19	PSELCD*1	Event Source Count Stop Enable 0: Disables count stop by the event D input 1: Enables count stop by the event D input	R/W
20	PSELCE*1	Event Source Count Stop Enable 0: Disables count stop by the event E input 1: Enables count stop by the event E input	R/W
21	PSELCF*1	Event Source Count Stop Enable 0: Disables count stop by the event F input 1: Enables count stop by the event F input	R/W
22	PSELCG*1 *2	Event Source Count Stop Enable 0: Disables count stop by the event G input 1: Enables count stop by the event G input	R/W
23	PSELCH*1 *2	Event Source Count Stop Enable 0: Disables count stop by the event H input 1: Enables count stop by the event H input	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTOP	Software Source Count Stop Enable 0: Disables count stop by the GTSTP register 1: Enables count stop by the GTSTP register	R/W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to unit 08 (LLPP0 and LLPP1)

20.3.7 GTCSR : General PWM Timer Clear Source Select Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CCLR	—	—	—	—	—	—	—	CSELCH	CSELCG	CSELCF	CSELCE	CSELCD	CSELCC	CSELCB	CSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CSCBFAHL[1:0]		CSCBRAHL[1:0]		CSCAFBHL[1:0]		CSCARBHL[1:0]		CSGTRGDFR[1:0]		CSGTRGCFR[1:0]		CSGTRGBFR[1:0]		CSGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CSGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTETRGA signal 1 0: The counter is cleared at a falling edge of the GTETRGA signal 1 1: The counter is cleared at both edges of the GTETRGA signal	R/W
3:2	CSGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTETRGB signal 1 0: The counter is cleared at a falling edge of the GTETRGB signal 1 1: The counter is cleared at both edges of the GTETRGB signal	R/W
5:4	CSGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTIOC10_0A output Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTETRGC signal 1 0: The counter is cleared at a falling edge of the GTETRGC signal 1 1: The counter is cleared at both edges of the GTETRGC signal	R/W
7:6	CSGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTIOC10_1A output Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTETRGD signal 1 0: The counter is cleared at a falling edge of the GTETRGD signal 1 1: The counter is cleared at both edges of the GTETRGD signal	R/W
9:8	CSCARBHL[1:0]	GTIOCm_nA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nA signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter is cleared at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter is cleared at a rising edge of the GTIOCm_nA signal	R/W
11:10	CSCAFBHL[1:0]	GTIOCm_nA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nA signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter is cleared at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter is cleared at a falling edge of the GTIOCm_nA signal	R/W
13:12	CSCBRAHL[1:0]	GTIOCm_nB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nB signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter is cleared at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter is cleared at a rising edge of the GTIOCm_nB signal	R/W
15:14	CSCBFAHL[1:0]	GTIOCm_nB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nB signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter is cleared at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter is cleared at a falling edge of the GTIOCm_nB signal	R/W
16	CSELCA[*1]	Event Source Count Clear Enable 0: Disables counter clear by the event A input 1: Enables counter clear by the event A input	R/W
17	CSELCB[*1]	Event Source Count Clear Enable 0: Disables counter clear by the event B input 1: Enables counter clear by the event B input	R/W
18	CSELCC[*1]	Event Source Count Clear Enable 0: Disables counter clear by the event C input 1: Enables counter clear by the event C input	R/W
19	CSELCD[*1]	Event Source Count Clear Enable 0: Disables counter clear by the event D input 1: Enables counter clear by the event D input	R/W

Bit	Symbol	Function	R/W
20	CSELCE*1	Event Source Count Clear Enable 0: Disables counter clear by the event E input 1: Enables counter clear by the event E input	R/W
21	CSELCF*1	Event Source Count Clear Enable 0: Disables counter clear by the event F input 1: Enables counter clear by the event F input	R/W
22	CSELCG*1 *2	Event Source Count Clear Enable 0: Disables counter clear by the event G input 1: Enables counter clear by the event G input	R/W
23	CSELCH*1 *2	Event Source Count Clear Enable 0: Disables counter clear by the event H input 1: Enables counter clear by the event H input	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CCLR	Software Source Count Clear Enable 0: Disables counter clear by the GTCLR register 1: Enables counter clear by the GTCLR register	R/W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to unit 08 (LLPP0 and LLPP1)

20.3.8 GTUPSR : General PWM Timer Count-Up Source Select Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	USELCH	USELCG	USELCF	USELCE	USELCD	USELCC	USELCB	USELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	USCBFAHL[1:0]		USCBRAHL[1:0]		USCAFBHL[1:0]		USCARBHL[1:0]		USGTRGDFR[1:0]		USGTRGCFR[1:0]		USGTRGBFR[1:0]		USGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	USGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTETRGA signal 1 0: The counter is incremented at a falling edge of the GTETRGA signal 1 1: The counter is incremented at both edges of the GTETRGA signal	R/W
3:2	USGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTETRGB signal 1 0: The counter is incremented at a falling edge of the GTETRGB signal 1 1: The counter is incremented at both edges of the GTETRGB signal	R/W
5:4	USGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTIOC10_0A output Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTETRGC signal 1 0: The counter is incremented at a falling edge of the GTETRGC signal 1 1: The counter is incremented at both edges of the GTETRGC signal	R/W
7:6	USGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTIOC10_1A output Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTETRGD signal 1 0: The counter is incremented at a falling edge of the GTETRGD signal 1 1: The counter is incremented at both edges of the GTETRGD signal	R/W

Bit	Symbol	Function	R/W
9:8	USCARBHL[1:0]	GTIOCm_nA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nA signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter is incremented at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter is incremented at a rising edge of the GTIOCm_nA signal	R/W
11:10	USCAFBHL[1:0]	GTIOCm_nA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nA signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter is incremented at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter is incremented at a falling edge of the GTIOCm_nA signal	R/W
13:12	USCBRAHL[1:0]	GTIOCm_nB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nB signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter is incremented at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter is incremented at a rising edge of the GTIOCm_nB signal	R/W
15:14	USCBFAHL[1:0]	GTIOCm_nB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nB signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter is incremented at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter is incremented at a falling edge of the GTIOCm_nB signal	R/W
16	USELCA*1	Event Source Count-Up Enable 0: Disables count-up by the event A input 1: Enables count-up by the event A input	R/W
17	USELCB*1	Event Source Count-Up Enable 0: Disables count-up by the event B input 1: Enables count-up by the event B input	R/W
18	USELCC*1	Event Source Count-Up Enable 0: Disables count-up by the event C input 1: Enables count-up by the event C input	R/W
19	USELCD*1	Event Source Count-Up Enable 0: Disables count-up by the event D input 1: Enables count-up by the event D input	R/W
20	USELCE*1	Event Source Count-Up Enable 0: Disables count-up by the event E input 1: Enables count-up by the event E input	R/W
21	USELCF*1	Event Source Count-Up Enable 0: Disables count-up by the event F input 1: Enables count-up by the event F input	R/W
22	USELCG*1 *2	Event Source Count-Up Enable 0: Disables count-up by the event G input 1: Enables count-up by the event G input	R/W
23	USELCH*1 *2	Event Source Count-Up Enable 0: Disables count-up by the event H input 1: Enables count-up by the event H input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to unit 08 (LLPP0 and LLPP1)

20.3.9 GTDNSR : General PWM Timer Count-Down Source Select Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DSELCH	DSELCG	DSELCF	DSELCE	DSELCD	DSELCC	DSELCB	DSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DSCBFAHL[1:0]		DSCBRAHL[1:0]		DSCAFBHL[1:0]		DSCARBHL[1:0]		DSGTRGDFR[1:0]		DSGTRGCFR[1:0]		DSGTRGBFR[1:0]		DSGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	DSGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTETRGA signal 1 0: The counter is decremented at a falling edge of the GTETRGA signal 1 1: The counter is decremented at both edges of the GTETRGA signal	R/W
3:2	DSGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTETRGB signal 1 0: The counter is decremented at a falling edge of the GTETRGB signal 1 1: The counter is decremented at both edges of the GTETRGB signal	R/W
5:4	DSGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTIOC10_0A output Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTETRGC signal 1 0: The counter is decremented at a falling edge of the GTETRGC signal 1 1: The counter is decremented at both edges of the GTETRGC signal	R/W
7:6	DSGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTIOC10_1A output Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTETRGD signal 1 0: The counter is decremented at a falling edge of the GTETRGD signal 1 1: The counter is decremented at both edges of the GTETRGD signal	R/W
9:8	DSCARBHL[1:0]	GTIOCm_nA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nA signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter is decremented at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter is decremented at a rising edge of the GTIOCm_nA signal	R/W
11:10	DSCAFBHL[1:0]	GTIOCm_nA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nA signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter is decremented at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter is decremented at a falling edge of the GTIOCm_nA signal	R/W

Bit	Symbol	Function	R/W
13:12	DSCBRAHL[1:0]	GTIOCm_nB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nB signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter is decremented at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter is decremented at a rising edge of the GTIOCm_nB signal	R/W
15:14	DSCBFAHL[1:0]	GTIOCm_nB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nB signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter is decremented at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter is decremented at a falling edge of the GTIOCm_nB signal	R/W
16	DSELCA*1	Event Source Count-Down Enable 0: Disables count-down by the event A input 1: Enables count-down by the event A input	R/W
17	DSELCB*1	Event Source Count-Down Enable 0: Disables count-down by the event B input 1: Enables count-down by the event B input	R/W
18	DSELCC*1	Event Source Count-Down Enable 0: Disables count-down by the event C input 1: Enables count-down by the event C input	R/W
19	DSELCD*1	Event Source Count-Down Enable 0: Disables count-down by the event D input 1: Enables count-down by the event D input	R/W
20	DSELCE*1	Event Source Count-Down Enable 0: Disables count-down by the event E input 1: Enables count-down by the event E input	R/W
21	DSELCF*1	Event Source Count-Down Enable 0: Disables count-down by the event F input 1: Enables count-down by the event F input	R/W
22	DSELCG*1 *2	Event Source Count-Down Enable 0: Disables count-down by the event G input 1: Enables count-down by the event G input	R/W
23	DSELCH*1 *2	Event Source Count-Down Enable 0: Disables count-down by the event H input 1: Enables count-down by the event H input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to unit 08 (LLPP0 and LLPP1)

20.3.10 GTICASR : General PWM Timer Input Capture Source Select Register A

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	ASELCH	ASELCG	ASELCF	ASELCE	ASELCD	ASELCC	ASELCB	ASELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASCBFAHL[1:0]		ASCBRAHL[1:0]		ASCAFBHL[1:0]		ASCARBHL[1:0]		ASGTRGDFR[1:0]		ASGTRGCFR[1:0]		ASGTRGBFR[1:0]		ASGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ASGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGA signal 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGA signal 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGA signal	R/W
3:2	ASGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGB signal 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGB signal 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGB signal	R/W
5:4	ASGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTIOC10_0A output Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGC signal 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGC signal 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGC signal	R/W
7:6	ASGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTIOC10_1A output Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGD signal 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGD signal 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGD signal	R/W
9:8	ASCARBHL[1:0]	GTIOCm_nA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nA signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCm_nA signal	R/W

Bit	Symbol	Function	R/W
11:10	ASCAFBHL[1:0]	GTIOCm_nA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nA signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCm_nA signal	R/W
13:12	ASCBRAHL[1:0]	GTIOCm_nB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nB signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCm_nB signal	R/W
15:14	ASCBFAHL[1:0]	GTIOCm_nB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nB signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCm_nB signal	R/W
16	ASELCA*1	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the event A input 1: Enables GTCCRA input capture by the event A input	R/W
17	ASELCB*1	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the event B input 1: Enables GTCCRA input capture by the event B input	R/W
18	ASELCC*1	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the event C input 1: Enables GTCCRA input capture by the event C input	R/W
19	ASELCD*1	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the event D input 1: Enables GTCCRA input capture by the event D input	R/W
20	ASELCE*1	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the event E input 1: Enables GTCCRA input capture by the event E input	R/W
21	ASELCF*1	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the event F input 1: Enables GTCCRA input capture by the event F input	R/W
22	ASELCG*1 *2	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the event G input 1: Enables GTCCRA input capture by the event G input	R/W
23	ASELCH*1 *2	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the event H input 1: Enables GTCCRA input capture by the event H input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to unit 08 (LLPP0 and LLPP1)

20.3.11 GTICBSR : General PWM Timer Input Capture Source Select Register B

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	BSELCH	BSELCG	BSELCF	BSELCE	BSELCD	BSELCC	BSELCB	BSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSCBFAHL[1:0]		BSCBRAHL[1:0]		BSCAFBHL[1:0]		BSCARBHL[1:0]		BSGTRGDFR[1:0]		BSGTRGCFR[1:0]		BSGTRGBFR[1:0]		BSGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	BSGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGA signal 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGA signal 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGA signal	R/W
3:2	BSGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGB signal 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGB signal 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGB signal	R/W
5:4	BSGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTIOC10_0A output Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGC signal 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGC signal 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGC signal	R/W
7:6	BSGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTIOC10_1A output Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGD signal 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGD signal 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGD signal	R/W
9:8	BSCARBHL[1:0]	GTIOCm_nA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nA signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCm_nA signal	R/W

Bit	Symbol	Function	R/W
11:10	BSCAFBHL[1:0]	GTIOCm_nA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nA signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven low 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCm_nA signal while the GTIOCm_nB pin is driven high 1 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCm_nA signal	R/W
13:12	BSCBRAHL[1:0]	GTIOCm_nB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCm_nB signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCm_nB signal	R/W
15:14	BSCBFAHL[1:0]	GTIOCm_nB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCm_nB signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven low 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCm_nB signal while the GTIOCm_nA pin is driven high 1 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCm_nB signal	R/W
16	BSELCA*1	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the event A input 1: Enables GTCCRB input capture by the event A input	R/W
17	BSELCB*1	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the event B input 1: Enables GTCCRB input capture by the event B input	R/W
18	BSELCC*1	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the event C input 1: Enables GTCCRB input capture by the event C input	R/W
19	BSELCD*1	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the event D input 1: Enables GTCCRB input capture by the event D input	R/W
20	BSELCE*1	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the event E input 1: Enables GTCCRB input capture by the event E input	R/W
21	BSELCF*1	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the event F input 1: Enables GTCCRB input capture by the event F input	R/W
22	BSELCG*1 *2	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the event G input 1: Enables GTCCRB input capture by the event G input	R/W
23	BSELCH*1 *2	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the event H input 1: Enables GTCCRB input capture by the event H input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to unit 08 (LLPP0 and LLPP1)

20.3.12 GTCR : General PWM Timer Control Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SWMD[2:0]			—	—	TPCS[3:0]				—	—	—	—	MD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ICDS	—	—	—	—	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CST	Count Start The start/stop of the counter can also be set from other registers (GTSTR, GTSSR, GTSTP, GTPSR) and the counter status is reflected. 0: Count operation is stopped 1: Count operation is started	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	ICDS	Input Capture Operation Select at Count Stop 0: Input capture is operated during count stop 1: Input capture is not operated during count stop	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
18:16	MD[2:0]	Mode Select 0 0 0: Sawtooth-wave PWM mode (single buffer or double buffer possible) 0 0 1: Sawtooth-wave one-shot pulse mode (fixed buffer operation) 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 1 1 1: Setting prohibited	R/W
22:19	—	These bits are read as 0. The write value should be 0.	R/W
26:23	TPCS[3:0]	Timer Prescaler Select 0x0: Core Clock/1 0x1: Core Clock/2 0x2: Core Clock/4 0x3: Core Clock/8 0x4: Core Clock/16 0x5: Core Clock/32 0x6: Core Clock/64 0x7: Setting prohibited 0x8: Core Clock/256 0x9: Setting prohibited 0xA: Core Clock/1024 0xB: Setting prohibited 0xC: GTETRGA (GTETRGSA for SAFETY) (using the POEG) 0xD: GTETRGB (GTETRGSB for SAFETY) (using the POEG) 0xE: GTETRGC (GTIOC10_0A output for Safety) (using the POEG) 0xF: GTETRGD (GTIOC10_1A output for Safety) (using the POEG)	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31:29	SWMD[2:0]	Switch Mode Select 0 0 0: Hardware switch operation is disabled 0 0 1: Switch to upcount with offset reload 0 1 0: Switch to downcount with offset reload 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Switch to upcount without offset reload 1 1 0: Switch to downcount without offset reload 1 1 1: Setting prohibited	R/W

20.3.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	OABDTYT	OBDTYR	OBDTYF	OBDTY[1:0]		—	—	—	—	OADTYR	OADTYF	OADTY[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UDF	UD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	UD	Count Direction Setting 0: GTCNT counts down 1: GTCNT counts up	R/W
1	UDF	Forcible Count Direction Setting 0: Count direction is not forcibly set 1: Count direction is forcibly set	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	OADTY[1:0]	GTIOCm_nA Pin Output Duty Setting 0 x: Compare matches determine the duty cycle of the output on the GTIOCm_nA pin 1 0: The duty cycle of the output on the GTIOCm_nA pin is 0% 1 1: The duty cycle of the output on the GTIOCm_nA pin is 100%	R/W
18	OADTYF	GTIOCm_nA Pin Output Duty Forced Setting 0: Duty of the GTIOCm_nA pin output is not forcibly set 1: Duty of the GTIOCm_nA pin output is forcibly set	R/W
19	OADTYR	Output after Release of GTIOCm_nA Pin Output 0%/100% Duty Cycle Settings 0: The function selected by the GTIOR.GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting 1: The function selected by the GTIOR.GTIOA[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	OBDTY[1:0]	GTIOCm_nB Pin Output Duty Setting 0 x: Compare matches determine the duty cycle of the output on the GTIOCm_nB pin 1 0: The duty cycle of the output on the GTIOCm_nB pin is 0% 1 1: The duty cycle of the output on the GTIOCm_nB pin is 100%	R/W
26	OBDTYF	GTIOCm_nB Pin Output Duty Forced Setting 0: Duty of the GTIOCm_nB pin output is not forcibly set 1: Duty of the GTIOCm_nB pin output is forcibly set	R/W

Bit	Symbol	Function	R/W
27	OBDTYR	Output after Release of GTIOCm_nB Pin Output 0%/100% Duty Cycle Settings 0: The function selected by the GTIOR.GTIOB[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting 1: The function selected by the GTIOR.GTIOB[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting	R/W
28	OABDTYT	GTUDDTYC.OADTY[1:0] and GTUDDTYC.OBDTY[1:0] reflect timing setting in the triangle-wave mode Setting this bit to 1 is prohibited except in triangle-wave mode. 0: Reflect GTUDDTYC.OADTY[1:0] and GTUDDTYC.OBDTY[1:0] setting at trough of triangle-wave 1: Reflect GTUDDTYC.OADTY[1:0] and GTUDDTYC.OBDTY[1:0] setting at trough and crest of triangle-wave	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

20.3.14 GTIOR : General PWM Timer I/O Control Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCSB[1:0]		NFBEN	—	—	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NFCSA[1:0]		NFAEN	—	—	OADF[1:0]		OAE	OAHLD	OADFLT	—	GTIOA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	GTIOA[4:0]	GTIOCm_nA Pin Function Select See Table 20.6.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	OADFLT	GTIOCm_nA Pin Output Value Setting at the Count Stop 0: The GTIOCm_nA pin outputs low when counting is stopped 1: The GTIOCm_nA pin outputs high when counting is stopped	R/W
7	OAHLD	GTIOCm_nA Pin Output Retention at the Start/Stop Count 0: The GTIOCm_nA pin output level at start/stop of counting depends on the register setting 1: The GTIOCm_nA pin output level is retained at start/stop of counting	R/W
8	OAE	GTIOCm_nA Pin Output Enable 0: Pin output is disabled 1: Pin output is enabled	R/W
10:9	OADF[1:0]	GTIOCm_nA Pin Negate Value Setting 0 0: None of the following sources is specified 0 1: GTIOCm_nA pin is placed in the Hi-Z state in response to control for output negation 1 0: GTIOCm_nA pin is set to 0 in response to control for output negation 1 1: GTIOCm_nA pin is set to 1 in response to control for output negation	R/W
12:11	—	These bits are read as 0. The write value should be 0.	R/W
13	NFAEN	GTIOCm_nA Pin Input Noise Filter Enable 0: The noise filter for GTIOCm_nA pin input is disabled 1: The noise filter for GTIOCm_nA pin input is enabled	R/W

Bit	Symbol	Function	R/W
15:14	NFCSA[1:0]	GTIOCm_nA Pin Input Noise Filter Sampling Clock Select 0 0: Core clock 0 1: Core clock/4 1 0: Core clock/16 1 1: Core clock/64	R/W
20:16	GTIOB[4:0]	GTIOCm_nB Pin Function Select See Table 20.6.	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	OBDFLT	GTIOCm_nB Pin Output Value Setting at the Count Stop 0: The GTIOCm_nB pin outputs low when counting is stopped 1: The GTIOCm_nB pin outputs high when counting is stopped	R/W
23	OBHLD	GTIOCm_nB Pin Output Retention at the Start/Stop Count 0: The GTIOCm_nB pin output level at start/stop of counting depends on the register setting 1: The GTIOCm_nB pin output level is retained at start/stop of counting	R/W
24	OBE	GTIOCm_nB Pin Output Enable 0: Pin output is disabled 1: Pin output is enabled	R/W
26:25	OBDF[1:0]	GTIOCm_nB Pin Negate Value Setting 0 0: None of the following sources is specified 0 1: GTIOCm_nB pin is placed in the Hi-Z state in response to control for output negation 1 0: GTIOCm_nB pin is set to 0 in response to control for output negation 1 1: GTIOCm_nB pin is set to 1 in response to control for output negation	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W
29	NFBEN	GTIOCm_nB Pin Input Noise Filter Enable 0: The noise filter for GTIOCm_nB pin input is disabled 1: The noise filter for GTIOCm_nB pin input is enabled	R/W
31:30	NFCSB[1:0]	GTIOCm_nB Pin Input Noise Filter Sampling Clock Select 0 0: Core clock 0 1: Core clock/4 1 0: Core clock/16 1 1: Core clock/64	R/W

Table 20.6 Settings of GTIOA[4:0] bits, GTIOB[4:0] bits

GTIOA/GTIOB bits					Function		
[4]	[3]	[2]	[1]	[0]	bit [4]	bits [3:2]	bits [1:0]
0	0	0	0	0	Initial output is low	Output retained at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	0	1	0			High output at GTCCRA/GTCCRB compare match
0	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	0	1	0	0		Low output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	1	1	0			High output at GTCCRA/GTCCRB compare match
0	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	0	0	0		High output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	0	1	0			High output at GTCCRA/GTCCRB compare match
0	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	1	0	0		Output toggled at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	1	1	0			High output at GTCCRA/GTCCRB compare match
0	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	0	0	0	Initial output is high	Output retained at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Note:
- In sawtooth-wave mode, "end of a cycle" refers to an overflow (the value of the GTCNT counter changing from that of the GTPR register to 0x00000000 in up-counting), an underflow (the value of the GTCNT counter changing from 0x00000000 to that of the GTPR register in down-counting), or counter clearing. It refers to a trough in triangle-wave mode (the value of the GTCNT counter changing from 0x00000000 to 0x00000001).
- When the timing of an end of a cycle and the timing of a GTCCRA/GTCCRB register compare match are the same in a compare-match operation, the bit 3 and bit 2 settings are given priority in sawtooth-wave PWM mode, and the bit 1 and bit 0 settings are given priority in any other mode.
- During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), bit 3 and bit 2 settings become invalid.

20.3.15 GTINTAD : General PWM Timer Interrupt Output Setting Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	GRPABL	GRPABH	GRPDTE	—	—	GRP[1:0]		—	—	—	—	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	GTINTPR[1:0]		GTINTF	GTINTE	GTINTD	GTINTC	GTINTB	GTINTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GTINTA	GTCCRA Register Compare Match / Input Capture Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
1	GTINTB	GTCCRB Register Compare Match / Input Capture Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
2	GTINTC	GTCCRC Register Compare Match Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
3	GTINTD	GTCCRD Register Compare Match Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
4	GTINTE	GTCCRE Register Compare Match Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
5	GTINTF	GTCCRF Register Compare Match Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
7:6	GTINTPR[1:0]	GTPR Register Compare Match Interrupt Enable 0 0: Interrupt request is disabled 0 1: In sawtooth-wave mode, interrupt requests are enabled at overflows. In triangle-wave mode, interrupt requests are enabled at crests 1 0: In sawtooth-wave mode, interrupt requests are enabled at underflows. In triangle-wave mode, interrupt requests are enabled at troughs 1 1: In sawtooth-wave mode, interrupt requests are enabled at both overflows and underflows. In triangle-wave mode, interrupt requests are enabled at both crests and troughs	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	ADTRAUEN	GTADTRA Register Compare Match (Up-Counting) A/D Converter Start Request Enable 0: A/D converter start request is disabled 1: A/D converter start request is enabled	R/W
17	ADTRADEN	GTADTRA Register Compare Match (Down-Counting) A/D Converter Start Request Enable 0: A/D converter start request is disabled 1: A/D converter start request is enabled	R/W
18	ADTRBUEN	GTADTRB Register Compare Match (Up-Counting) A/D Converter Start Request Enable 0: A/D converter start request is disabled 1: A/D converter start request is enabled	R/W

Bit	Symbol	Function	R/W
19	ADTRBDEN	GTADTRB Register Compare Match (Down-Counting) A/D Converter Start Request Enable 0: A/D converter start request is disabled 1: A/D converter start request is enabled	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Stop Group Select Select the group to detect disabling of output (dead-time error or simultaneous driving of outputs to the high or low level) to POEG and to request of disabling of output from POEG. 0 0: Group A is selected 0 1: Group B is selected 1 0: Group C is selected 1 1: Group D is selected	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	GRPDTE	Dead Time Error Output Stop Detection Enable 0: Dead time error output stop detection is disabled 1: Dead time error output stop detection is enabled	R/W
29	GRPABH	Simultaneous High Output Stop Detection Enable (GTIOCm_nA pin and GTIOCm_nB output) 0: Simultaneous high output stop detection is disabled 1: Simultaneous high output stop detection is enabled	R/W
30	GRPABL	Simultaneous Low Output Stop Detection Enable (GTIOCm_nA pin and GTIOCm_nB output) 0: Simultaneous low output stop detection is disabled 1: Simultaneous low output stop detection is enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

20.3.16 GTST : General PWM Timer Status Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	OABLF	OABHF	DTEF	—	—	—	ODF	—	—	—	—	ADTRBDF	ADTRBUF	ADTRADF	ADTRAUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TUCF	—	—	—	—	ITCNT[2:0]			—	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
10:8	ITCNT[2:0]	GPTm_n_OVF/GPTm_n_UDF Interrupt Skipping Count Counter Counter for counting the number of times a timer interrupt has been skipped. The counter is incremented by 1 every time the interrupt source which is selected in GTITC.IVTC[1:0] is generated. Cleared when the interrupt skip count matches the specified count (GTITC.IVTT[2:0]) or when the count operation has stopped.	R
14:11	—	These bits are read as 0. The write value should be 0.	R/W
15	TUCF	Count Direction Flag 0: The GTCNT counter counts downward 1: The GTCNT counter counts upward	R

Bit	Symbol	Function	R/W
16	ADTRAUF	GTADTRA Register Compare Match (Up-Counting) A/D Converter Start Request Flag 0: No GTADTRA register compare match has occurred in up-counting 1: A GTADTRA register compare match has occurred in up-counting	R/W*1
17	ADTRADF	GTADTRA Register Compare Match (Down-Counting) A/D Converter Start Request Flag 0: No GTADTRA register compare match has occurred in down-counting 1: A GTADTRA register compare match has occurred in down-counting	R/W*1
18	ADTRBUF	GTADTRB Register Compare Match (Up-Counting) A/D Converter Start Request Flag 0: No GTADTRB register compare match has occurred in up-counting 1: A GTADTRB register compare match has occurred in up-counting	R/W*1
19	ADTRBDF	GTADTRB Register Compare Match (Down-Counting) A/D Converter Start Request Flag 0: No GTADTRB register compare match has occurred in down-counting 1: A GTADTRB register compare match has occurred in down-counting	R/W*1
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	ODF	Output Stop Request Flag This flag monitors the output stop request for the group selected by GTINTAD.GRP[1:0] bits. 0: No output stop request has occurred 1: An output stop request has occurred	R
27:25	—	These bits are read as 0. The write value should be 0.	R/W
28	DTEF	Dead Time Error Flag This flag returns to 0 when the timer output toggle point after the automatic addition of dead time is back within the period. 0: No dead time error has occurred 1: A dead time error has occurred	R
29	OABHF	Simultaneous High Output Flag When one of pins changes to 0, the flag changes to 0. 0: No simultaneous generation of high output both for the GTIOCm_nA and GTIOCm_nB pins has occurred 1: A simultaneous generation of high output both for the GTIOCm_nA and GTIOCm_nB pins has occurred	R
30	OABLF	Simultaneous Low Output Flag When one of pins changes to 1, the flag changes to 0. 0: No simultaneous generation of low output both for the GTIOCm_nA and GTIOCm_nB pins has occurred 1: A simultaneous generation of low output both for the GTIOCm_nA and GTIOCm_nB pins has occurred	R
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the flag. When clearing the ADTRAUF, ADTRADF, ADTRBUF, or ADTRBDF flag, be sure to write 0 only to the target flag or flags for clearing and to write 1 to the other flags not for clearing.

20.3.17 GTBER : General PWM Timer Buffer Enable Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	DBRTECB	—	DBRTECA	—	—	—	—	BD3	BD2	BD1	BD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BD0	GTCCRA/GTCCRB Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
1	BD1	GTPR Register Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
2	BD2	GTADTRA/GTADTRB Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
3	BD3	GTDVU/GTDVD Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	DBRTECA	GTCCRA Register Double Buffer Repeat Operation Enable 0: GTCCRA register double buffer repeat operation is disabled 1: GTCCRA register double buffer repeat operation is enabled	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	DBRTECB	GTCCRB Register Double Buffer Repeat Operation Enable 0: GTCCRB register double buffer repeat operation is disabled 1: GTCCRB register double buffer repeat operation is enabled	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CCRA[1:0]	GTCCRA Register Buffer Operation 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRA register ↔ GTCCRC register) 1 x: Double buffer operation (GTCCRA register ↔ GTCCRC register ↔ GTCCRD register)	R/W
19:18	CCRB[1:0]	GTCCRB Register Buffer Operation 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRB register ↔ GTCCRE register) 1 x: Double buffer operation (GTCCRB register ↔ GTCCRE register ↔ GTCCRF register)	R/W
21:20	PR[1:0]	GTPR Register Buffer Operation 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTPBR register → GTPR register) 1 x: Double buffer operation (GTPDBR register → GTPBR register → GTPR register)	R/W
22	CCRSWT	GTCCRA and GTCCRB Registers Forcible Buffer Operation Writing 1 to this bit forcibly performs buffer transfer of GTCCRA and GTCCRB registers. This bit automatically returns to 0 after the writing of 1. This bit is read as 0.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
25:24	ADTTA[1:0]	GTADTRA Register Buffer Transfer Timing Select 0 0: No transfer 0 1: (Triangle waves) Transfer at crest (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing 1 0: (Triangle waves) Transfer at trough (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing 1 1: (Triangle waves) Transfer at both crest and trough (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing	R/W
26	ADTDA	GTADTRA Register Double Buffer Operation 0: Single buffer operation (GTADTBRA register → GTADTRA register) 1: Double buffer operation (GTADTDBRA register → GTADTBRA register → GTADTRA register)	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
29:28	ADTTB[1:0]	GTADTRB Register Buffer Transfer Timing Select 0 0: No transfer 0 1: (Triangle waves) Transfer at crest (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing 1 0: (Triangle waves) Transfer at trough (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing 1 1: (Triangle waves) Transfer at both crest and trough (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing	R/W
30	ADTDB	GTADTRB Register Double Buffer Operation 0: Single buffer operation (GTADTBRB register → GTADTRB register) 1: Double buffer operation (GTADTDBRB register → GTADTBRB register → GTADTRB register)	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: The buffer operation mode is fixed in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough).

20.3.18 GTITC : General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x44

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADTBL	—	ADTAL	—	IVTT[2:0]			IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

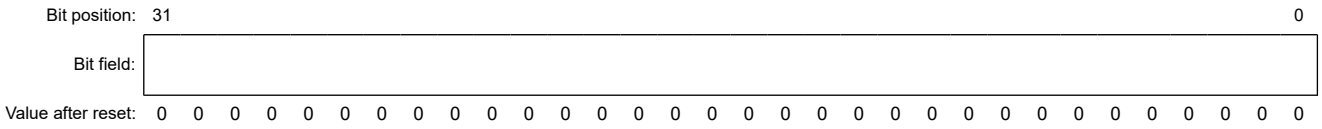
Bit	Symbol	Function	R/W
0	ITLA	GTCCRA Register Compare Match / Input Capture Interrupt Link 0: Not linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function 1: Linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function	R/W
1	ITLB	GTCCRB Register Compare Match / Input Capture Interrupt Link 0: Not linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function 1: Linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function	R/W
2	ITLC	GTCCRC Register Compare Match Interrupt Link 0: Not linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function 1: Linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function	R/W
3	ITLD	GTCCRD Register Compare Match Interrupt Link 0: Not linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function 1: Linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function	R/W
4	ITLE	GTCCRE Register Compare Match Interrupt Link 0: Not linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function 1: Linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function	R/W
5	ITLF	GTCCRF Register Compare Match Interrupt Link 0: Not linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function 1: Linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function	R/W

Bit	Symbol	Function	R/W
7:6	IVTC[1:0]	GPTm_n_OVF/GPTm_n_UDF Interrupt Skipping Function Select 0 0: Skipping is not performed 0 1: Both overflow and underflow for sawtooth waves and crest for triangle waves are counted and skipped 1 0: Both overflow and underflow for sawtooth waves and trough for triangle waves are counted and skipped 1 1: Both overflow and underflow for sawtooth waves and both crest and trough for triangle waves are counted and skipped	R/W
10:8	IVTT[2:0]	GPTm_n_OVF/GPTm_n_UDF Interrupt Skipping Count Select 0 0 0: Skipping is not performed 0 0 1: Skipping count of 1 0 1 0: Skipping count of 2 0 1 1: Skipping count of 3 1 0 0: Skipping count of 4 1 0 1: Skipping count of 5 1 1 0: Skipping count of 6 1 1 1: Skipping count of 7	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	ADTAL	GTADTRA Register A/D Converter Start Request Link 0: Not linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function 1: Linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	ADTBL	GTADTRB Register A/D Converter Start Request Link 0: Not linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function 1: Linked with GPTm_n_OVF/GPTm_n_UDF interrupt skipping function	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

20.3.19 GTCNT : General PWM Timer Counter

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x48

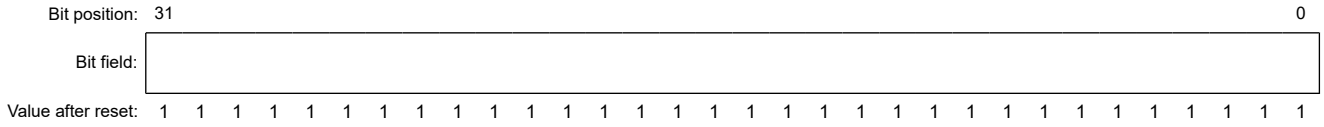


Bit	Symbol	Function	R/W
31:0	n/a	The GTCNT counter is a 32-bit readable/writable counter. There is one GTCNT counter for each channel. Writing is available only when counting is stopped, and thus writing is disabled while counting is in progress (when CST = 1). Access in 8-bit or 16-bit units to the GTCNT counter is prohibited, and it should be accessed in 32-bit units. Set the range of the GTCNT counter within the range of 0 ≤ GTCNT counter ≤ GTPR register.	R/W

20.3.20 GTCCRm : General PWM Timer Compare Capture Register m (m = A to F)

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x4C (GTCCRA), 0x50 (GTCCRB), 0x54 (GTCCRC)
0x58 (GTCCRE), 0x5C (GTCCRD), 0x60 (GTCCRF)

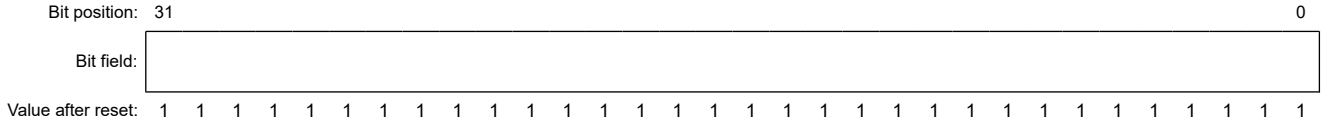


Bit	Symbol	Function	R/W
31:0	n/a	The GTCCRm register is a 32-bit readable/writable register. Access in 8-bit or 16-bit units to the GTCCRm register is prohibited, and it should be accessed in 32-bit units. The GTCCRA and GTCCRB registers are used for both output compare and input capture. The GTCCRC and GTCCRE registers are compare match registers, and can also function as buffer registers for the GTCCRA and GTCCRB registers. The GTCCRD and GTCCRF registers are compare match registers, and can also function as buffer registers for the GTCCRC and GTCCRE registers (double buffer registers for the GTCCRA and GTCCRB registers).	R/W

20.3.21 GTPR : General PWM Timer Cycle Setting Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x64

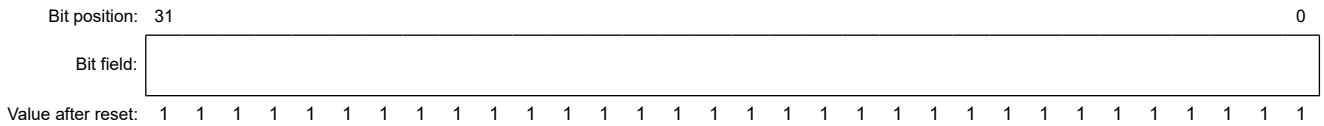


Bit	Symbol	Function	R/W
31:0	n/a	The GTPR register is a 32-bit readable/writable register that sets the maximum count value of the GTCNT counter. Access in 8-bit or 16-bit units to the GTPR register is prohibited, and it should be accessed in 32-bit units. In sawtooth-wave mode, the value of (GTPR register + 1) is the count period. In triangle-wave mode, the value of (GTPR register value × 2) is the count period.	R/W

20.3.22 GTPBR : General PWM Timer Cycle Setting Buffer Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x68



Bit	Symbol	Function	R/W
31:0	n/a	The GTPBR register is a 32-bit readable/writable register that functions as a buffer register for the GTPR register. Access in 8-bit or 16-bit units to the GTPBR register is prohibited, and it should be accessed in 32-bit units.	R/W

20.3.23 GTPDBR : General PWM Timer Cycle Setting Double-Buffer Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x6C

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTPDBR register is a 32-bit readable/writable register that functions as a buffer register for the GTPBR register (a double buffer register for the GTPR register). Access in 8-bit or 16-bit units to the GTPDBR register is prohibited, and it should be accessed in 32-bit units.	R/W

20.3.24 GTADTRm : A/D Converter Start Request Timing Register m (m = A, B)

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)

Offset address: 0x70 (GTADTRA)
0x7C (GTADTRB)

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTADTRm register is a 32-bit readable/writable register that sets the timing of A/D converter start request generation. Access in 8-bit or 16-bit units to the GTADTRm register is prohibited, and it should be accessed in 32-bit units. When the GTADTRm register value matches the GTCNT counter value, an A/D converter start request is generated.	R/W

Unit 10 (SAFETY) does not support A/D converter start request.

20.3.25 GTADTBRm : A/D Converter Start Request Timing Buffer Register m (m = A, B)

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)

Offset address: 0x74 (GTADTBRA)
0x80 (GTADTBRB)

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

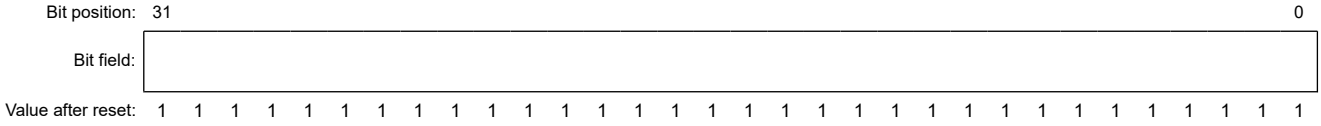
Bit	Symbol	Function	R/W
31:0	n/a	The GTADTBRm register is a 32-bit readable/writable register that functions as buffer register for the GTADTRm register. Access in 8-bit or 16-bit units to the GTADTBRm register is prohibited, and it should be accessed in 32-bit units.	R/W

Unit 10 (SAFETY) does not support A/D converter start request.

20.3.26 GTADTDBRm : A/D Converter Start Request Timing Double-Buffer Register m (m = A, B)

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)

Offset address: 0x78 (GTADTDBRA)
0x84 (GTADTDBRB)



Bit	Symbol	Function	R/W
31:0	n/a	The GTADTDBRm register is a 32-bit readable/writable register that functions as a buffer register for the GTADTBRm register (double buffer register for the GTADTRm register). Access in 8-bit or 16-bit units to the GTADTDBRm register is prohibited, and it should be accessed in 32-bit units.	R/W

Unit 10 (SAFETY) does not support A/D converter start request.

20.3.27 GTDTCR : General PWM Timer Dead Time Control Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x88

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TDFER	—	—	TDBDE	TDBUE	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDE	Negative-Phase Waveform Setting 0: GTCCRB register is set without using GTDVU and GTDVD registers 1: GTDVU and GTDVD registers are used to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB register	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TDBUE	GTDVU Register Buffer Operation Enable 0: GTDVU register buffer operation is disabled 1: GTDVU register buffer operation is enabled	R/W

Bit	Symbol	Function	R/W
5	TDBDE	GTDVD Register Buffer Operation Enable 0: GTDVD register buffer operation is disabled 1: GTDVD register buffer operation is enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	TDFER	GTDVD Register Setting 0: GTDVU and GTDVD registers are set separately 1: The value written to GTDVU register is automatically set to GTDVD register	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The GTDTCR register enables automatic setting of a compare match value for negative-phase waveform with dead time. This register is invalid during the event count operation.

20.3.28 GTDVm : General PWM Timer Dead Time Value Register m (m = U, D)

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x8C (GTDVU)
0x90 (GTDVD)

Bit position:	31																													0
Bit field:																														
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTDVm register is a 32-bit readable/writable register that sets the dead time for generating PWM waveforms with dead time. Access in 8-bit or 16-bit units to the GTDVm register is prohibited, and it should be accessed in 32-bit units. The GTDVU register is used for up-counting. The GTDVD register is used for down-counting. The setting which the value for the GTDVm register is equal to or larger than the value for the GTPR register is prohibited. In addition, when using the automatic dead time setting function, do not set a value that makes a change point of the waveform exceeding the count period. The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. When using the GTDVm register, writing to the GTCCRB register is prohibited. When setting this register to 0, waveforms without dead time are output. When the GTDTCR.TDFER bit is 1, the writing to the GTDVD register has no effect. At this time, when the GTDVD register is read, the value for the GTDVU register is read.	R/W

20.3.29 GTDBm : General PWM Timer Dead Time Value Buffer Register m (m = U, D)

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x94 (GTDBU)
0x98 (GTDBD)

Bit position:	31																													0
Bit field:																														
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTDBm register is a 32-bit readable/writable register that functions as a buffer register for the GTDVm register. Access in 8-bit or 16-bit units to the GTDBm register is prohibited, and it should be accessed in 32-bit units.	R/W

20.3.30 GTSOS : General PWM Timer Output Protection Function Status Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0x9C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOS[1:0]	
Value after reset:	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SOS[1:0]	Output Protection Function Status 0 0: Normal operation 0 1: Protected state (GTCCRA = 0 is set during transfer at trough or crest) 1 0: Protected state (GTCCRA ≥ GTPR is set during transfer at trough) 1 1: Protected state (GTCCRA ≥ GTPR is set during transfer at crest)	R
31:2	—	These bits are read as 0.	R

The GTSOS register is a status register that indicates the status of the output protection function. The output protection function is enabled only when the dead time is automatically set (the GTDTCR.TDE bit is 1) in triangle-wave mode.

20.3.31 GTSOTR : General PWM Timer Output Protection Function Temporary Release Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0xA0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOTR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SOTR	Output Protection Function Temporary Release When the output is protected, the GTIOCm_nB pin output is temporarily released from the protected state. The protected state can be released only for the case of GTSOS.SOS[1:0] bits are 10b (protected state in which GTCCRA ≥ GTPR has occurred during transfer at trough). 0: Protected state is not released 1: Protected state is released	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

20.3.32 GTADSMR : General PWM Timer A/D Conversion Start Request Signal Monitoring Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)

Offset address: 0xA4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ADSMEN1	—	—	—	—	—	—	ADSMS1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADSMEN0	—	—	—	—	—	—	ADSMS0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADSMS0[1:0]	A/D Conversion Start Request Signal Monitor 0 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	ADSMEN0	A/D Conversion Start Request Signal Monitor 0 Output Enabling 0: Output of A/D conversion start request signal monitor 0 is disabled 1: Output of A/D conversion start request signal monitor 0 is enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
17:16	ADSMS1[1:0]	A/D Conversion Start Request Signal Monitor 1 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	ADSMEN1	A/D Conversion Start Request Signal Monitor 1 Output Enabling 0: Output of A/D conversion start request signal monitor 1 is disabled 1: Output of A/D conversion start request signal monitor 1 is enabled	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Unit 10 (SAFETY) does not support A/D conversion start request.

The GTADSMR register is used to control monitors for the A/D conversion start request signal that is synchronized with a frame period.

When the output of the same A/D converter start request signal monitoring output is enabled for multiple channels, ORed signals will be output.

20.3.33 GTEITC : General PWM Timer Extended Interrupt Skipping Counter Control Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0xA8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EITCNT2[3:0]				EITCNT2IV[3:0]				EIVTT2[3:0]				—	—	EIVTC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EITCNT1[3:0]				—	—	—	—	EIVTT1[3:0]				—	—	EIVTC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	EIVTC1[1:0]	Extended Interrupt Skipping Counter 1 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in sawtooth-wave mode and counting crests in triangle-wave mode 1 0: Counting both at overflow or underflow in sawtooth-wave mode and counting troughs in triangle-wave mode 1 1: Counting both at overflow or underflow in sawtooth-wave mode and counting both crests and troughs in triangle-wave mode	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	EIVTT1[3:0]	Extended Interrupt Skipping 1 Skipping Count Setting Skipping count for the extended interrupt skipping 1	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
15:12	EITCNT1[3:0]	Extended Interrupt Skipping Counter 1 The counting is incremented by 1 every time a count source selected by the EIVTC1[1:0] bits is generated. When the count source is generated while the EIVTT1[3:0] bits match, these bits are cleared.	R
17:16	EIVTC2[1:0]	Extended Interrupt Skipping Counter 2 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in sawtooth-wave mode and counting crests in triangle-wave mode 1 0: Counting both at overflow or underflow in sawtooth-wave mode and counting troughs in triangle-wave mode 1 1: Counting both at overflow or underflow in sawtooth-wave mode and counting both crests and troughs in triangle-wave mode	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
23:20	EIVTT2[3:0]	Extended Interrupt Skipping 2 Skipping Count Setting	R/W
27:24	EITCNT2IV[3:0]	Extended Interrupt Skipping Counter 2 Initial Value The EITCNT2IV[3:0] bits are only writable when a value other than 00b is written to the EIVTC2[1:0] bits that are 00b.	R/W
31:28	EITCNT2[3:0]	Extended Interrupt Skipping Counter 2 The counting is incremented by 1 every time a count source selected by the EIVTC2[1:0] bits is generated. When the count source is generated while the EIVTT2[3:0] bits match, these bits are cleared.	R

20.3.34 GTEITLI1 : General PWM Timer Extended Interrupt Skipping Setting Register 1

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0xAC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	EITLU[2:0]			—	EITLV[2:0]			—	EITLF[2:0]			—	EITLE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	EITLD[2:0]			—	EITLC[2:0]			—	EITLB[2:0]			—	EITLA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	EITLA[2:0]	GTCCRA Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 20.7.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EITLB[2:0]	GTCCRB Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 20.7.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	EITLC[2:0]	GTCCRC Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 20.7.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	EITLD[2:0]	GTCCRD Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 20.7.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	EITLE[2:0]	GTCCRE Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 20.7.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	EITLF[2:0]	GTCCRF Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 20.7.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	EITLV[2:0]	Overflow Interrupt Extended Skipping Function Select See Table 20.7.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	EITLU[2:0]	Underflow Interrupt Extended Skipping Function Select See Table 20.7.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTEITLI1 register sets the extended skipping function for interrupts such as compare match/input capture, overflow, and underflow.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register.

The setting is invalid during the event count operation.

Table 20.7 Setting the function select for the GTEITLI1 register

EITLy[2:0] bits	Function
000b	Do not perform an extended interrupt skipping
001b	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than 0. An interrupt is output in the period of EITCNT1[3:0] bits = 0000b.
010b	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than 0. An interrupt is output in the period of EITCNT2[3:0] bits = 0000b.
011b	Skip an interrupt in the period the value for the extended interrupt skipping counter 1 or 2 is other than 0. An interrupt is output in the period of EITCNT1[3:0] bits = 0000b and EITCNT2[3:0] bits = 0000b.
100b	Do not set this value
101b	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count. An interrupt is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits.
110b	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count. An interrupt is output in the period of EITCNT2[3:0] bits = EIVTT2[3:0] bits.
111b	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count. An interrupt is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits and EITCNT2[3:0] bits = EIVTT2[3:0] bits.

Note:
- y = A, B, C, D, E, F, V, U; k = 1, 2
- When the intended skipping counter is set as not to count (EIVTCk[1:0] bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.
- When the EITLy[2:0] bits are set to 011b or 111b, or when one of the skipping counters is set as not to count, skipping is not performed.

20.3.35 GTEITLI2 : General PWM Timer Extended Interrupt Skipping Setting Register 2

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0xB0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	EADTBL[2:0]			—	EADTAL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	EADTAL[2:0]	GTADTRA A/D Converter Start Request Extended Skipping Function Select See Table 20.8.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EADTBL[2:0]	GTADTRB A/D Converter Start Request Extended Skipping Function Select See Table 20.8.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The GTEITLI2 register sets the extended skipping function for A/D converter start requests.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register.

The setting is invalid during the event count operation.

Table 20.8 Setting the function select for the GTEITLI2 register

EADTmL[2:0] bits	Function
000b	Do not perform an extended interrupt skipping
001b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 1 is other than 0. An A/D converter start request is output in the period of EITCNT1[3:0] bits = 0000b.
010b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 2 is other than 0. An A/D converter start request is output in the period of EITCNT2[3:0] bits = 0000b.
011b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0. An A/D converter start request is output in the period of EITCNT1[3:0] bits = 0000b and EITCNT2[3:0] bits = 0000b.
100b	Do not set this value
101b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count. An A/D converter start request is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits.
110b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count. An A/D converter start request is output in the period of EITCNT2[3:0] bits = EIVTT2[3:0] bits.
111b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count. An A/D converter start request is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits and EITCNT2[3:0] bits = EIVTT2[3:0] bits.

Note:
- m = A, B; k = 1, 2
- When the intended skipping counter is set as not to count (EIVTCk[1:0] bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.
- When the EADTmL[2:0] bits are set to 011b or 111b, or when one of the skipping counters is set as not to count, skipping is not performed.

20.3.36 GTEITLB : General PWM Timer Extended Buffer Transfer Skipping Setting Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0xB4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	EBTLDVD[2:0]			—	EBTLDVU[2:0]			—	EBTLADB[2:0]			—	EBTLADA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	EBTLPR[2:0]			—	EBTLCB[2:0]			—	EBTLCA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	EBTLCA[2:0]	GTCCRA Register Buffer Transfer Extended Skipping Function Select See Table 20.9.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EBTLCB[2:0]	GTCCRB Register Buffer Transfer Extended Skipping Function Select See Table 20.9.	R/W

Bit	Symbol	Function	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	EBTLPR[2:0]	GTPR Register Buffer Transfer Extended Skipping Function Select See Table 20.9.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	EBTLADA[2:0]	GTADTRA Register Buffer Transfer Extended Skipping Function Select See Table 20.9.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	EBTLADB[2:0]	GTADTRB Register Buffer Transfer Extended Skipping Function Select See Table 20.9.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	EBTLDVU[2:0]	GTDVU Register Buffer Transfer Extended Skipping Function Select See Table 20.9.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	EBTLDVD[2:0]	GTDVD Register Buffer Transfer Extended Skipping Function Select See Table 20.9.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTEITLB register sets the extended skipping function for buffer transfers.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register.

The setting is invalid during the event count operation.

Table 20.9 Setting the function select for the GTEITLB register

EBTLx[2:0] bits	Function
000b	Do not perform an extended interrupt skipping
001b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than 0. A buffer transfer is output in the period of EITCNT1[3:0] bits = 0000b.
010b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than 0. A buffer transfer is output in the period of EITCNT2[3:0] bits = 0000b.
011b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0. A buffer transfer is output in the period of EITCNT1[3:0] bits = 0000b and EITCNT2[3:0] bits = 0000b.
100b	Do not set this value
101b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count. A buffer transfer is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits.
110b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count. A buffer transfer is output in the period of EITCNT2[3:0] bits = EIVTT2[3:0] bits.
111b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count. A buffer transfer is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits and EITCNT2[3:0] bits = EIVTT2[3:0] bits.

Note:
- x = CA, CB, PR, ADA, ADB, DVU, DVD; k = 1, 2
- When the intended skipping counter is set as not to count (EIVTCk[1:0] bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.
- When the EBTLx[2:0] bits are set to 011b or 111b, or when one of the skipping counters is set as not to count, skipping is not performed.

20.3.37 GTSECSR : General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0xD0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	SECSEL6	SECSEL5	SECSEL4	SECSEL3	SECSEL2	SECSEL1	SECSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SECSEL0	Channel 0 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
1	SECSEL1	Channel 1 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
2	SECSEL2	Channel 2 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
3	SECSEL3	Channel 3 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
4	SECSEL4*1	Channel 4 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
5	SECSEL5*1 *2	Channel 5 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
6	SECSEL6*1 *2	Channel 6 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to 08 (LLPP0 and LLPP1)

The GTSECSR register selects an intended channel for updating an operation enable bit by the GTSECR register.

A bit position for the GTSECSR register indicates a channel number. The GTSECSR register of each channel is a common register, and writing 1 to a bit in the GTSECSR register in any channel and updating it changes a channel, related to the position of the bit written with 1 by the GTSECSR register, to be simultaneously controlled of the operation enable bit by the GTSECR register.

20.3.38 GTSECR : General PWM Timer Operation Enable Bit Simultaneous Control Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0xD4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SBDDD	SBDAD	SBDPD	SBDCD	—	—	—	—	SBDDE	SBDAE	SBDPE	SBDCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SBDCE	GTCCR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTCCR buffer operations 1: Enable GTCCR register buffer operations simultaneously	R/W
1	SBDPE	GTPR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTPR buffer operations 1: Enable GTPR register buffer operations simultaneously	R/W
2	SBDAE	GTADTR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTADTR buffer operations 1: Enable GTADTR register buffer operations simultaneously	R/W
3	SBDDE	GTDV Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTDV buffer operations 1: Enable GTDV register buffer operations simultaneously	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	SBDCD	GTCCR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTCCR buffer operations 1: Disable GTCCR register buffer operations simultaneously	R/W
9	SBDPD	GTPR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTPR buffer operations 1: Disable GTPR register buffer operations simultaneously	R/W
10	SBDAD	GTADTR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTADTR buffer operations 1: Disable GTADTR register buffer operations simultaneously	R/W
11	SBDDD	GTDV Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTDV buffer operations 1: Disable GTDV register buffer operations simultaneously	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The GTSECR register simultaneously updates the value for operation enable bits of a channel set by the GTSECSR register.

Writing 1 to a bit in the GTSECR register in any channel and updating it updates an operation enable bit for all channels, related to the position of the bit written with 1 by the all GTSECSR registers. It is reflected in the GTBER.BDn (n = 0 to 3) bits of simultaneous channels. Setting enable and disable bits for the same operation enable bit to 1 in the GTSECR is prohibited.

A bit written to 1 is automatically cleared. When the GTSECR is read, 0 is read.

Access in 8-bit or 16-bit units to the GTSECR register is prohibited, and it should be accessed in 32-bit units.

20.3.39 GTSWSR : General PWM Timer Switch Source Select Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0xD8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	CSEL CH	WSEL CG	WSEL CF	WSEL CE	WSEL CD	WSEL CC	WSEL CB	WSEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	WSGTRGD[1:0]		WSGTRGC[1:0]		WSGTRGB[1:0]		WSGTRGA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	WSGTRGA[1:0]	GTETRGA Signal Edge Select to Switch Counter (GTETRGSA Signal for SAFTY) 0 0: The GTETRGA signal is not used as a trigger to switch the counter 0 1: The counter is switched at a rising edge of the GTETRGA signal 1 0: The counter is switched at a falling edge of the GTETRGA signal 1 1: The counter is switched at both edges of the GTETRGA signal	R/W
3:2	WSGTRGB[1:0]	GTETRGB Signal Edge Select to Switch Counter (GTETRGSB Signal for SAFTY) 0 0: The GTETRGB signal is not used as a trigger to switch the counter 0 1: The counter is switched at a rising edge of the GTETRGB signal 1 0: The counter is switched at a falling edge of the GTETRGB signal 1 1: The counter is switched at both edges of the GTETRGB signal	R/W
5:4	WSGTRGC[1:0]	GTETRGC Signal Edge Select to Switch Counter (GTIOC10_0A output Signal for SAFTY) 0 0: The GTETRGC signal is not used as a trigger to switch the counter 0 1: The counter is switched at a rising edge of the GTETRGC signal 1 0: The counter is switched at a falling edge of the GTETRGC signal 1 1: The counter is switched at both edges of the GTETRGC signal	R/W
7:6	WSGTRGD[1:0]	GTETRGD Signal Edge Select to Switch Counter (GTIOC10_1A output Signal for SAFTY) 0 0: The GTETRGD signal is not used as a trigger to switch the counter 0 1: The counter is switched at a rising edge of the GTETRGD signal 1 0: The counter is switched at a falling edge of the GTETRGD signal 1 1: The counter is switched at both edges of the GTETRGD signal	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	WSELCA*1	Event Source Counter Switch Enable 0: Disables count switch by the event A input 1: Enables count switch by the event A input	R/W
17	WSELCB*1	Event Source Counter Switch Enable 0: Disables count switch by the event B input 1: Enables count switch by the event B input	R/W
18	WSELCC*1	Event Source Counter Switch Enable 0: Disables count switch by the event C input 1: Enables count switch by the event C input	R/W
19	WSELCD*1	Event Source Counter Switch Enable 0: Disables count switch by the event D input 1: Enables count switch by the event D input	R/W
20	WSELCE*1	Event Source Counter Switch Enable 0: Disables count switch by the event E input 1: Enables count switch by the event E input	R/W
21	WSELCF*1	Event Source Counter Switch Enable 0: Disables count switch by the event F input 1: Enables count switch by the event F input	R/W

Bit	Symbol	Function	R/W
22	WSELCG*1 *2	Event Source Counter Switch Enable 0: Disables count switch by the event G input 1: Enables count switch by the event G input	R/W
23	CSELCH*1 *2	Event Source Counter Switch Enable 0: Disables count switch by the event H input 1: Enables count switch by the event H input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Not available for unit 10 (SAFETY)
Note 2. Not available for unit 00 to unit 08 (LLPP0 and LLPP1)

The GTSWSR register sets a counter switch source for the GTCNT counter. Count switch can be executed whether the counter is running (GTCR.CST = 1) or stopped (GTCR.CST = 0). Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD, (GTETRGSA and GTETRGSB for SAFETY) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

20.3.40 GTSWOS : General PWM Timer Switch Offset Setting Register

Base address: GPT0h_i = 0x9000_2000 + 0x1000 × h + 0x0100 × i (h = 0 to 5, i = 0 to 4)
GPT0j_k = 0x9010_0000 + 0x1000 × (j - 6) + 0x0100 × k (j = 6 to 8, k = 0 to 4)
GPT09_h = 0x8000_0000 + 0x0100 × h (h = 0 to 6)
GPT10_y = 0x8100_0000 + 0x0100 × y (y = 0 to 3)

Offset address: 0xDC

Bit position:	31																													0
Bit field:																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The GTSWOS register is a readable/writable register that sets the offset value of GTCNT counter which is reloaded by count switch operation. Access in 8-bit or 16-bit units to the GTSWOS register is prohibited, and it should be accessed in 32-bit units. The GTSWOS register is writable even while the GTCNT counter operation. If the GTSWOS register writing and the event input of count switch occur at the same time, the GTSWOS register setting before writing is reloaded to GTCNT counter. Write the GTSWOS register earlier than the event input of count switch to ensure that the written GTSWOS register value is reloaded by the count switch operation.

20.3.41 GTIOCSEL : General PWM Timer Input Capture Signal Select Register

Base address: GPT_IC = 0x8029_0000

Offset address: 0x0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	GT08_ ISEL	GT07_ ISEL	GT06_ ISEL	GT05_ ISEL	GT04_ ISEL	GT03_ ISEL	GT02_ ISEL	GT01_ ISEL	GT00_ ISEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GT00_ISEL	Input Capture Signal Select for GPT00_4 channel 0: GTIOC00_4A / GTIOC00_4B input signals are selected 1: GTIOC00_3A / GTIOC00_3B input signals are selected	R/W
1	GT01_ISEL	Input Capture Signal Select for GPT01_4 channel 0: GTIOC01_4A / GTIOC01_4B input signals are selected 1: GTIOC01_3A / GTIOC01_3B input signals are selected	R/W

Bit	Symbol	Function	R/W
2	GT02_ISEL	Input Capture Signal Select for GPT02_4 channel 0: GTIOC02_4A / GTIOC02_4B input signals are selected 1: GTIOC02_3A / GTIOC02_3B input signals are selected	R/W
3	GT03_ISEL	Input Capture Signal Select for GPT03_4 channel 0: GTIOC03_4A / GTIOC03_4B input signals are selected 1: GTIOC03_3A / GTIOC03_3B input signals are selected	R/W
4	GT04_ISEL	Input Capture Signal Select for GPT04_4 channel 0: GTIOC04_4A / GTIOC04_4B input signals are selected 1: GTIOC04_3A / GTIOC04_3B input signals are selected	R/W
5	GT05_ISEL	Input Capture Signal Select for GPT05_4 channel 0: GTIOC05_4A / GTIOC05_4B input signals are selected 1: GTIOC05_3A / GTIOC05_3B input signals are selected	R/W
6	GT06_ISEL	Input Capture Signal Select for GPT06_4 channel 0: GTIOC06_4A / GTIOC06_4B input signals are selected 1: GTIOC06_3A / GTIOC06_3B input signals are selected	R/W
7	GT07_ISEL	Input Capture Signal Select for GPT07_4 channel 0: GTIOC07_4A / GTIOC07_4B input signals are selected 1: GTIOC07_3A / GTIOC07_3B input signals are selected	R/W
8	GT08_ISEL	Input Capture Signal Select for GPT08_4 channel 0: GTIOC08_4A / GTIOC08_4B input signals are selected 1: GTIOC08_3A / GTIOC08_3B input signals are selected	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The GTIOCSEL register specifies input capture signals for channel 4 of GPTm units (m = 00 to 08).

20.4 Operation

20.4.1 Basic Operation

The timer in each channel performs periodic count operation by count clock or hardware source. The GTCNT counter performs up-counting, or down-counting. The timer period is controlled by the GTPR register. When the GTCNT counter value matches the values for GTCCRA or GTCCRB register, the GTIOCm_nA or GTIOCm_nB pin outputs ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)) can be changed respectively. Also, the GTCCRA or GTCCRB register can be used as the input capture register by hardware source. The GTCCRC and GTCCRD registers can be performed as buffer registers for GTCCRA register, and the GTCCRE and GTCCRF registers can be performed as buffer registers for GTCCRB register.

20.4.1.1 Counter Operation

(1) Count Start/Count Stop

The counter in each channel starts counting operation when the corresponding GTCR.CST bit is set to 1, and stops counting when the bit is set to 0.

CST bit value can be changed by the following sources:

- Writing to the GTCR register
- Writing 1 to the bit corresponding to the channel number for the GTSTR register while the GTSSR.CSTRT bit is 1
- Writing 1 to the bit corresponding to the channel number for the GTSTP register while the GTPSR.CSTOP bit is 1
- Hardware source specified by the GTSSR register
- Hardware source specified by the GTPSR register

(2) Periodic Count Operation (In Up-Counting by the Count Clock)

The counter in each channel starts up-counting when the corresponding CST bit is set to 1 while the GTUPSR register and the GTDNSR register are 0x00000000. When the value of the GTCNT counter changes from that of the GTPR register

to 0x00000000 (an overflow) while the GTINTAD.GTINTPR[0] bit is 1, a GPTm_n_OVF interrupt request is also issued. After GTCNT counter overflows, up-counting is resumed from 0x00000000.

Figure 20.2 shows an example of periodic count operation in up-counting by count clock.

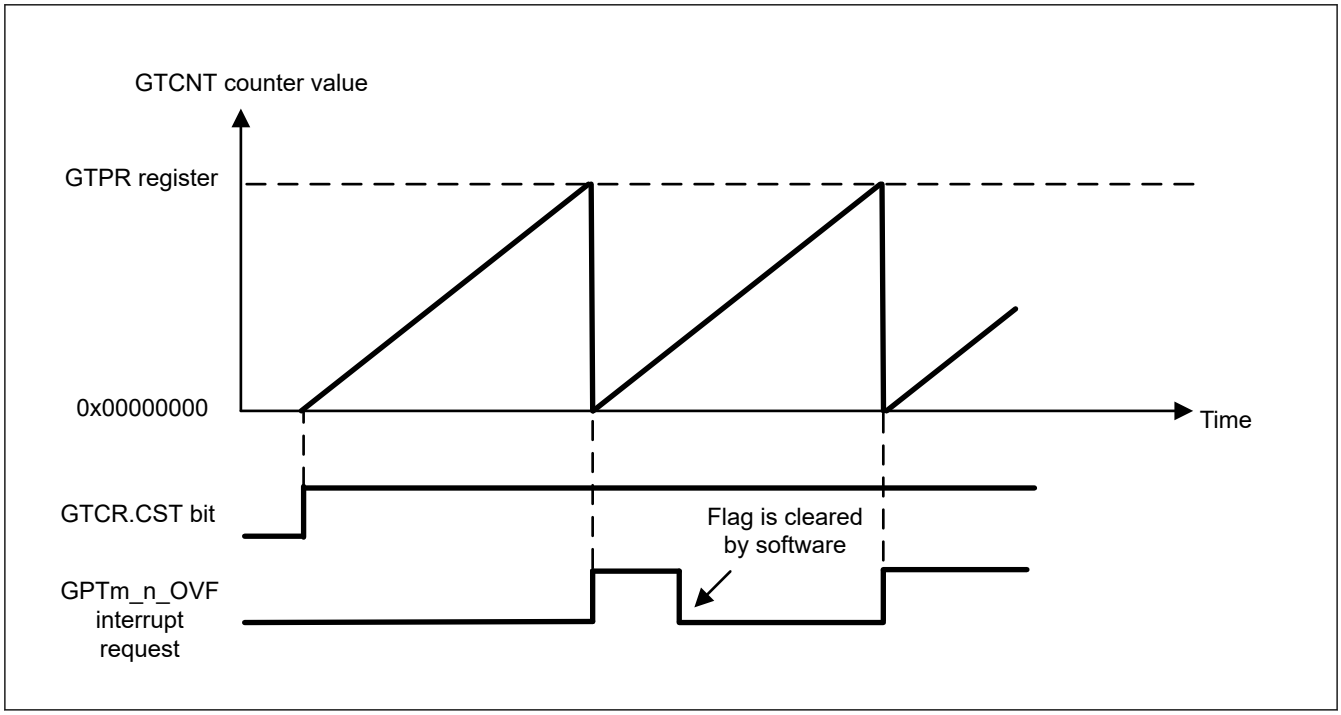


Figure 20.2 Example of periodic count operation (in up-counting by the count clock)

The following example shows the setting for periodic count operation in up-counting by count clock.

1. Set operating mode
   Set the operating mode with GTCR.MD[2:0] bits. In Figure 20.2, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction with the GTUDDTYC register. In Figure 20.2, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter. In Figure 20.2, 0x00000000 is set.
6. Start count operation
   Set the GTCR.CST bit to 1 to start count operation.

(3) Periodic Count Operation (In Down-Counting by the Count Clock)

The counter in each channel can start down-counting when the corresponding GTUDDTYC.UD bit is set in a state of the GTUPSR register and the GTDNSR register as 0x00000000. When the value of the GTCNT counter changes from 0x00000000 to that of the GTPR register (an underflow) while the GTINTAD.GTINTPR[1] bit is 1, a GPTm_n_UDF interrupt request is also issued. After GTCNT counter underflows, down-counting is resumed from GTPR register value.

Figure 20.3 shows an example of periodic count operation in down-counting by count clock.

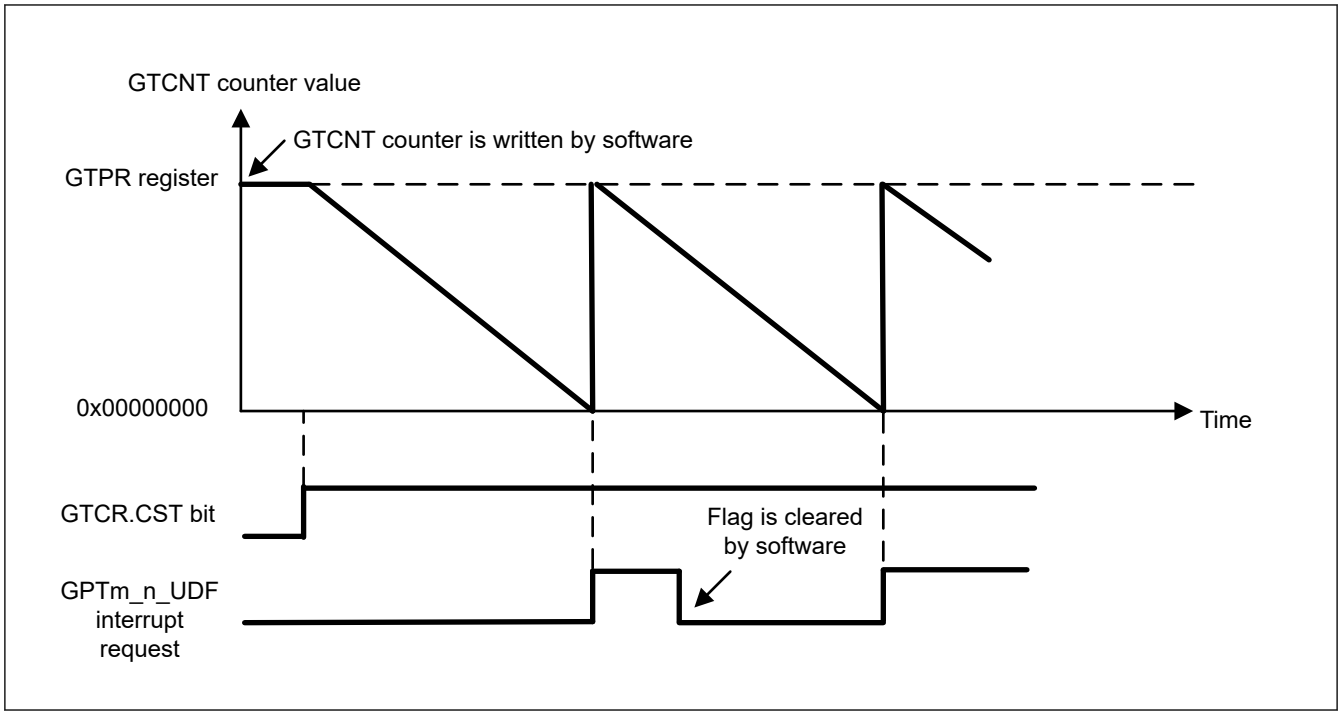


Figure 20.3 Example of periodic count operation (in down-counting by the count clock)

The following example shows the setting for periodic count operation in down-counting by count clock.

1. Set operating mode
   Set the operating mode with GTCR.MD[2:0] bits. In Figure 20.3, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction with the GTUDDTYC register. In Figure 20.3, the lower 2 bits of the GTUDDTYC register is set to 10b and then to 00b for down-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter. In Figure 20.3, the GTPR register value is set.
6. Start count operation
   Set the GTCR.CST bit to 1 to start count operation.

(4) Event Count Operation (In Up-Counting by Hardware Source)

The counter in each channel can start up-counting by hardware source by setting the GTUPSR register. When the GTUPSR register is set, the count clock selected by the GTCR.TPCS[3:0] bits and the count direction set by the GTUDDTYC.UD bit are invalid. If the hardware source in up-counting and the hardware source in down-counting are generated at the same time, the GTCNT counter value does not change.

Operation due to overflow in up-counting by hardware source is the same as the periodic count operation by the count clock.

To perform up-counting by hardware source, start count operation by setting the GTCR.CST bit to 1. At the start, the count operation is synchronized with the count clock selected by GTCR.TPCS[3:0] bits, set the CST bit to 1, and wait until the first cycle of the count clock elapses to start the up-counting operation. To start up-counting after 1 PCLKGPTL (LLPP)/PCLKM (NONSAFETY, SAFETY) from setting the CST bit to 1, set the GTCR.TPCS[3:0] bits to 0000b.

Figure 20.4 shows an example of up-counting operation by hardware source.

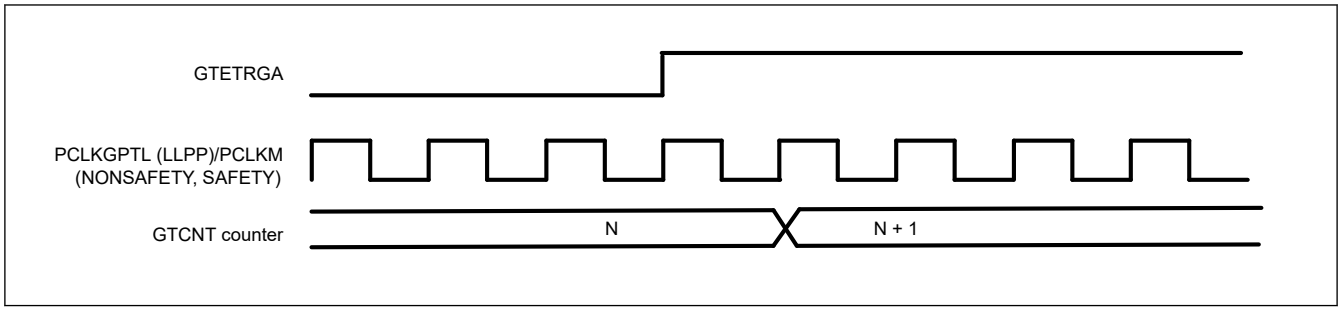


Figure 20.4 Example of event count operation (up-counting of rising edges of the input on the GTETRGA pin)

The following example shows the setting for event count operation in up-counting by a hardware source.

1. Set count-up source
   Select the count-up source with the GTUPSR register.
2. Period setting
   Set a period in the GTPR register.
3. Set initial value for counter
   Set the initial value in the GTCNT counter.
4. Start count operation
   Set the GTCR.CST bit to 1 to start count operation.

(5) Event Count Operation (In Down-Counting by Hardware Source)

The counter in each channel can start down-counting by hardware source by setting the GTDNSR register. When the GTDNSR register is set, the count clock selected by the GTCR.TPCS[3:0] bits and the count direction set by the GTUDDTYC.UD bit are invalid. If the hardware source in up-counting and the hardware source in down-counting are generated at the same time, the GTCNT counter value does not change.

Operation due to underflow in down-counting by hardware source is the same as the periodic count operation by the count clock.

To perform down-counting by hardware source, start count operation by setting the GTCR.CST bit to 1. At the start, the count operation is synchronized with the count clock selected by GTCR.TPCS[3:0] bits, set the CST bit to 1, and wait until the first cycle of the count clock elapses to start down-counting operation. To start down-counting after 1 PCLKGPTL (LLPP)/PCLKM (NONSAFETY, SAFETY) from setting the CST bit to 1, set the GTCR.TPCS[3:0] bits to 0000b.

Figure 20.5 shows an example of down-counting operation by hardware source.

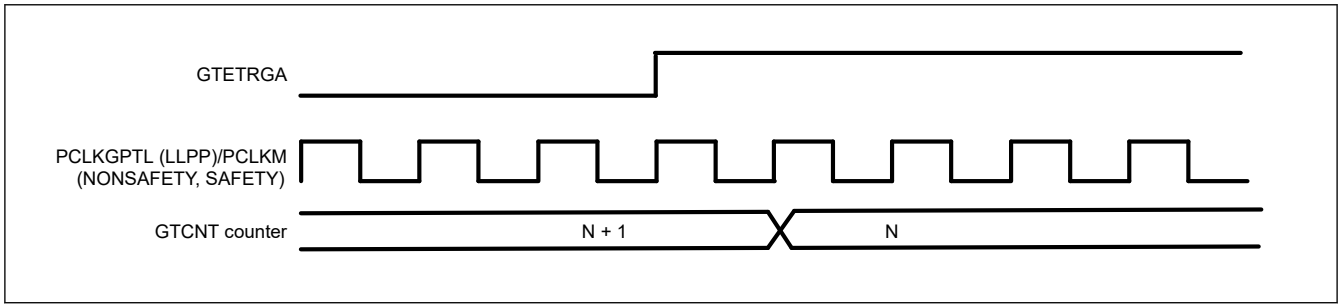


Figure 20.5 Example of event count operation (down-counting of rising edges of the input on the GTETRGA pin)

The following example shows the setting for event count operation in down-counting by hardware source.

1. Set count-down source
   Select the count-down source with the GTDNSR register.
2. Period setting
   Set a period in the GTPR register.
3. Set initial value for counter
   Set the initial value in the GTCNT counter.

4. Start count operation
   Set the GTCR.CST bit to 1 to start count operation.

(6) Counter Clearing Operation

The counter in each channel can be cleared by the following sources:

- Writing to the GTCNT counter
- Writing 1 to the bit corresponding to the channel number for the GTCLR register while the GTCSR.CCLR bit is 1
- Hardware source specified by the GTCSR register

Write access during counting (when CST = 1) is disabled.

GTCLR register can be written to and cleared by hardware source whether the count operation is performed (GTCR.CST bit = 1) or is stopped (CST bit = 0). When the count direction flag is set as decrement (GTST.TUCF flag = 0) in sawtooth-wave mode selected with GTCR.MD[2:0] bits, the GTCNT counter value for writing to the GTCLR register and for clearing by hardware source becomes the GTPR register value whether the counter is in operation or is stopped. The GTCNT counter value becomes 0x00000000 for other settings.

When the event count operation is set (at least one of the bits for the GTUPSR or the GTDNSR register is set as 1), writing to the GTCLR register and clearing by hardware source are performed right after the clear source is generated (the operation is based on PCLKGPTL (LLPP)/PCLKM (NONSAFETY, SAFETY)). For other settings, clearing operation is synchronized with the count clock selected with the GTCR.TPCS[3:0] bits.

20.4.1.2 Waveform Output by Compare Match Counter Operation

Compare match refers to when the GTCNT counter value matches the GTCCRA or GTCCRB register value. The output of the corresponding GTIOCm_nA or GTIOCm_nB pin ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)) can be set to be driven low, high, or toggled in synchronization with the counter clock (including in the case of event counting) after the match.

In addition, the GTIOCm_nA or GTIOCm_nB pin output can be driven low, high, or toggled at the "end of the cycle" which is determined by the GTPR register. The end of the cycle is as follows:

- For sawtooth waves in up-counting: When the GTCNT counter value changes from the GTPR register value to 0x00000000 (overflow)
- For sawtooth waves in down-counting: When the GTCNT counter value changes from 0x00000000 to the GTPR register value (underflow)
- For sawtooth waves with the GTCNT counter clear
- For triangle waves: When the GTCNT counter value changes from 0x00000000 to 0x00000001 (trough)

(1) Low Output and High Output

Figure 20.6 shows an example of low output and high output operation by a compare match of the GTCNT counter and GTCCRA register, and of the GTCNT counter and GTCCRB register.

In this example, the GPT00_0.GTCNT counter performs up-counting, and settings are made so that high is output from the GTIOC00_0A pin by a GPT00_0.GTCCRA register compare match, and low is output from the GTIOC00_0B pin by a GPT00_0.GTCCRB register compare match. The pin level does not change when the specified level and pin level match.

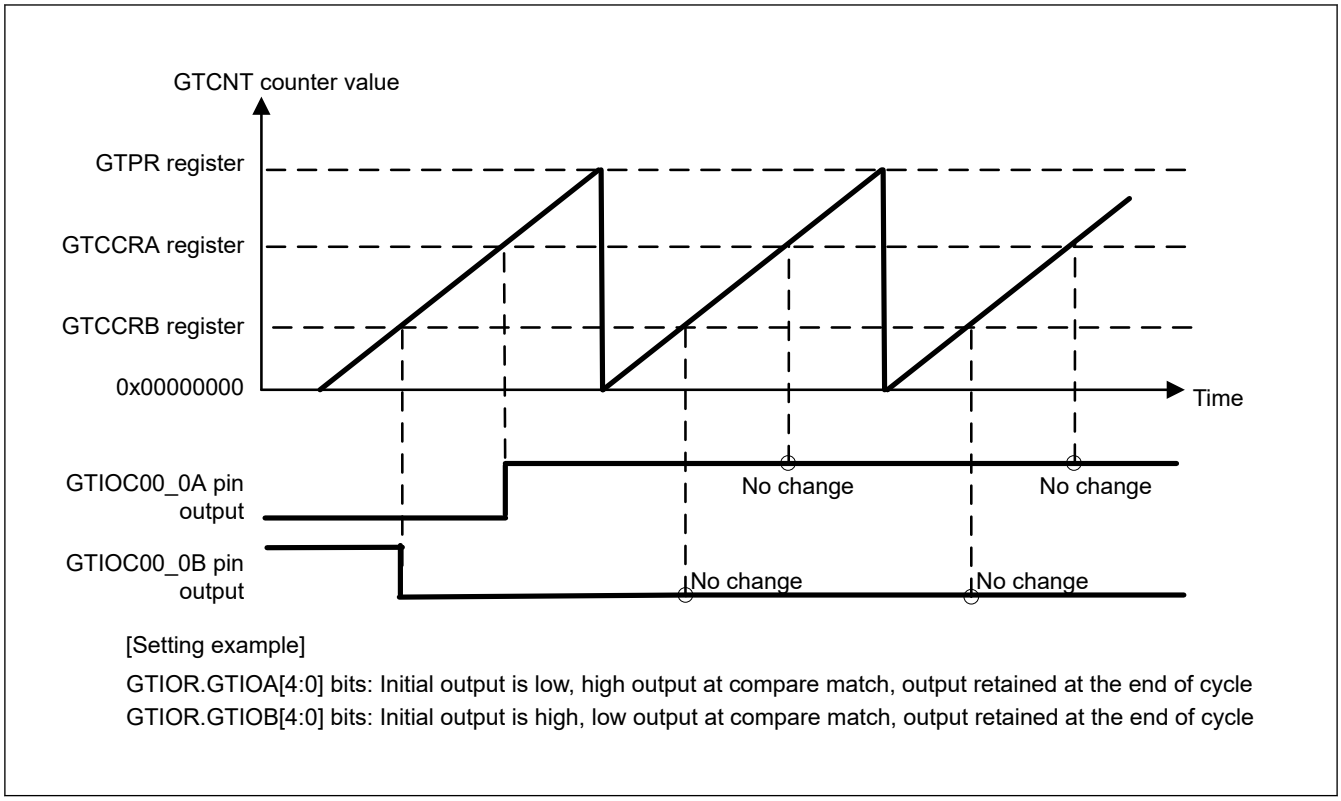


Figure 20.6 Example of low output and high output operation pin

The following example shows the setting for low/high output operation.

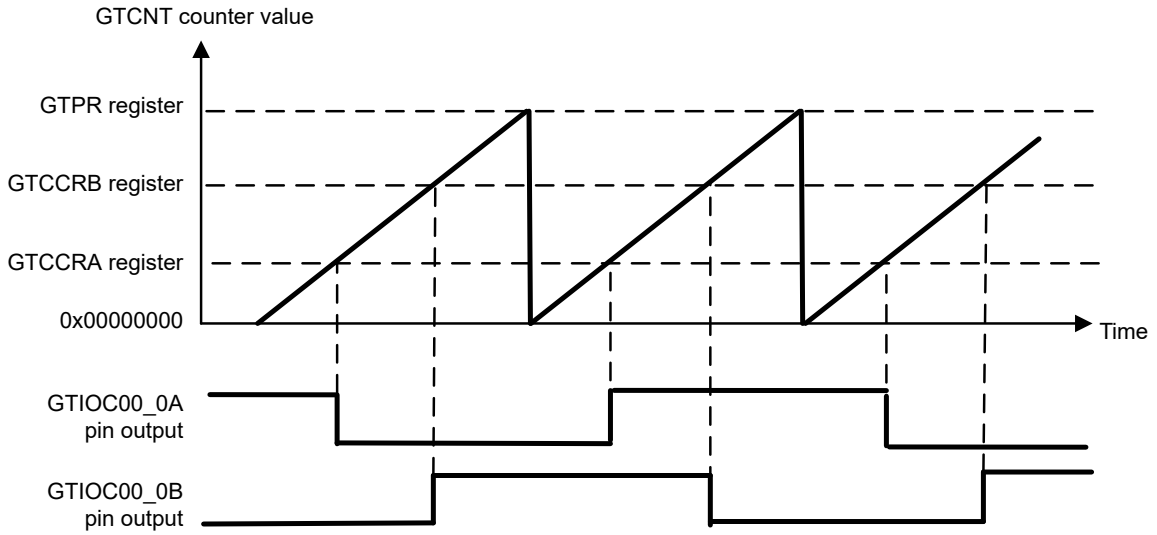
1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits. In Figure 20.6, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 20.6, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCm_nA and GTIOCm_nB pin function
   Set the GTIOCm_nA and GTIOCm_nB pin function with GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register.
   In Figure 20.6, GTIOA[4:0] bits = 00010b and GTIOB[4:0] bits = 10001b.
7. Enable GTIOCm_nA and GTIOCm_nB pin output
   Set to enable the GTIOCm_nA and GTIOCm_nB pin output with the OAE and OBE bits in the GTIOR register.
8. Set compare match value
   Set compare match values in the GTCCRA and GTCCRB registers.
9. Set count operation
   Set the GTCR.CST bit to 1 to start count operation.

(2) Toggled Output

Figure 20.7 and Figure 20.8 show examples of toggle output operations by compare match between the GPT00_0.GTCNT counter and the GTCCRA and GTCCRB registers.

In Figure 20.7, the GPT00_0.GTCNT counter performs up-counting. Settings are made so that the GTIOC00_0A pin output by a GPT00_0.GTCCRA register compare match and GTIOC00_0B pin output by a GPT00_0.GTCCRB register compare match are toggled.

In Figure 20.8, the GPT00_0.GTCNT counter performs up-counting. Settings are made so that the GTIOC00_0A pin output is toggled by a compare match of GPT00_0.GTCCRA register and the GTIOC00_0B pin output is toggled at the end of the cycle.



[Setting example]

GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at the end of cycle

GTIOR.GTIOB[4:0] bits: Initial output is low, output toggled at compare match, output retained at the end of cycle

Figure 20.7 Example of toggled output operation (1)

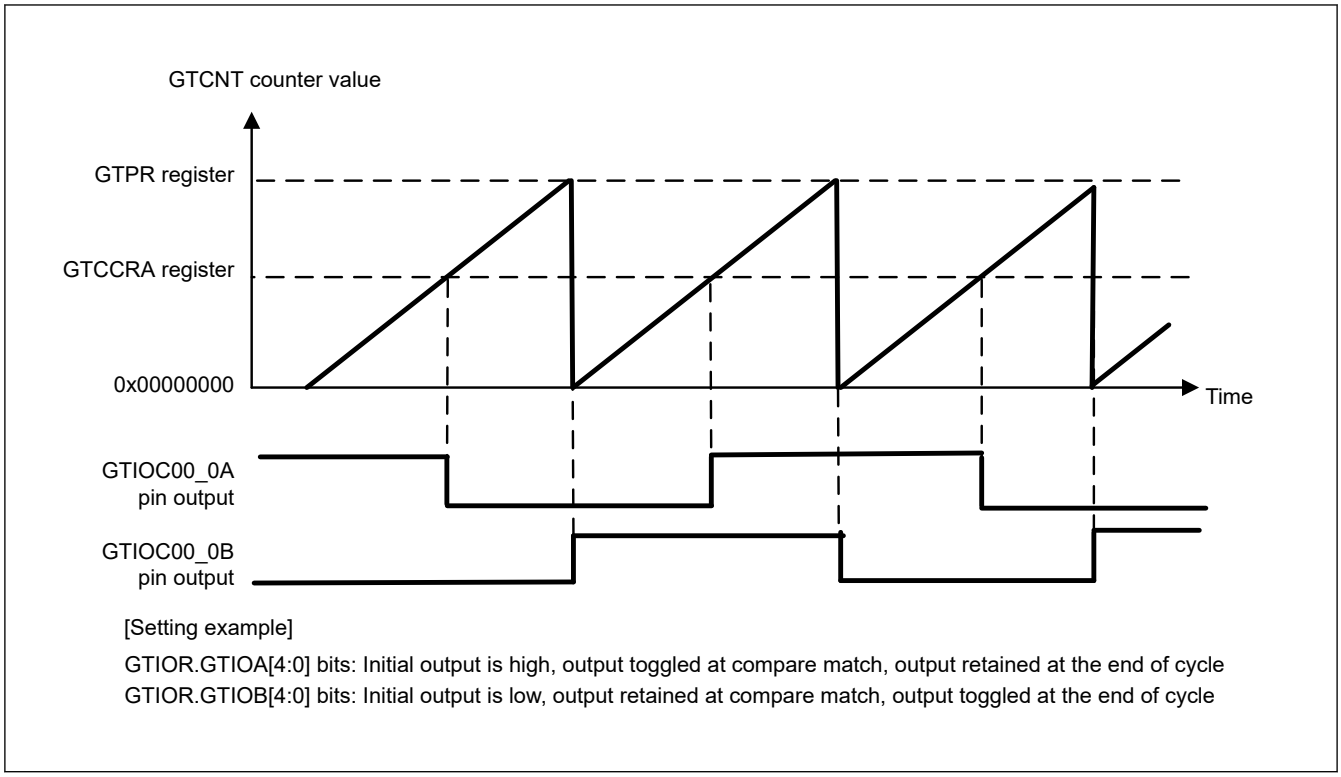


Figure 20.8 Example of toggled output operation (2)

The following example shows the setting for toggled output operation.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.7 and Figure 20.8, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 20.7 and Figure 20.8, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCm_nA and GTIOCm_nB pin function
   Set the GTIOCm_nA and GTIOCm_nB pin function with GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register.
   In Figure 20.7, GTIOA[4:0] bits = 10011b and GTIOB[4:0] bits = 00011b.
   In Figure 20.8, GTIOA[4:0] bits = 10011b and GTIOB[4:0] bits = 01100b.
7. Enable GTIOCm_nA and GTIOCm_nB pin output
   Set to enable the GTIOCm_nA and GTIOCm_nB pin output with the OAE and OBE bits in the GTIOR register.
8. Set compare match value
   Set compare match values in the GTCCRA and GTCCRB registers.
9. Set count operation
   Set the GTCR.CST bit to 1 to start count operation.

20.4.1.3 Input Capture Function

Through detecting hardware sources selected by the GTICASR and GTICBSR registers, the GTCNT counter values can be transferred to the GTCCRA and GTCCRB registers, respectively.

Figure 20.9 shows an example of input capture function operation.

In this example, the GPT00_0.GTCNT counter performs up-counting by count clock, and settings are made so that an input capture is performed at both edges of the GTIOC00_0A pin input in the GTCCRA register and at the rising edge of the GTIOC00_0B pin input in the GTCCRB register.

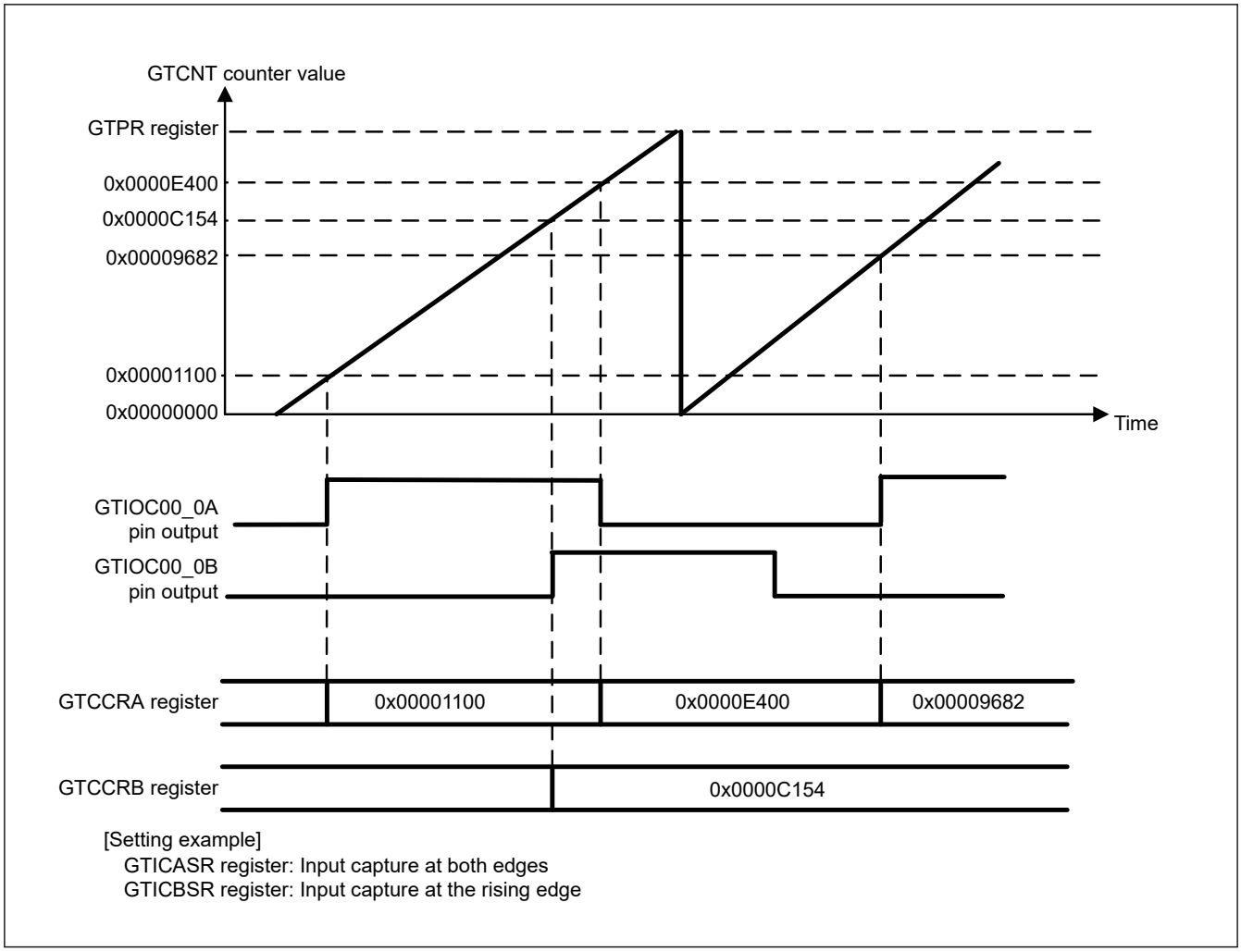


Figure 20.9 Example of input capture operation

The following example shows the setting for input capture operation.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.9, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.9, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Select input capture source
   Select the input capture sources with the GTICASR and GTICBSR registers.
   In Figure 20.9, GTICASR = 0x00000F00 and GTICBSR = 0x00003000.

7. Set count operation
   Set the GTCR.CST bit to 1 to start count operation.

20.4.2 Buffer Operation

The following buffer operations can be set with the GTBER register:

- With the GTPR, GTPBR, and GTPDBR registers used together
- With the GTCCRA, GTCCRC, and GTCCRD registers used together
- With the GTCCRB, GTCCRE, and GTCCRF registers used together
- With the GTADTRA, GTADTBRA, and GTADTDBRA registers used together
- With the GTADTRB, GTADTBRB, and GTADTDBRB registers used together

The following buffer operations can be set with the GTDTCR register:

- With the GTDVU and GTDBU registers used together
- With the GTDVD and GTDBD registers used together

20.4.2.1 GTPR Register Buffer Operation

The GTPBR register can function as a buffer register for the GTPR register, and the GTPDBR register can function as a buffer register for the GTPBR register (double buffer register for the GTPR register).

The buffer transfer is performed at an overflow (in up-counting) or underflow (in down-counting) in sawtooth-wave mode or in event counting, or at a trough in triangle-wave mode.

When in sawtooth-wave mode and in event counting, the buffer transfer is performed even when the counter clearing occurs during the count is in operation.

- Hardware source clearing (clear source selected with bits [23:0] of GTCSR register)
- Clearing by software (when the GTCLR.CCLRn bit is written to 1 while GTCSR.CCLR bit is set to 1) (n = 0 to 17)

To set the GTPR register to function as double buffer, set the GTBER.PR[1:0] bits to 10b or 11b. For single buffer operation, set 01b. For not to function as buffer, set 00b.

Figure 20.10 to Figure 20.12 show examples of the GTPR register buffer operation.

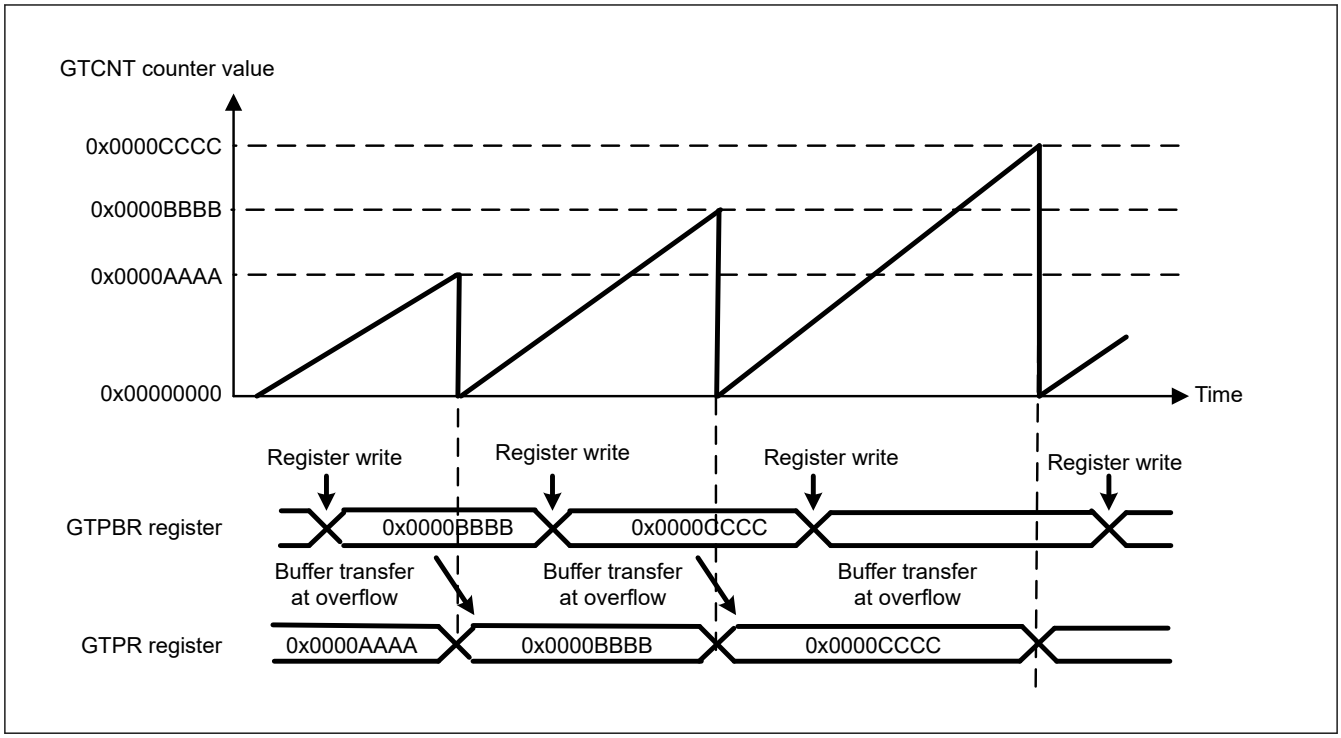


Figure 20.10 Example of the GTPR register buffer operation (sawtooth waves in up-counting)

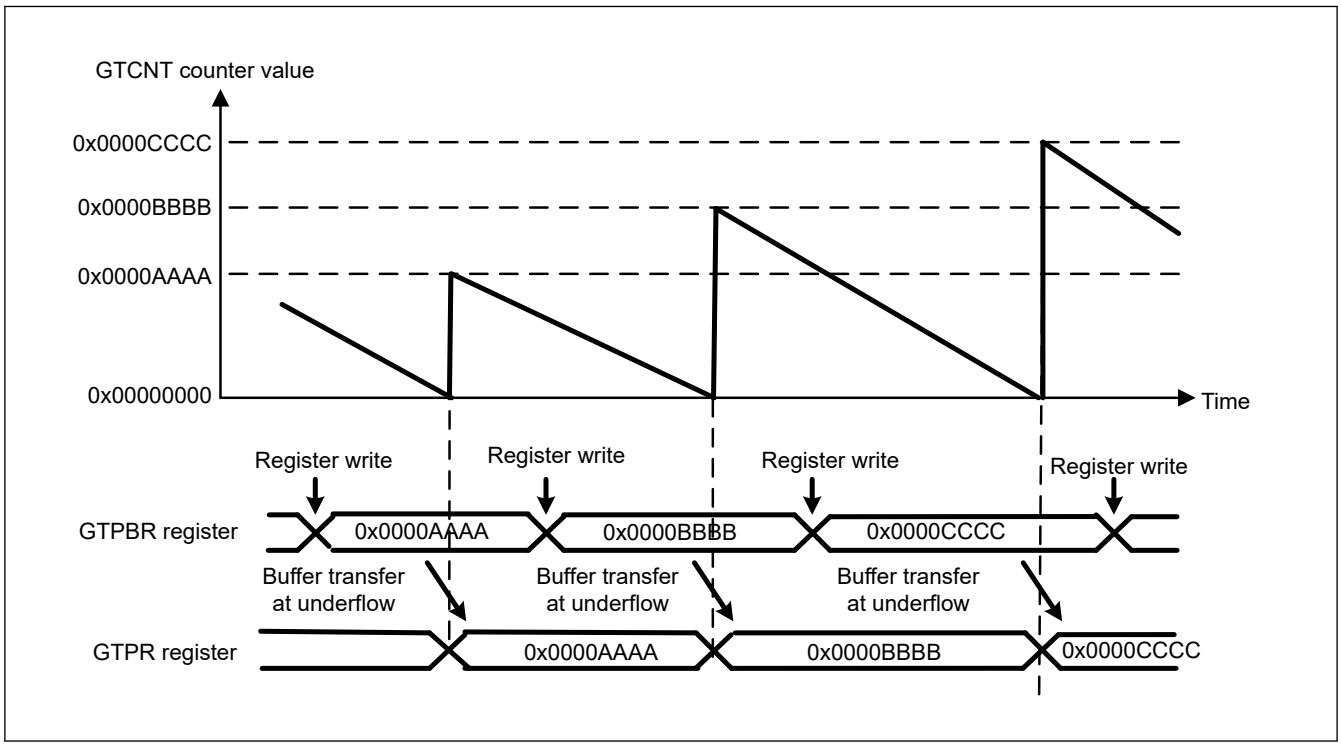


Figure 20.11 Example of the GTPR register buffer operation (sawtooth waves in down-counting)

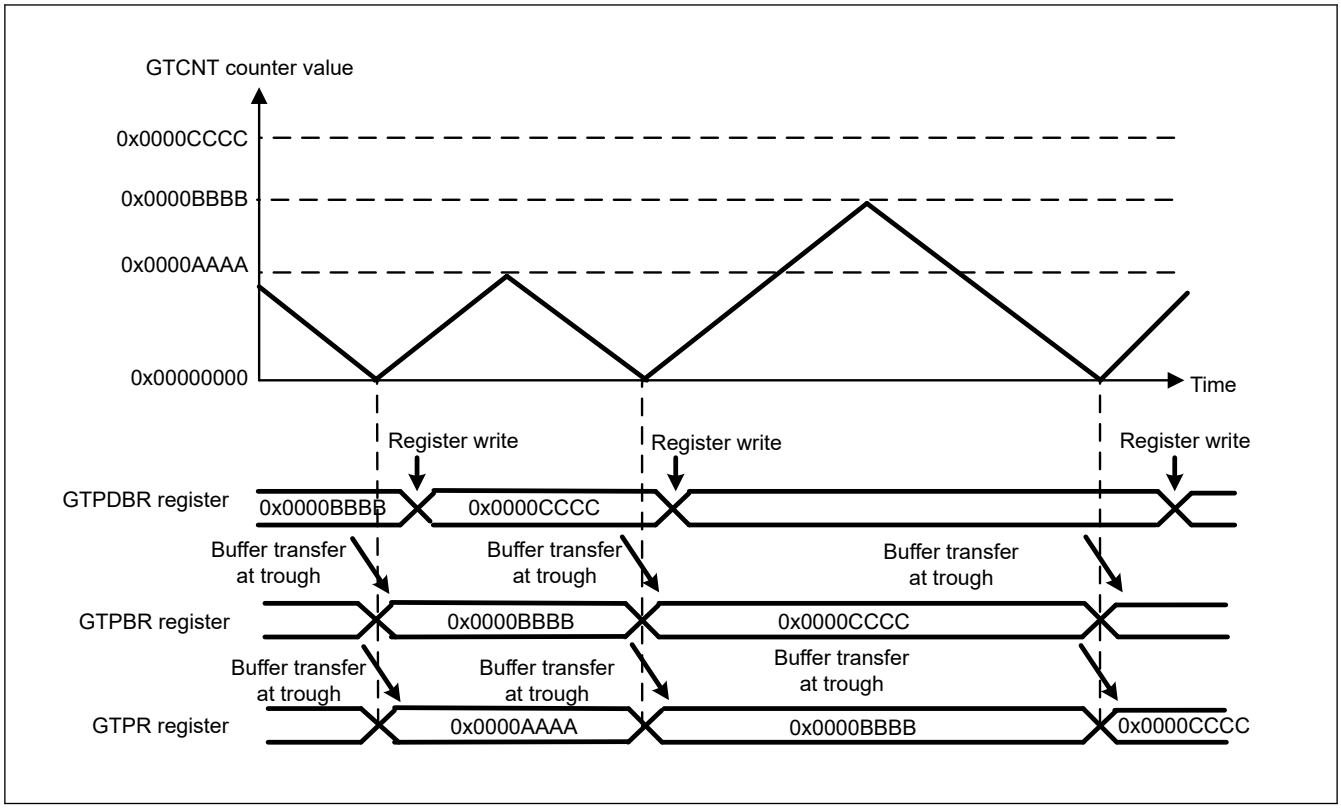


Figure 20.12 Example of the GTPR register double buffer operation (triangle waves)

The following example shows the setting for the GTPR register buffer operation.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.10 and Figure 20.11, 000b (sawtooth-wave PWM mode) is set, and in Figure 20.12, 100b (triangle-wave PWM mode 1) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.10, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting. In Figure 20.11, the lower 2 bits of the GTUDDTYC register is set to 10b and then to 10b for down-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set buffer operation
   Set buffer operation with the GTBER.PR[1:0] bits.
   In Figure 20.10 and Figure 20.11, PR[1:0] bits = 01b. In Figure 20.12, PR[1:0] bits = 1xb.
7. Set buffer value
   For buffer operation, set a period value for the next cycle from the current cycle in the GTPBR register. For double buffer operation, also set a period value for the cycle after the next cycle in the GTPDBR register.
8. Set count operation
   Set the GTCR.CST bit to 1 to start count operation.
9. Set buffer value by cycle
   For buffer operation, set a period value for the next cycle from the current in the GTPBR register. For double buffer operation, set a period value for the cycle after the next cycle in the GTPDBR register.

20.4.2.2 Buffer Operation for the GTCCRA and GTCCRB Registers

The GTCCRC register can function as the GTCCRA buffer register and the GTCCRD register can function as the GTCCRC buffer register (double buffer register for the GTCCRA register). Similarly, the GTCCRE register can function as the GTCCRB buffer register and the GTCCRF register can function as the GTCCRE buffer register (double buffer register for the GTCCRB register).

To set the GTCCRA or GTCCRB register to function as a double buffer, set the GTBER.CCRA[1:0] or GTBER.CCRB[1:0] bits to 10b or 11b. For single buffer operation, set 01b. Not to function as buffer, set 00b.

The following describes buffer operation during output compare and input capture operation.

(1) When the GTCCRA or GTCCRB Register Functions as Output Compare Register

Buffer transfer is performed for the following three cases:

- Buffer transfer at an overflow (in up-counting) and underflow (in down-counting)
  Buffer transfer is performed at an overflow (in up-counting) or an underflow (in down-counting) in sawtooth-wave mode and in event counting, and at a trough (triangle-wave PWM mode 1) or at a crest/trough (triangle-wave PWM mode 2) in triangle-wave mode.
- Buffer transfer by counter clearing
  In sawtooth-wave mode and in event counting, the buffer transfer by similar counter clearing sources in counting operation and in section 20.4.2.1. GTPR Register Buffer Operation, is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).
  In triangle-wave mode, buffer transfer by counter clearing is not performed.
- Forcible buffer transfer
  In both sawtooth-wave mode and triangle-wave mode, buffer transfer of the GTCCRA and GTCCRB registers is forcibly performed by writing 1 to the GTBER.CCRSWT bit while the counting is stopped.
  In sawtooth-wave one-shot pulse mode and triangle-wave PWM mode 3, buffer transfer from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B is also performed using the forcible buffer operation.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1 in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffers to the GTCCRm (m = A, B) registers is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD0 bit or buffer transfer extended skipping function (function for repeated double-buffer operation while buffer transfer is disabled). For details, see section 20.9.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 20.13 to Figure 20.15 show examples of the GTCCRA and GTCCRB registers buffer operations.

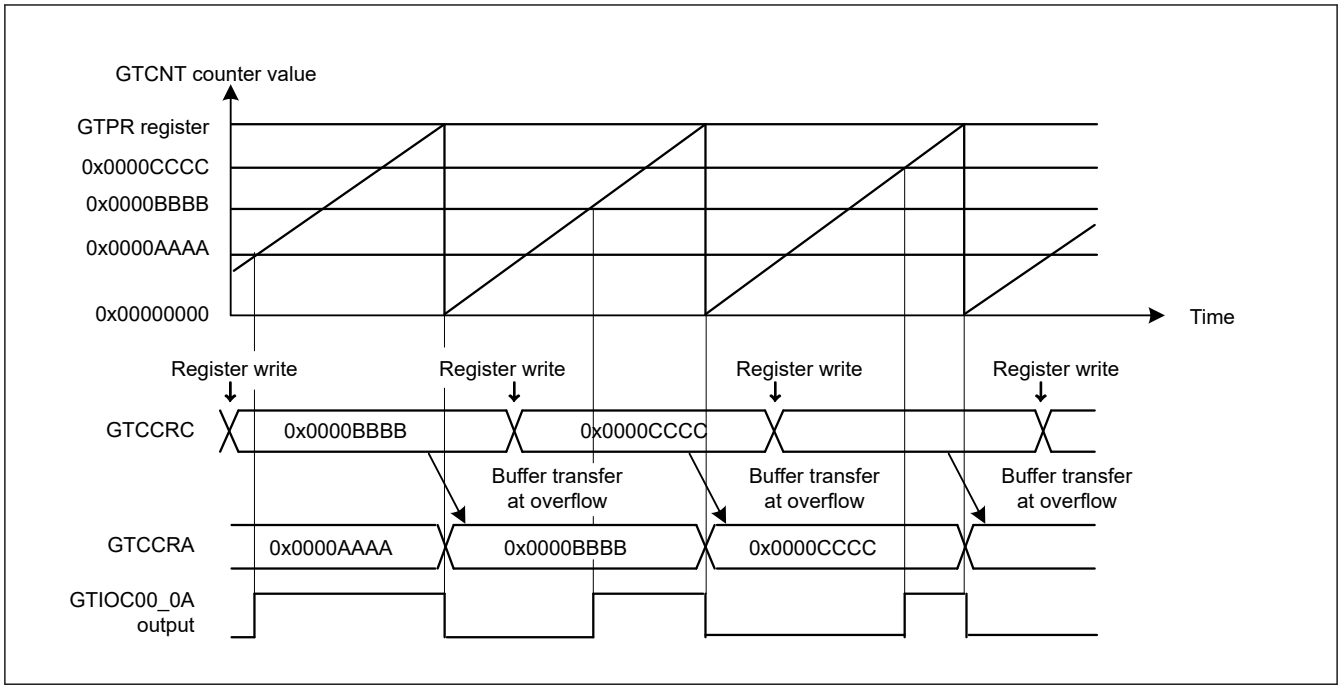


Figure 20.13 Example of GTCCRA and GTCCRB registers buffer operation (output compare, sawtooth waves in up-counting, high output at GTCCRA register compare match, low output at the end of the cycle)

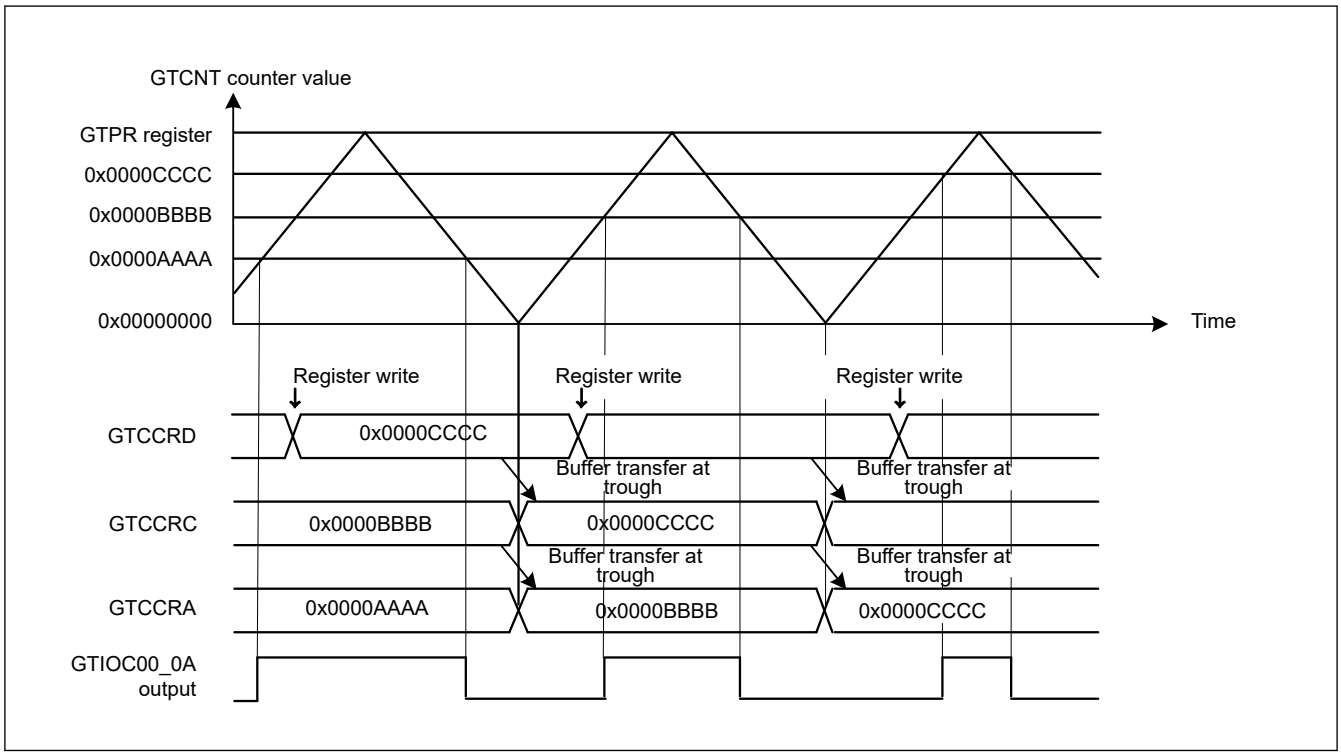


Figure 20.14 Example of GTCCRA and GTCCRB registers double buffer operation (output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA register compare match, output retained at the end of the cycle)

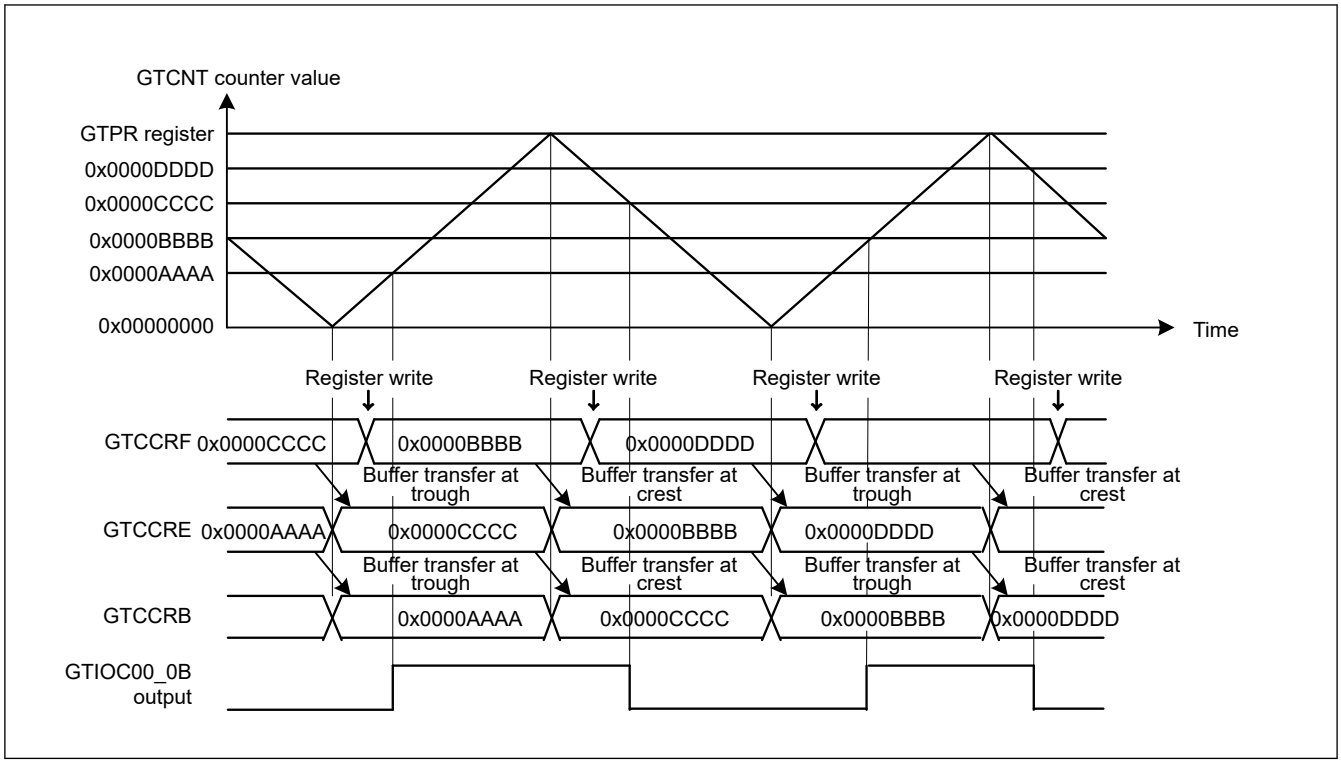


Figure 20.15 Example of GTCCRA and GTCCRB registers double buffer operation (output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB register compare match, output retained at the end of the cycle)

The following example shows setting for buffer operation of the GTCCRA and GTCCRB registers for output compare.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.13, 000b (sawtooth-wave PWM mode) is set, in Figure 20.14, 100b (triangle-wave PWM mode 1) is set, and in Figure 20.15, 101b (triangle-wave PWM mode 2) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 20.13, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCm_nA and GTIOCm_nB pin function
   Set the GTIOCm_nA and GTIOCm_nB pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register.
   In Figure 20.13, GTIOA[4:0] bits = 00110b, in Figure 20.14, GTIOA[4:0] bits = 00011b, and in Figure 20.15, GTIOB[4:0] bits = 00011b.
7. Enable GTIOCm_nA and GTIOCm_nB pin output
   Set to enable the GTIOCm_nA and GTIOCm_nB pin output with the OAE and OBE bits in the GTIOR register.
8. Set buffer operation
   Set buffer operation with the GTBER.CCRA[1:0] bits and CCRB[1:0] bits.
   In Figure 20.13, CCRA[1:0] bits = 01b, in Figure 20.14, CCRA[1:0] bits = 1xb, and in Figure 20.15, CCRB[1:0] bits = 1xb.
9. Set compare match value
   Set the GTIOCm_nA pin changing point in the GTCCRA register and GTIOCm_nB pin changing point in the GTCCRB register.

10. Set buffer value
    For buffer operation, set the GTIOCm_nA pin and GTIOCm_nB pin changing points in one cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCm_nA pin and GTIOCm_nB pin changing points in two cycles after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.
11. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
12. Set buffer value by cycle
    For buffer operation, set the GTIOCm_nA pin and GTIOCm_nB pin changing points in one cycle after the current cycle (in sawtooth-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCm_nA pin and GTIOCm_nB pin changing points in two cycles after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.

(2) When the GTCCRA or GTCCRB Register Functions as Input Capture Register

The timing of transfer to the buffers is when an input capture is generated. When an input capture is generated, the GTCNT counter value is transferred to GTCCRA and GTCCRB registers and the stored GTCCRA and GTCCRB registers values are transferred to buffer registers. Buffer transfer by counter clearing is not performed at input capture.

Figure 20.16 and Figure 20.17 show examples of buffer operation of the GTCCRA and GTCCRB registers.

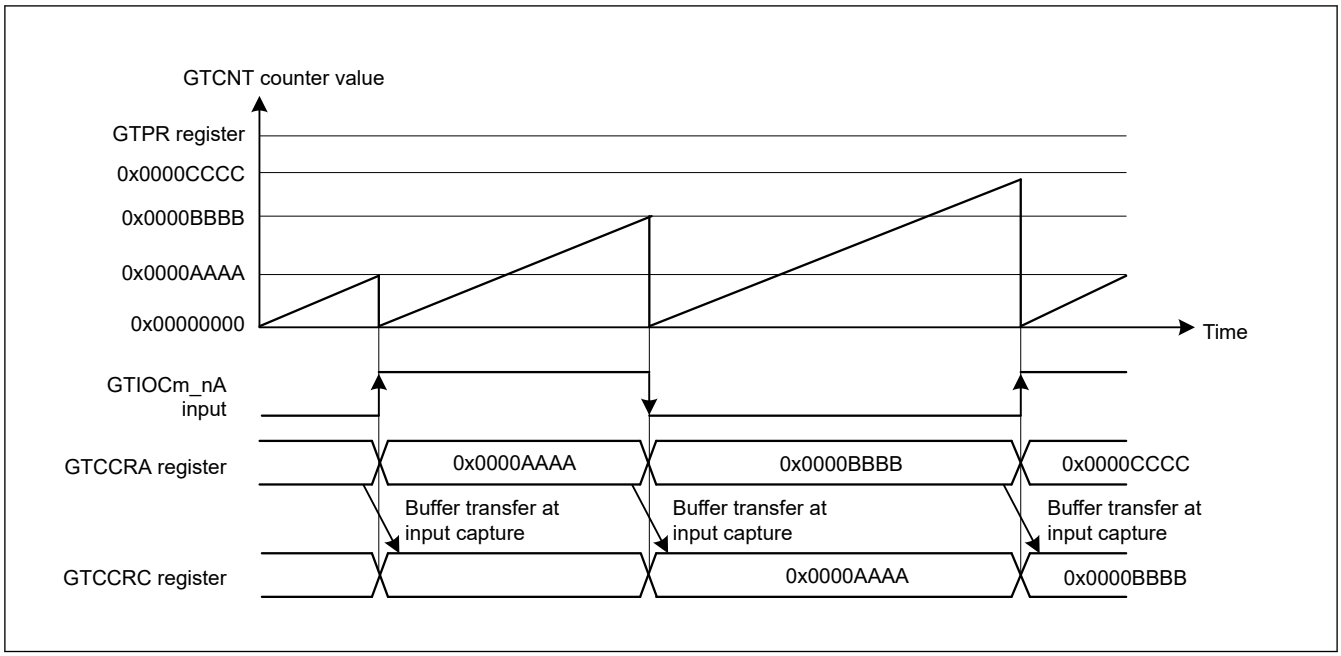


Figure 20.16 Example of buffer operation of the GTCCRA and GTCCRB registers (counting up to produce a sawtooth wave, and clearing the GTCNT counter and input capture on both edges of the input on the GTIOCm_nA pin) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

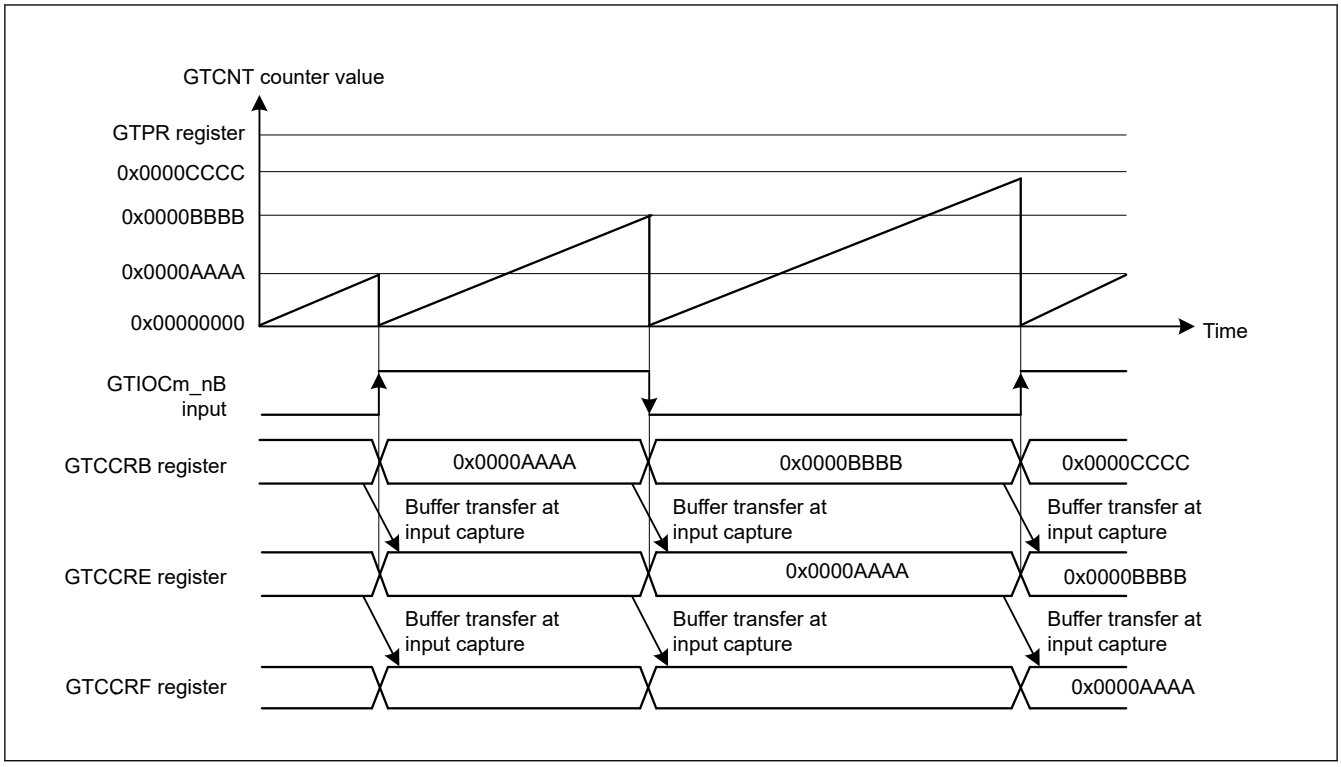


Figure 20.17 Example of double buffer operation of the GTCCRA and GTCCRB registers (counting up to produce a sawtooth wave, and clearing the GTCNT counter and input capture on both edges of the input on the GTIOCm_nB pin) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

The following example shows setting for buffer operation of the GTCCRA and GTCCRB registers for input capture.

1. Set operating mode and counter clear source
   Set the operating mode with GTCR.MD[2:0] and counter clear source with the GTCSR register. In Figure 20.16, MD[2:0] = 000b (sawtooth-wave PWM mode) and GTCSR register = 0x00000F00 and in Figure 20.17, MD[2:0] bits = 000b (sawtooth-wave PWM mode) and GTCSR register = 0x0000F000.
2. Select count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 20.16, the lower 2 bits of GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set input capture source
   Select input capture sources with the GTICASR and GTICBSR registers. In Figure 20.16, GTICASR register = 0x00000F00 and in Figure 20.17, GTICBSR register = 0x0000F000.
7. Set buffer operation
   Set buffer operation with the GTBER.CCRA[1:0] bits and CCRB[1:0] bits.
   In Figure 20.16, CCRA[1:0] bits = 01b and in Figure 20.17, CCRB[1:0] bits = 1xb.
8. Set count operation
   Set the GTCR.CST bit to 1 to start count operation.

20.4.2.3 Buffer Operation for the GTADTRA and GTADTRB Registers

The GTADTBRA register can function as the GTADTRA buffer register and the GTADTDBRA register can function as the GTADTBRA buffer register (double buffer register for the GTADTRA register). Similarly, the GTADTBRB register can

function as the GTADTRB buffer register and the GTADTDBRB register can function as the GTADTBRB buffer register (double buffer register for the GTADTRB register).

To set the GTADTRA or GTADTRB register to function as a double buffer, set the GTBER.ADTDA or ADTDB bit to 1. For single buffer operation, set 0. Not to function as buffer, set the GTBER.ADTTA[1:0] or ADTTB[1:0] bits to 00b. The buffer transfer timing can be set with the ADTTA[1:0] and ADTTB[1:0] bits to an overflow (in up-counting) or an underflow (in down-counting) in sawtooth-wave mode, with ADTTA[1:0] and ADTTB[1:0] bits to 01b for a crest, to 10b for a trough, or to 11b for both crest and trough in triangle-wave mode.

In sawtooth-wave mode, when the ADTTA[1:0] and ADTTB[1:0] bits are set to value other than 00b and in count operation, the buffer transfer, by similar counter clearing sources in section 20.4.2.1. GTPR Register Buffer Operation, is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).

Figure 20.18 to Figure 20.20 show examples of buffer operation of the GTADTRA and GTADTRB registers.

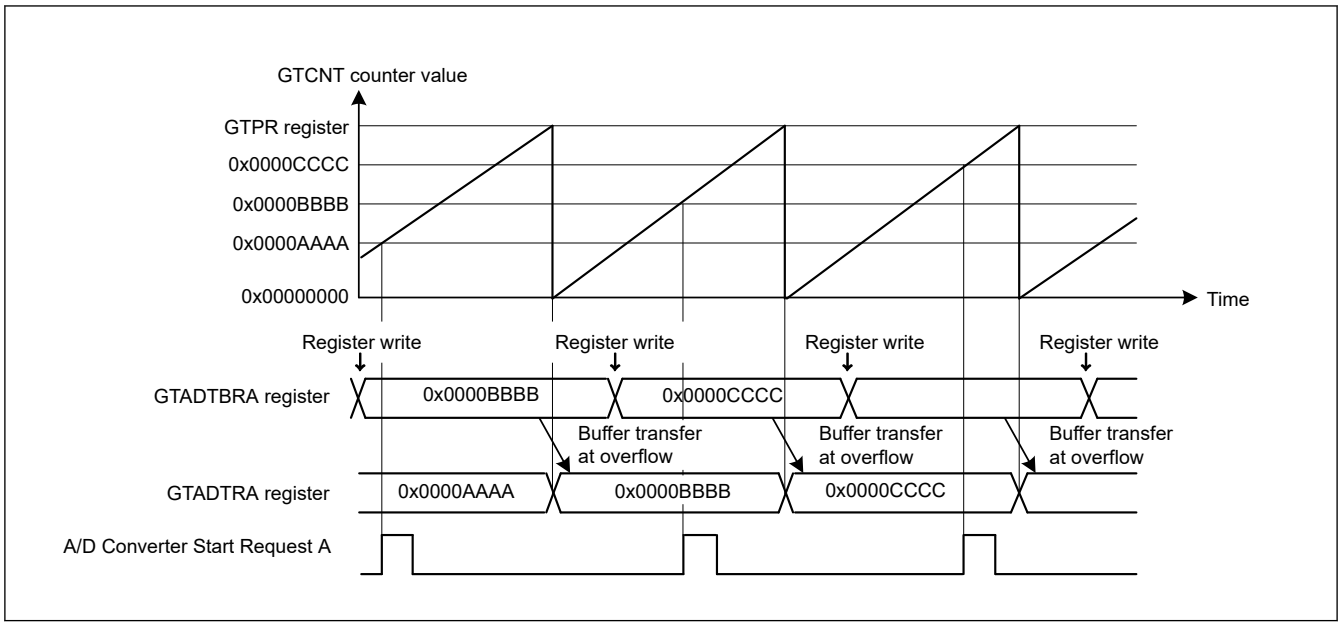


Figure 20.18 Example of buffer operation of the GTADTRA and GTADTRB registers (sawtooth waves in up-counting, A/D converter start request generated by up-counting)

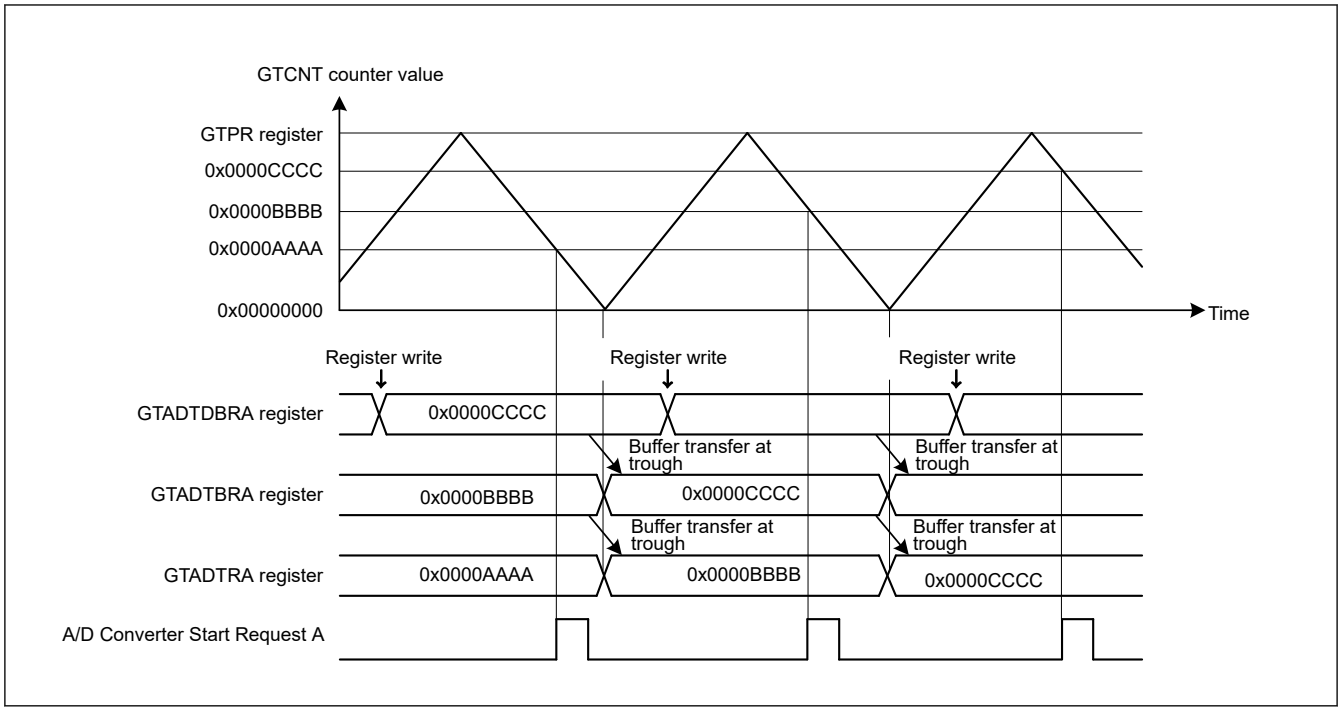


Figure 20.19 Example of double buffer operation of the GTADTRA and GTADTRB registers (triangle waves, buffer transfer at troughs, A/D converter start request generated by down-counting)

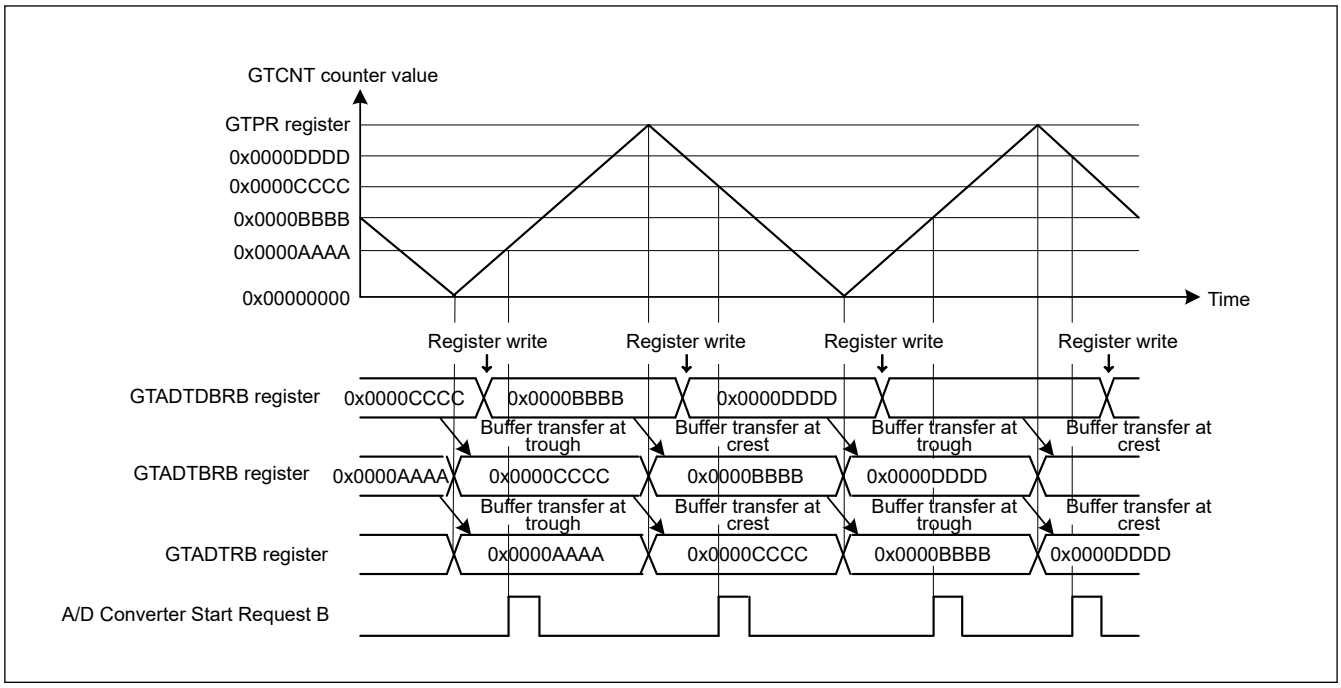


Figure 20.20 Example of double buffer operation of the GTADTRA and GTADTRB registers (triangle waves, buffer transfer at both troughs and crests, A/D converter start request generated by both up-counting and down-counting)

The following example shows setting for buffer operation of the GTADTRA and GTADTRB registers.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.18, 000b (sawtooth-wave PWM mode) is set, and in Figure 20.19 and Figure 20.20, 100b, 101b, or 110b (triangle-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.

In Figure 20.18, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.

3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set buffer operation
   Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in the GTBER register. In Figure 20.18, ADTTA[1:0] bits = 01b, 10b, or 11b and ADTDA bit = 0. In Figure 20.19, ADTTA[1:0] bits = 10b and ADTDA bit = 1, and in Figure 20.20, ADTTB[1:0] bits = 11b and ADTDB bit = 1.
7. Set compare match value
   Set the A/D converter start request point in the GTADTRA and GTADTRB registers.
8. Set buffer value
   For buffer operation, set the A/D converter start request point in one cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D converter start request points in two cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.
9. Enable A/D converter start request
   Set the enable A/D converter start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. In Figure 20.18, ADTRAUEN bit = 1, in Figure 20.19, ADTRADEN bit = 1, and in Figure 20.20, ADTRBUEN bit = 1 and ADTRBDEN bit = 1 .
10. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
11. Set buffer value for each cycle
    For buffer operation, set the A/D converter start request point in one cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, set the A/D converter start request points in two cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

20.4.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOCm_nA pin or GTIOCm_nB pin ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)) by a compare match between the GTCNT counter and the GTCCRA or GTCCRB register.

By setting the GTDTCR, GTDVU, and GTDVD registers, the compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

(1) Sawtooth-Wave PWM Mode

In sawtooth-wave PWM mode, the GTCNT counter performs sawtooth-wave (half-wave) operation by setting the period in the GTPR register and a PWM waveform is output to the GTIOCm_nA or GTIOCm_nB pin when a GTCCRA or GTCCRB register compare match occurs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

Figure 20.21 shows an example of sawtooth-wave PWM mode operation.

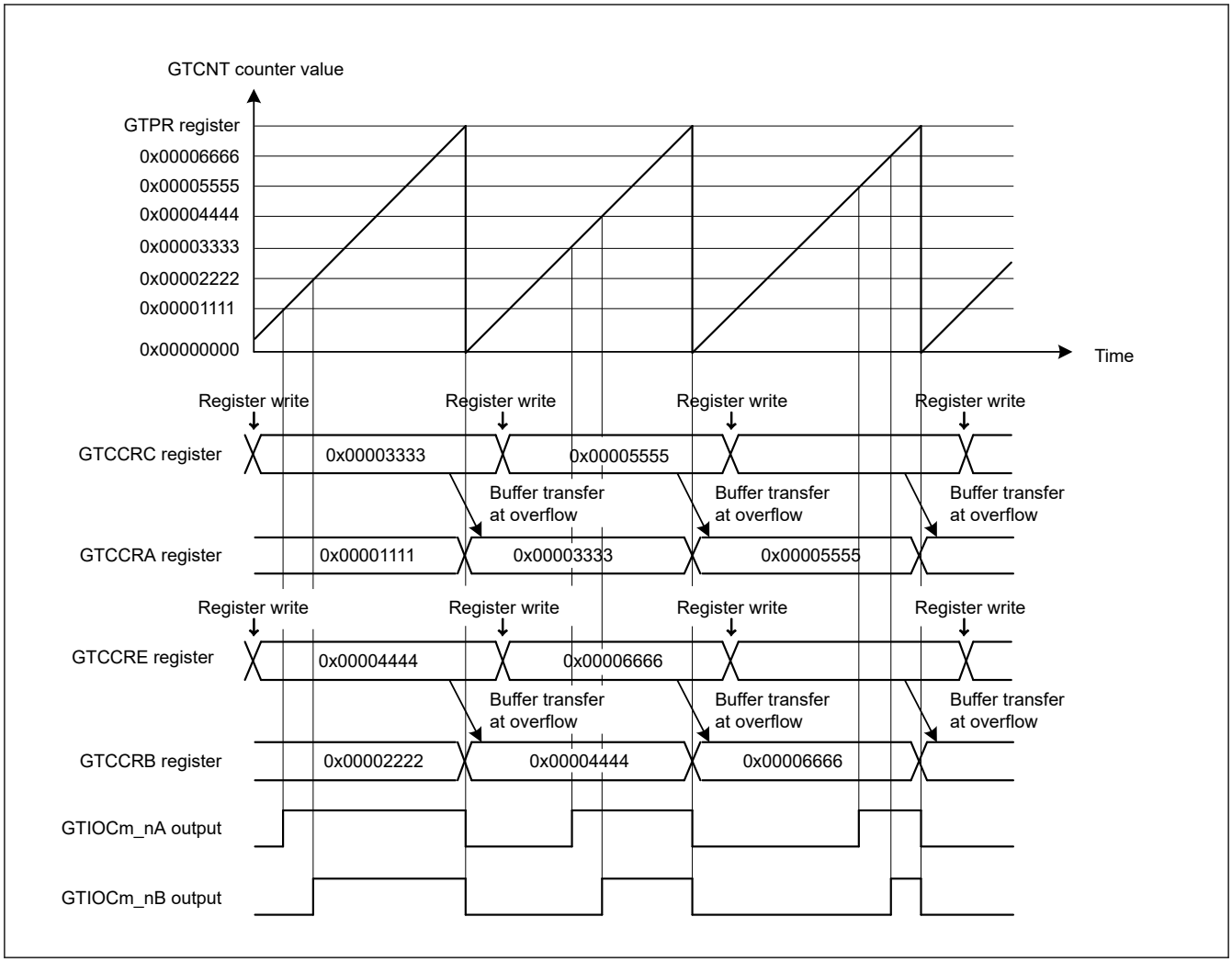


Figure 20.21 Example of sawtooth-wave PWM mode operation (up-counting, buffer operation, high output at GTCCRA/GTCCRB register compare match, low output at the end of the cycle) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

The following example shows setting for sawtooth-wave PWM mode.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.21, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 20.21, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCm_nA and GTIOCm_nB pin function
   Set the GTIOCm_nA and GTIOCm_nB pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register.
   In Figure 20.21, GTIOA[4:0] bits = 00110b and GTIOB[4:0] bits = 00110b.
7. Enable GTIOCm_nA and GTIOCm_nB pin output
   Set to enable the GTIOCm_nA and GTIOCm_nB pin output with the OAE and OBE bits in the GTIOR register.
8. Set buffer operation

Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register.
In Figure 20.21, CCRA[1:0] bits = 01b and CCRB[1:0] bits = 01b.

9. Set compare match value
   Set the GTIOCm_nA pin changing point in the GTCCRA register and the GTIOCm_nB pin changing point in the GTCCRB register.
10. Set buffer value
    For buffer operation, set the GTIOCm_nA and the GTIOCm_nB pin changing points in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCm_nA pin and GTIOCm_nB pin changing points in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
11. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
12. Set buffer value for each cycle
    For buffer operation, set the GTIOCm_nA pin and GTIOCm_nB pin changing point in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively.
    For double buffer operation, set the GTIOCm_nA pin and GTIOCm_nB pin changing points in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: (m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)

(2) Sawtooth-Wave One-Shot Pulse Mode

The sawtooth-wave one-shot pulse mode is a mode in which the period is set in the GTPR register, the GTCNT counter performs sawtooth-wave (half-wave) operation and a PWM waveform is output to the GTIOCm_nA or GTIOCm_nB pin ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)) at a compare match of the GTCCRA or GTCCRB register with buffer operation fixed.

Buffer operation in sawtooth-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from the following registers:

- GTCCRC register to GTCCRA register at the end of the cycle
- GTCCRE register to GTCCRB register at the end of the cycle
- GTCCRD register to temporary register A at the end of the cycle
- GTCCRF register to temporary register B at the end of the cycle
- Temporary register A to GTCCRA register at a GTCCRA register compare match
- Temporary register B to GTCCRB register at a GTCCRB register compare match

The pin output value can be selected from low output, high output, or toggle output separately for a compare match and the end of the cycle according to the GTIOR register setting.

The temporary register A and the temporary register B can perform forcible buffer transfer from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B respectively by writing 1 to the GTBER.CCRSW bit while the count is stopped.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

When the GTBER.DBRTECm (m = A, B) bit is set to 1, transfer from an intermediate buffer to the GTCCRm register is repeated on a cyclic basis by using the temporary register x (x = C, E) and temporary register m which operate as intermediate buffers for the GTCCRx and GTCCRm registers, respectively, even while buffer transfer is disabled (repeated double buffer operation function during disabling of buffer transfer). For details, see section 20.9.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 20.22 shows an example of sawtooth-wave one-shot pulse mode operation.

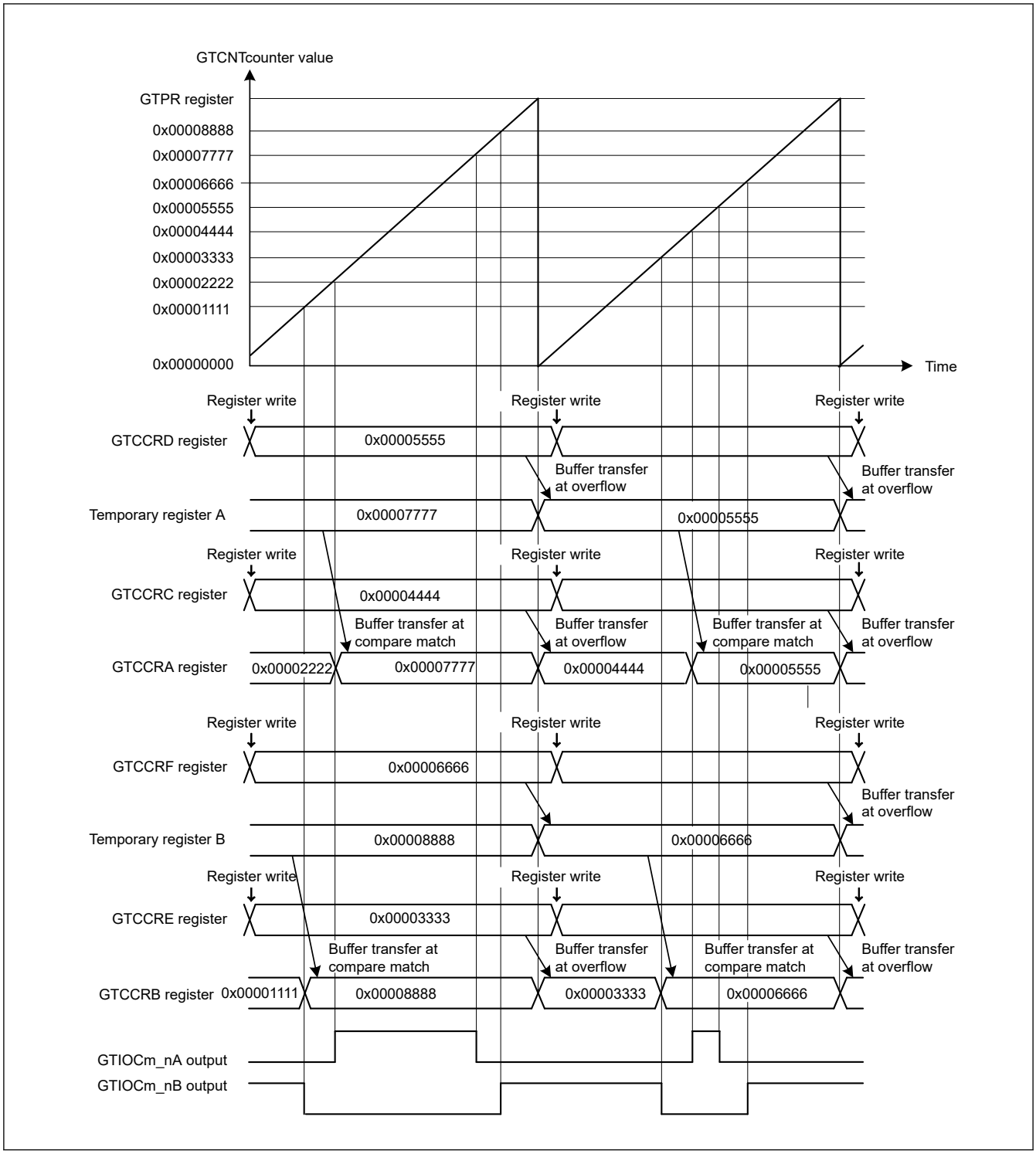


Figure 20.22 Example of sawtooth-wave one-shot pulse mode operation (up-counting, low output from the GTIOCm_nA pin and high output from the GTIOCm_nB pin at initial output, output toggled at GTCCRA/GTCCRB register compare match, output retained at the end of the cycle)

The following example shows setting for sawtooth-wave one-shot PWM mode.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits. In Figure 20.22, 001b (sawtooth-wave PWM one-shot mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 20.22, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.

3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCm_nA and GTIOCm_nB pin function
   Set the GTIOCm_nA and GTIOCm_nB pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.22, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
7. Enable GTIOCm_nA and GTIOCm_nB pin output
   Set to enable the GTIOCm_nA and GTIOCm_nB pin output with the OAE and OBE bits in the GTIOR register.
8. Set buffer value
   Set the GTIOCm_nA pin changing point immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCm_nB pin changing point in the GTCCRE and GTCCRF registers.
9. Set forcible buffer transfer
   Set the GTBER.CCRSWT bit to 1 to forcibly transfer buffer register data.
10. Set buffer value
    For buffer operation, set the GTIOCm_nA pin changing point for the next cycle from the current cycle to the GTCCRC and GRCCRD registers, and set the GTIOCm_nB pin changing point for the next cycle from the current cycle to the GTCCRE and GTCCRF registers.
11. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
12. Set buffer value by cycle
    For buffer operation, set the GTIOCm_nA pin changing point for next cycle from the current cycle to the GTCCRC and GTCCRD registers and set the GTIOCm_nB pin changing point for the next cycle from the current cycle to the GTCCRE and GTCCRF registers.

Note: (m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)

(3) Triangle-Wave PWM Mode 1 (32-Bit Transfer at Trough)

The triangle-wave PWM mode 1 is a mode in which the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCm_nA or GTIOCm_nB pin ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)) when a GTCCRA or GTCCRB register compare match occurs.

Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

Figure 20.23 shows an example of triangle-wave PWM mode 1 operation.

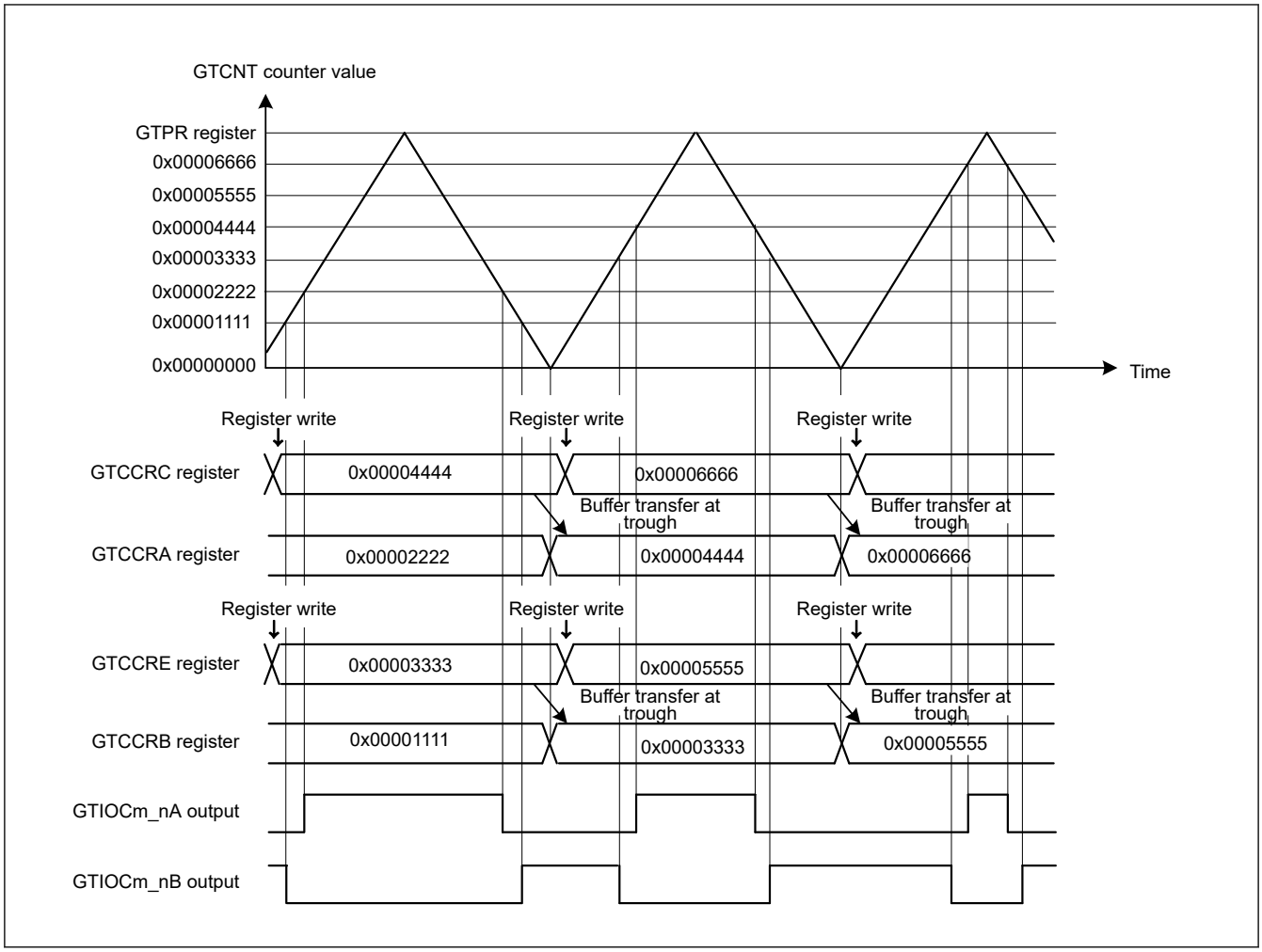


Figure 20.23 Example of triangle-wave PWM mode 1 operation (buffer operation, low output from the GTIOCm_nA pin and high output from the GTIOCm_nB pin at initial output, output toggled at GTCCRA/GTCCRB register compare match, output retained at the end of the cycle) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

The following example shows setting for triangle-wave PWM mode 1.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.23, 100b (triangle-wave PWM mode 1) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
3. Period setting
   Set a period in the GTPR register.
4. Set initial value for counter
   Set the initial value in the GTCNT counter.
5. Set GTIOCm_nA and GTIOCm_nB pin function
   Set the GTIOCm_nA and GTIOCm_nB pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register.
   In Figure 20.23, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
6. Enable GTIOCm_nA and GTIOCm_nB pin output
   Set to enable the GTIOCm_nA and GTIOCm_nB pin output with the OAE and OBE bits in the GTIOR register.
7. Set buffer operation
   Set buffer operation with CCRA[1:0] and CCRB[1:0] bits in the GTBER register.
   In Figure 20.23, CCRA[1:0] bits = 01b and CCRB[1:0] bits = 01b.
8. Set compare match value

Set the GTIOCm_nA pin changing point in the GTCCRA register and GTIOCm_nB pin changing point in the GTCCRB register.

9. Set buffer value
   For buffer operation, set the GTIOCm_nA pin and GTIOCm_nB pin changing points in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCm_nA pin and GTIOCm_nB pin changing points in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
10. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
11. Set buffer value by cycle
    For buffer operation, set the GTIOCm_nA pin and GTIOCm_nB pin changing points in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively.
    For double buffer operation, set the GTIOCm_nA pin and GTIOCm_nB pin changing points in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: (m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)

(4) Triangle-Wave PWM Mode 2 (32-Bit Transfer at Crest and Trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2, the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCm_nA or GTIOCm_nB pin ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)) when a GTCCRA or GTCCRB register compare match occurs. The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

Figure 20.24 shows an example of triangle-wave PWM mode 2 operation.

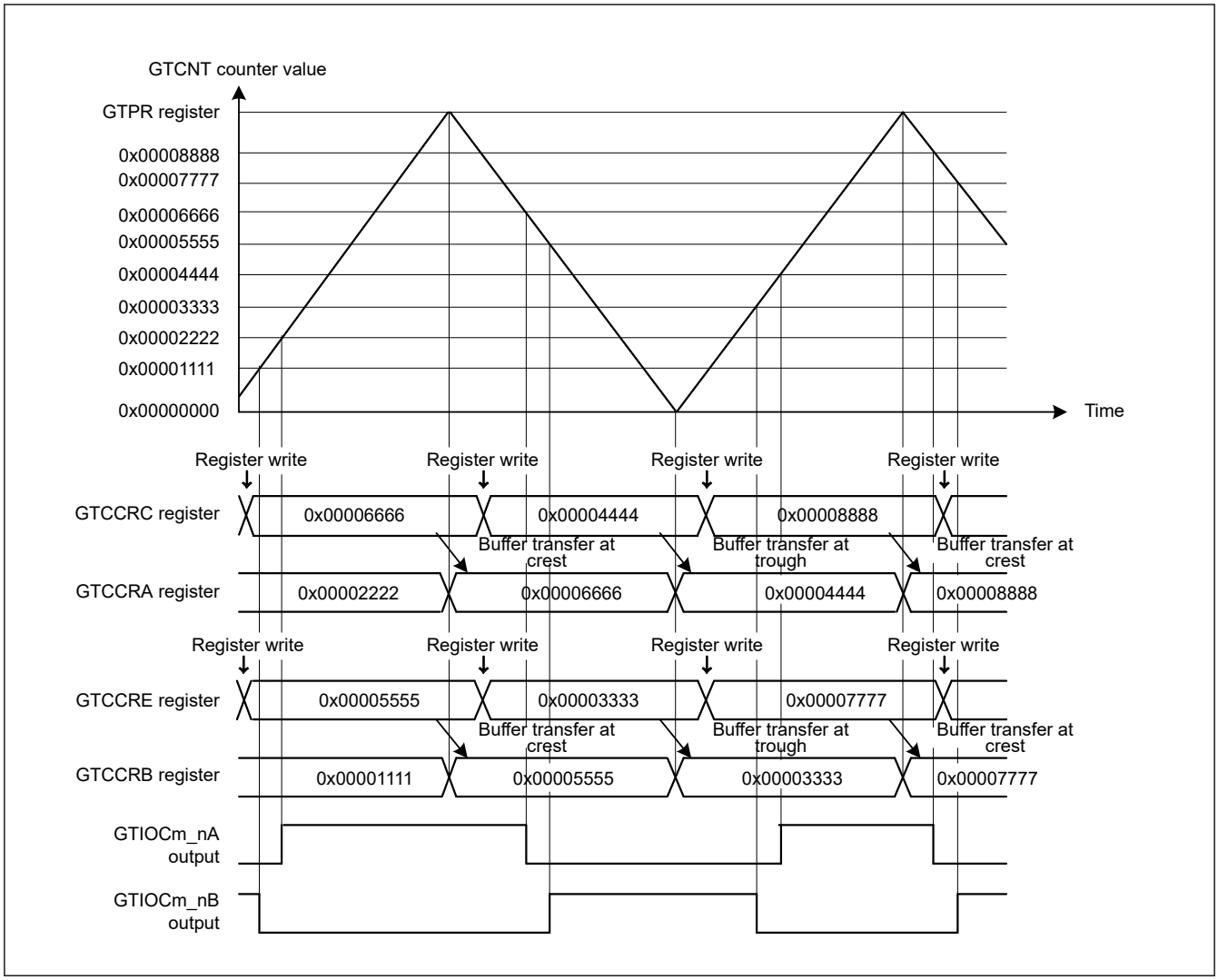


Figure 20.24 Example of triangle-wave PWM mode 2 operation (buffer operation, low output from the GTIOCm_nA pin and high output from the GTIOCm_nB pin at initial output, output toggled at GTCCRA/GTCCRB register compare match, output retained at the end of the cycle) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

The following example shows setting for triangle-wave PWM mode 2.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.24, 101b (triangle-wave PWM mode 2) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
3. Period setting
   Set a period in the GTPR register.
4. Set initial value for counter
   Set the initial value in the GTCNT counter.
5. Set GTIOCm_nA and GTIOCm_nB pin function
   Set the GTIOCm_nA and GTIOCm_nB pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register.
   In Figure 20.24, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
6. Enable GTIOCm_nA and GTIOCm_nB pin output
   Set to enable the GTIOCm_nA and GTIOCm_nB pin output with the OAE and OBE bits in the GTIOR register.
7. Set buffer operation

Set buffer operation with CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 20.24, CCRA[1:0] bits = 01b and CCRB[1:0] bits = 01b.

8. Set compare match value
   Set the GTIOCm_nA pin changing point in the GTCCRA register and GTIOCm_nB pin changing point in the GTCCRB register.
9. Set buffer value
   For buffer operation, set the GTIOCm_nA pin and GTIOCm_nB pin changing points in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively.
   For double buffer operation, also set the GTIOCm_nA pin and GTIOCm_nB pin changing points in one cycle after the current cycle in the GTCCRD and GTCCRF registers, respectively.
10. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
11. Set buffer value by half cycle
    For buffer operation, set the GTIOCm_nA pin and GTIOCm_nB pin changing points in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively.
    For double buffer operation, also set the GTIOCm_nA pin and GTIOCm_nB pin changing points in one cycle after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: (m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)

(5) Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough)

The triangle-wave PWM mode 3 is a mode in which the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCm_nA or GTIOCm_nB pin ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)) at a compare match of the GTCCRA or GTCCRB register with buffer operation fixed. Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the following registers:

- GTCCRC register to GTCCRA register at the trough
- GTCCRE register to GTCCRB register at the trough
- GTCCRD register to temporary register A at the trough
- GTCCRF register to temporary register B at the trough
- Temporary register A to GTCCRA register at the crest
- Temporary register B to GTCCRB register at the crest

The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

When the GTBER.DBRTECm (m = A, B) bit is set to 1, transfer from an intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis using the temporary register x (x = C, E) and temporary register m (m = A, B) which operate as intermediate buffers for the GTCCRx (x = C, E) and GTCCRm registers, respectively, even while buffer transfer is disabled (repeated double buffer operation function during disabling of buffer transfer). For details, see section 20.9.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 20.25 shows an example of triangle-wave PWM mode 3 operation.

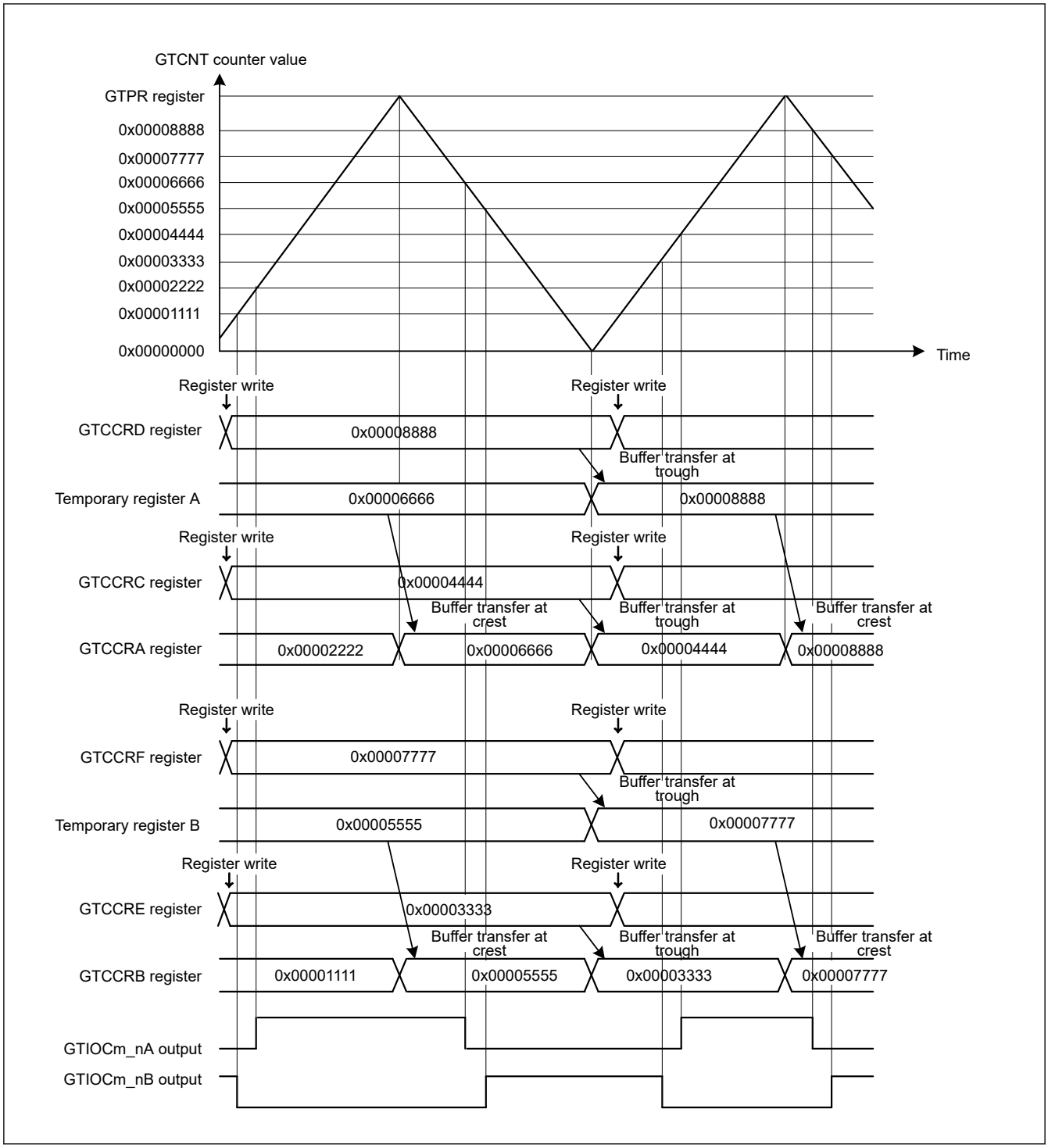


Figure 20.25 Example of triangle-wave PWM mode 3 operation (low output from the GTIOCm_nA pin and high output from the GTIOCm_nB pin at initial output, output toggled at GTCCRA/GTCCRB register compare match, output retained at the end of the cycle) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

The following example shows setting for triangle-wave PWM mode 3.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.25, 110b (triangle-wave PWM mode 3) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
3. Period setting

Set a period in the GTPR register.

4. Set initial value for counter
   Set the initial value in the GTCNT counter.
5. Set GTIOCm_nA and GTIOCm_nB pin function
   Set the GTIOCm_nA and GTIOCm_nB pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.25, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
6. Enable GTIOCm_nA and GTIOCm_nB pin output
   Set to enable the GTIOCm_nA and GTIOCm_nB pin output with the OAE and OBE bits in the GTIOR register.
7. Set buffer value
   Set the GTIOCm_nA pin changing point immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCm_nB pin changing point in the GTCCRE and GTCCRF registers.
8. Set forcible buffer transfer
   Set the GTBER.CCRSWT bit to 1 to forcibly transfer buffer register data.
9. Set buffer value
   For buffer operation, set the GTIOCm_nA pin changing point for the next cycle from the current cycle to the GTCCRC and GRCCRD registers, and set the GTIOCm_nB pin changing point for the next cycle from the current cycle to the GTCCRE and GTCCRF registers.
10. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
11. Set buffer value by cycle
    For buffer operation, set the GTIOCm_nA pin changing point for next cycle from the current cycle to the GTCCRC and GTCCRD registers and set the GTIOCm_nB pin changing point for the next cycle from the current cycle to the GTCCRE and GTCCRF registers.

Note: (m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)

20.4.4 Automatic Dead Time Setting Function

By setting the GTDTCR register, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (the GTCCRA register value) and specified dead time values (the GTDVU and GTDVD registers values) can automatically be set to the GTCCRB register.

The automatic dead time setting function can be used in sawtooth-wave one-shot pulse mode and all the triangle PWM modes.

Dead time can be separately set for the first half and second half of a waveform. Dead time for the changing point in the first half of a negative waveform is set in the GTDVU register and that in the second half is set in the GTDVD register. The same dead time can also be set for the first and second halves by setting the GTDTCR.TDFER bit to 1.

The GTDBU register can be used as a buffer register of the GTDVU register, and the GTDBD register can be used as a buffer register of the GTDVD register. Buffer transfer is performed at the end of the cycle (in sawtooth-wave mode: either of an overflow of the GTCNT counter (up-counting), an underflow (down-counting), or the GTCNT counter clearing in triangle-wave mode: a trough).

The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. Writing to the GTCCRB register is prohibited when the automatic dead time setting function is used.

Do not set the dead-time that makes the change point of the waveform exceeding the count period. When a dead-time setting which would generate a dead-time error is made, adjust the change points of the positive- and negative-phase waveforms to generate waveforms with secured dead-time as shown in Table 20.10. The adjusted change point of the negative-phase waveform is automatically set in the GTCCRB register. An internal signal is used to determine the change point of the positive-phase waveform, thus the value of the GTCCRA register is not updated by the adjusted value.

In sawtooth-wave one-shot pulse mode, if the order of the change point becomes inconsistent by adjustment of the waveform change point due to occurrence of dead time errors, or if the change point exceeds the count period even after the adjustment, the complementary relation between the positive- and negative-phases cannot be guaranteed.

In triangle-wave PWM mode, if dead-time exceeds the count period by setting 0x00000000 or a value greater than or equal to the setting value of the GTPR register is set in the GTCCRA register, output change is controlled by the output protection function (see section 20.9.4. GTIOCm_nA and GTIOCm_nB Protection Function for GTIOCm_nA and GTIOCm_nB Pin

Output ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))). When GTCCRA register is greater than or equal to [GTPR register + GTDVm (m = U, D) register], [GTPR register − 1] is set in the GTCCRB register as the upper limit.

Automatic setting for a dead time value to the GTCCRB register is performed at the next count clock after the register value for calculating the automatic setting value is updated. In triangle-wave mode, it can also be done at the next count clock from the current crest.

Table 20.10 Adjustment of the waveform change point when a dead-time error occurs

Mode	Count direction	Period	Condition for dead-time error	Change point of the positive-phase waveform after adjustment	Change point of the negative-phase waveform after adjustment
Sawtooth-wave one-shot pulse mode	Up-counting	First half	GTCCRA − GTDVU < 0	GTDVU	0
		Second half	GTCCRA + GTDVD > GTPR	GTPR − GTDVD	GTPR
	Down-counting	First half	GTCCRA + GTDVU > GTPR	GTPR − GTDVU	GTPR
		Second half	GTCCRA − GTDVD < 0	GTDVD	0
Triangle-wave PWM mode 1/2/3	Up-counting	(First half)	GTCCRA − GTDVU ≤ 0	GTDVU + 1	1
	Down-counting	(Second half)	GTCCRA − GTDVD < 0	GTDVD	0

Figure 20.26 to Figure 20.28 show examples of automatic dead time setting function operation.

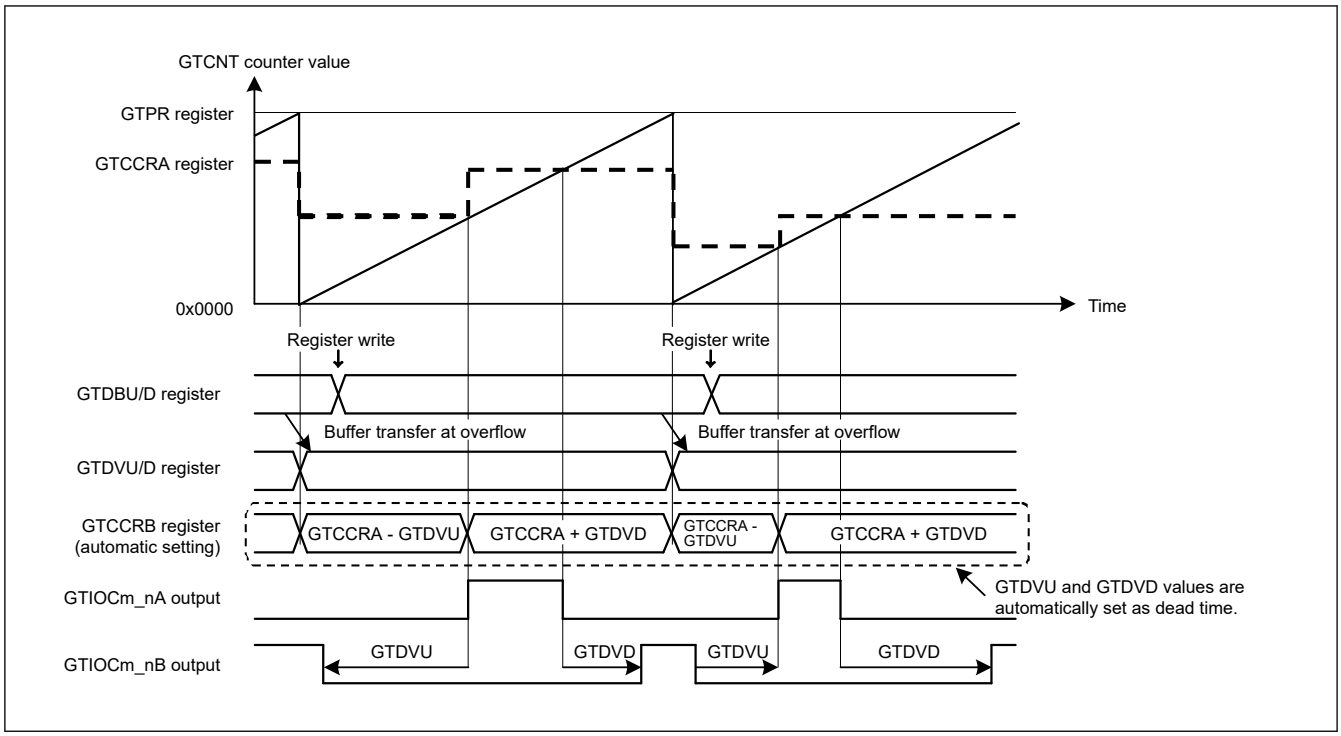


Figure 20.26 Example of automatic dead time setting function operation (sawtooth-wave one-shot pulse mode, up-counting, GTDVU and GTDVD registers set to buffer operation, active-level: high) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

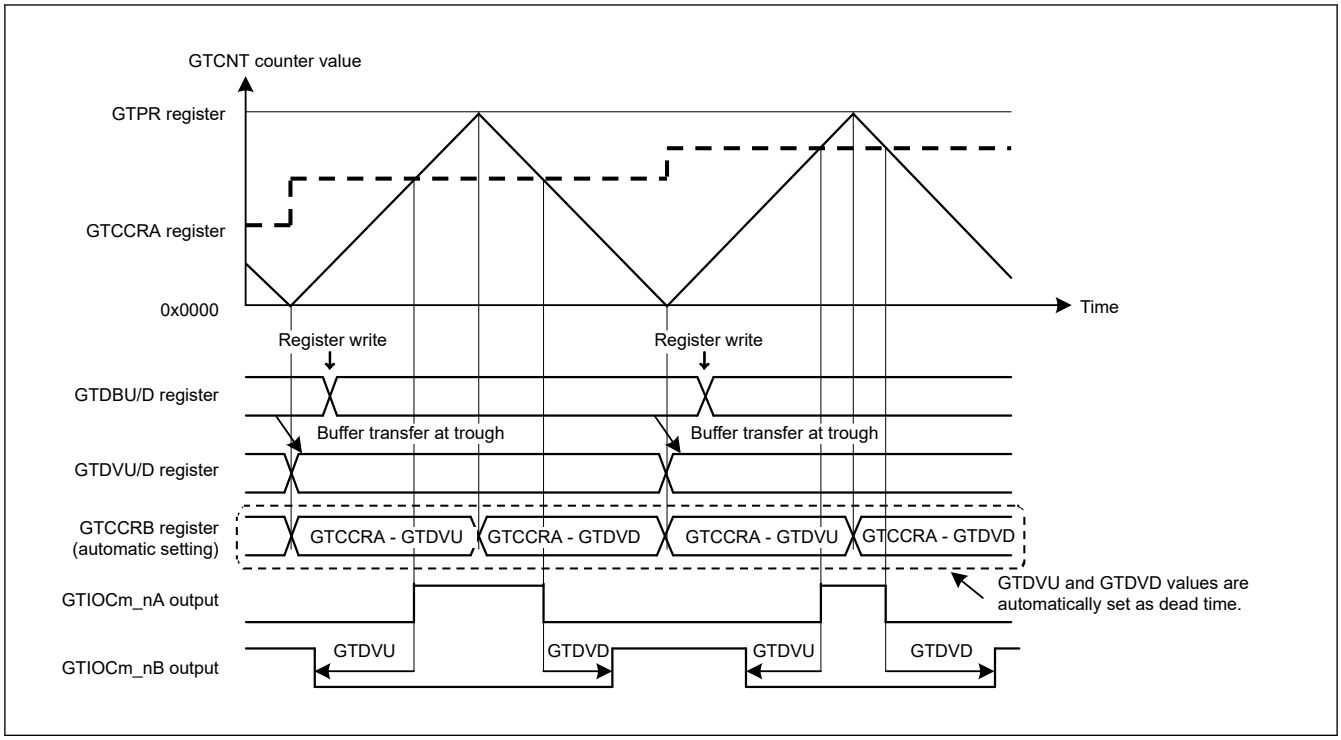


Figure 20.27 Example of automatic compare-match value setting function with dead time (triangle-wave PWM mode 1, GTDVU and GTDVD registers set to buffer operation, active-level: high)

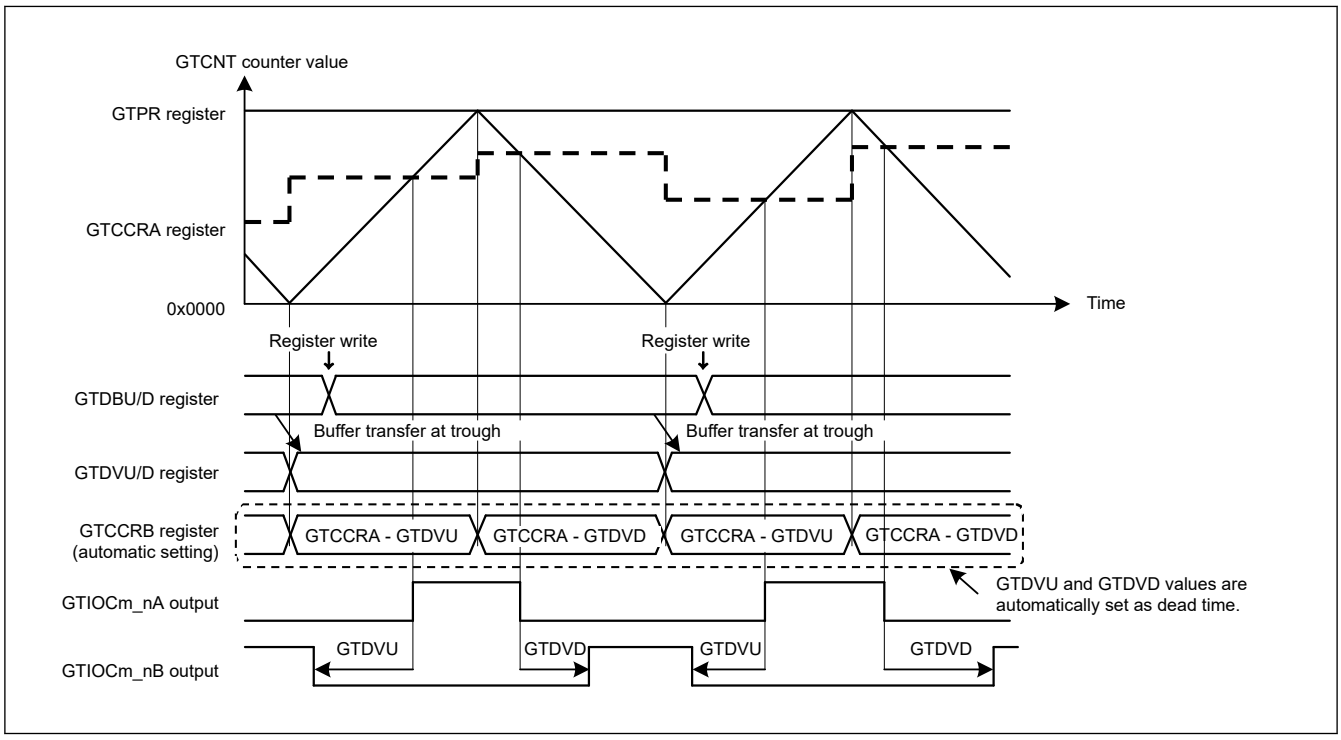


Figure 20.28 Example of automatic compare-match value setting function with dead time (triangle-wave PWM mode 2 or 3, GTDVU and GTDVD registers set to buffer operation, active-level: high)

The following example shows setting for automatic dead time setting function (sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3).

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.26, 001b (sawtooth-wave one-shot pulse mode) is set. In Figure 20.28, 110b (triangle-wave PWM mode 3) is set.

2. Select count direction
   Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.26, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCm_nA and GTIOCm_nB pin function
   Set the GTIOCm_nA and GTIOCm_nB pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.26 and Figure 20.28, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
7. Enable GTIOCm_nA and GTIOCm_nB pin output
   Set to enable the GTIOCm_nA and GTIOCm_nB pin output with the OAE and OBE bits in the GTIOR register.
8. Set buffer value for compare match
   Set the GTIOCm_nA pin changing point immediately after the count start in the GTCCRC and GTCCRD registers.
9. Set forcible buffer transfer for compare match
   Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly to the GTCCRA register.
10. Set buffer value for compare match
    Set the GTIOCm_nA pin changing point in one cycle after the count start in the GTCCRC and GTCCRD registers.
11. Set automatic dead time setting function
    Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
12. Set buffer operation for dead time setting
    Set buffer operation with the TDBUE and TDBDE bits in the GTDTCR register.
13. Set dead time value
    Set the first half dead time value in the GTDVU register and the second half dead time in the GTDVD register. When the GTDVU register is set with the GTDTCR.TDFER bit set to 1, the same value is also set to the GTDVD register, the same dead time value can be set for the first and second halves.
14. Set buffer value for dead time
    For buffer operation, set the first half dead time in one cycle after the current cycle in the GTDBU register and the second half dead time in the GTDBD register.
15. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
16. Set buffer value by cycle
    Set the GTIOCm_nA pin changing point for the next cycle from the current cycle to the GTCCRC and GTCCRD registers.
    When the dead time register is used for buffer operation, set the dead time value in the first half of the next cycle from the current cycle to the GTDBU register and the dead time value in the second half to the GTDBD register.

Note: (m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)

The following example shows setting for automatic dead time setting function (triangle-wave PWM mode 1 or 2).

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.27, 100b (triangle-wave PWM mode 1) is set.
   In Figure 20.28, 101b (triangle-wave PWM mode 2) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
3. Period setting
   Set a period in the GTPR register.
4. Set initial value for counter

Set the initial value in the GTCNT counter.

5. Set GTIOCm_nA and GTIOCm_nB pin function
   Set the GTIOCm_nA and GTIOCm_nB pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.27 and Figure 20.28, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
6. Enable GTIOCm_nA and GTIOCm_nB pin output
   Set to enable the GTIOCm_nA and GTIOCm_nB pin output with the OAE and OBE bits in the GTIOR register.
7. Set buffer operation for compare match
   Set buffer operation with the GTBER.CCRA[1:0] bits.
8. Set the value for compare match
   Set the GTIOCm_nA pin changing point to the GTCCRA register.
9. Set buffer value for compare match
   For buffer operation, set the GTIOCm_nA pin changing point in one cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRC register. For double buffer operation, also set the GTIOCm_nA pin changing point in two cycles after the current cycle (in triangle-wave PWM mode 1) or one cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRD register.
10. Set automatic dead time setting function
    Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
11. Set buffer operation for dead time setting
    Set buffer operation with the TDBUE and TDBDE bits in the GTDTCR register.
12. Set dead time value
    Set the first half dead time value in the GTDVU register and the second half dead time in the GTDVD register. When the GTDVU register is set with the GTDTCR.TDFER bit set to 1, the same value is also set to the GTDVD register, the same dead time value can be set for the first and second halves.
13. Set buffer value for dead time
    For buffer operation, set the first half dead time in one cycle after the current cycle in the GTDBU register and the second half dead time in the GTDBD register.
14. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
15. Set buffer value by cycle
    For buffer operation of compare match register, set the GTIOCm_nA pin changing point in one cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRC register. For double buffered operation of compare match register, set the GTIOCm_nA pin changing point in two cycles after the current cycle (in triangle-wave PWM mode 1) or one cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRD register.
    In the same way, set the dead time value in the first half of the cycle after the current cycle in the GTDBU register and the dead time in the second half in the GTDBD register.

Note: (m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)

20.4.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the GTUDDTYC.UD bit.

In sawtooth-wave mode, if the UD bit is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the UD bit is modified while count operation is stopped and the GTUDDTYC.UDF bit is 0, the UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while count operation is stopped, the UD bit value at that time is reflected at the start of counting.

In triangle-wave mode, changing the value of the GTUDDTYC.UD bit during counting does not switch the direction of counting. Likewise, changing the value of the GTUDDTYC.UD bit while the GTUDDTYC.UDF bit is 0 and counting is stopped does not actually update the value of the bit. If 1 is written to the GTUDDTYC.UDF bit while counting is stopped, the value of the GTUDDTYC.UD bit at that time is reflected from the time counting is started.

When counting direction is switched during count operation in sawtooth-wave mode, the value of GTPR register after the start of up-counting is reflected in the count period in up-counting operation, and the value of GTPR register after the start of down-counting is reflected in the count period in down-counting operation.

Figure 20.29 shows an example of the count direction changing function operation.

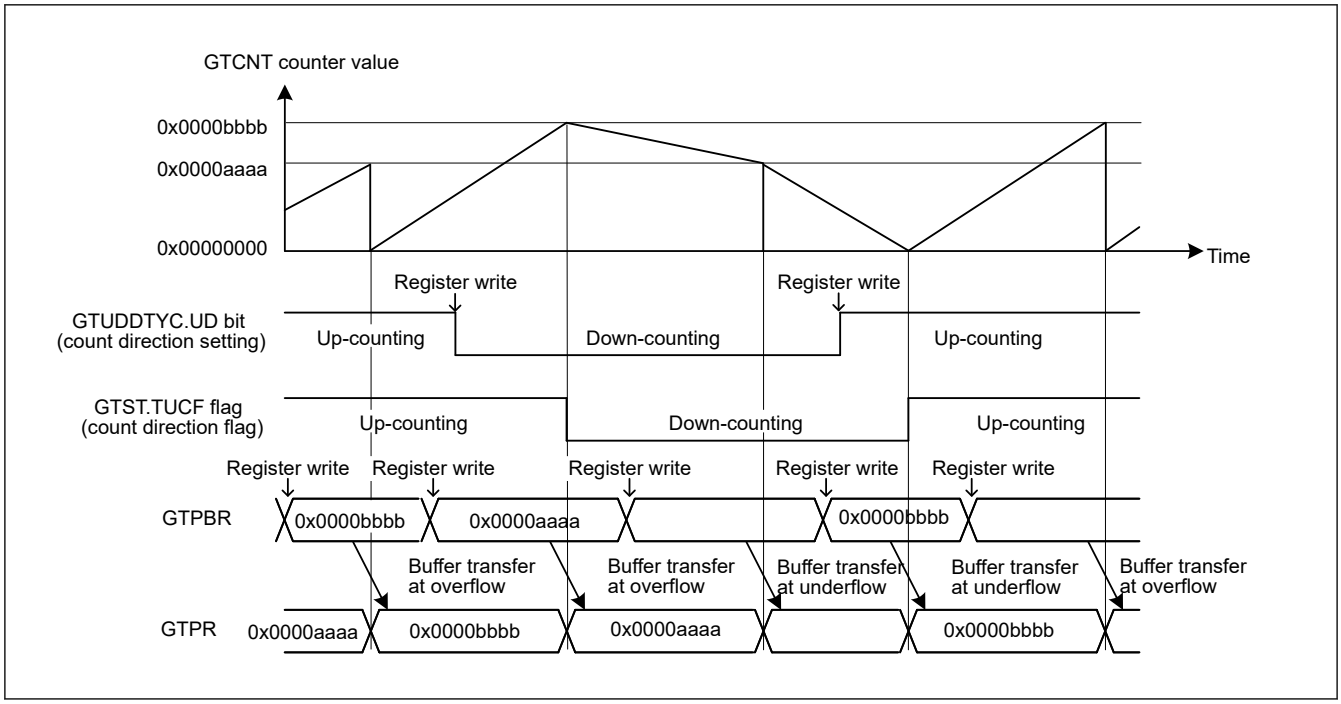


Figure 20.29 Example of count direction changing function operation (during buffer operation)

20.4.6 Function of Output Duty 0% and 100%

Changing the value of the GTUDDTYC.OADTY[1:0] bits and GTUDDTYC.OBDTY[1:0] bits specifies the output duty setting on the GTIOCm_nA and GTIOCm_nB pins to 0% or 100% ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)).

In sawtooth-wave mode, if the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed during the count operation, the output duty-cycle setting is reflected at an overflow (when changed during up-counting) or an underflow (when changed during down-counting). When the GTUDDTYC.OADTYF bit or GTUDDTYC.OBDTYF bit is 0 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the output duty-cycle setting changed at the start of counting is not reflected, but the output duty-cycle setting changed at an overflow or underflow is reflected. When the OADTYF bit or OBDTYF bit is 1 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the value of the OADTY[1:0] bits or OBDTY[1:0] bits at that time is reflected at the start of counting.

In triangle-wave mode, if the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed during the count operation, the output duty-cycle setting is reflected at trough of triangle-wave.

When the OADTYF bit or OBDTYF bit is 0 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the output duty-cycle setting changed at the start of counting is not reflected, but the output duty-cycle setting changed at trough is reflected. When the OADTYF bit or OBDTYF bit is 1 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the value of the OADTY[1:0] bits or OBDTY[1:0] bits at that time is reflected at the start of counting.

During the operation under the settings of duty cycle for 0% or 100%, compare match operation inside the GPT continues to perform interrupt outputs and buffer operations.

When the output duty setting is changed from 0% or 100% due to a compare match, the GTIOR.GTIOA[3:2] bits and the GTUDDTYC.OADTYR bits determine the level output on the GTIOCm_nA pin and GTIOR.GTIOB[3:2] bits and GTUDDTYC.OBDTYR bits determine the level output on the GTIOCm_nB pin at the end of the cycle. When the GTIOA[3:2] and GTIOB[3:2] bits are set as 01b, output is driven low at the end of the cycle, and when set as 10b, output is driven high at the end of the cycle.

The output is retained at the end of the cycles when the GTIOA[3:2] or GTIOB[3:2] bits are 00b, and toggled at the end of the cycles when the GTIOA[3:2] or GTIOB[3:2] bits are 11b. A value to be output is selectable in the OADTYR or OBDTYR bit when the corresponding GTIOA[3:2] or GTIOB[3:2] bits are either 00b or 11b.

Table 20.11 lists the output values at the end of the cycle when the output setting is changed from duty 0% or 100% to compare match.

Table 20.11 Output value after release of duty 0% or 100%

GTIOR.GTIOm[3:2]	Value at compare match output at the end of the cycle when masking by a duty cycle 0% or 100%	GTUDDTYC.OmDTYR bit at duty 0% setting		GTUDDTYC.OmDTYR bit at duty 100% setting	
		0	1	0	1
00b (output retained at the end of the cycle)	0	0	0	1	0
	1	0	1	1	1
01b (low output at the end of the cycle)	—	0	0	0	0
10b (high output at the end of the cycle)	—	1	1	1	1
11b (toggle output at the end of the cycle)	0	1	1	0	1
	1	1	0	0	0

Note: m = A, B

Figure 20.30 shows an example of operation of the output of the duty cycle 0% or 100%.

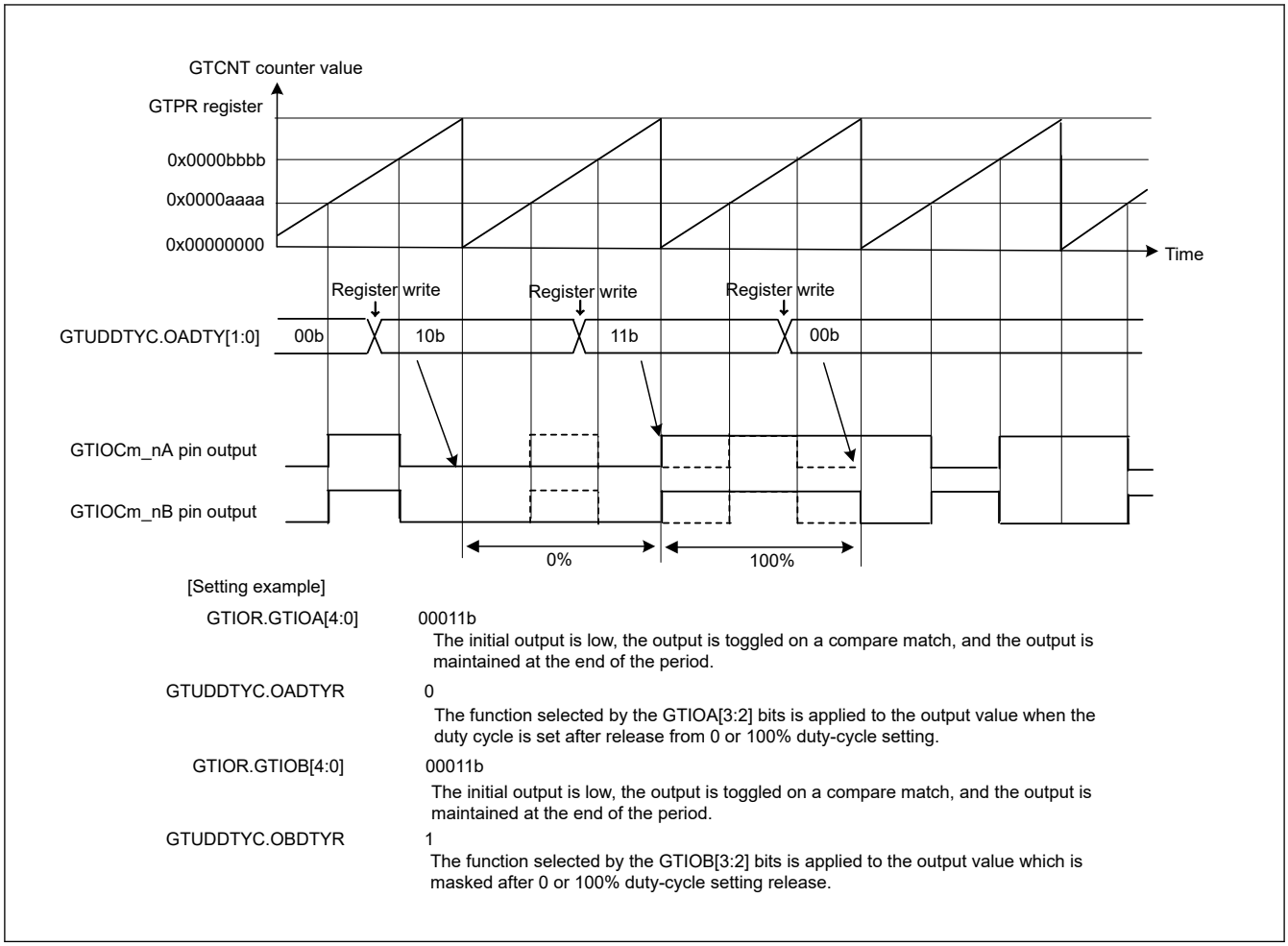


Figure 20.30 Example of operation of output of duty cycle 0% or 100%

20.4.7 Hardware Count Start/Count Stop and Clear Operation

The GTCNT counter can be started, stopped, or cleared by hardware sources. There are 3 types of hardware sources, including external trigger input, ELC event input, and GTIOCm_nA and GTIOCm_nB pin input ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)).

20.4.7.1 Hardware Start Operation

The GTCNT counter can be started by a hardware source. Select a hardware source to start counting using the GTSSR register, and enable to start counting.

Figure 20.31 shows an example of count start operation by a hardware source.

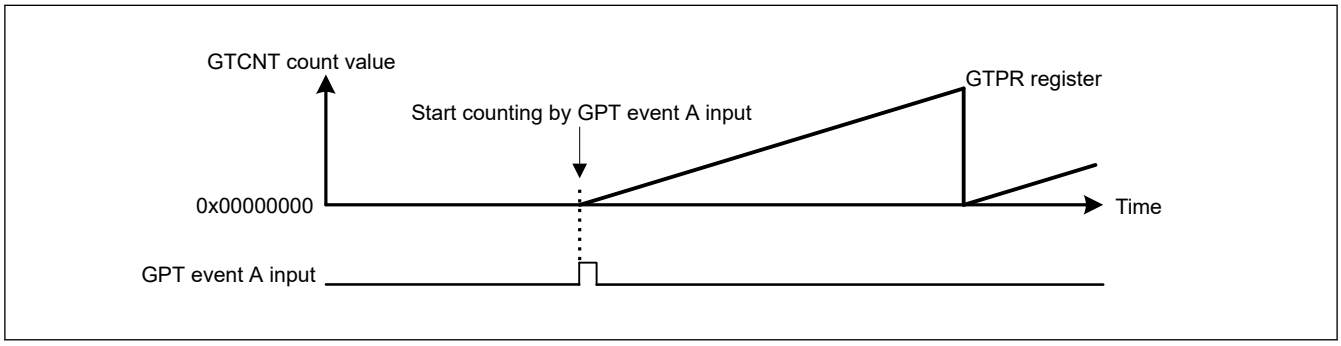


Figure 20.31 Example of count start operation by hardware source (at starting by GPT event A)

The following example shows setting for count start operation by hardware source.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.31, 000b (sawtooth-wave PWM mode 1) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 20.31, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
   In Figure 20.31, 0x00000000 is set.
6. Set hardware count start
   Select a hardware source to start count operation with the GTSSR register.
   In Figure 20.31, GTSSR.SSELCA bit = 1.
7. Start hardware source operation
   Set operation of the hardware source selected by the GTSSR register and start counting.
   In Figure 20.31, the GPT event A input operation is set.

20.4.7.2 Hardware Stop Operation

The GTCNT counter can be stopped by a hardware source. Select a hardware source to stop counting with the GTPSR register, and enable to stop counting.

Figure 20.32 shows an example of count stop operation by a hardware source. In this example, the count operation is stopped by the GPT event A input, and is restarted by the GPT event B input.

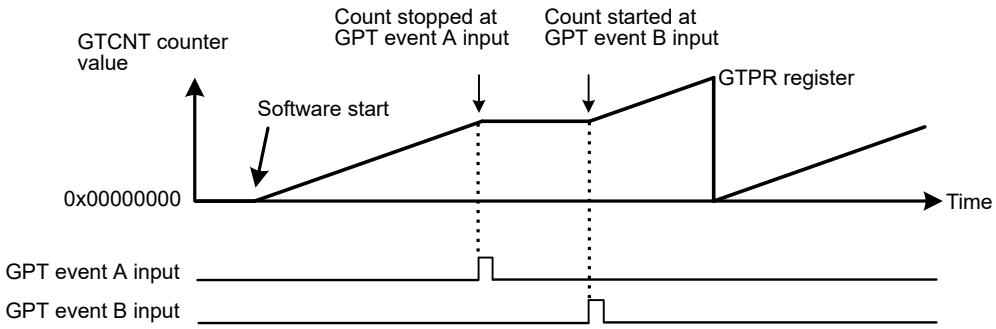


Figure 20.32 Example of count stop operation by hardware source (started by software, stopped at GPT event A input, and restarted at GPT event B input)

The following example shows setting for count stop operation by hardware source.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.32, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 20.32, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.

5. Set initial value for counter
   Set the initial value in the GTCNT counter.
   In Figure 20.32, 0x00000000 is set.
6. Set hardware count start
   Select a hardware source for starting count operation with the GTSSR register.
   In Figure 20.32, GTSSR.SSELCB bit = 1.
7. Set hardware count stop
   Select a hardware source for starting count operation with the GTPSR register.
   In Figure 20.32, GTPSR.PSELCA bit = 1.
8. Start hardware source operation
   Set operation of the hardware source selected by the GTSSR or GTPSR register then start and stop counting.
   In Figure 20.32, the GPT event A and GPT event B input operations are set.

Figure 20.33 shows an example of count start/stop operation by a hardware source.

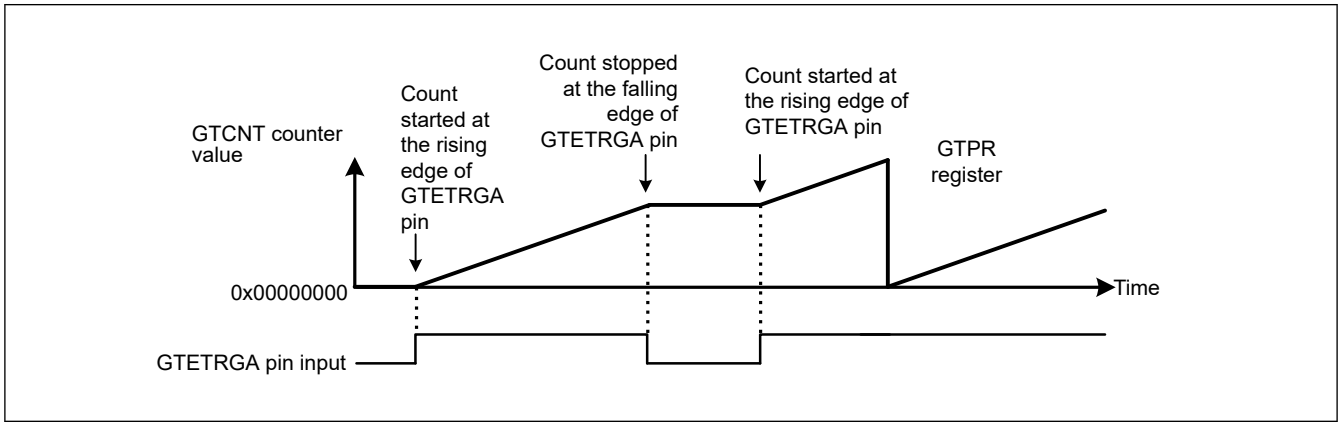


Figure 20.33 Example of count start/stop operation by hardware source (started at rising edge of GTETRGA pin input, stopped at falling edge of GTETRGA pin input)

The following example shows setting for count start/stop operation by hardware source.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.33, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 20.33, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
   In Figure 20.33, 0x00000000 is set.
6. Set hardware count start
   Select a hardware source for starting count operation with the GTSSR register.
   In Figure 20.33, GTSSR.SSGTRGAFR[1:0] bits = 01b.
7. Set hardware count stop
   Select a hardware source for stopping count operation with the GTPSR register.
   In Figure 20.33, GTPSR.PSGTRGAFR[1:0] bits = 10b.
8. Start hardware source operation
   Set operation of the hardware source selected by the GTSSR or GTPSR register then start and stop counting.
   In Figure 20.33, the GTETRGA pin operation is set.

20.4.7.3 Hardware Clear Operation

The GTCNT counter can be cleared by a hardware source. Select a hardware source to clear the counter with the GTCSR register, and enable to clear the counter.

Note: The GPTm_n_OVF/GPTm_n_UDF interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 20.34 and Figure 20.35 show examples of the GTCNT counter clearing operation by a hardware source.

In this example, the GTCNT counter is started at the GPT event A input, and is stopped/cleared at the GPT event B input.

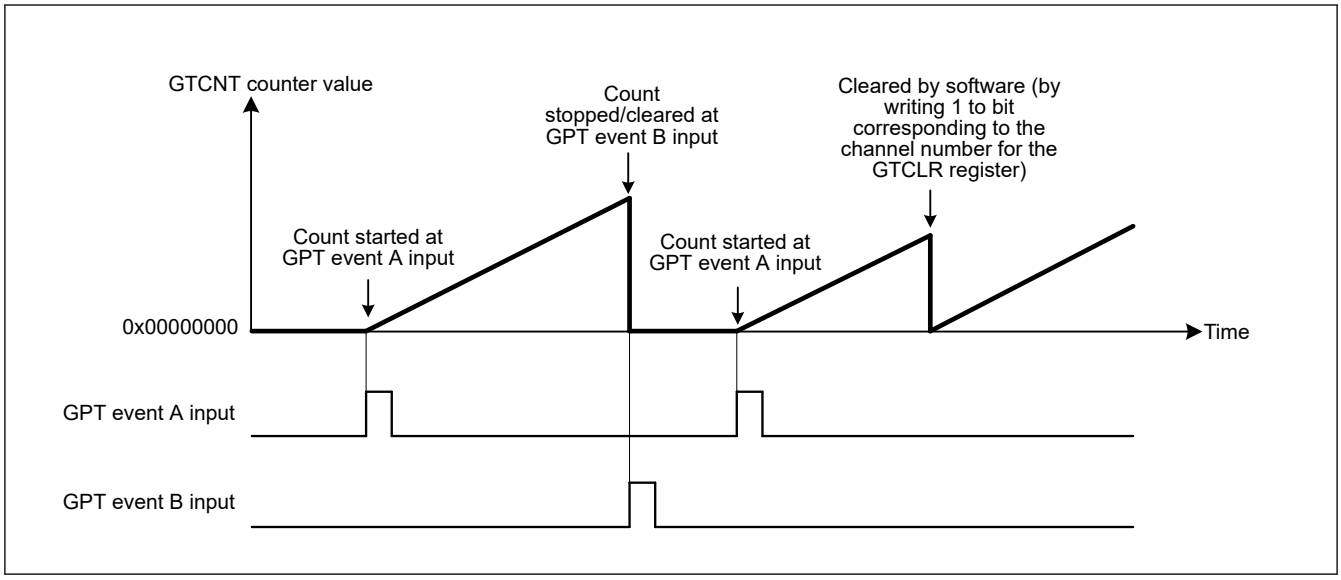


Figure 20.34 Examples of counter clearing operation by hardware source (sawtooth-wave up-counting, started at GPT event A input, and count stopped/counter cleared at GPT event B input)

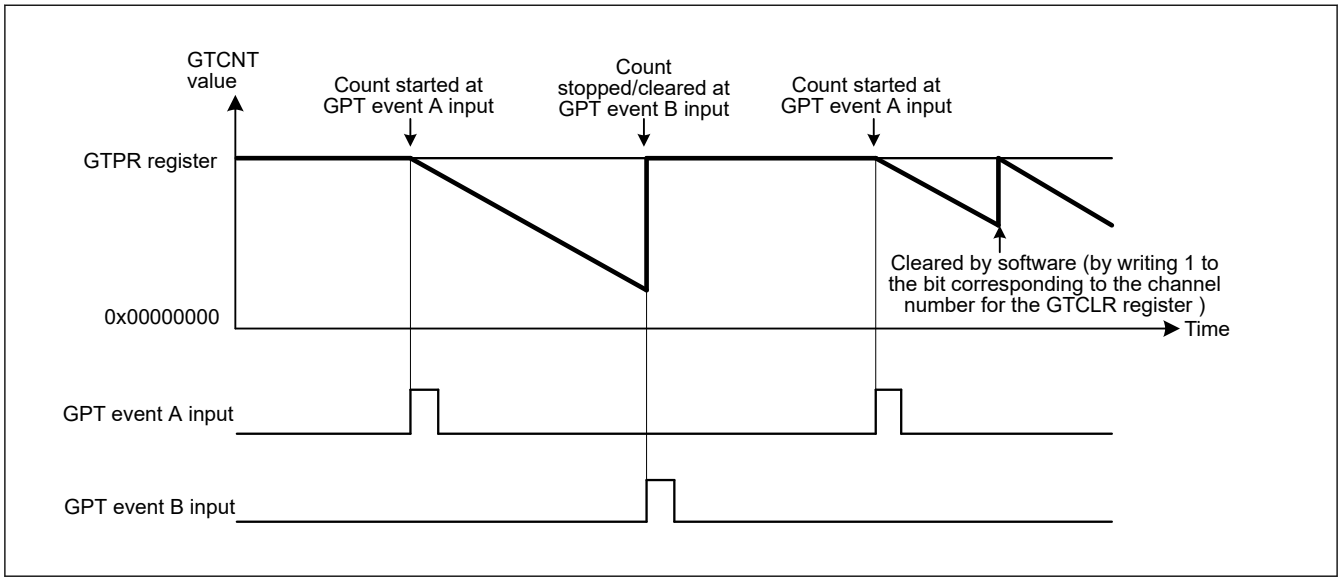


Figure 20.35 Examples of counter clearing operation by hardware source (sawtooth-wave down-counting, started at GPT event A input, and count stopped/counter cleared at GPT event B input)

The following example shows setting for counter clearing operation by hardware source.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 20.34 and Figure 20.35, 000b (sawtooth-wave PWM mode 1) is set.
2. Set count direction

Select the count direction (up or down) with the GTUDDTYC register.
In Figure 20.34, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
In Figure 20.35, the lower 2 bits of the GTUDDTYC register is set to 10b and then to 00b for down-counting.

3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
   In Figure 20.34, 0x00000000 is set.
   In Figure 20.35, the GTPR register value is set.
6. Set hardware count start
   Select a hardware source for starting count operation with the GTSSR register and wait for count start by the hardware source.
   In Figure 20.34 and Figure 20.35, GTSSR.SSELCA bit = 1.
7. Set hardware count stop
   Select a hardware source for stopping count operation with the GTPSR register and wait for count stop by the hardware source.
   In Figure 20.34 and Figure 20.35, GTPSR.PSELCB bit = 1.
8. Set hardware count clear
   Select a hardware source to clear the counter with the GTCSR register and wait for count clearing by the hardware source.
   In Figure 20.34 and Figure 20.35, GTPSR.CSELCB bit = 1.
9. Start hardware source operation
   Set operation of the hardware source selected by the GTSSR, GTPSR, or GTCSR register then start, stop, or clear counting.
   In Figure 20.34 and Figure 20.35, the GPT event A and B inputs operations are set.

The GPTm_n_OVF/GPTm_n_UDF interrupt (overflow/underflow interrupt) does not occur when the counter is cleared by a hardware source. In the same way, the GPTm_n_OVF/GPTm_n_UDF interrupt does not occur when the counter is cleared by software.

Figure 20.36 shows the relationship between counter clearing by a hardware source and the GPTm_n_OVF interrupt.

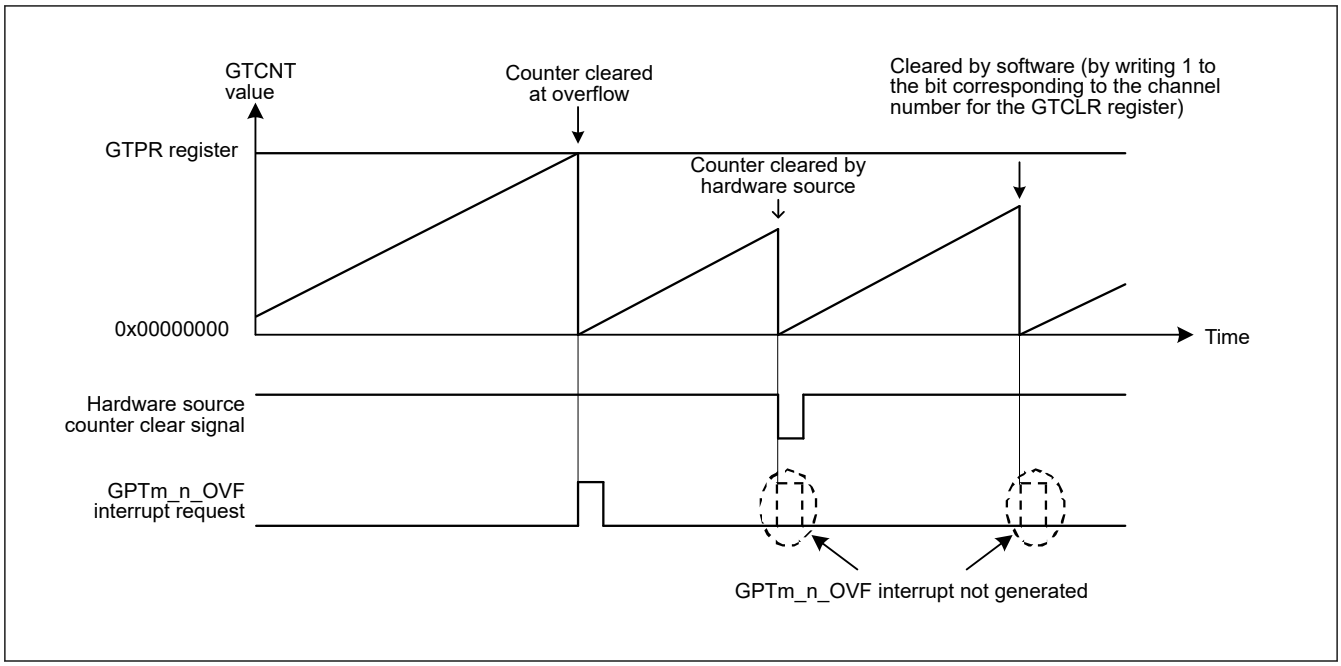


Figure 20.36 Relationship between counter clearing by hardware source and GPTm_n_OVF interrupt

20.4.8 Hardware Count Switch Operation

In the triangle-wave PWM mode, the GTCNT counter can be switched by a hardware source. Select the count switch operation with the GTCR.SWMD[2:0] bits and select a hardware source to switch the counter with the GTSWSR register and enable to switch the counter.

When the offset reload is enabled by the GTCR.SWMD[2:0] bits, the offset value of the GTCNT counter reloaded by count switch is set by the GTSWOS register in the range of 0 to GTPR.

If the count switch and counter clear occur at the same time, the counter clear has priority.

Figure 20.37 to Figure 20.40 show examples of the GTCNT count switch operation by a hardware source.

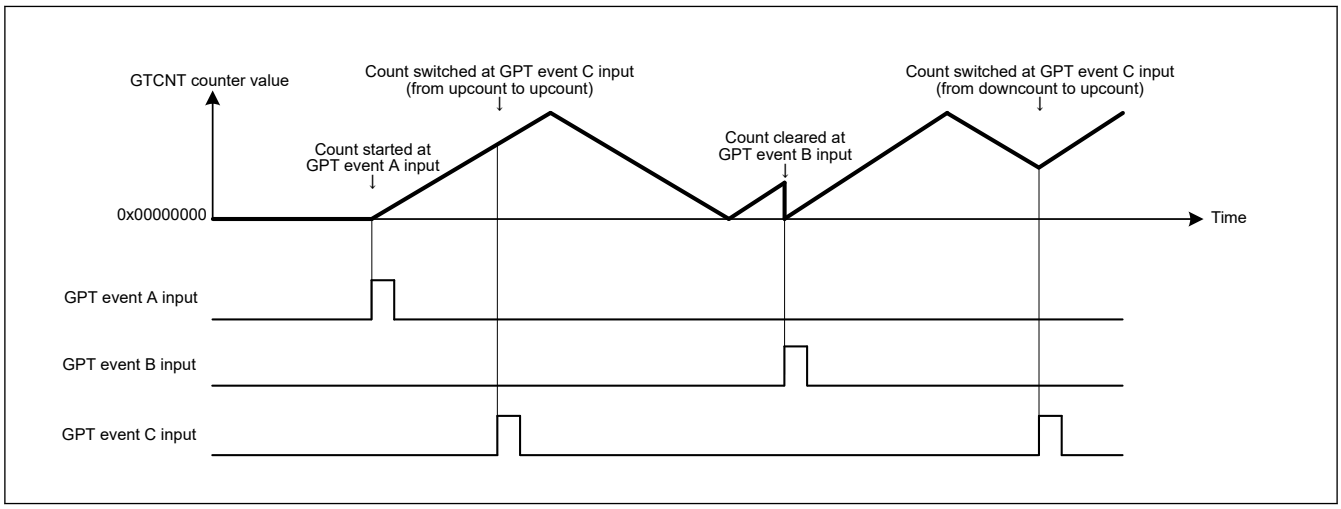


Figure 20.37 Examples of count switch operation by hardware source (switched to upcount without reload at GPT event C input)

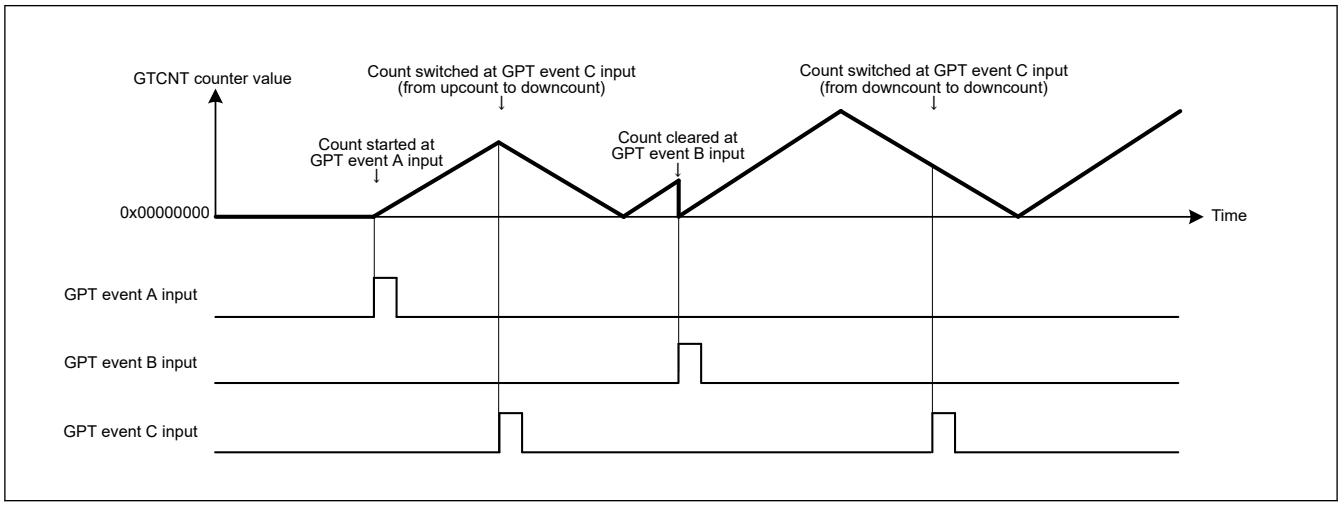


Figure 20.38 Examples of count switch operation by hardware source (switched to downcount without reload at GPT event C input)

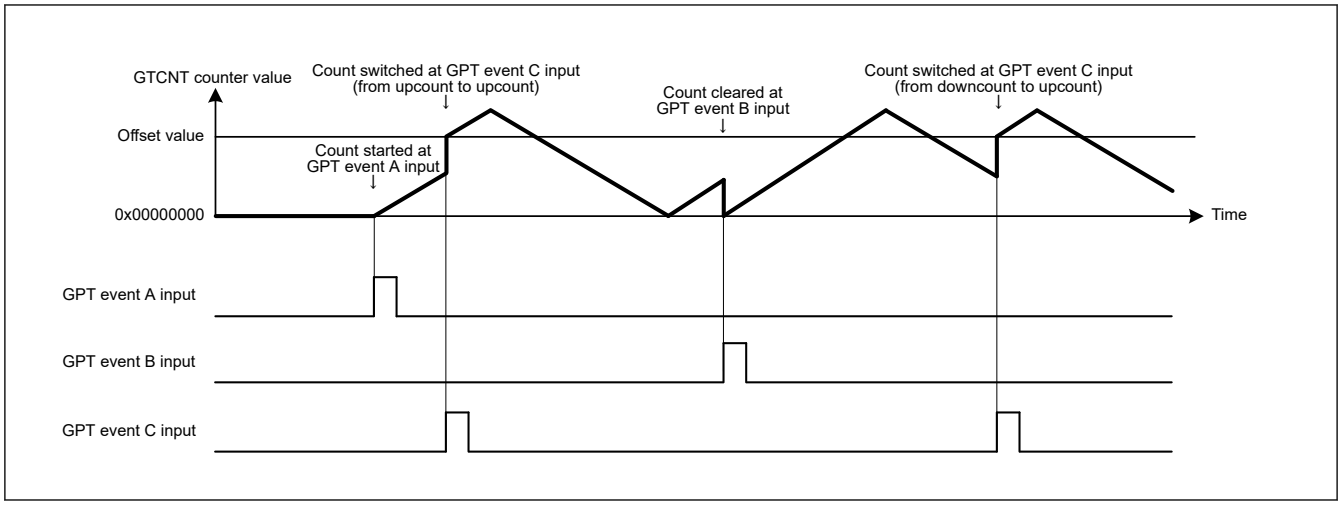


Figure 20.39 Examples of count switch operation by hardware source (switched to upcount with reload at GPT event C input)

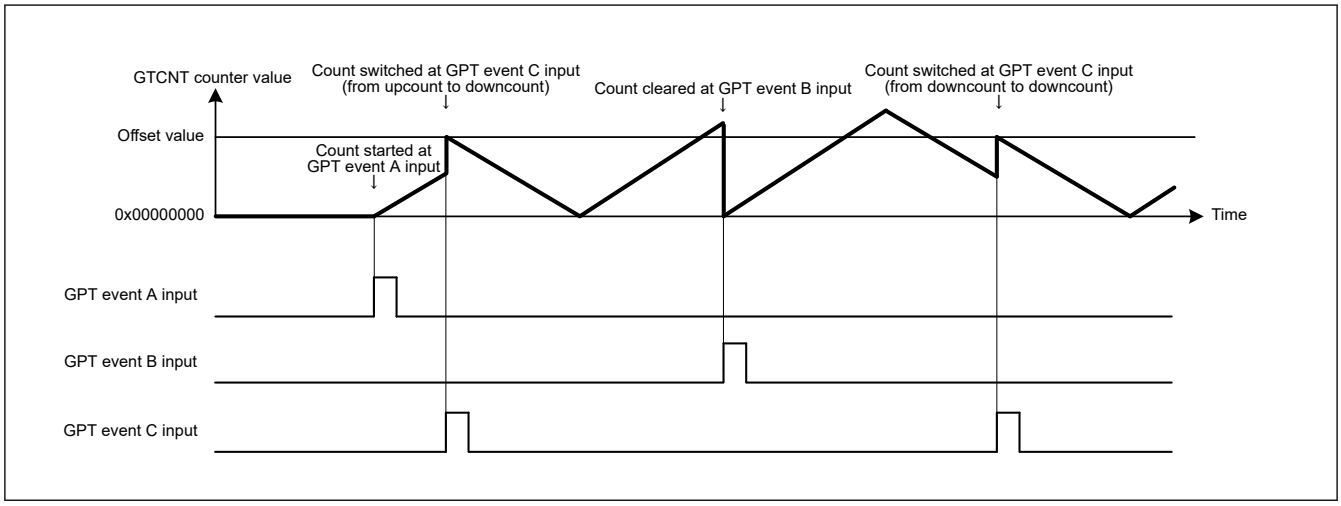


Figure 20.40 Examples of count switch operation by hardware source (switched to downcount with reload at GPT event C input)

The following procedure shows the setting example. In this example, the GTCNT counter started at the GPT event A input, is stopped/cleared at the GPT event B input, and is switched at the GPT event C input.

1. Set operating mode
   Set the operating mode to the triangle-wave PWM mode with the GTCR.MD[2:0] bits.
   In Figure 20.37, Figure 20.38, Figure 20.39, and Figure 20.40, 100b (triangle-wave PWM mode 1) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
3. Period setting
   Set a period in the GTPR register.
4. Set initial value for counter
   Set the initial value in the GTCNT counter.
   In Figure 20.37, Figure 20.38, Figure 20.39, and Figure 20.40, 0x00000000 is set.
5. Set switch mode
   Set the switch mode with the GTCR.SWMD[2:0] bits.
   In Figure 20.37, 101b is set, in Figure 20.38, 110b is set, in Figure 20.39, 001b is set, in Figure 20.40, 010b is set.
6. Set hardware count start
   Select a hardware source for starting count operation with the GTSSR register and wait for count start by the hardware source.
   In Figure 20.37, Figure 20.38, Figure 20.39, and Figure 20.40, GTSSR.SSELCA bit = 1.

7. Set hardware count stop
   Select a hardware source for stopping count operation with the GTPSR register and wait for count stop by the hardware source.
8. Set hardware count clear
   Select a hardware source to clear the counter with the GTCSR register and wait for counter clearing by the hardware source.
   In Figure 20.37, Figure 20.38, Figure 20.39, and Figure 20.40, GTCSR.CSELCB bit = 1.
9. Set hardware count switch
   Select a hardware source to switch the counter with the GTSWSR register and wait for count switch by the hardware source.
   In Figure 20.37, Figure 20.38, Figure 20.39, and Figure 20.40, GTSWSR.WSELCC bit = 1.
10. Set hardware source operation
    Set operation of the hardware source selected with the GTSSR register, GTPSR register, GTCSR register, or GTSWSR register and start, stop, clear, or switch the counter.
    In Figure 20.37, Figure 20.38, Figure 20.39, and Figure 20.40, the GPT event A, GPT event B, and GPT event C input operations are set.

The compare match operation is valid even when the count switch and the compare match occur at the same time. Similarly, even when the count switch occurs at the crest (GTCNT counter matches GTPR register value during upcounting) or trough (GTCNT counter is 0 during downcounting) of triangle-waves, overflow or underflow interrupt is output and the buffer transfer of crest or trough functions normally.

If the count switch is performed at the crest or trough without the offset reload, the crest or trough occurs and then the count direction changes to downcount or upcount regardless of the count switch setting. Also, if the count switch is performed at the count clear state (GTCNT counter value is 0 during upcount) without the offset reload, the count direction continues upcount regardless of the count switch setting.

If the switch mode is set to Switch to upcount with offset reload with the SWMD[2:0] bits and the offset is set to 0 with the GTSWOS register, the count switch functions the same as the counter clear and the GTCNT counter does not operate as the trough. Also, if the switch mode is set to Switch to downcount with offset reload and the offset is set to the GTPR value, the GTCNT counter does not operate as the crest. On the other hand, if the switch mode is set to Switch to upcount with offset reload and the offset is set to the GTPR value, the GTCNT counter operates as the crest, if the switch mode is set to Switch to downcount with offset reload and the offset is set to 0, the GTCNT counter operates as the trough, and in these cases, the interrupt outputs and the buffer transfers of the crest and the trough functions normally.

If the crest or the trough due to the count switch occurs at the timing when the GTCNT counter value is the crest or the trough, the buffer transfer and the interrupt output at the crest or trough will occur twice in a row.

Table 20.12 Triangle-waves condition after the count switch with offset reload

Count direction of switch mode (GTCR.SWMD[2:0])	Offset (GTSWOS)	The triangle-waves condition after the count switch			
		GTCNT	GTST.TUCF	Triangle-waves condition	Buffer transfer/ Interrupt output
Upcount	0	GTSWOS register value	1	Counter clear (start of upcount)	Nothing
	1 to GTPR - 1			Upcount	Nothing
	GTPR			Crest (end of upcount)	Buffer transfer available (only Triangle-Wave PWM Mode 2)/ GPTm_n_OVF Output
Downcount	0		0	Trough (end of downcount)	Buffer transfer available/ GPTm_n_UDF Output
	1 to GTPR - 1			Downcount	Nothing
	GTPR			Start of downcount	Nothing

Figure 20.41 and Figure 20.42 show examples of the buffer operation due to the count switch when the offset is set to 0 and GTPR. Figure 20.43 shows example of the buffer operation due to the count switch when the offset is set other than the crest and trough.

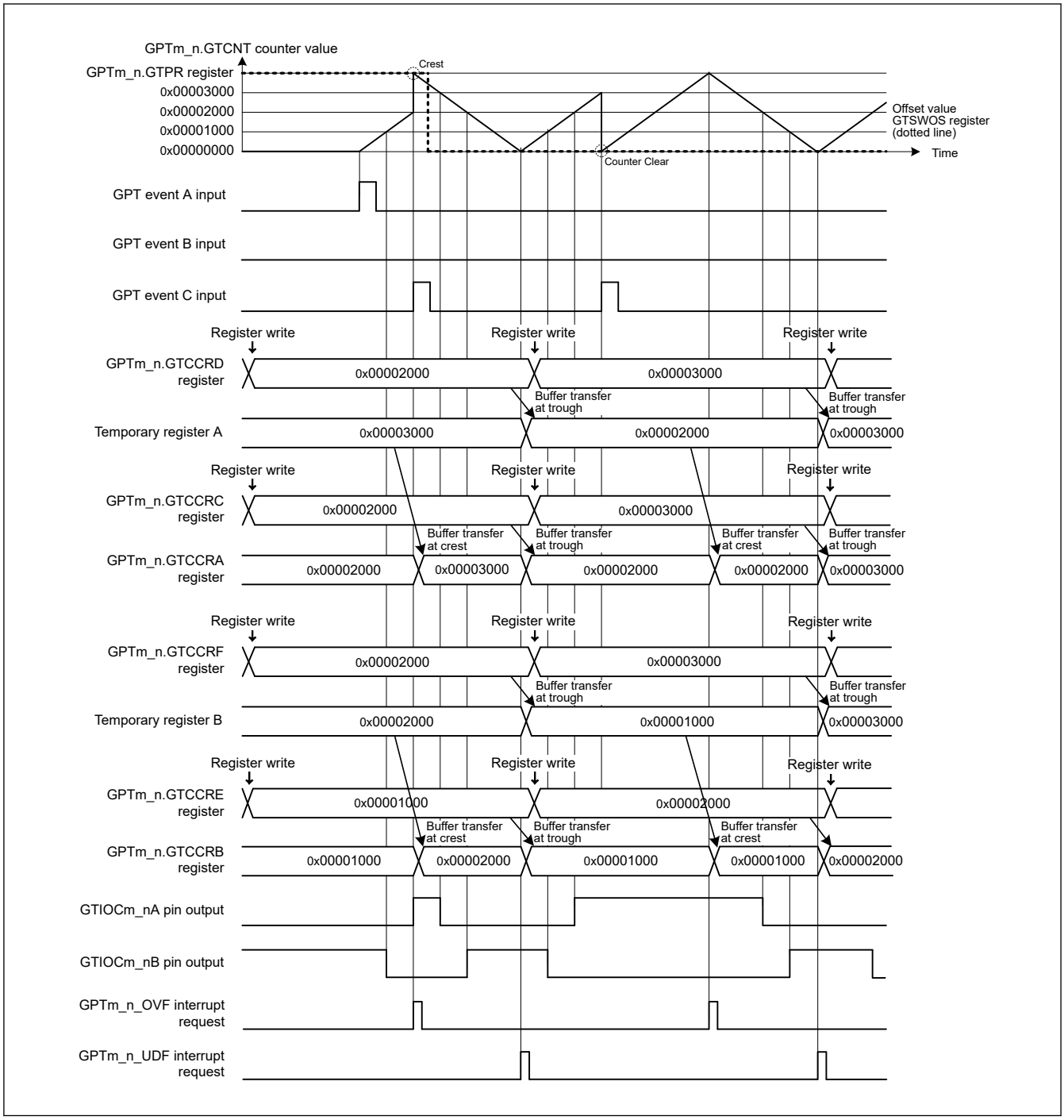


Figure 20.41 Example of the buffer operation due to the count switch (triangle-wave PWM mode 3, offset is set to 0 and GTPR, and switch to upcount)

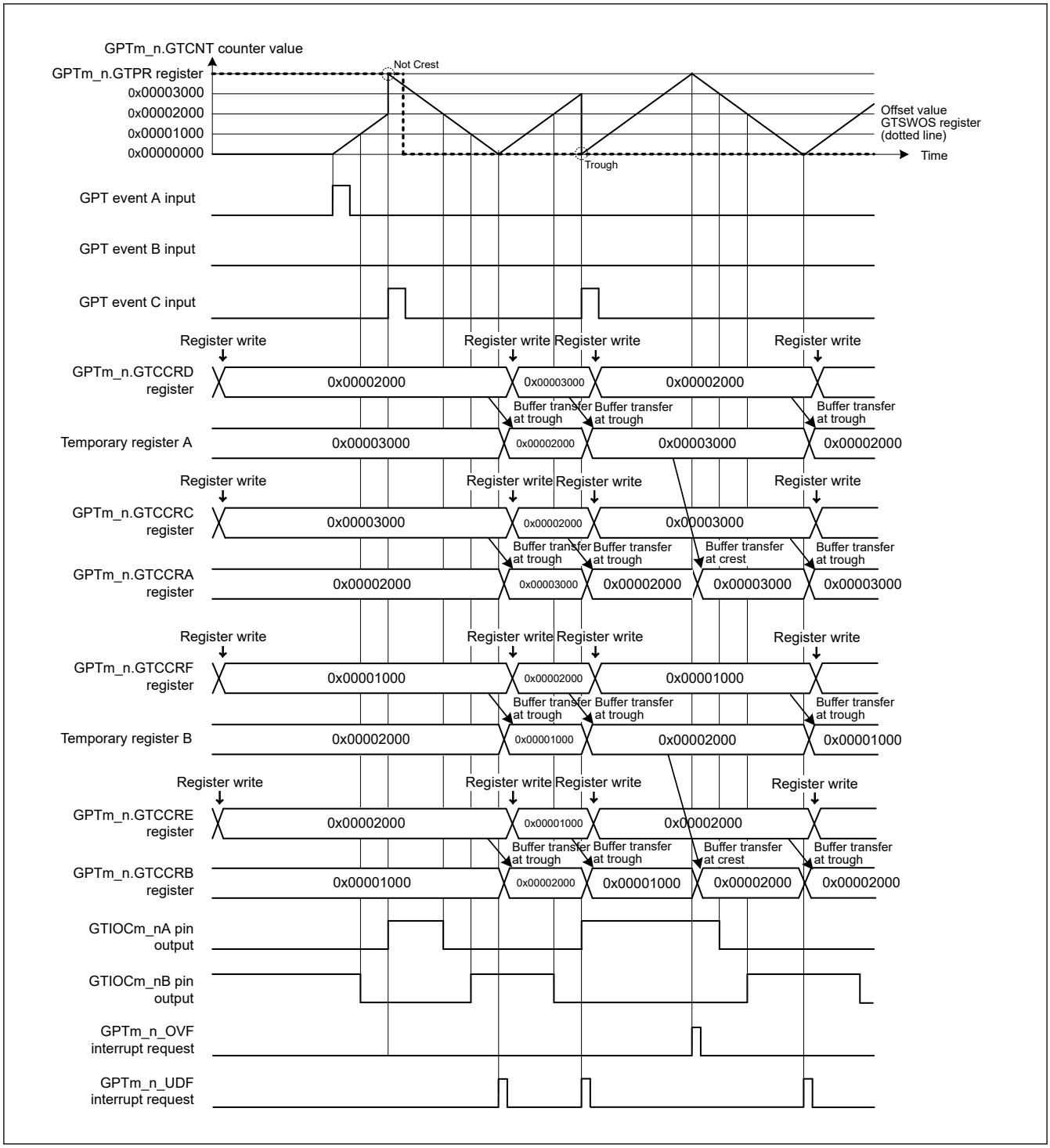


Figure 20.42 Example of the buffer operation due to the count switch (triangle-wave PWM mode 3, offset is set to 0 and GTPR, and switch to downcount)

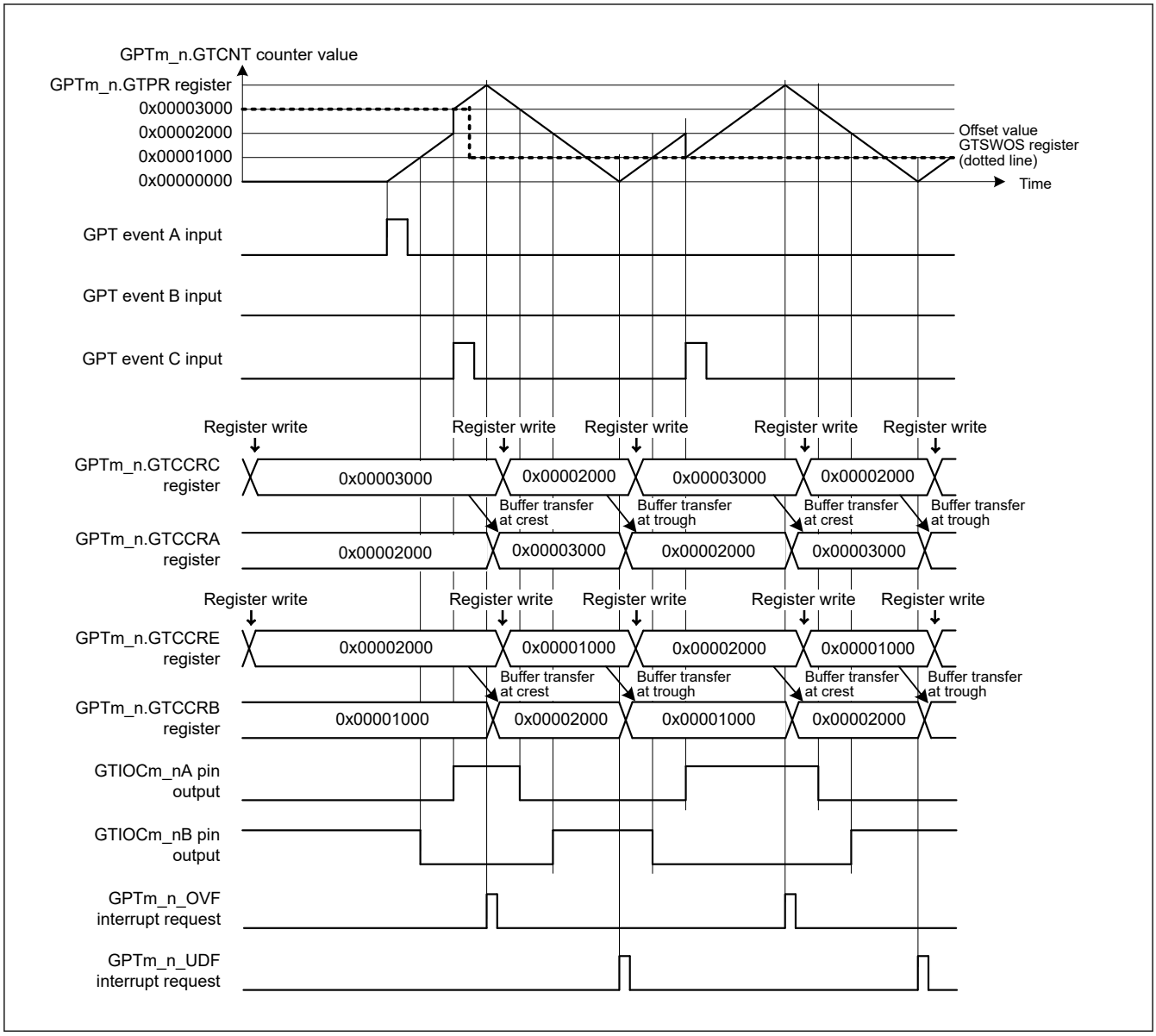


Figure 20.43 Example of the buffer operation due to the count switch (triangle-wave PWM mode 2, no crest and trough occurrence due to the count switch)

When the counter value skips the compare match value by the count switch, no compare match occurs and the GTIOCm_nA and GTIOCm_nB pin output does not change. Figure 20.44 shows example of the operation when the compare match is skipped by switch operation.

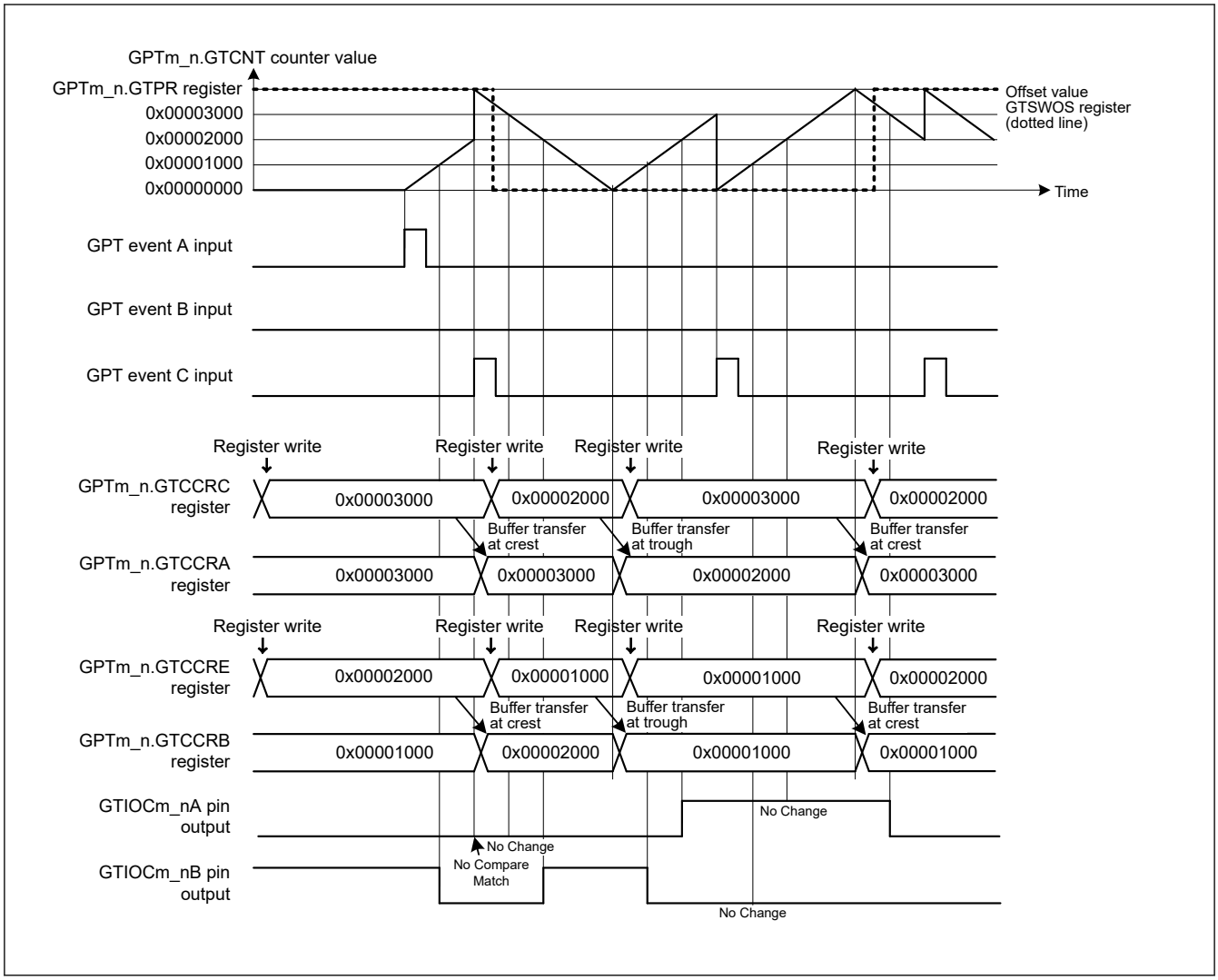


Figure 20.44 Example of the exception operation due to the count switch (switch to upcount with offset reload by GPT event C)

Figure 20.45 and Figure 20.46 show examples of the timing of operations to switch the counter in response to a rising edge of the input on the GTETRGA pin when a clock signal produced by frequency-dividing the Core clock signal is used as the counter clock for the GTCNT counter.

The GTCNT counter is switched when counting is in progress after the GPT has detected the internal switching signal.

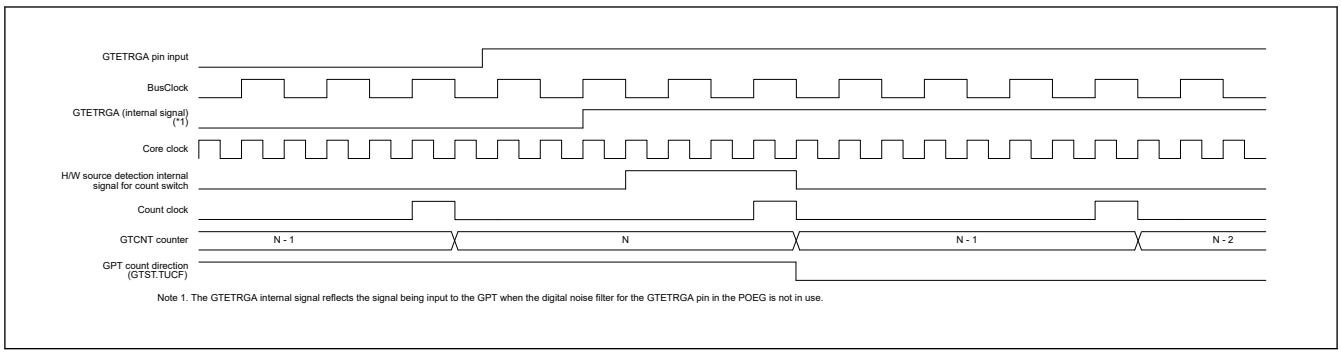


Figure 20.45 Example of the timing of operations for count switch in response to a rising edge of the input on the GTETRGA pin (switch to downcount without offset reload during the counting)

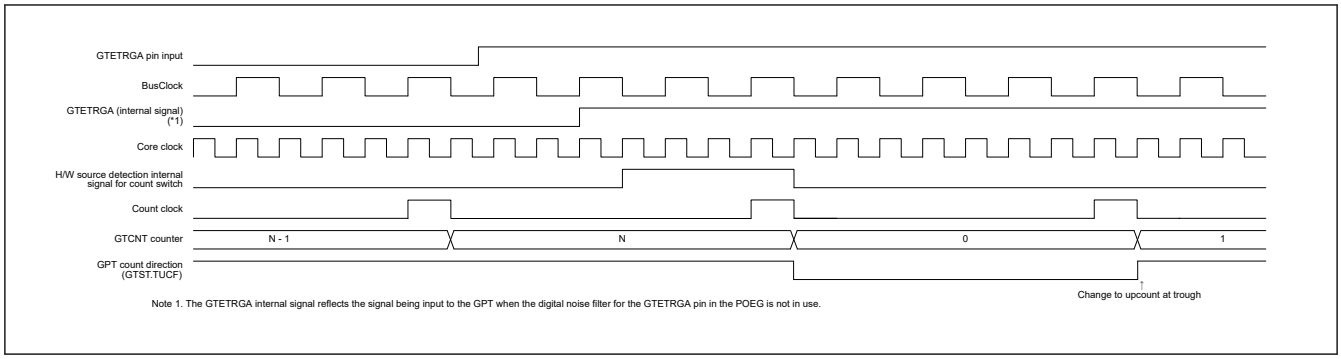


Figure 20.46 Examples of the Timing of Operations for Count Switch in Response to a Rising Edge of the Input on the GTETRGA Pin (switch to downcount with offset reload during the counting and offset value is 0)

Figure 20.47 and Figure 20.48 show examples of the timing of count switch operations in response to the input of an event signal from the GPT event A when counting is triggered by a hardware source.

This is an example of operations to switch the GPT00_1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT00_0.GTCCRA register. This is selected as a trigger for output to the GPT by the ELC as event source A.

The GPT00_0 compare match A signal is synchronized with Core clock. The ELC receives the signal in synchronization with BusClock, and outputs the GPT event source A signal after 1 cycle of BusClock has elapsed. The GTCNT counter is switched in synchronization with Core clock after the GPT has detected the internal switching signal.

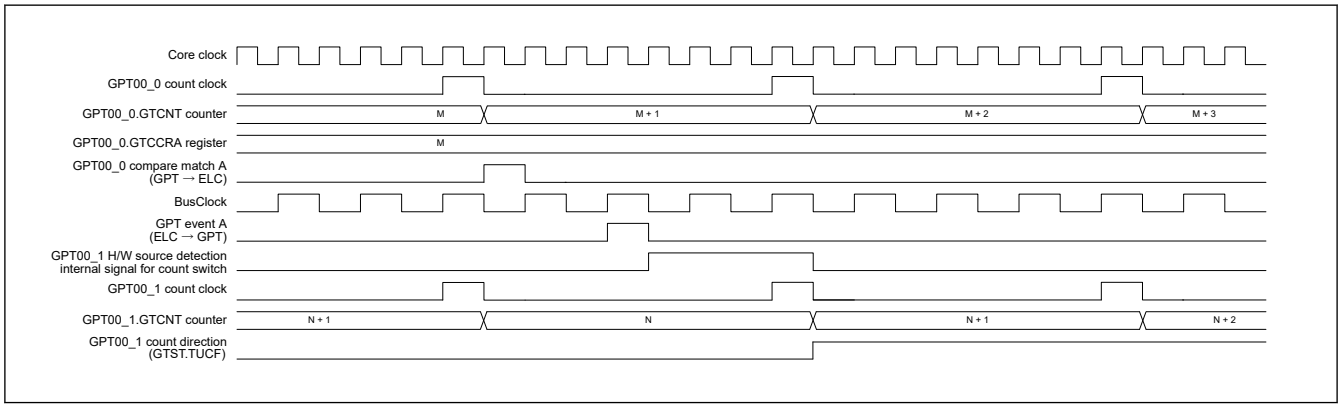


Figure 20.47 Examples of the Timing of Count Switch Operations in Response to Event Input from the GPT event A (switch to upcount without offset reload during the counting)

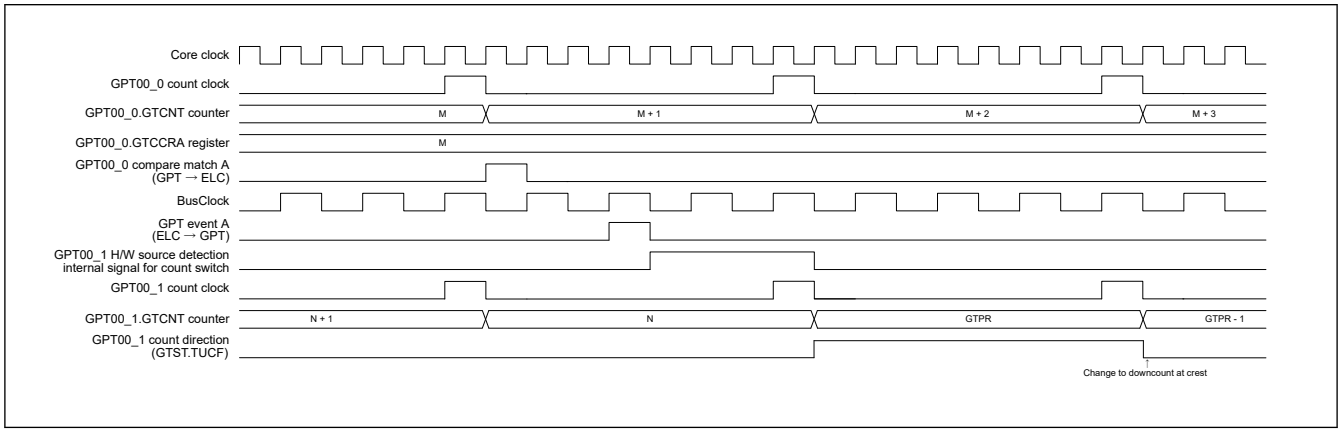


Figure 20.48 Examples of the Timing of Count Switch Operations in Response to Event Input from the GPT event A (switch to upcount with offset reload during the counting and offset value is GTPR)

20.4.9 Synchronous Operation

Synchronous operation on channels such as synchronous start/stop/clear/switch operation can be performed.

20.4.9.1 Synchronous Operation by Software

The count operations for respective channels can be started, stopped and cleared simultaneously by setting multiple bits of the GTSTR, GTSTP, and GTCLR registers to 1 at the same time.

Count start with phase differences among channels is possible by setting the GTCNT counter value before counting starts and then by setting simultaneously multiple bits in the GTSTR register to 1.

The Figure 20.49 shows an example of simultaneous start/stop/clear of four channels by software, and Figure 20.50 shows an example of phase shift start among four channels by software.

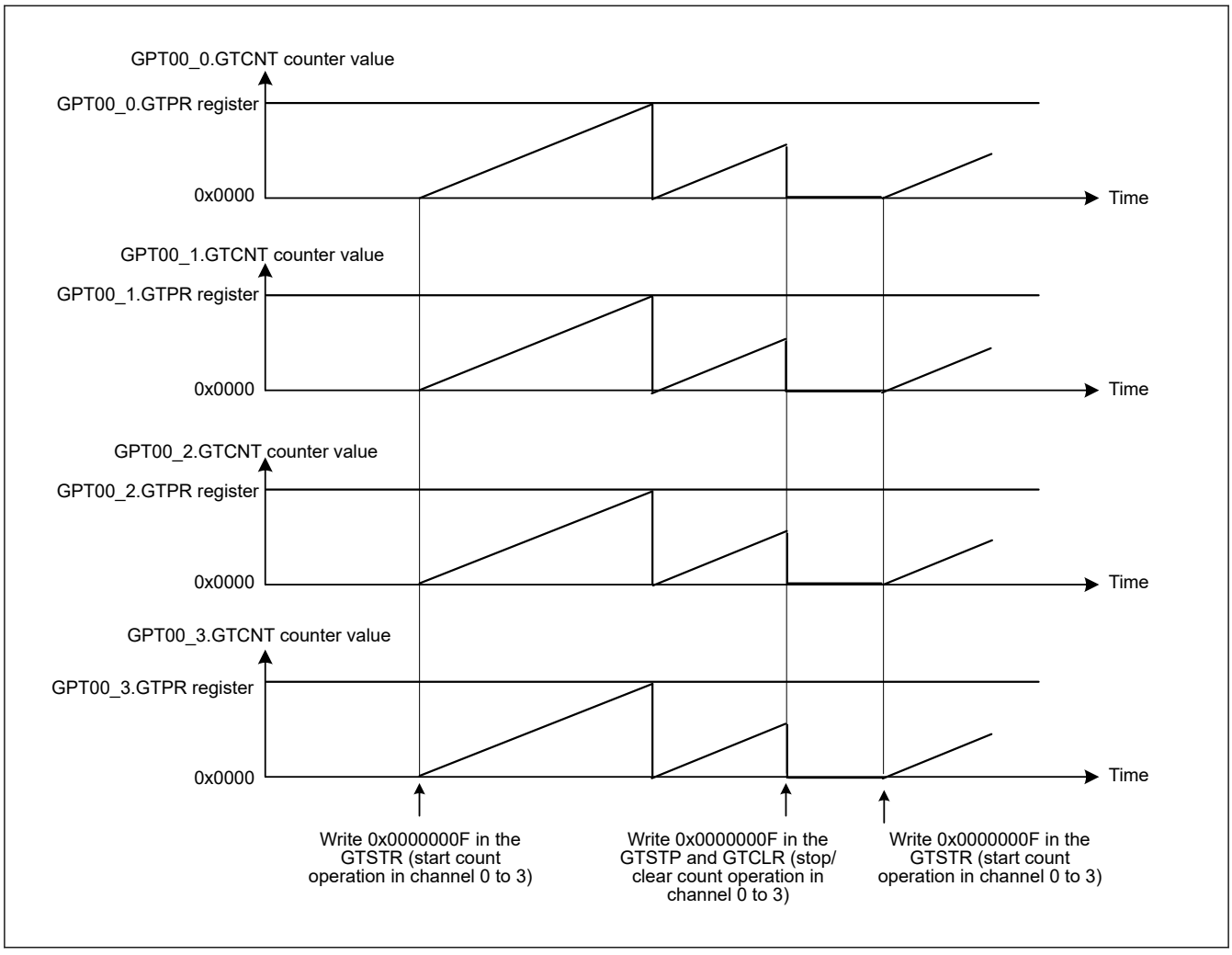


Figure 20.49 Example of simultaneous start/stop/clear operation by software with the same count period (GTPR register value)

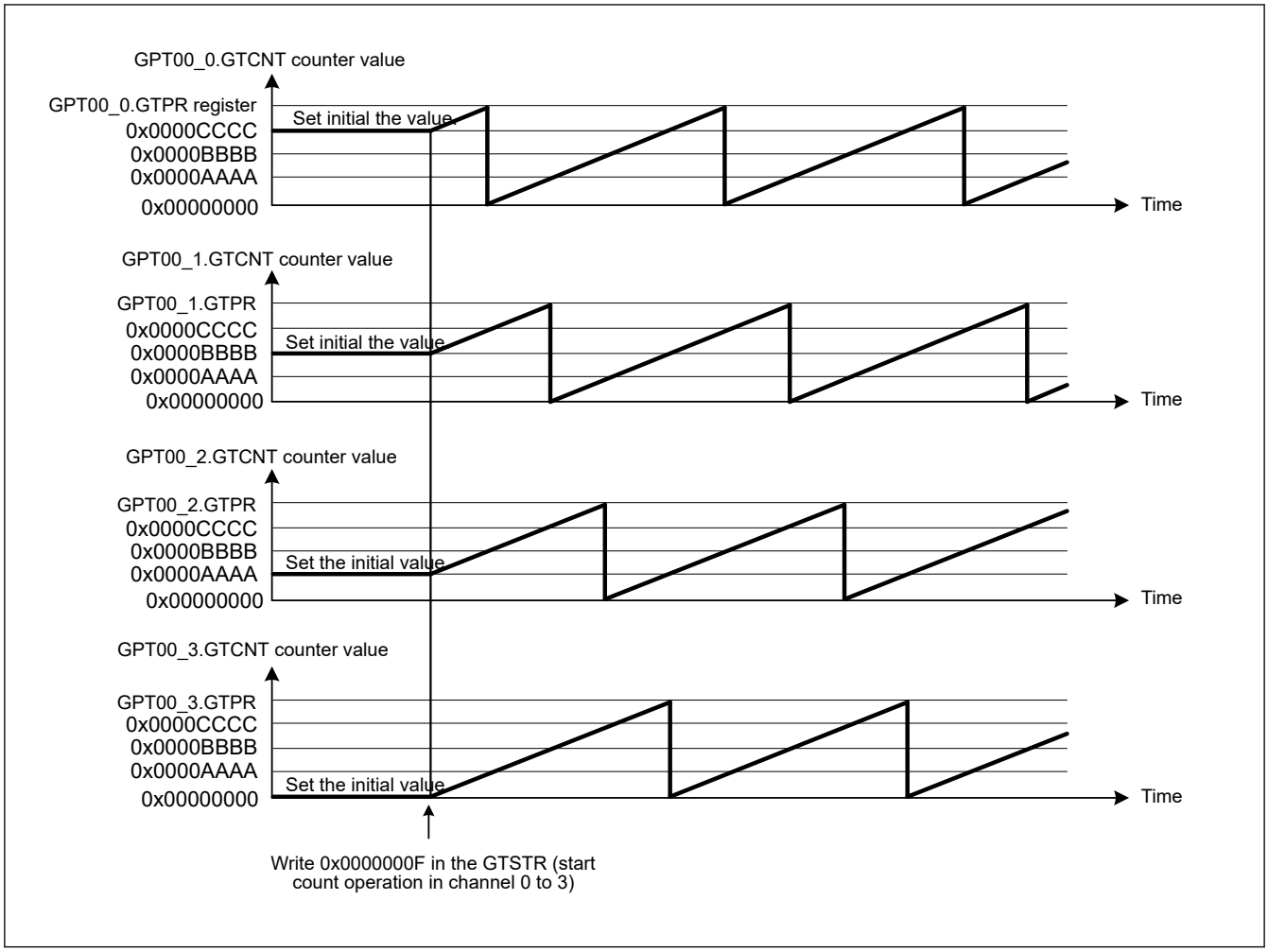


Figure 20.50 Example of software phase shift start with same count period (GTPR register value)

20.4.9.2 Synchronous Operation by Hardware Source

The count operations for respective channels can be started/stopped/cleared/switched simultaneously by hardware sources such as external trigger input and ELC event input.

Synchronous operation by the GTIOCm_nA and GTIOCm_nB pin inputs ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)) can be performed by setting the ELC event with input capture as a hardware source.

Figure 20.51 shows an example of simultaneous start/stop/clear/switch operation of four channels by hardware source.

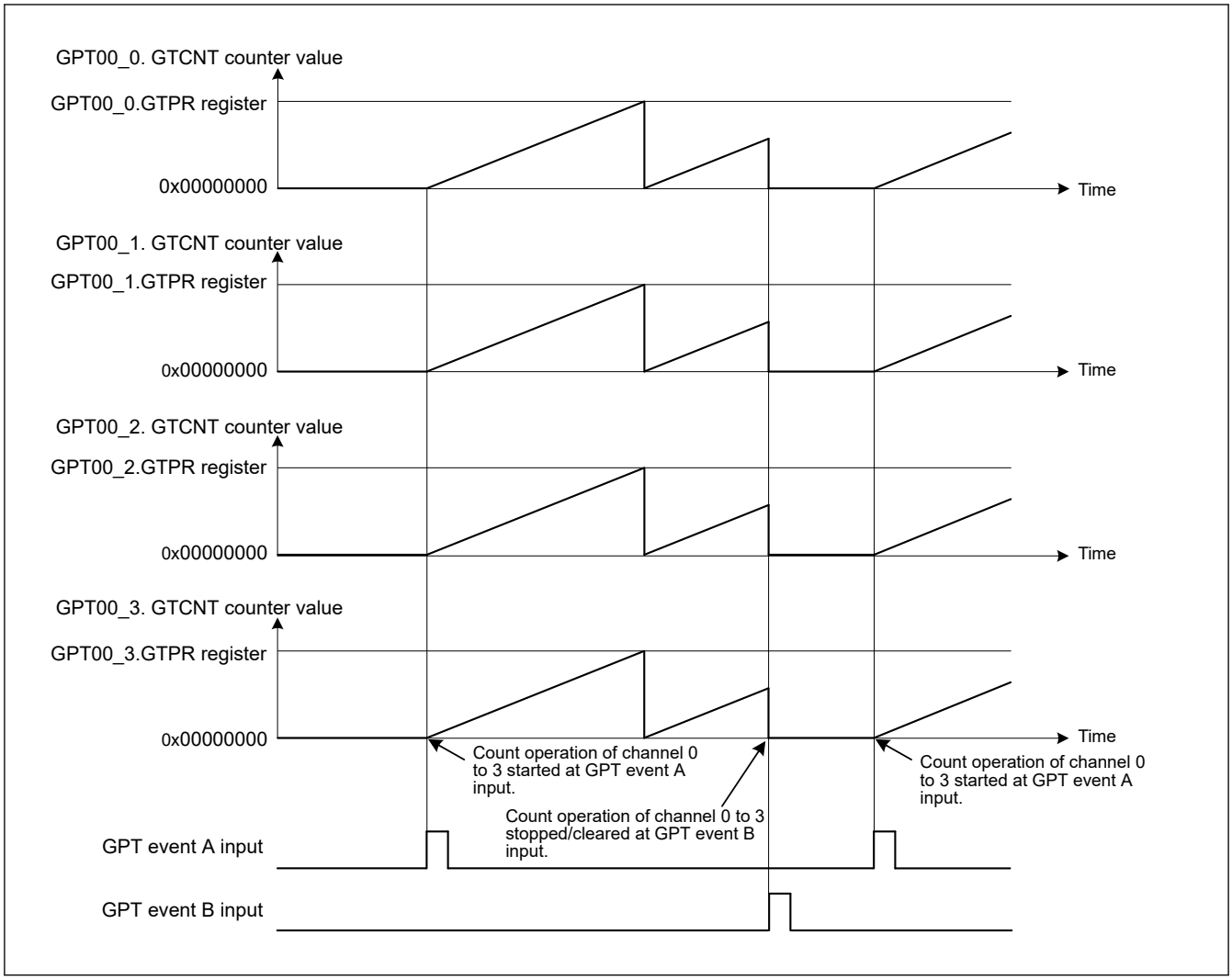


Figure 20.51 Example of simultaneous start/stop/clear operation by hardware source with same count period (GTPR register value)

The following example shows setting for simultaneous start by hardware source.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits of the corresponding channel.
   In Figure 20.51, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 20.51, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits of the corresponding channel.
4. Period setting
   Set a period in the GTPR register of the corresponding channel.
5. Set initial value for counter
   Set the initial value in the GTCNT counter of the corresponding channel.
   In Figure 20.51, 0x00000000 is set.
6. Set hardware count start
   Select a hardware source to start the count operation with the GTSSR register and wait for count start by the hardware source.
   In Figure 20.51, GTSSR.SSELCA bit = 1.
7. Set hardware count stop

Select a hardware source to stop count operation with the GTPSR register and wait for count stop by the hardware source.
In Figure 20.51, GTPSR.PSELCB bit = 1.

8. Set hardware count clearing
Select a hardware source to clear the counter with the GTCSR register and wait for counter clear by the hardware source.
In Figure 20.51, GTCSR.CSELCB bit = 1.

9. Start hardware source operation
Set operation of the hardware source selected by the GTSSR, GTPSR, and GTCSR registers and start counting, stop counting, or clearing the counter.
In Figure 20.51, the GPT event A and B inputs are set.

20.4.10 PWM Output Operation Examples

(1) Synchronous PWM Output

The GPT can output two phases of linked PWM waveforms for one channel or 14 phases of linked PWM waveforms for a maximum of seven channels by one GPT unit or 36 phases of linked PWM waveforms for a maximum of 52 channels by plural GPT units by synchronizing operation of the channels.

Figure 20.52 shows an example in which four channels perform synchronous operation in sawtooth-wave PWM mode and eight phases of PWM waveforms are output. The GTIOC00_nA pin (n = 0 to 3) is set so that it outputs low as the initial output, high at a GTCCRA register compare match, and low at the end of the cycle. The GTIOC00_nB pin is set so that it outputs low as the initial output, high at a GTCCRB register compare match, and low at the end of the cycle.

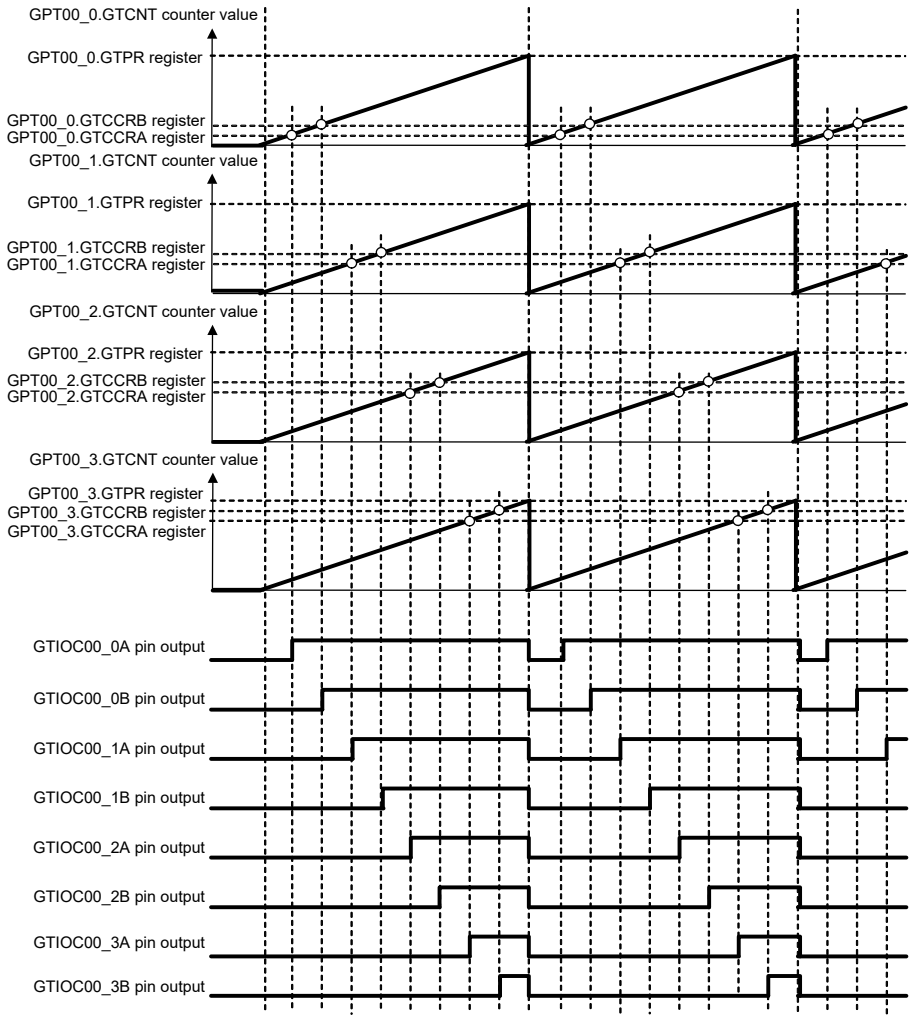


Figure 20.52 Example of synchronous PWM output

(2) 3-Phase Sawtooth-Wave Complementary PWM Output

Figure 20.53 shows an example in which three channels perform synchronous operation in sawtooth-wave PWM mode and 3-phase complementary PWM waveforms are output. The GTIOC00_nA pin (n = 0 to 2) is set so that it outputs low as the initial output, high at a GTCCRA register compare match, and low at the end of the cycle. The GTIOC00_nB pin is set so that it outputs high as the initial output, low at a GTCCRB register compare match, and high at the end of the cycle.

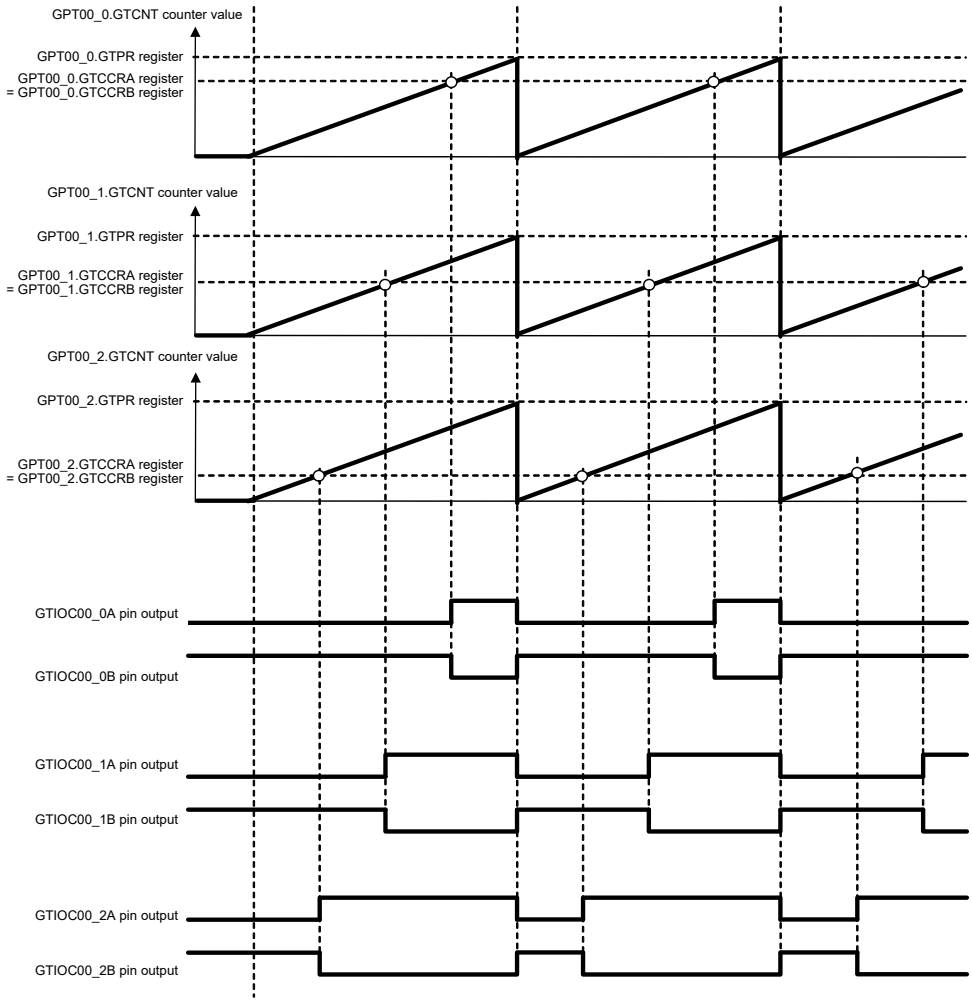


Figure 20.53 Example of 3-phase sawtooth-wave complementary PWM output

(3) 3-Phase Sawtooth-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 20.54 shows an example in which three channels perform synchronous operation in sawtooth-wave one-shot pulse mode with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOC00_nA pin (n = 0 to 2) is set so that it outputs low as the initial output, toggles the output at a GTCCRA register compare match, and retains the output at the end of the cycle. The GTIOC00_nB pin is set so that it outputs high as the initial output, toggles the output at a GTCCRB register compare match, and retains the output at the end of the cycle.

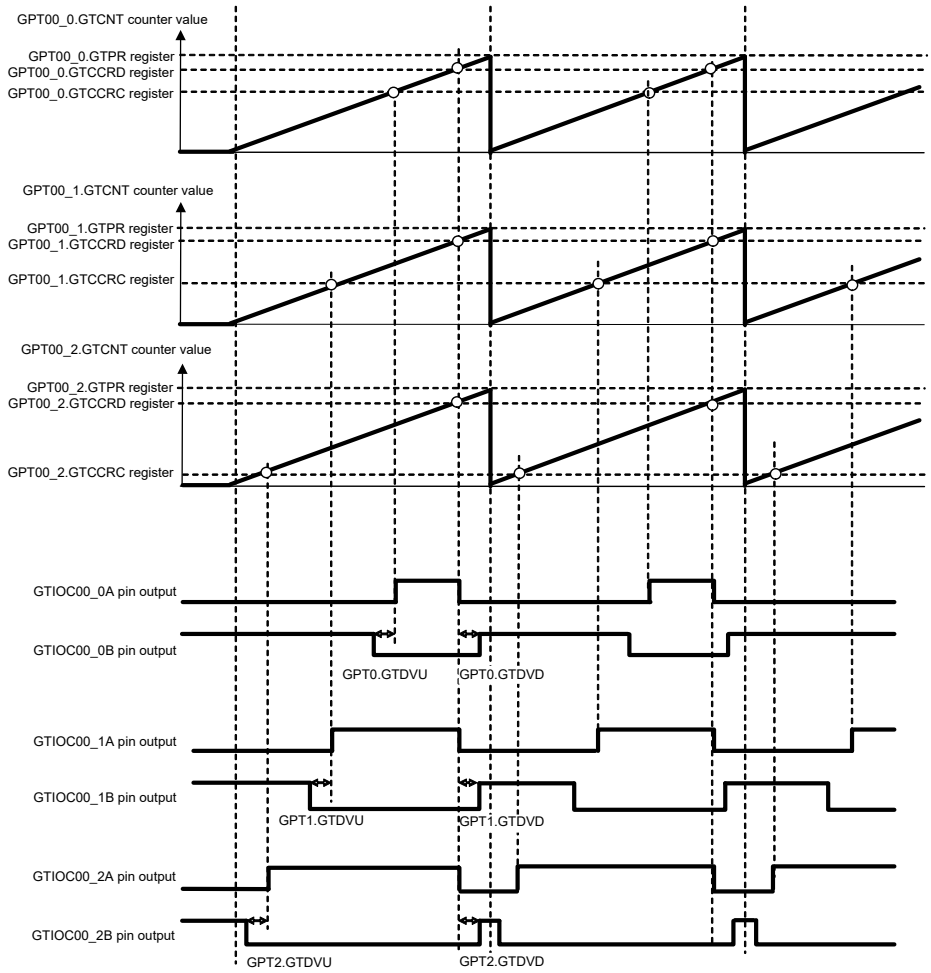


Figure 20.54 Example of 3-phase sawtooth-wave complementary PWM output with automatic dead time setting

(4) 3-Phase Triangle-Wave Complementary PWM Output

Figure 20.55 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOC00_nA pin (n = 0 to 2) is set so that it outputs low as the initial output, toggles the output at a GTCCRA register compare match, and retains the output at the end of the cycle. The GTIOC00_nB pin is set so that it outputs high as the initial output, toggles the output at a GTCCRB register compare match, and retains the output at the end of the cycle.

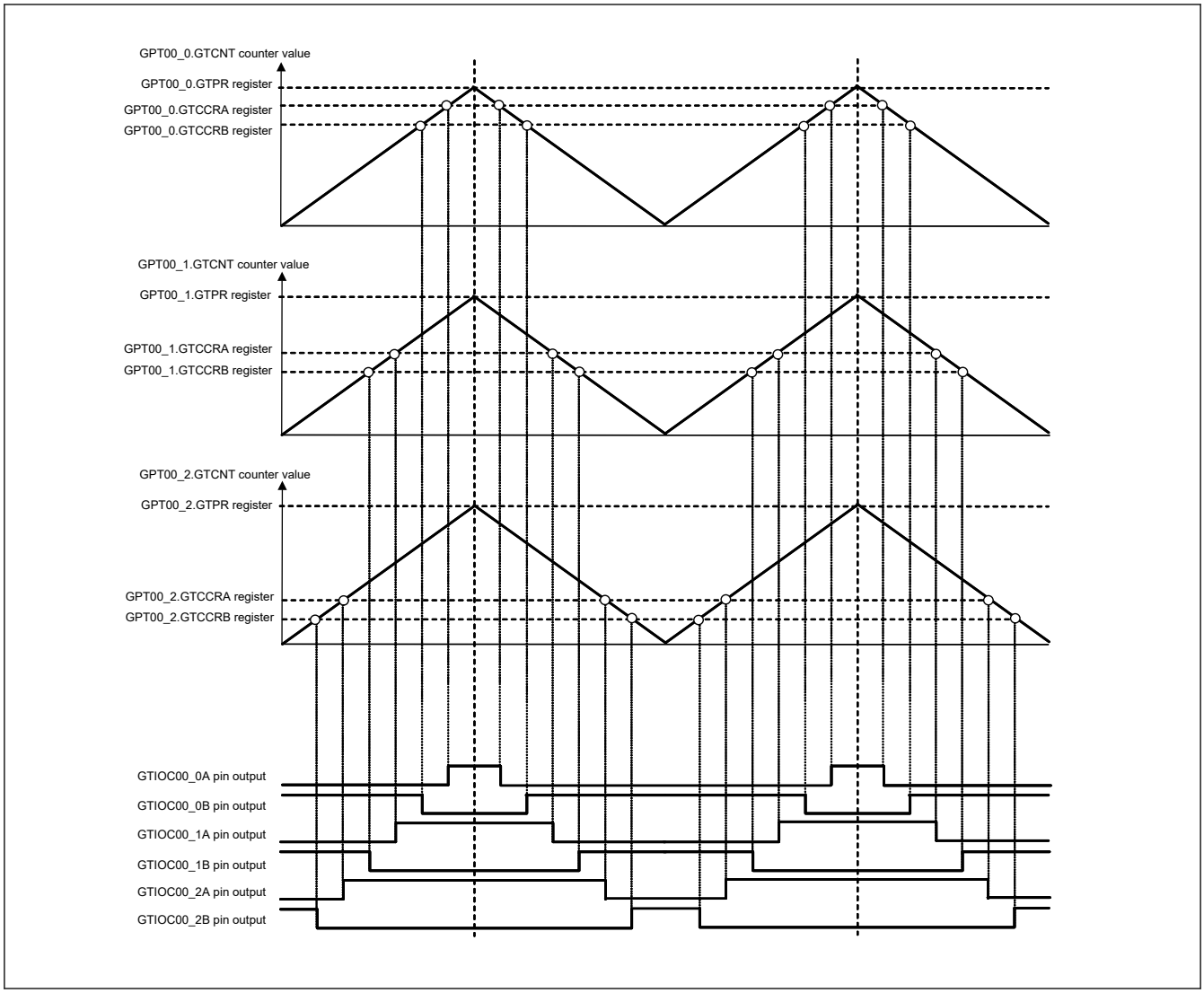


Figure 20.55 Example of 3-phase triangle-wave complementary PWM output

(5) 3-Phase Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 20.56 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOC00_nA pin (n = 0 to 2) is set so that it outputs low as the initial output, toggles the output at a GTCCRA register compare match, and retains the output at the end of the cycle. The GTIOC00_nB pin is set so that it outputs high as the initial output, toggles the output at a GTCCRB register compare match, and retains the output at the end of the cycle.

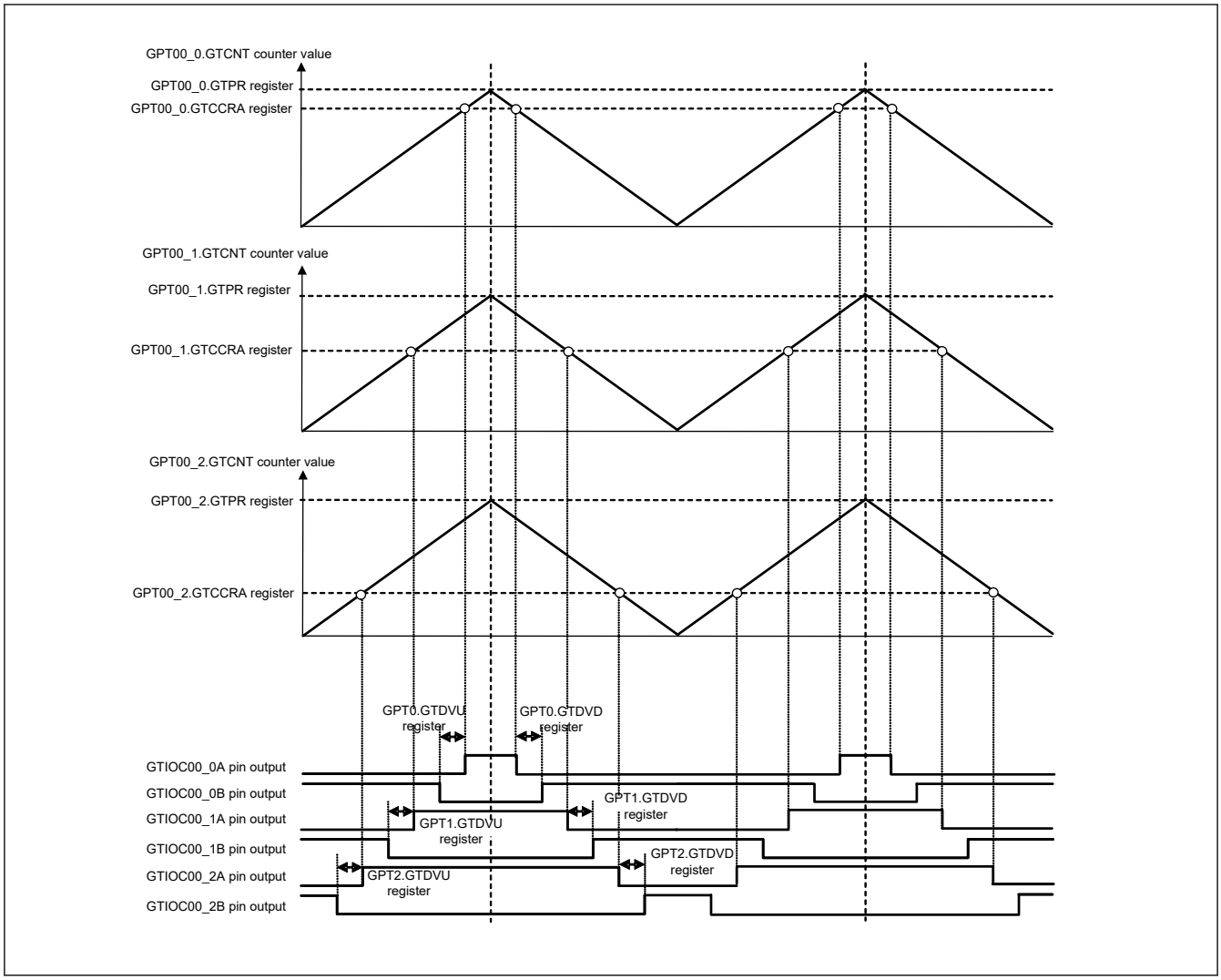


Figure 20.56 Example of 3-phase triangle-wave complementary PWM output with automatic dead time setting

(6) 3-Phase Asymmetric Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 20.57 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 3 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOC00_nA pin (n = 0 to 2) is set so that it outputs low as the initial output, toggles the output at a GTCCRA register compare match, and retains the output at the end of the cycle. The GTIOC00_nB pin is set so that it outputs high as the initial output, toggles the output at a GTCCRB register compare match, and retains the output at the end of the cycle.

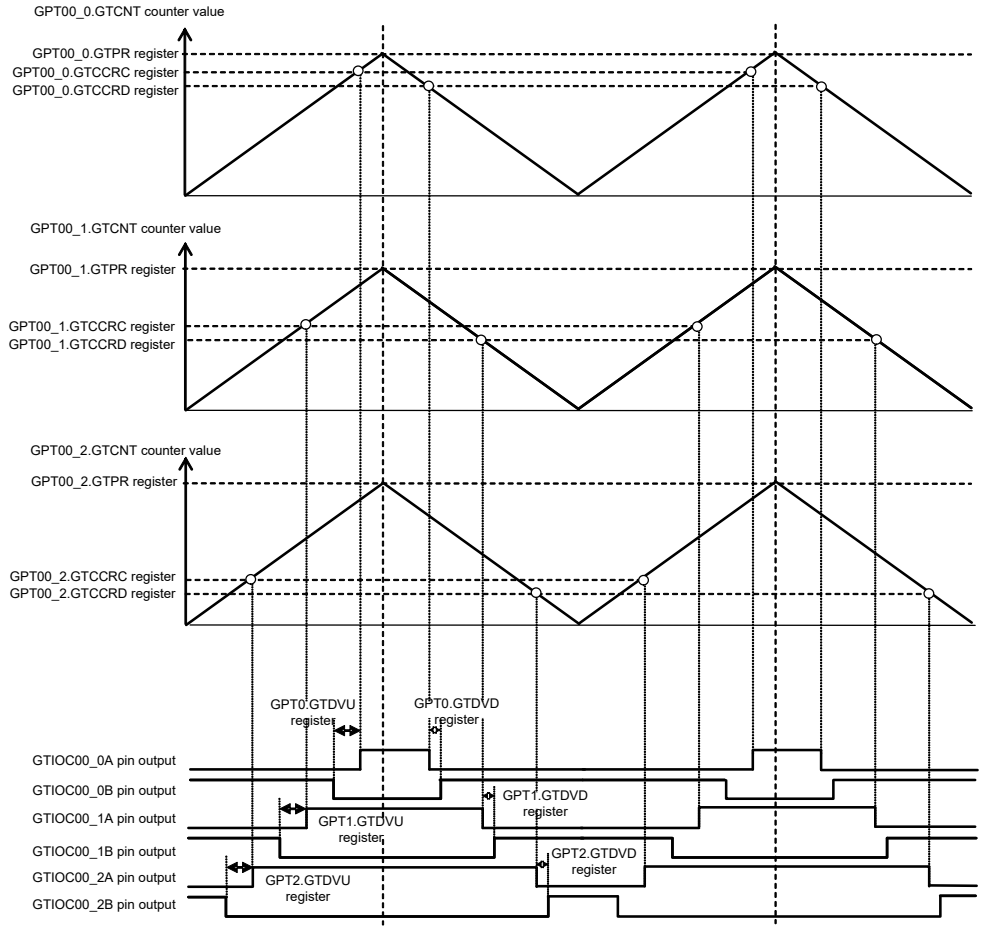


Figure 20.57 Example of 3-Phase Asymmetric Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

20.4.11 Phase Counting Mode

The GTCNT counter can be up-counting or down-counting by detecting the phase difference between the GTIOCm_nA pin input ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)) and the GTIOCm_nB pin input. Detection of the phase difference can be performed by setting the relationship between the edge and the level of the GTIOCm_nA pin input and the GTIOCm_nB pin input using the GTUPSR and GTDNSR registers for input combination. See section 20.4.1.1. Counter Operation.

Figure 20.58 to Figure 20.67 show the examples of setting procedure for the phase counting mode 1 to 5, and Table 20.13 to Table 20.22 show the setting of the GTUPSR and GTDNSR registers including up-counting and down-counting conditions.

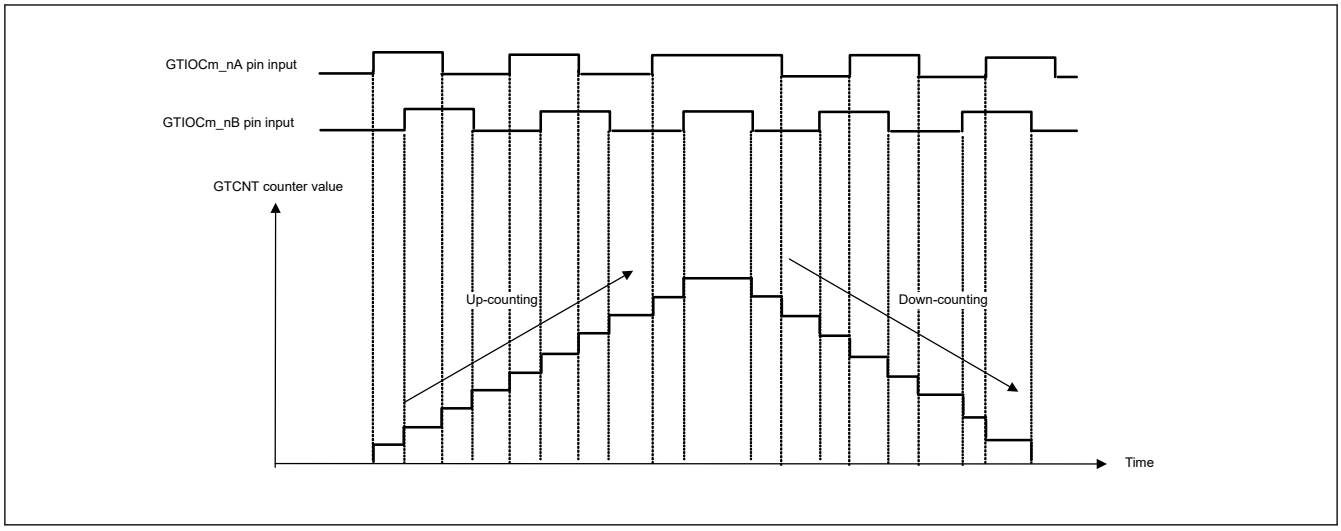


Figure 20.58 Example of setting procedure for the phase counting mode 1 ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

Table 20.13 Up-counting and down-counting conditions for phase counting mode 1

GTIOCm_nA pin input	GTIOCm_nB pin input	Operation	Setting of register
High	↑	Up-counting	GTUPSR = 0x00006900 GTDNSR = 0x00009600
Low	↓		
↑	Low		
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High		
↓	Low		

Note: n = 0 to 17

↑ : Rising edge

↓ : Falling edge

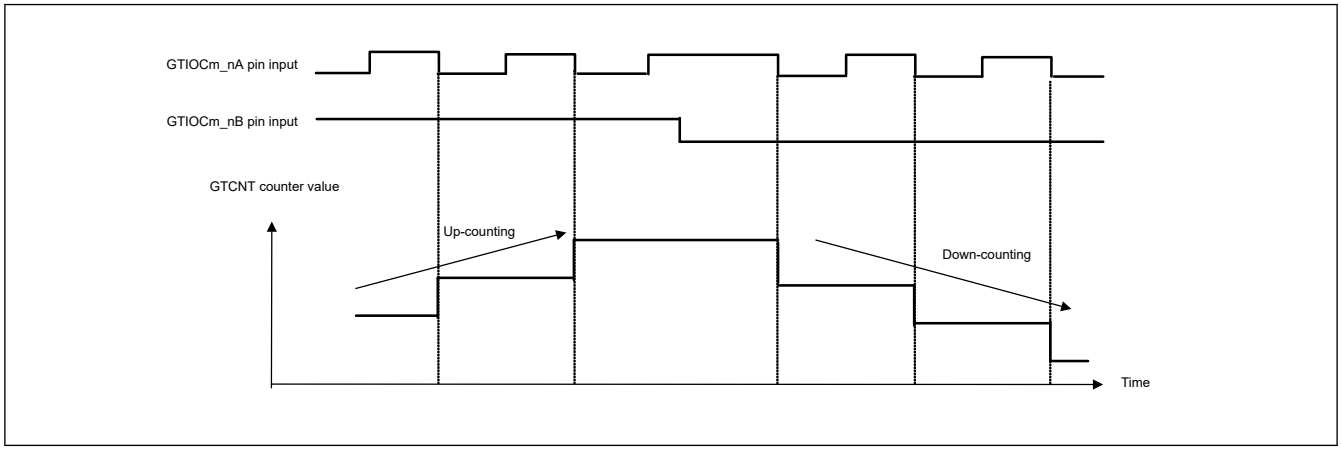


Figure 20.59 Example of setting procedure for phase counting mode 2 (00) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

Table 20.14 Up-counting and down-counting conditions for phase counting mode 2 (00)

GTIOCm_nA pin input	GTIOCm_nB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x00000800 GTDNSR = 0x00000400
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Down-counting	

Note: n = 0 to 17

↑ : Rising edge

↓ : Falling edge

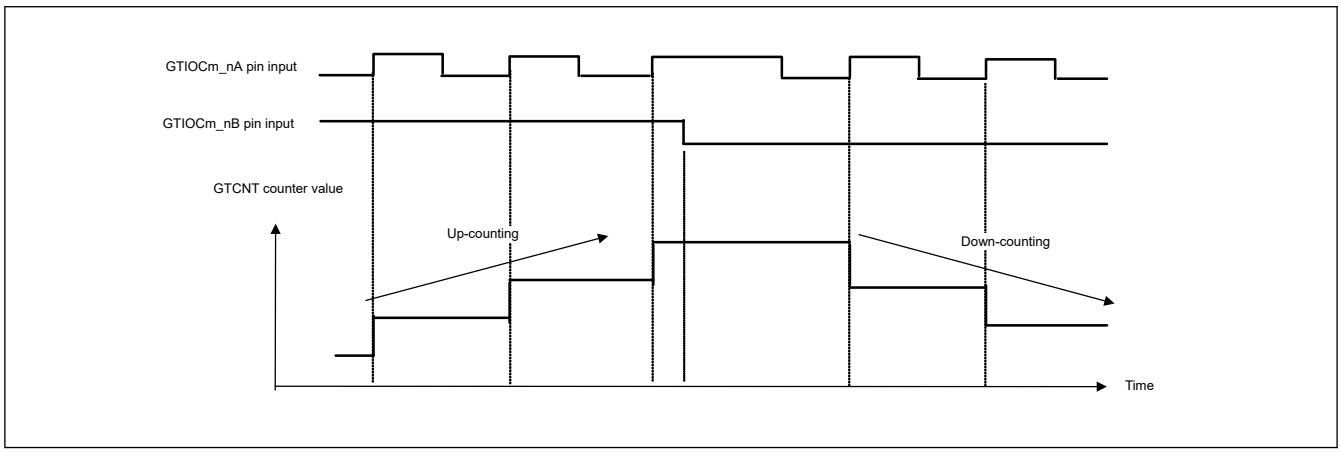


Figure 20.60 Example of setting procedure for the phase counting mode 2 (01) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

Table 20.15 Up-counting and down-counting conditions for phase counting mode 2 (01)

GTIOCm_nA pin input	GTIOCm_nB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x00000200 GTDNSR = 0x00000100
Low	↓		
↑	Low	Down-counting	
↓	High	Don't care	
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

Note: n = 0 to 17

↑ : Rising edge

: Falling edge

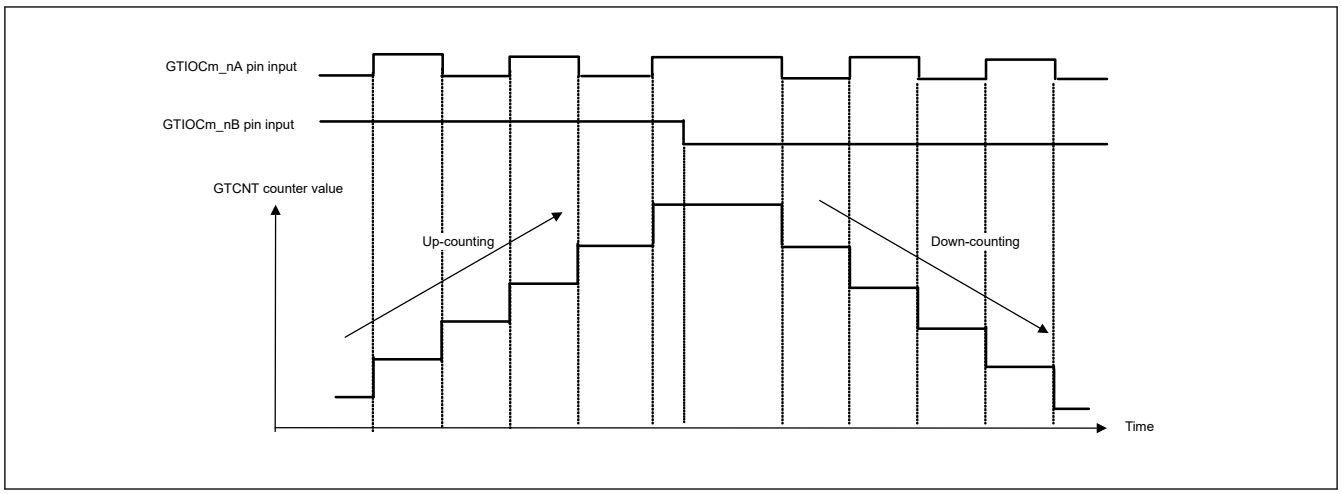


Figure 20.61 Example of setting procedure for the phase counting mode 2 (1x) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

Table 20.16 Up-counting and down-counting conditions for phase counting mode 2 (1x)

GTIOCm_nA pin input	GTIOCm_nB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x00000A00 GTDNSR = 0x00000500
Low	↓		
↑	Low	Down-counting	
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High	Up-counting	
↓	Low	Down-counting	

Note: n = 0 to 17

: Rising edge

: Falling edge

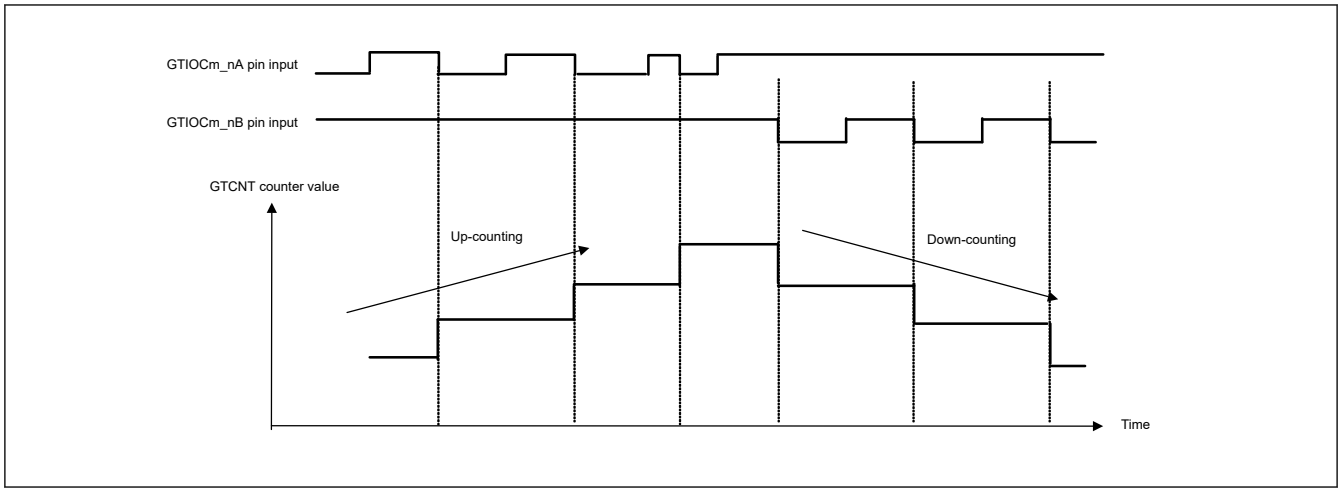


Figure 20.62 Example of setting procedure for the phase counting mode 3 (00) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

Table 20.17 Up-counting and down-counting conditions for the phase counting mode 3 (00)

GTIOCm_nA pin input	GTIOCm_nB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x00000800 GTDNSR = 0x00008000
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

Note: n = 0 to 17

↑ : Rising edge

↓ : Falling edge

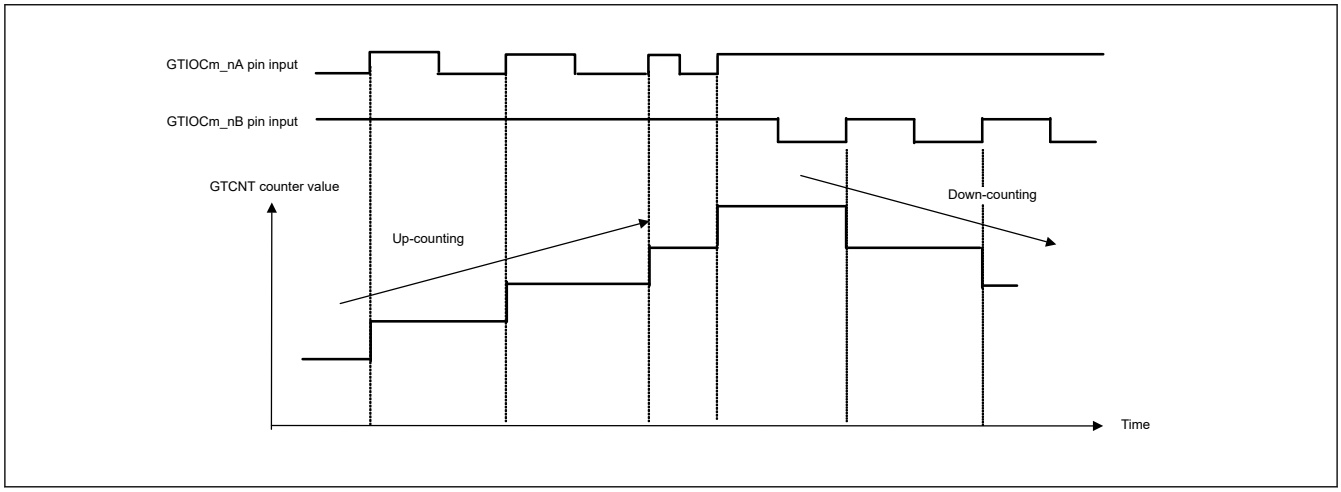


Figure 20.63 Example of setting procedure for the phase counting mode 3 (01) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

Table 20.18 Up-counting and down-counting conditions for the phase counting mode 3 (01)

GTIOCm_nA pin input	GTIOCm_nB pin input	Operation	Setting of register
High	↓	Down-counting	GTUPSR = 0x00000200 GTDNSR = 0x00002000
Low	↓	Don't care	
↑	Low		
↓	High		
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

Note: n = 0 to 17

↑ : Rising edge

↓ : Falling edge

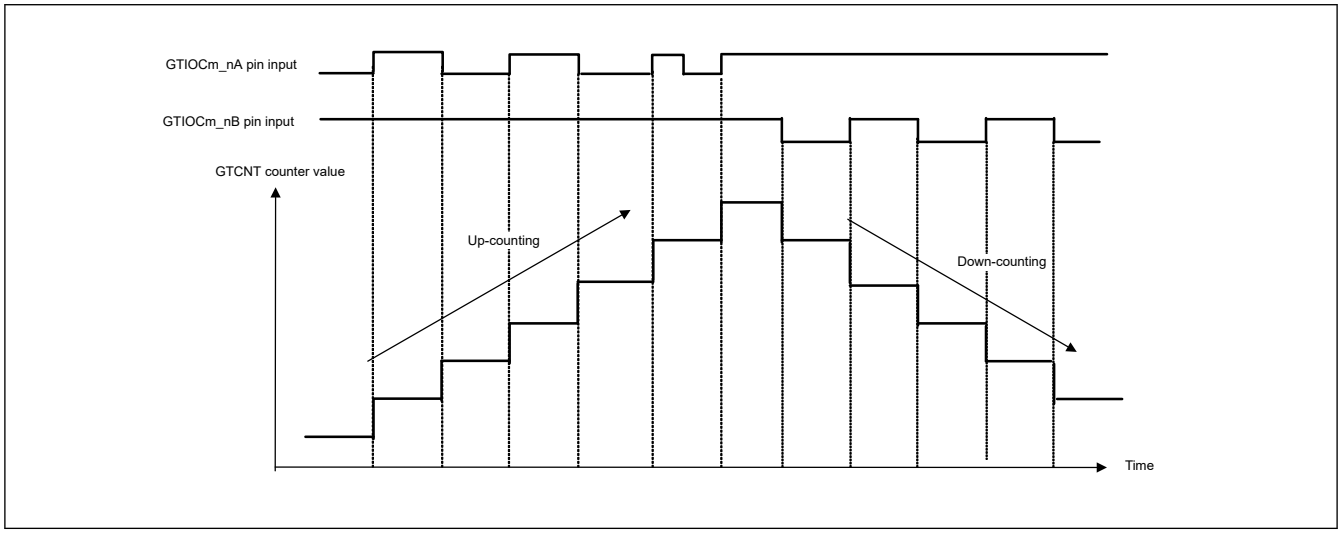


Figure 20.64 Example of setting procedure for phase counting mode 3 (1x) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

Table 20.19 Up-counting and down-counting conditions for phase counting mode 3 (1x)

GTIOCm_nA pin input	GTIOCm_nB pin input	Operation	Setting of register
High	↑	Down-counting	GTUPSR = 0x00000A00 GTDNSR = 0x0000A000
Low	↓	Don't care	
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High	Up-counting	
↓	Low	Don't care	

Note: n = 0 to 17

: Rising edge

: Falling edge

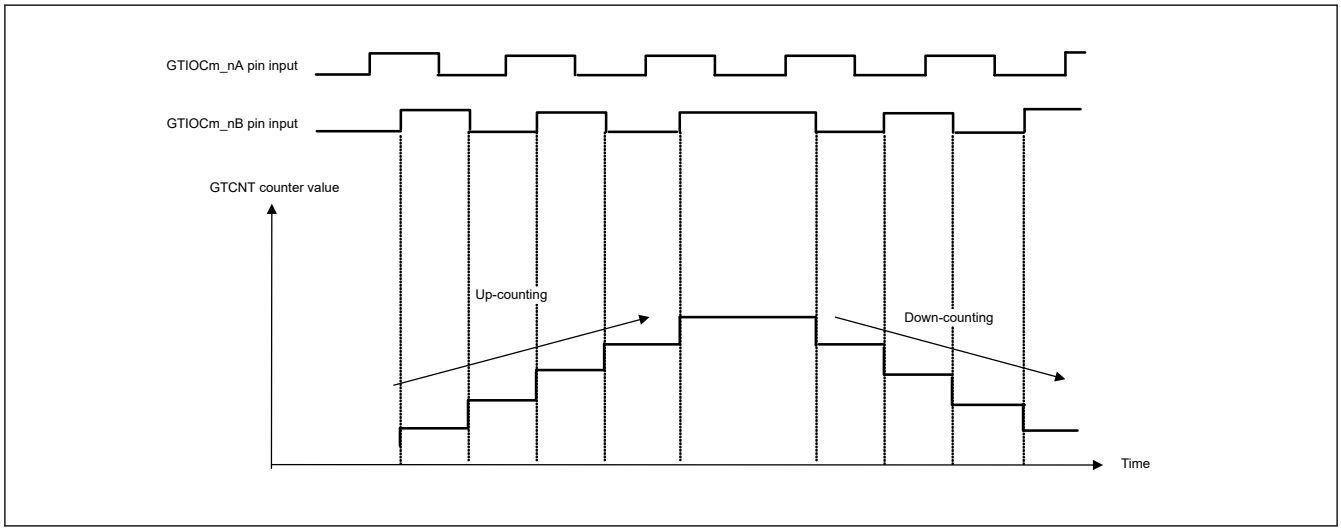


Figure 20.65 Example of setting procedure for phase counting mode 4 ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

Table 20.20 Up-counting and down-counting conditions for phase counting mode 4

GTIOCm_nA pin input	GTIOCm_nB pin input	Operation	Setting of register
High	↑	Up-counting	GTUPSR = 0x00006000 GTDNSR = 0x00009000
Low	↓		
↑	Low	Don't care	
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High	Don't care	
↓	Low		

Note: n = 0 to 17

: Rising edge

: Falling edge

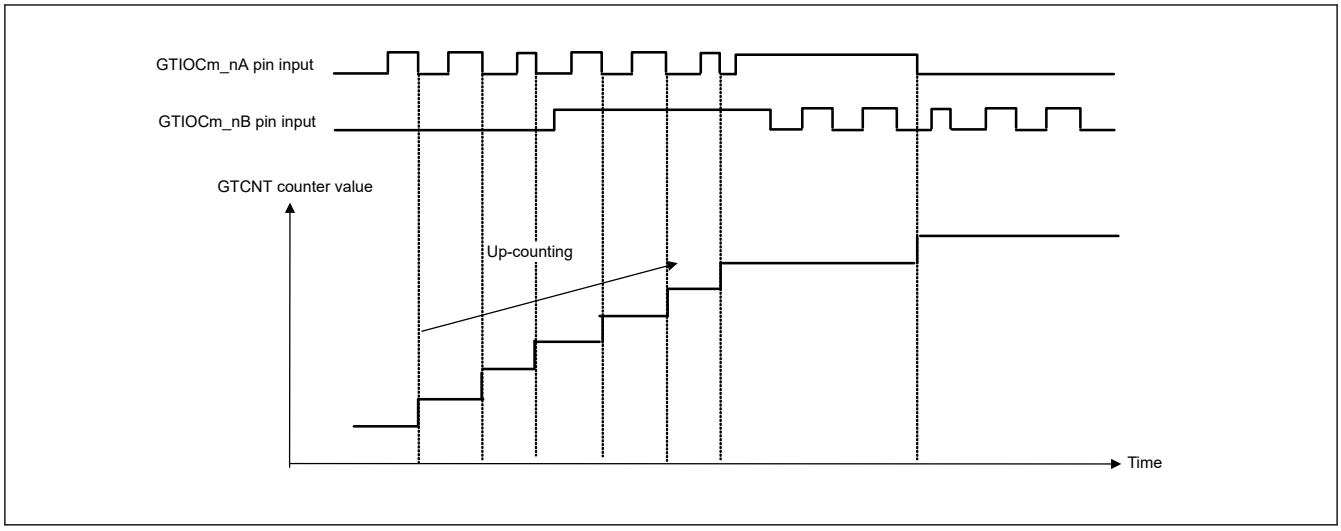


Figure 20.66 Example of setting procedure for the phase counting mode 5 (0x) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

Table 20.21 Up-counting and down-counting conditions for phase counting mode 5 (0x)

GTIOCm_nA pin input	GTIOCm_nB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x00000C00 GTDNSR = 0x00000000
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Up-counting	

Note: n = 0 to 17

↑ : Rising edge

↓ : Falling edge

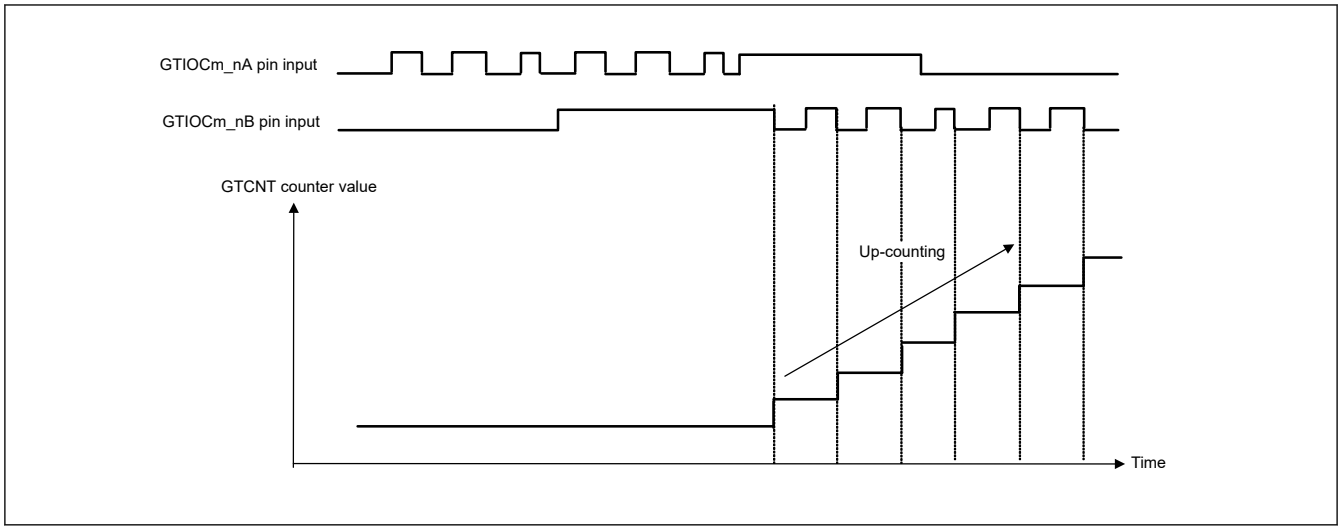


Figure 20.67 Example of setting procedure for phase counting mode 5 (1x) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

Table 20.22 Up-counting and down-counting conditions for phase counting mode 5 (1x)

GTIOCm_nA pin input	GTIOCm_nB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x0000C000 GTDNSR = 0x00000000
Low	↓	Up-counting	
↑	Low	Don't care	
↓	High		
High	↓	Up-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

Note: n = 0 to 17

↑ : Rising edge

↓ : Falling edge

20.5 Interrupt Sources

20.5.1 Interrupt Sources

There are four kinds of interrupt sources:

- GTCCRm register (m = A to F) input capture / compare match
- GTADTR register compare match (only used for ELC)
- GTCNT counter overflow (GTPR register compare match) / underflow
- Dead time error

Each interrupt source has its own status flag and control bit for generating interrupt request signals, where the generation of interrupt request signals can be enabled or disabled individually.

When an interrupt source condition is satisfied, the corresponding status flag in GTST is set to 1 and an interrupt request is generated if the corresponding interrupt request enable or disable bit in the GTINTAD register is 1.

The corresponding status flag in the GTST register is writable to clear. If flag set and flag clear occur at the same time, flag clear has priority over flag set. Both dead time error (DTEF flag) and those flags are automatically updated by internal state.

For details, see section 12, Interrupt Controller (ICU). Table 20.23 shows a list of the GPT interrupt sources.

Table 20.23 Interrupt sources

Channel	Name	Interrupt source	Interrupt flag
GPTm_n	GPTm_n_CCMPA	GPTm_n.GTCCRA register input capture / compare match	—
	GPTm_n_CCMPB	GPTm_n.GTCCRB register input capture / compare match	—
	GPTm_n_CMPC	GPTm_n.GTCCRC register compare match	—
	GPTm_n_CMPD	GPTm_n.GTCCRD register compare match	—
	GPTm_n_CMPE	GPTm_n.GTCCRE register compare match	—
	GPTm_n_CMPF	GPTm_n.GTCCRF register compare match	—
	GPTm_n_DTE*1	GPTm_n dead time error	GPTm_n.GTST.DTEF
	GPTm_n_ADTRGA*1	GPTm_n.GTADTRA register compare match	GPTm_n.GTST.ADTRAUF GPTm_n.GTST.ADTRADF
	GPTm_n_ADTRGB*1	GPTm_n.GTADTRB register compare match	GPTm_n.GTST.ADTRBUF GPTm_n.GTST.ADTRBDF
	GPTm_n_OVF	GPTm_n.GTCNT counter overflow (GPTm_n.GTPR register compare match)	—
	GPTm_n_UDF	GPTm_n.GTCNT counter underflow	—

Note: (m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)
Note 1. Except m = 10 (Unit 10, SAFETY)

(1) GPTm_n_CCMPA Interrupt ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

When the GTINTAD.GTINTA bit is 1, an interrupt request is generated under the following conditions:

- When the GTCCRA register functions as a compare match register, the GTCNT counter value matches with the GTCCRA register
- When the GTCCRA register functions as an input capture register, the input capture signal causes transfer of the GTCNT counter value to the GTCCRA register

(2) GPTm_n_CCMPB Interrupt ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

When the GTINTAD.GTINTB bit is 1, an interrupt request is generated under the following conditions:

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register
- When the GTCCRB register functions as an input capture register, the input capture signal causes transfer of the GTCNT counter value to the GTCCRB register

(3) GPTm_n_CMPC Interrupt ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

When the GTINTAD.GTINTC bit is 1, an interrupt request is generated under the following condition:

- When the GTCCRC register functions as a compare match register, the GTCNT counter value matches with the GTCCRC register

A compare match is not performed and thus interrupt is not requested under the following conditions:

- GTCR.MD[2:0] bits = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] bits = 01b, 10b, 11b (buffer operation with the GTCCRC register)

(4) GPTm_n_CMPD Interrupt ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

When the GTINTAD.GTINTD bit is 1, an interrupt request is generated under the following condition:

- When the GTCCRD register functions as a compare match register, the GTCNT counter value matches with the GTCCRD register

A compare match is not performed and thus interrupt is not requested under the following conditions:

- GTCR.MD[2:0] bits = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] bits = 10b, 11b (buffer operation with the GTCCRD register)

(5) GPTm_n_CMPE Interrupt ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

When the GTINTAD.GTINTE bit is 1, an interrupt request is generated under the following condition:

- When the GTCCRE register functions as a compare match register, the GTCNT counter value matches with the GTCCRE register.

A compare match is not performed and thus interrupt is not requested under the following conditions:

- GTCR.MD[2:0] bits = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] bits = 01b, 10b, 11b (buffer operation with the GTCCRE register)

(6) GPTm_n_CMPF Interrupt ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

When the GTINTAD.GTINTF bit is 1, an interrupt request is generated under the following condition:

- When the GTCCRF register functions as a compare match register, the GTCNT counter value matches with the GTCCRF register.

A compare match is not performed and thus interrupt is not requested under the following conditions:

- GTCR.MD[2:0] bits = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] bits = 10b, 11b (buffer operation with the GTCCRF register)

(7) GPTm_n_ADTRGA Interrupt ((m = 00 to 08 and n = 0 to 4) or (m = 09 and n = 0 to 6))

When the GTCNT counter value matches with the GTADTRA register, an interrupt request is generated under the following conditions:

- In up-counting, the interrupt enable bit (ADTRAUEN) in the GTINTAD register is 1
- In down-counting, the interrupt enable bit (ADTRADEN) in the GTINTAD register is 1

In event count operation, this interrupt request is not generated.

(8) GPTm_n_ADTRGB Interrupt ((m = 00 to 08 and n = 0 to 4) or (m = 09 and n = 0 to 6))

When the GTCNT counter value matches with the GTADTRB register, an interrupt request is generated under the following conditions:

- In up-counting, the interrupt enable bit (ADTRBUEN) in the GTINTAD register is 1
- In down-counting, the interrupt enable bit (ADTRBDEN) in the GTINTAD register is 1

In event count operation, this interrupt request is not generated.

(9) GPTm_n_OVF Interrupt ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

When the GTINTAD.GTINTPR[0] bit is 1, an interrupt request is generated under the following conditions:

- In sawtooth-wave mode, interrupt requests are enabled at overflows (GTCNT counter value changes from GTPR register value to 0 during up-counting)
- In triangle-wave mode, interrupt requests are enabled at crests (GTCNT counter value changes from GTPR register value to GTPR register value minus 1)
- In count operation by a hardware source, interrupt requests are enabled at overflows (GTCNT counter value changes from GTPR register value to 0 during up-counting)

(10) GPTm_n_UDF Interrupt ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), and (m = 10 and n = 0 to 3))

When the GTINTAD.GTINTPR[1] bit is 1, an interrupt request is generated under the following conditions:

- In sawtooth-wave mode, interrupt requests are enabled at underflows (GTCNT counter value changes from 0 to GTPR register value during down-counting)
- In triangle-wave mode, interrupt requests are enabled at troughs (GTCNT counter value changes from 0 to 1)
- In counting by a hardware source, interrupt requests are enabled at underflows (GTCNT counter value changes from 0 to GTPR register value during down-counting)

(11) GPTm_n_DTE Interrupt ((m = 00 to 08 and n = 0 to 4) or (m = 09 and n = 0 to 6))

When automatic dead time setting is made, the GTST.DTEF flag becomes 1 when the timer output toggle point with dead time added exceeds the count period. If GTINTAD.GRPDTE bit is 1 at this time, a dead time error interrupt request is generated.

In addition, when the timer output toggle point with dead time added is back within the count period, the GTST.DTEF flag changes from 1 to 0.

Table 20.24 Relationship between interrupt signals and interrupt enable bits

Interrupt signal	Interrupt enable bit	Interrupt status flag
GPTm_n_CCMPA	GPTm_n.GTINTAD.GTINTA bit	—
GPTm_n_CCMPB	GPTm_n.GTINTAD.GTINTB bit	—
GPTm_n_CMPC	GPTm_n.GTINTAD.GTINTC bit	—
GPTm_n_CMPD	GPTm_n.GTINTAD.GTINTD bit	—
GPTm_n_CMPE	GPTm_n.GTINTAD.GTINTE bit	—
GPTm_n_CMPF	GPTm_n.GTINTAD.GTINTF bit	—
GPTm_n_DTE*1	GPTm_n.GTINTAD.GRPDTE bit	GPTm_n.GTST.DTEF
GPTm_n_ADTRGA*1	GPTm_n.GTINTAD.ADTRAUEN bit	GPTm_n.GTST.ADTRAUF bit
	GPTm_n.GTINTAD.ADTRADEN bit	GPTm_n.GTST.ADTRADF bit
GPTm_n_ADTRGB*1	GPTm_n.GTINTAD.ADTRBUEN bit	GPTm_n.GTST.ADTRBUF bit
	GPTm_n.GTINTAD.ADTRBDEN bit	GPTm_n.GTST.ADTRBDF bit
GPTm_n_OVF	GPTm_n.GTINTAD.GTINTPR[1:0] bits	—
GPTm_n_UDF		—

Note: (m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)
Note 1. Except m = 10 (Unit 10, SAFETY)

20.5.2 Interrupt and A/D Conversion Start Request Skipping Function

20.5.2.1 Interrupt Skipping Function by GTITC Register

By setting the GTITC register, the GTCNT counter overflow (GTPR register compare match) interrupt (GPTm_n_OVF) and underflow interrupt (GPTm_n_UDF) can be skipped. Other interrupts and A/D converter start request signals can be skipped in coordination with the GPTm_n_OVF/GPTm_n_UDF skipping function. However, the dead time error interrupts cannot be linked with the GPTm_n_OVF/GPTm_n_UDF skipping function.

The interrupt skipping function is related only to the GTITC register setting, and is not related to setting of the GTINTAD register interrupt enable bit.

When both troughs and crests are counted and skipped in triangle-wave mode, if the number of times of skipping is odd, GPTm_n_OVF/GPTm_n_UDF interrupt requests cannot be generated at troughs only or at crests only depending on the skipping counter start timing. Therefore, to count both troughs and crests and generate the GPTm_n_OVF/GPTm_n_UDF interrupts at troughs only or crests only in triangle-wave mode, the number of times of skipping should be even.

Similarly, in sawtooth-wave mode, when both overflows and underflows are counted and skipped with the count direction changed, GPTm_n_OVF/GPTm_n_UDF interrupt requests cannot be generated at overflows only or at underflows only.

Therefore, to count both overflows and underflows with the count direction changed and generate the GPTm_n_OVF/ GPTm_n_UDF interrupts at overflows only or underflows only in sawtooth-wave mode, the skipping state should be carefully checked before using.

When changing the skipping count, be sure to release the skipping count setting (GTITC.IVTC[1:0] bits = 00b).

Figure 20.68 to Figure 20.73 show examples of skipping function operation.

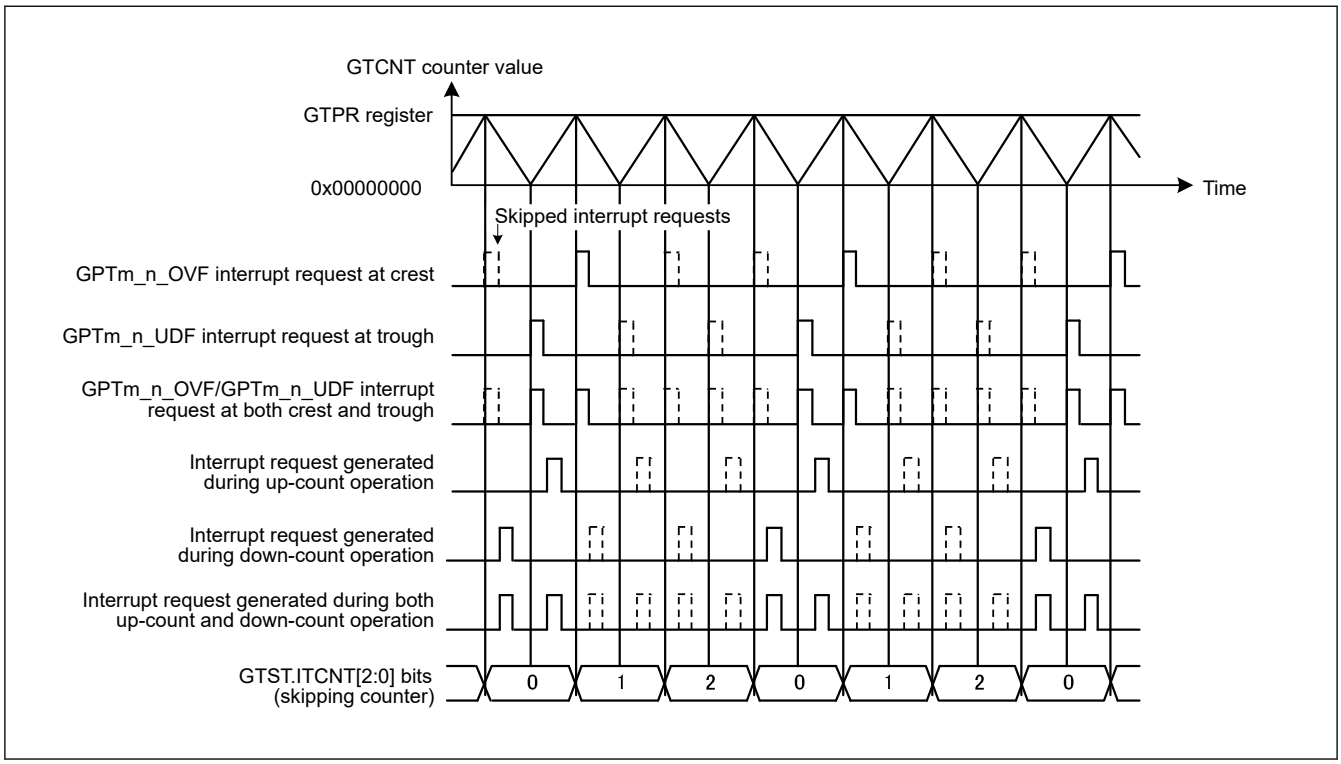


Figure 20.68 Example of interrupt skipping function operation (triangle waves, counting and skipping crests, skipping count = 2)

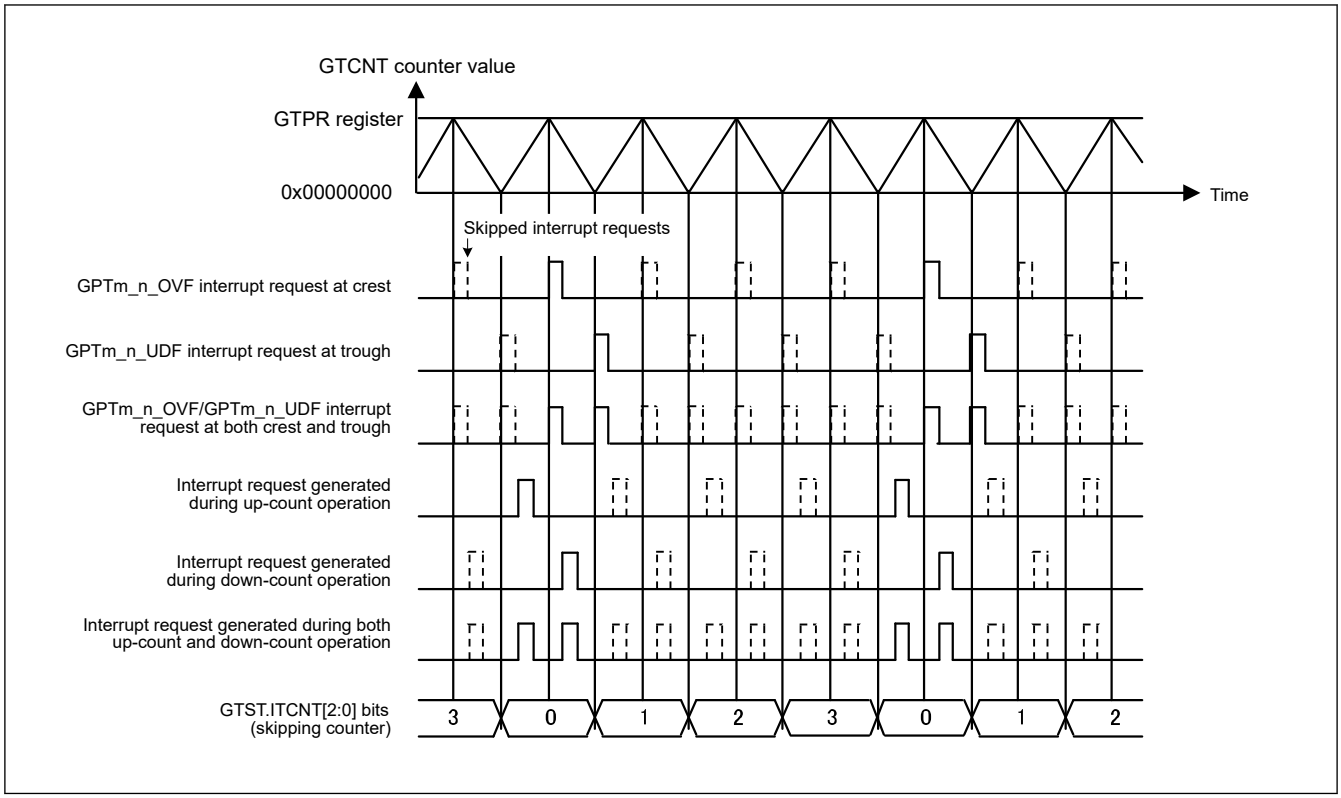


Figure 20.69 Example of interrupt skipping function operation (triangle waves, counting and skipping troughs, skipping count = 3)

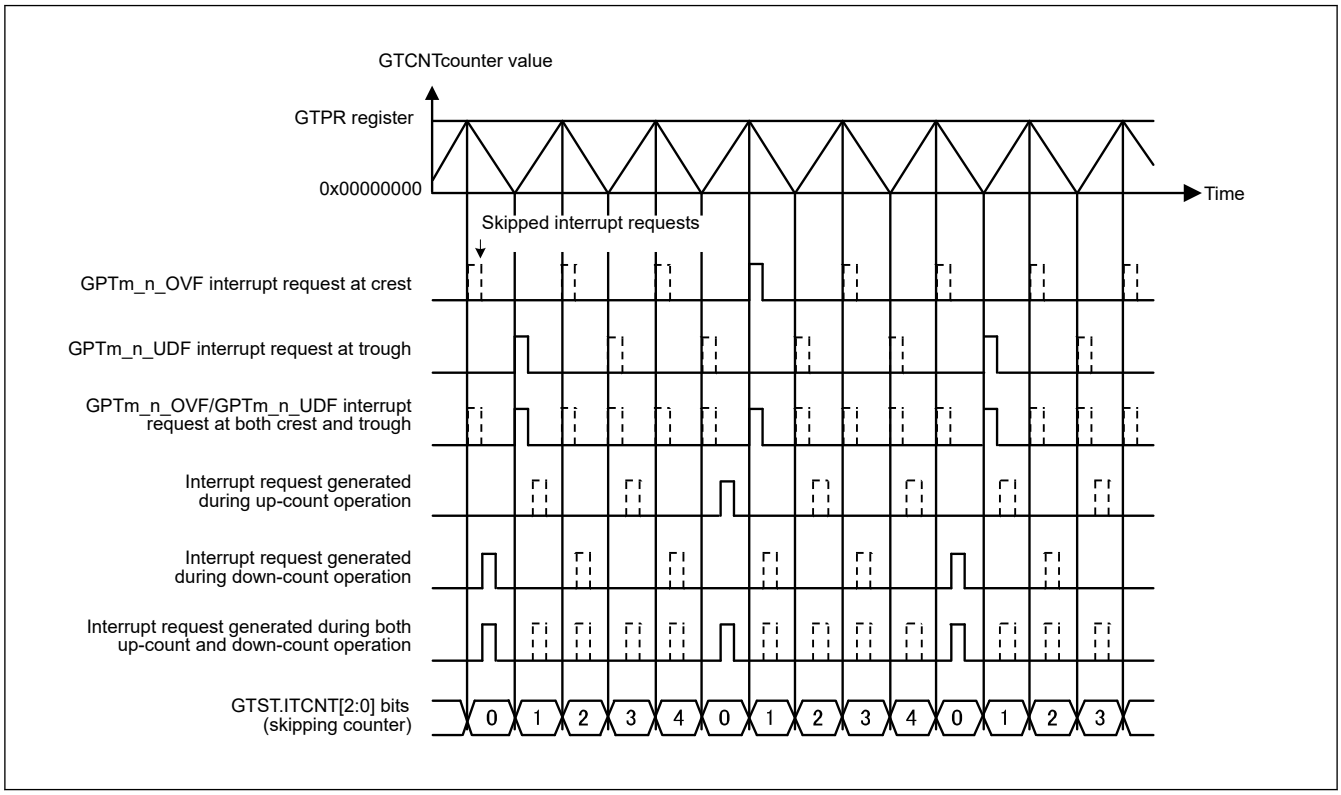


Figure 20.70 Example of interrupt skipping function operation (triangle waves, counting and skipping both troughs and crests, skipping count = 4)

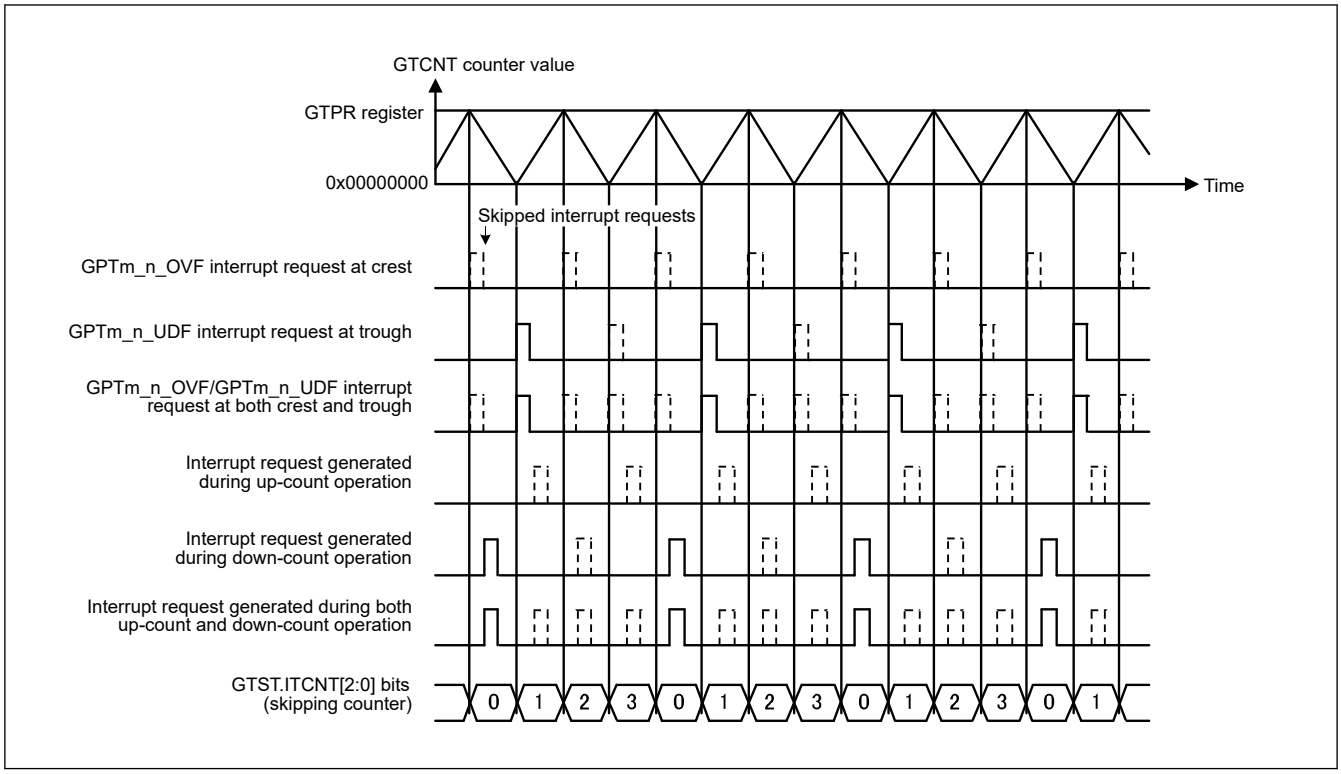


Figure 20.71 Example of interrupt skipping function operation (triangle waves, counting and skipping both troughs and crests, skipping count = 3, skipping started at up-counting)

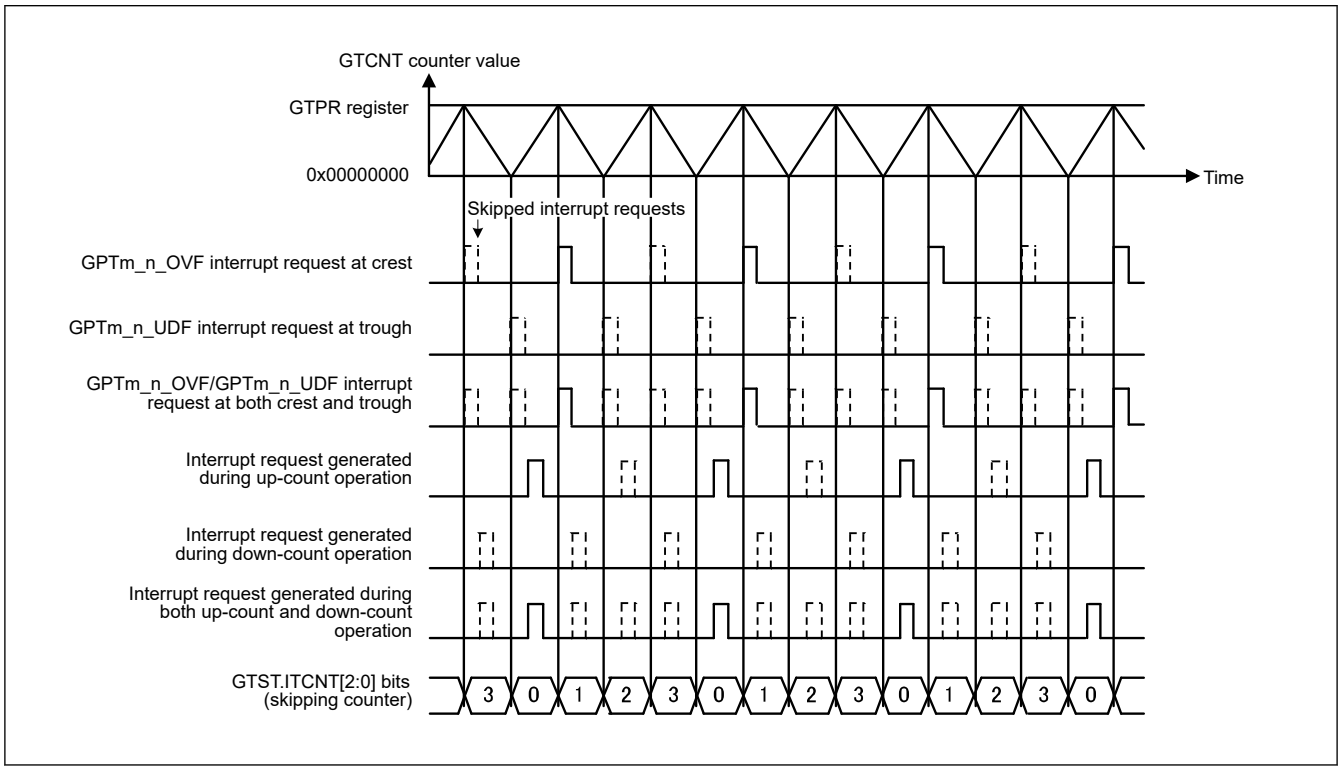


Figure 20.72 Example of interrupt skipping function operation (triangle waves, counting and skipping both troughs and crests, skipping count = 3, skipping started at down-counting)

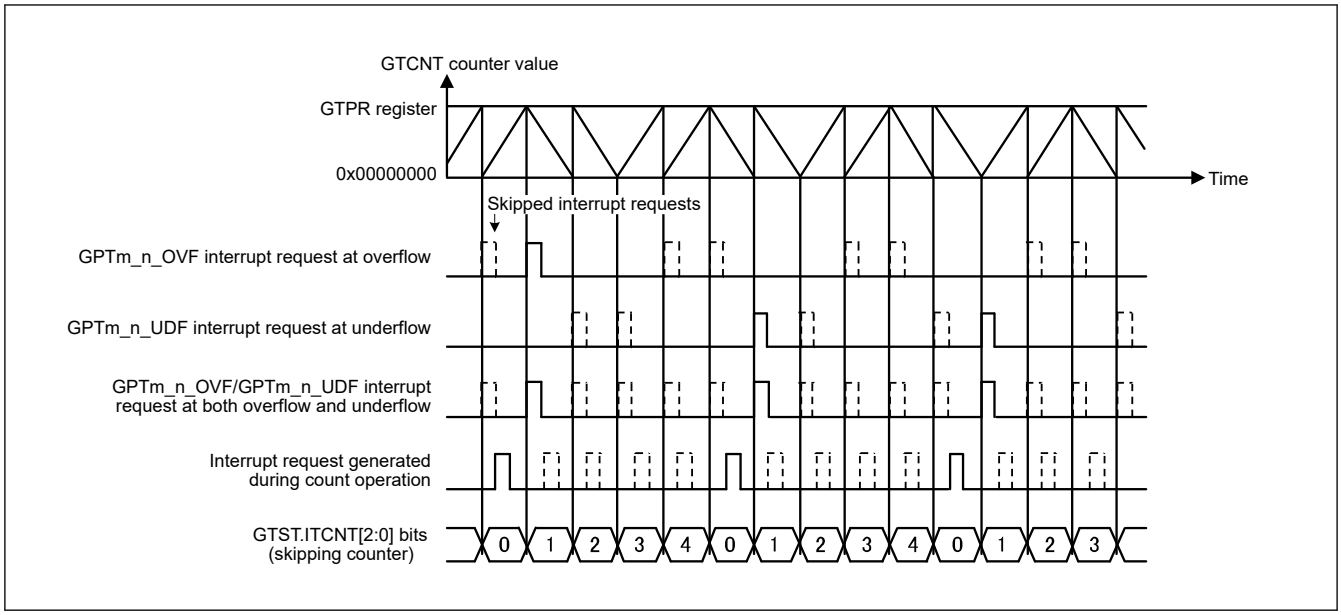


Figure 20.73 Example of interrupt skipping function operation (sawtooth waves, operation with count direction changed, counting and skipping both overflows and underflows, skipping count = 4)

20.6 A/D Converter Start Request

An A/D converter start request can be issued at a compare match between the GTCNT counter and the GTADTRA or GTADTRB register, up-counting only, down-counting only, or both up-counting and down-counting can be specified by setting the GTINTAD register.

During event count operation, the A/D converter start request cannot be generated.

The A/D converter start request is not output directly to the A/D converter but output the event signal to the ELC.

The GTADTRA and GTADTRB registers each has two buffer registers. Buffer operation with the GTADTRA register used together with the GTADTBRA and GTADTDBRA registers, and buffer operation with the GTADTRB register used together with the GTADTBRB and GTADTDBRB registers can be performed.

Figure 20.74 shows an example of A/D converter start request operation.

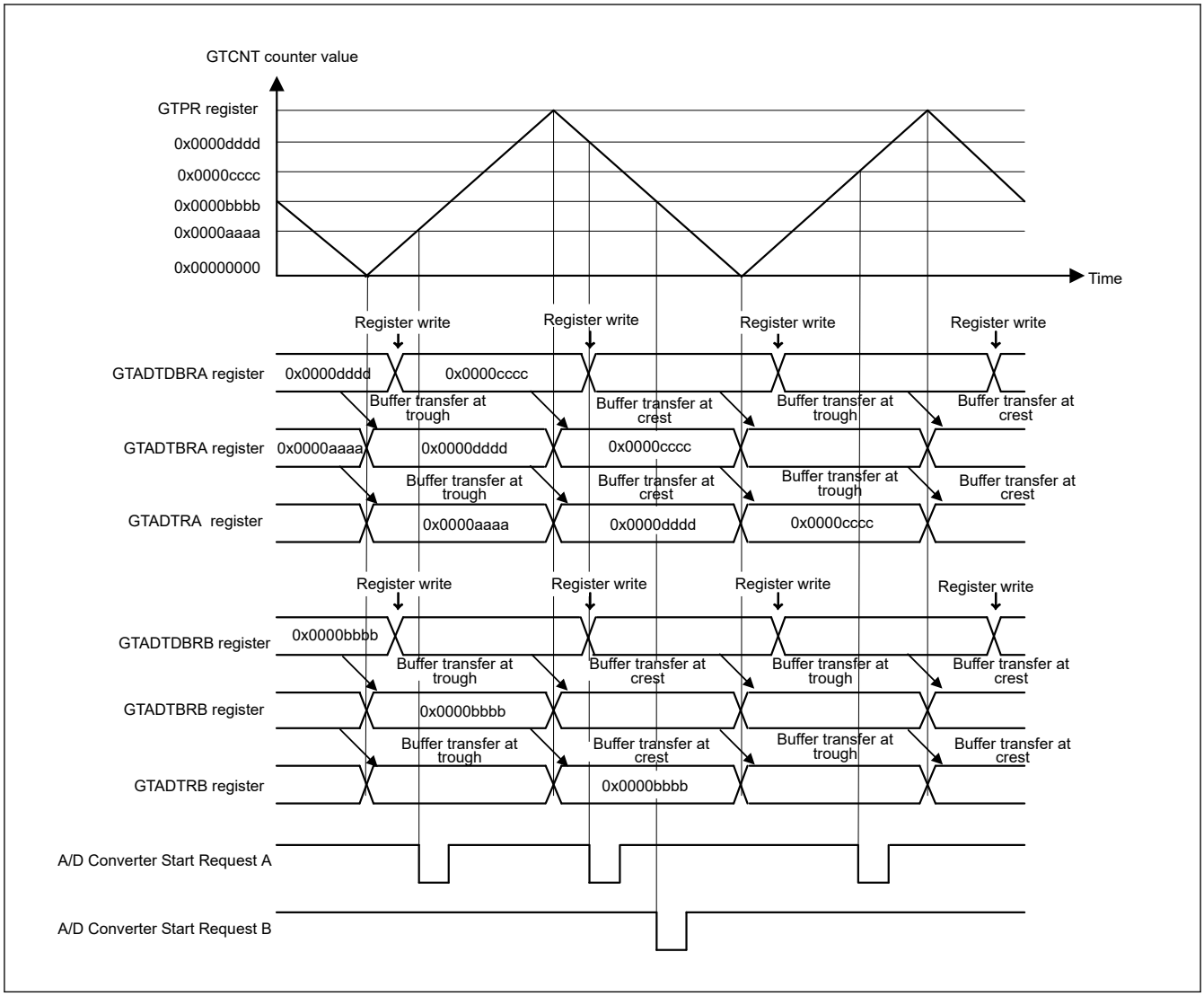


Figure 20.74 Example of A/D converter start request timing operation (triangle waves, double buffer operation, buffer transfer at both troughs and crests, A/D converter start request by GTADTRA register at both up-counting and down-counting, A/D converter start request by GTADTRB register at down-counting)

The following example shows setting for A/D converter start request operation.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits of the corresponding channel.
   In Figure 20.74, 100b, 101b, or 110b (triangle-wave PWM mode) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits of corresponding channel.
3. Period setting
   Set a period in the GTPR register of corresponding channel.
4. Set initial value for counter
   Set the initial value in the GTCNT counter of the corresponding channel.
5. Set buffer operation
   Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in the GTBER register.
   In Figure 20.74, ADTTA[1:0] bits = 11b, ADTTB[1:0] bits = 11b, ADTDA bit = 1, and ADTDB bit = 1.
6. Set compare match value
   Set the A/D converter start request point in the GTADTRA and GTADTRB registers.
7. Set buffer value

For buffer operation, set the A/D converter start request point in one cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers, respectively.
For double buffer operation, set the A/D converter start request point in two cycles after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers, respectively.

8. Set A/D converter start request for monitoring
   Select the A/D converter start request signal to be monitored with the ADSMS0[1:0] and ADSMS1[1:0] bits in the GTADSMR register from the GTADSM0 and GTADSM1 pins and enable output of the A/D converter start request signal being monitored to the ADSMEN0 and ADSMEN1 bits in the GTADSMR register.
9. Enable A/D converter start request
   Set to enable A/D converter start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register.
   In Figure 20.74, ADTRAUEN bit = 1, ADTRADEN bit = 1, ADTRBUEN bit = 0, and ADTRBDEN bit = 1.
10. Start count operation
    Set the GTCR.CSR bit to 1 to start count operation.
11. Set buffer value by cycle
    For buffer operation, set the A/D converter start request point in one cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers, respectively.
    For double buffer operation, also set the A/D converter start request point in two cycles after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers, respectively.

20.7 Operations Linked by the ELC

20.7.1 Event Signal Output to the ELC

GPT is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).

The event signal can be output regardless of the setting of the corresponding interrupt request enable bits except the A/D converter start request.

The A/D converter start request during up-counting/down-counting can be enabled/disabled individually with the A/D converter start request enable bit to output both interrupts and events output to ELC.

The corresponding channels in LLPP (LLPP0 and LLPP1) and NONSAFETY respectively have 10 event signals to the ELC in the GPT, and they are shown below. A dead time error interrupt does not have a corresponding event signal.

- Generation of compare match A interrupt
- Generation of compare match B interrupt
- Generation of compare match C interrupt
- Generation of compare match D interrupt
- Generation of compare match E interrupt
- Generation of compare match F interrupt
- Generation of overflow interrupt
- Generation of underflow interrupt
- Generation of A/D converter start request A (Only used for ELC)
- Generation of A/D converter start request B (Only used for ELC)

20.7.2 Operations in Response to Receiving Event Signals from the ELC

The GPT can perform the following operations by the signals for sources A to F for unit in LLPP (LLPP0 and LLPP1) and A to H for unit in NONSAFETY which are output from the ELC. Each signal is provided to all channels. Select an event source for the desired operation by a relevant source select register in a channel:

- Operation of count start/count stop/counter clear
- Operation of switch and up-counting/down-counting
- Operation of input capture A and B

See operation by hardware sources in section 20.4. Operation for respective operations.

20.8 Noise Filter Function

Each pin for use in input capture to the GPT is equipped with a noise filter. The noise filter samples input signals at the sampling clock (Core clock) and removes the pulses of which length is less than 3 sampling cycles.

The noise filter functionality includes enabling and disabling of the noise filter for each pin and setting of the sampling clock for each channel.

Figure 20.75 shows the timing of noise filtering.

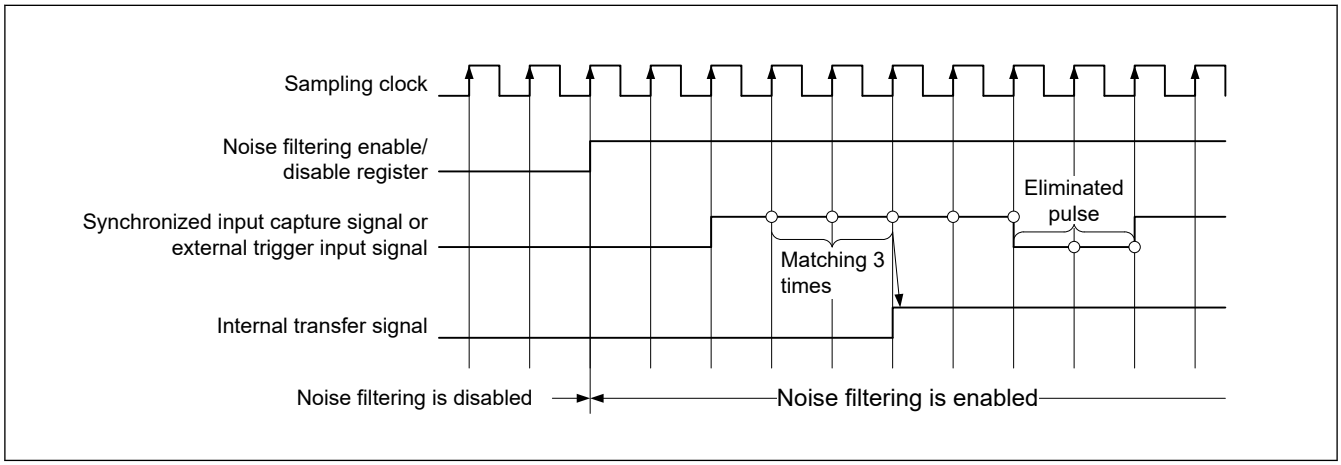


Figure 20.75 Timing of noise filtering

20.9 Protection Function

20.9.1 Write-Protection for Registers

To avoid incorrect writing to registers, write access to registers can be enabled or disabled per channel by setting the GTWP.WP bit for the given channel.

The WP bits enable or disable writing to the following registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTSWSR, GTSWOS.

Every bit in registers GTSTR, GTSTP, and GTCLR, which can update the corresponding registers in other channels and can be updated by any of the corresponding registers in other channels conversely, can be protected by setting the GTWP.STRWP, STPWP, and CLRWP bits, respectively, per channel.

Likewise, writing to the GTSECSR and GTSECR registers, which can control all channel in group by writing to the GTSECSR and GTSECR registers of a given channel, can be enabled or disabled by the setting of the GTWP.CMNWP bit.

Protection by the GTWP register only targets writing operation by the CPU. Update to the register generated in related to the CPU writing is not protected.

20.9.2 Disabling of Buffer Operation

If the timing of buffer register write is too late for the buffer transfer timing, buffer operation can be suspended by setting the BD0, BD1, BD2, and BD3 bits in the GTBER register.

Specifically, buffer transfer can be temporarily disabled, even though a buffer transfer condition is generated during buffer register write, by setting the BD0, BD1, BD2, and BD3 bits to 1 (buffer operation disabled) before buffer register write, and setting the bits to 0 (buffer operation enabled) after completion of writing to all the buffer registers.

The BD0, BD1, BD2, and BD3 bits can be set on channel basis by writing directly to the GTBER register or can be set simultaneously by setting the GTSECR register for multiple channels which were set by the GTSECSR register. Figure 20.76 shows an example of operation for disabling buffer operation by writing to the GTBER register.

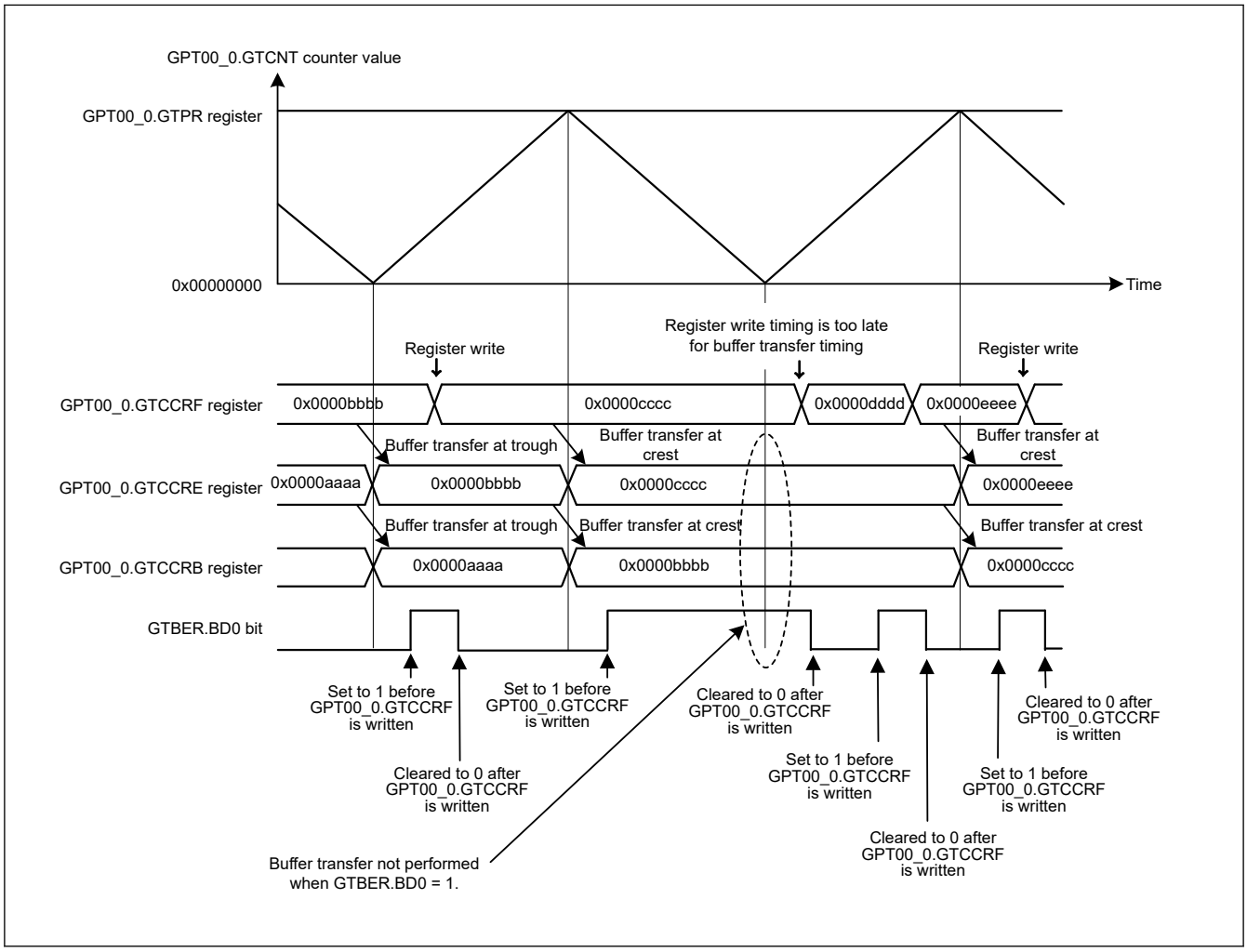


Figure 20.76 Example of operation for disabling buffer operation (triangle waves, double buffer operation, buffer transfer at both troughs and crests)

20.9.2.1 Simultaneous Control of Buffer Operations of Multiple Channels

The GTBER.BDn bit can be set by writing directly to the GTBER register per channel or by making settings in the GTSECR register for multiple channels that are already set in the GTSECSR register.

Use the following procedure to simultaneously set the GTBER.BDn bit of multiple channels.

1. Select the channels to simultaneously set the GTSECSR register
   Set the GTSECSR register so that the values at the bit positions for the corresponding channels for simultaneously setting of the GTBER.BDn bits become 1. All GTSECSR registers can be updated by writing to the GTSECSR register of any channel.
2. Simultaneously set the GTBER.BDn bits by updating the GTSECR register

In the GTSECR register, set the operation of the GTBER.BDn bit (enabling or disabling of buffer operation) that are to be simultaneously set. Writing to the GTSECR register from any channel updates the GTBER.BDn bit in all channels corresponding to the bits set as 1 in the GTSECSR register, in accordance with the value of the GTSECR register.

Figure 20.77 shows an example of simultaneously controlling the enabling or disabling of buffer operation for multiple channels.

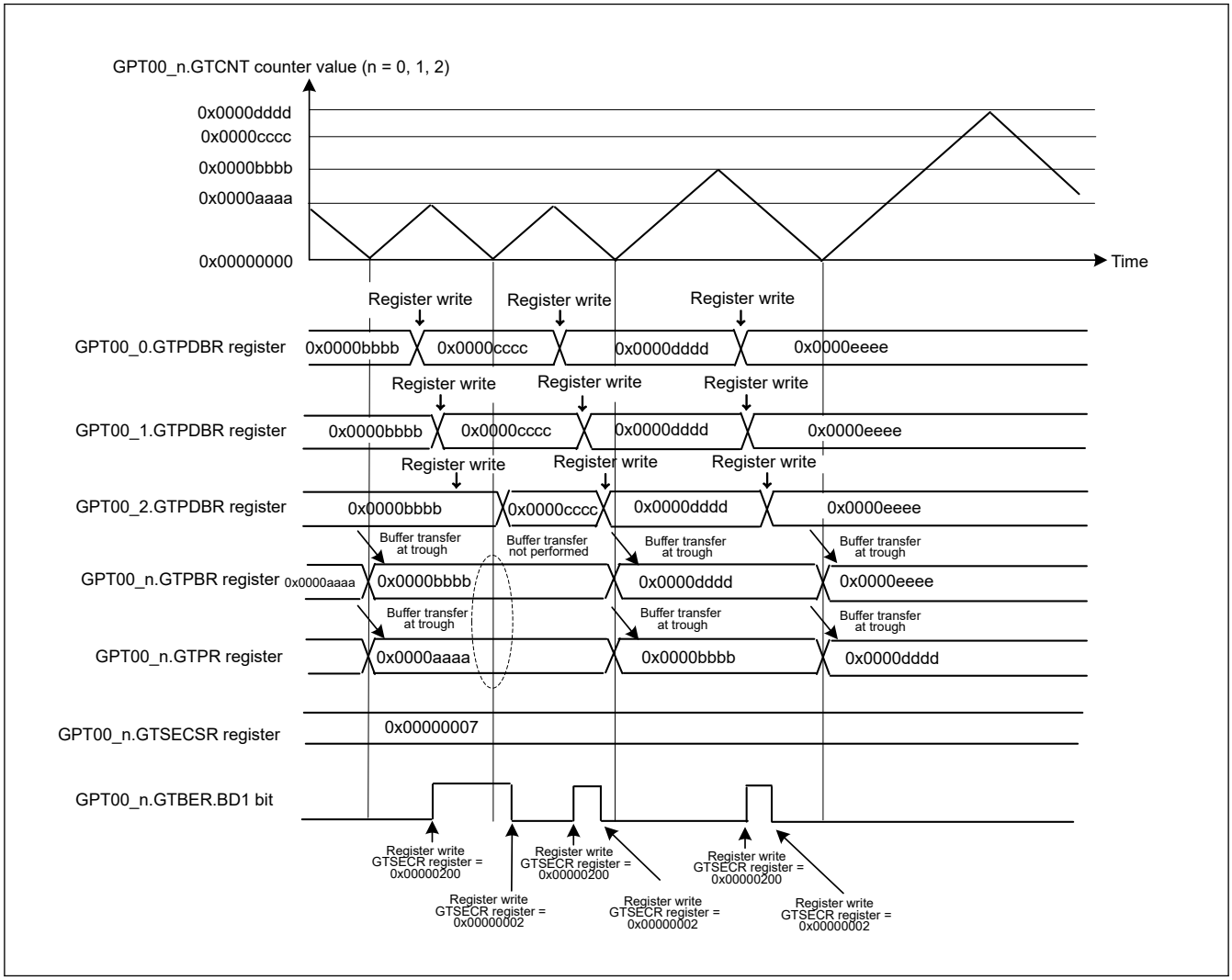


Figure 20.77 Example of multiple channel operation for disabling buffer operation (triangle waves, double buffer operation)

20.9.2.2 Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer

When a GTBER.DBRTECm (m = A, B) bit is set to 1 in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD0 bit or by the buffer transfer extended skipping function.

(1) In Sawtooth-Wave One-Shot Pulse Mode

In sawtooth-wave one-shot pulse mode, the compare match value for the first half of a waveform is stored in temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) registers.

Table 20.25 lists the types of buffer transfer of the GTCCR register during counting in sawtooth-wave one-shot pulse mode.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer.

In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E) when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 20.25 GTCCR buffer transfer operation in sawtooth-wave one-shot pulse mode during GTCNT counting

GTBER.DBRTECm	Buffer transfer	Timing of transfer				
		GTCCRx ↓ GTCCRm	GTCCRx ↓ Temporary register x	Temporary register x ↓ GTCCRm	GTCCRy ↓ Temporary register m	Temporary register m ↓ GTCCRm
0	Transfer enabled period	Overflow or underflow	No transfer	No transfer	Overflow or underflow	GTCCRm compare match
	Transfer disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer enabled period	Overflow or underflow	Overflow or underflow	Not Transferred	Overflow or underflow	GTCCRm compare match
	Transfer disabled period	No transfer	No transfer	Overflow or underflow	No transfer	GTCCRm compare match

Note: m = A, B; x = C, E; y = D, F

Figure 20.78 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer is disabled in sawtooth-wave one-shot pulse mode.

Figure 20.79 shows the operation of generating transfer-disabled period by updating the GTBER.BD0 bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in sawtooth-wave one-shot pulse mode.

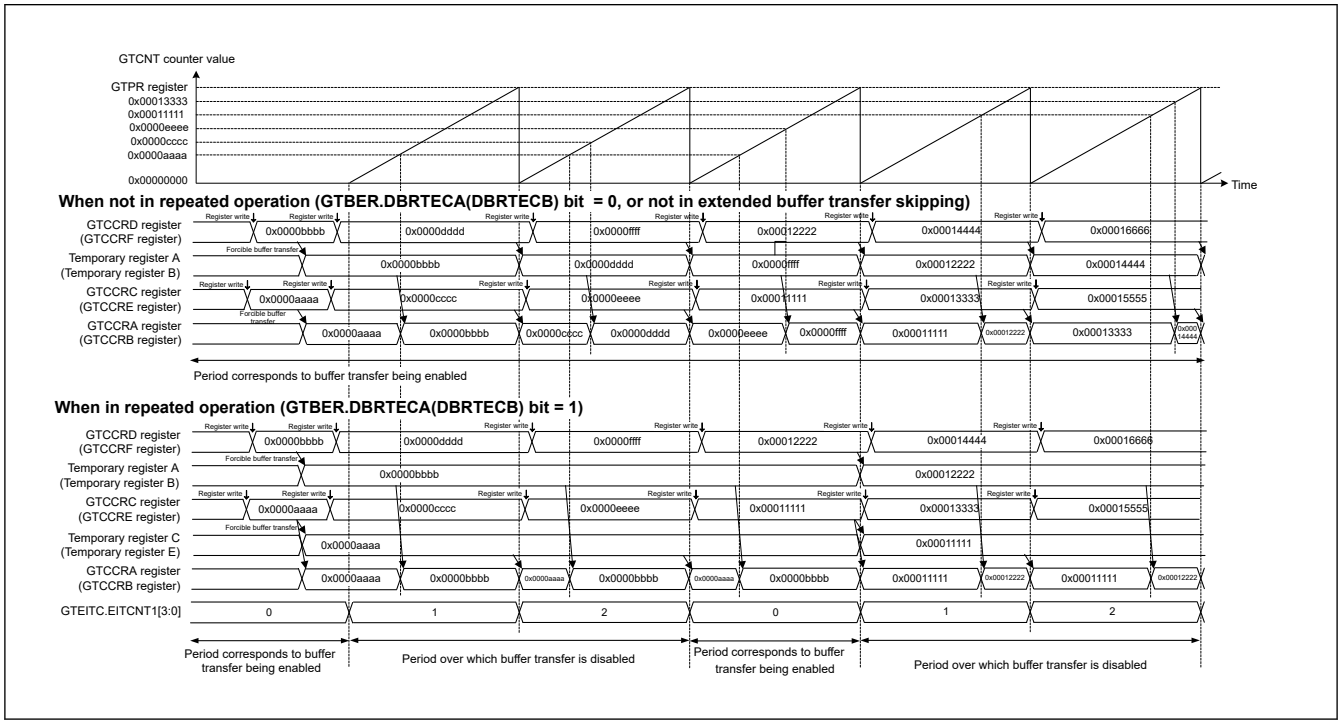


Figure 20.78 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (sawtooth-wave one-shot pulse mode, using extended buffer transfer skipping, GTBER.BD0 is constantly 0)

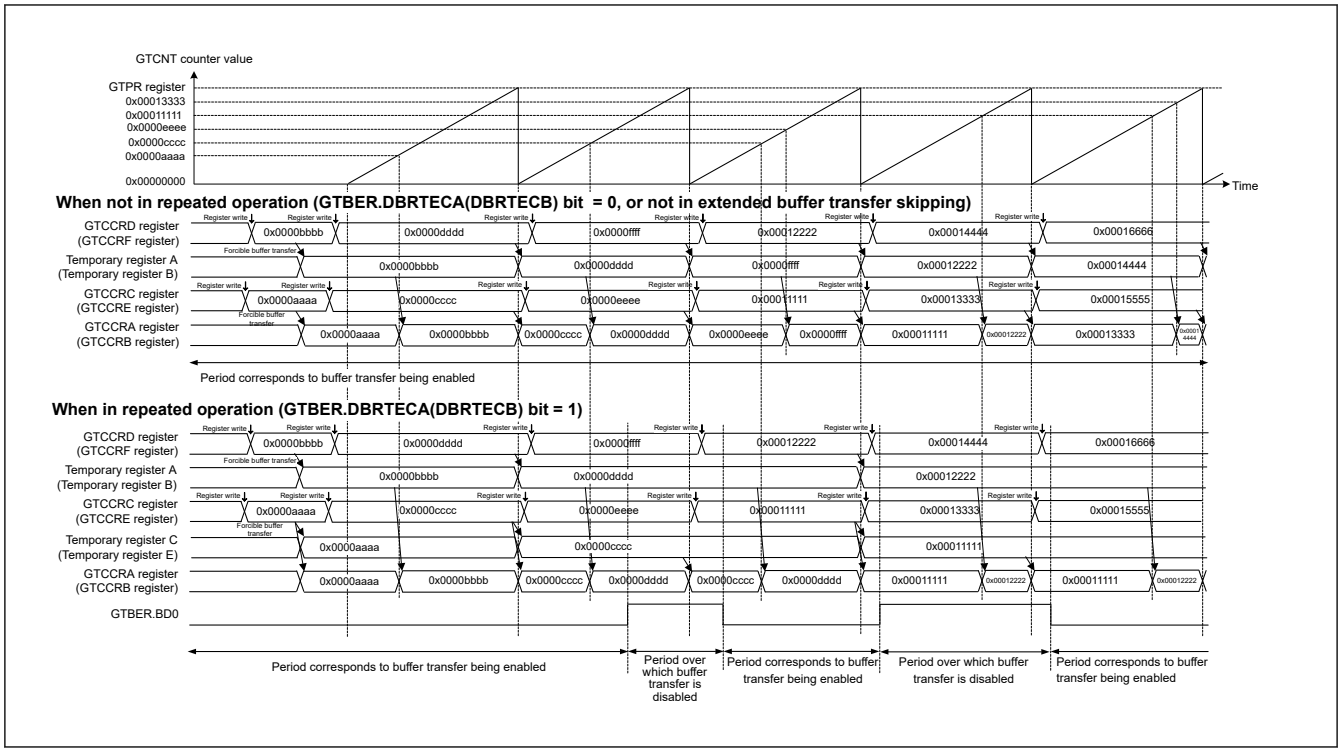


Figure 20.79 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (sawtooth-wave one-shot pulse mode, updating the GTBER.BD0 bit)

(2) In Triangle-Wave PWM Mode 3

In triangle-wave PWM mode 3, the compare match value for the first half of a waveform is stored in the temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) register.

Table 20.26 lists the types of buffer transfer of the GTCCR register during counting operation in triangle-wave PWM mode 3.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer. In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E), when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is set to 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 20.26 GTCCR buffer transfer operation in triangle-wave PWM mode 3 during GTCNT counting

GTBER.DBRTECm	Buffer transfer	Timing of transfer				
		GTCCRx ↓ GTCCRm	GTCCRx ↓ temporary register x	temporary register x ↓ GTCCRm	GTCCRy ↓ temporary register m	temporary register m ↓ GTCCRm
0	Transfer enabled period	Trough	No transfer	No transfer	Trough	Crest
	Transfer disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer enabled period	Trough	Trough	No transfer	Trough	Crest
	Transfer disabled period	No transfer	No transfer	Trough	No transfer	Crest

Note: m = A, B; x = C, E; y = D, F

Figure 20.80 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer is disabled in triangle-wave PWM mode 3.

Figure 20.81 shows the operation of generating transfer-disabled period by updating the GTBER.BD0 bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in triangle-wave PWM mode 3.

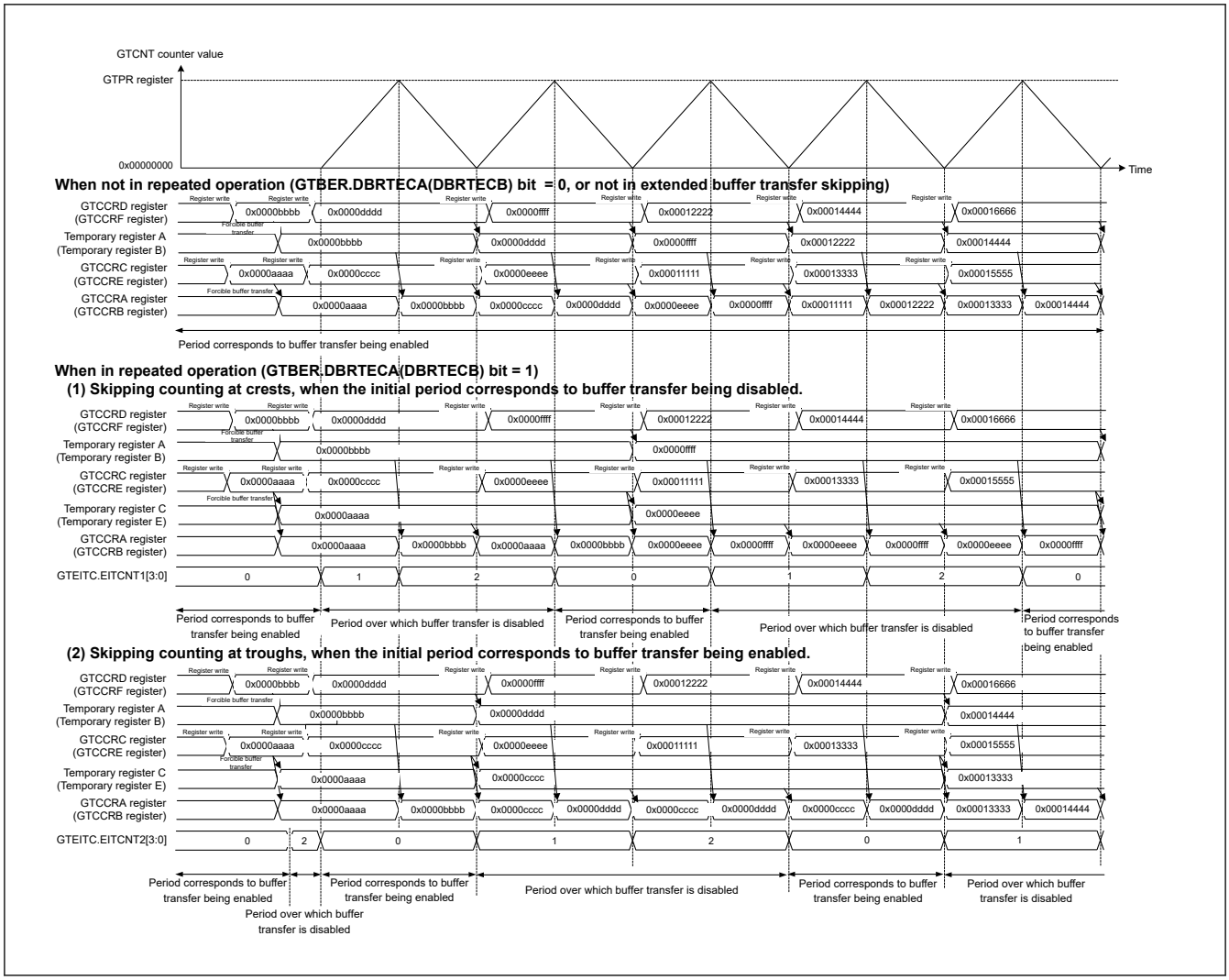


Figure 20.80 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (triangle-wave PWM mode 3, using extended buffer transfer skipping, GTBER.BD0 is constantly 0)

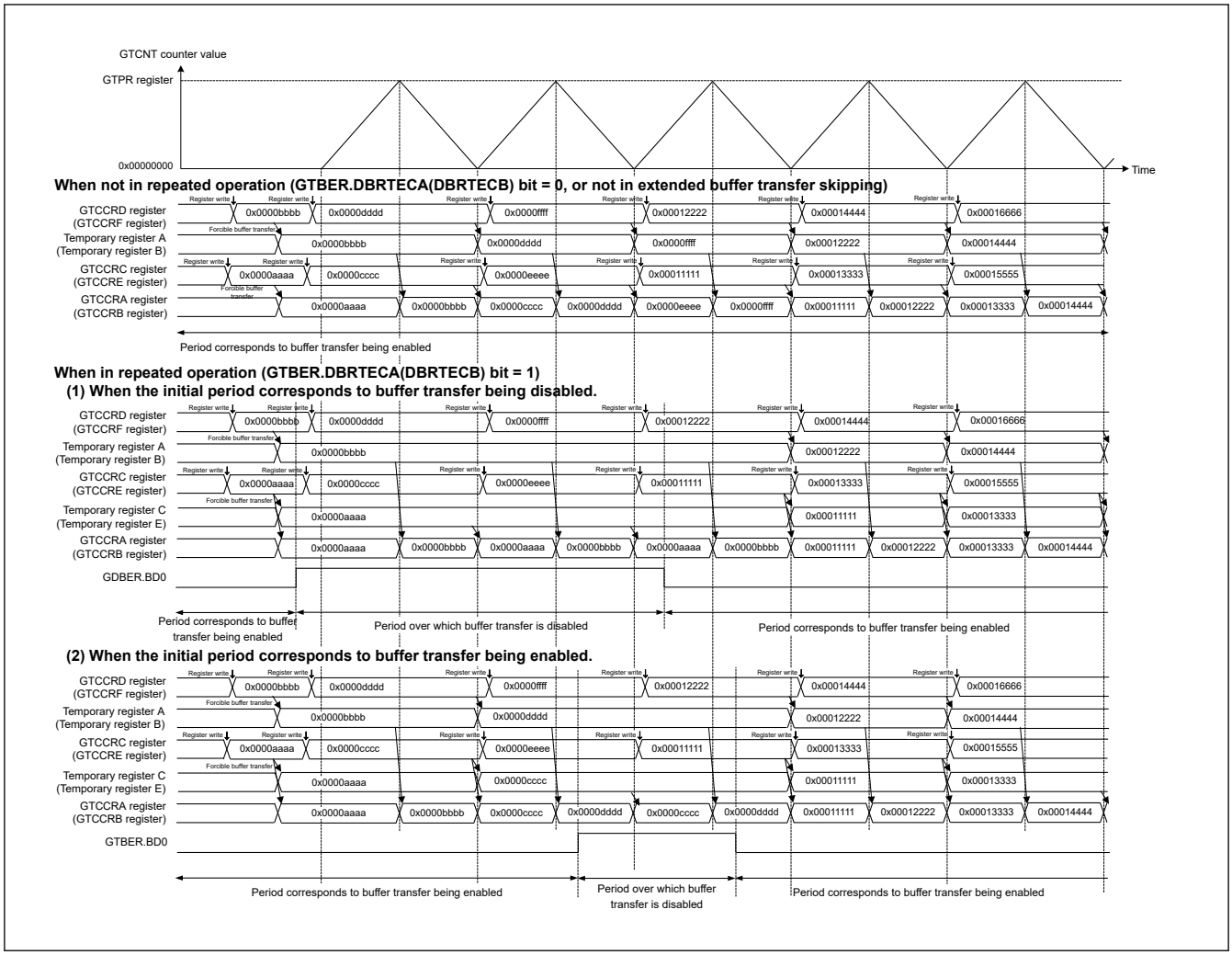


Figure 20.81 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (triangle-wave PWM mode 3, updating the GTBER.BD0 bit)

20.9.2.3 Repeated Double-Buffered Operation When Disabling GTADTR Buffer Transfer

When GTBER.DBRTEADm (m = A, B) is set to 1 for triangle waves using GTADTRm double buffer (GTBER.ADTDm bit = 1) with the buffer transfer timing of both crest and trough (GTBER.ADTTm[1:0] bits = 11b), transfer from the intermediate buffer to the GTADTRm register is repeated on a cyclic basis even while transfer is disabled by the setting of the GTBER.BD[2] bit or by the extended buffer transfer skipping function.

The A/D converter start request timing value to be transferred repeatedly are stored into the following registers:

- The A/D converter start request timing value to be used in the first half of period is stored into temporary register ADm as the intermediate buffer of the GTADTBRm register
- The compare match value to be used in the second half of period is stored into the GTADTBRm register.

These values are transferred to the GTADTRm register alternately. The GTBER.DBRTSADm bit selects either the first half or the second half of the period.

Table 20.27 lists the types of buffer transfer of the GTADTR register during counting operation in triangle waves.

While counting is stopped, set the temporary register ADm (m = A, B) to the A/D converter start request timing value of first half of period after GTCNT counting start. When the setting of the GTBER.DBRTEADm bit is set to 1, values written by the CPU to the GTADTRm registers are reflected as the values of temporary register ADm.

When the period right after GTCNT counting start is the second half, set the GTADTRm register to the A/D converter start request timing value of second half of period during GTCNT counting stop. When the period after GTCNT counting start is the second half of the period and buffer transfer is disabled, set the GTADTRm register to the same value as the GTADTBRm register.

Setting the GTADTRm register to the value of second half of period during GTCNT counting stop should be set by CPU under the condition that the GTBER.DBRTEADm bit is temporary set to 0 after setting temporary register ADm to the value of first half of the period. After setting the GTADTRm register, set the GTBER.DBRTEADm bit to 1 before GTCNT counting start.

Table 20.27 GTADTR buffer transfer operation of triangle-wave during GTCNT counting

GTBER.DBRTEADm	GTBER.DBRTSADm bit	Buffer transfer	Timing of transfer			
			GTADTDBRm ↓ GTADTBRm	GTADTBRm ↓ GTADTRm	GTADTDBRm ↓ Temporary register ADm	Temporary register ADm ↓ GTADTRm
0	—	Transfer enabled period	Both crest and trough	Both crest and trough	No transfer	No transfer
		Transfer disabled period	No transfer	No transfer	No transfer	No transfer
1	0 (trough period)	Transfer enabled period	Both crest and trough	Both crest and trough	Crest	No transfer
		Transfer disabled period	No transfer	Crest	No transfer	Trough
	1 (crest period)	Transfer enabled period	Both crest and trough	Both crest and trough	Trough	No transfer
	—	Transfer disabled period	No transfer	Trough	No transfer	Crest

Figure 20.82 shows the operation of repeated double buffer operation when the GTADTR buffer transfer disabled for triangle-wave which the period is defined by trough and the period right after GTCNT counting start is the first half.

Figure 20.83 shows the operation of repeated double buffer operation when the GTADTR buffer transfer disabled for triangle-wave which the period is defined by crest and the period right after GTCNT counting start is the second half.

Figure 20.84 shows the operation of repeated double buffer operation when the GTADTR buffer transfer disabled for triangle-wave which the period is defined by trough and the period right after GTCNT counting start is the second half.

Figure 20.85 shows the operation of repeated double buffer operation when the GTADTR buffer transfer disabled for triangle-wave which the period is defined by crest and the period right after GTCNT counting start is the first half.

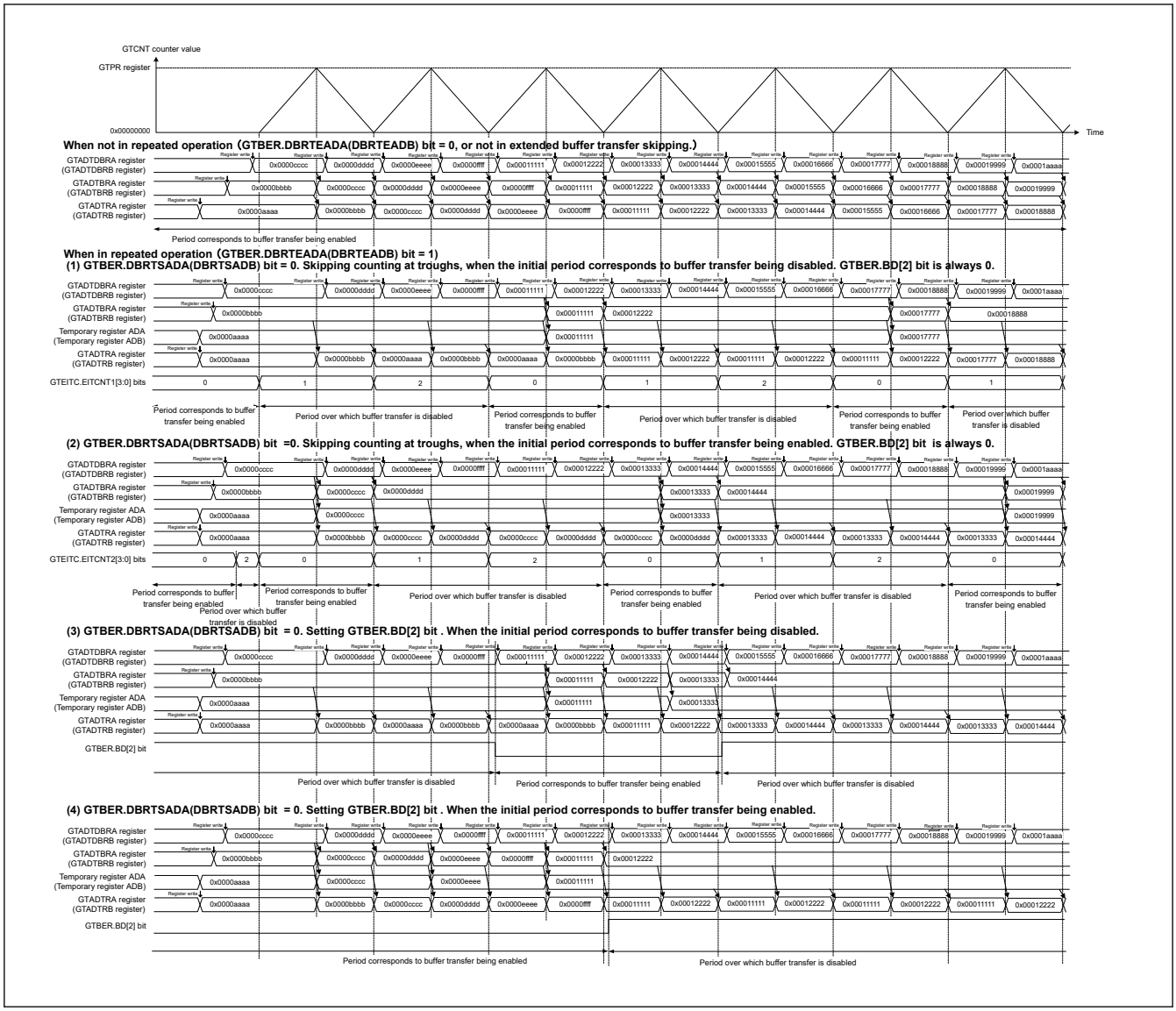


Figure 20.82 Example of repeated double-buffer operation when GTADTR buffer transfer is disabled (triangle-wave, repeating period between troughs, first half of period at GTCNT counting start)

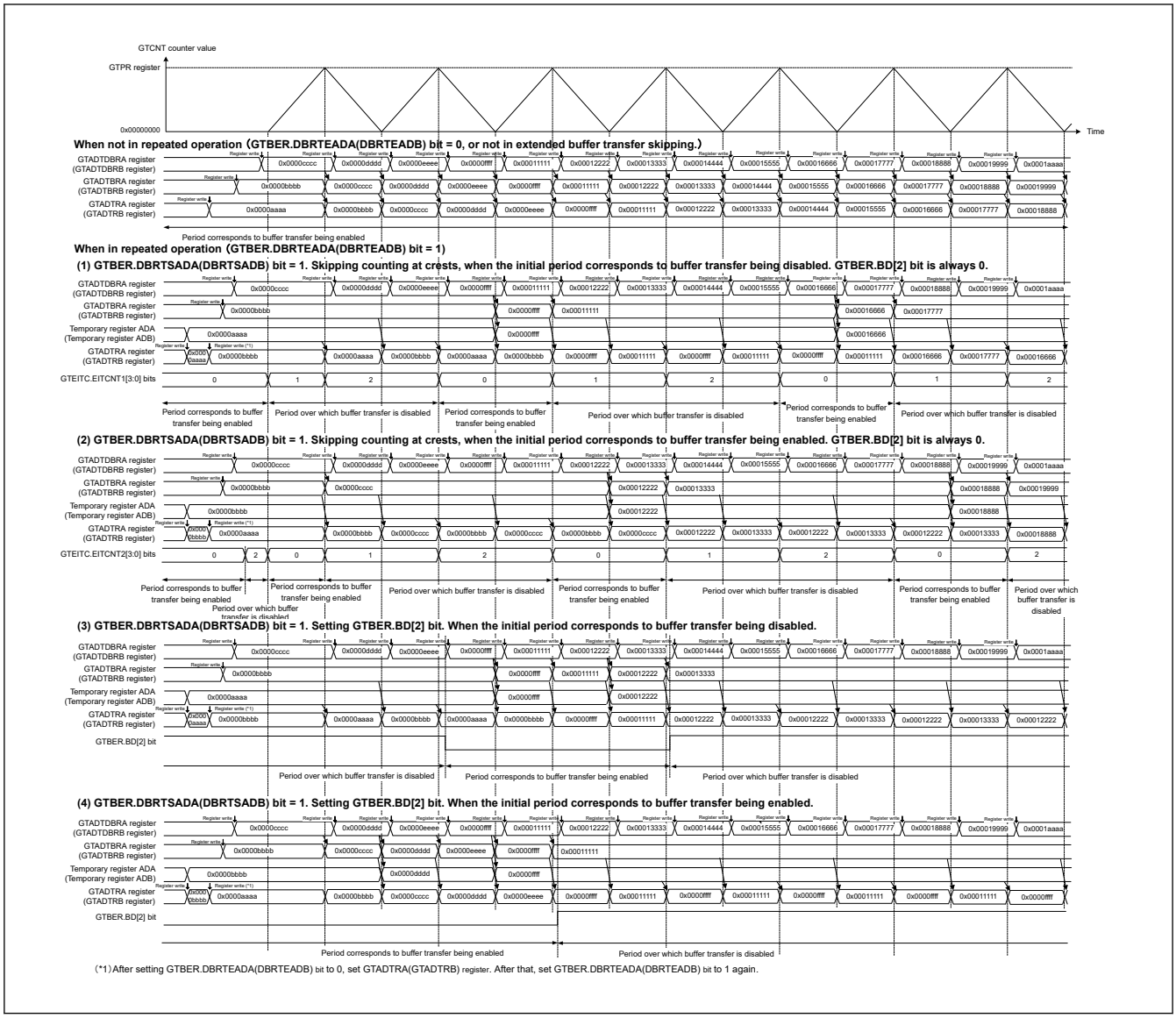


Figure 20.83 Example of repeated double-buffer operation when GTADTR buffer transfer is disabled (triangle-wave, repeating period between crests, second half of period at GTCNT counting start)

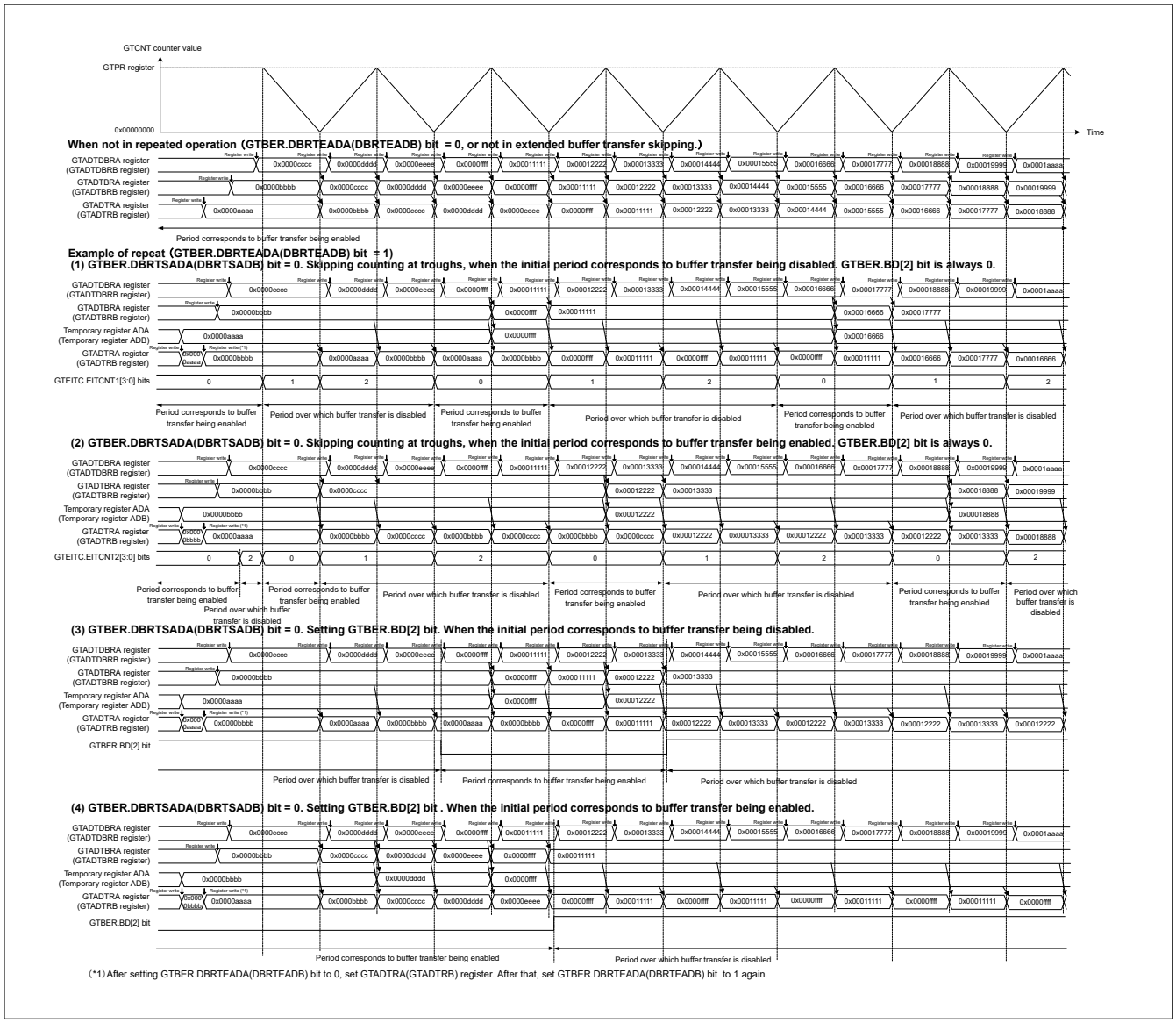


Figure 20.84 Example of repeated double-buffer operation when GTADTR buffer transfer is disabled (triangle-wave, repeating period between troughs, second half of period at GTCNT counting start)

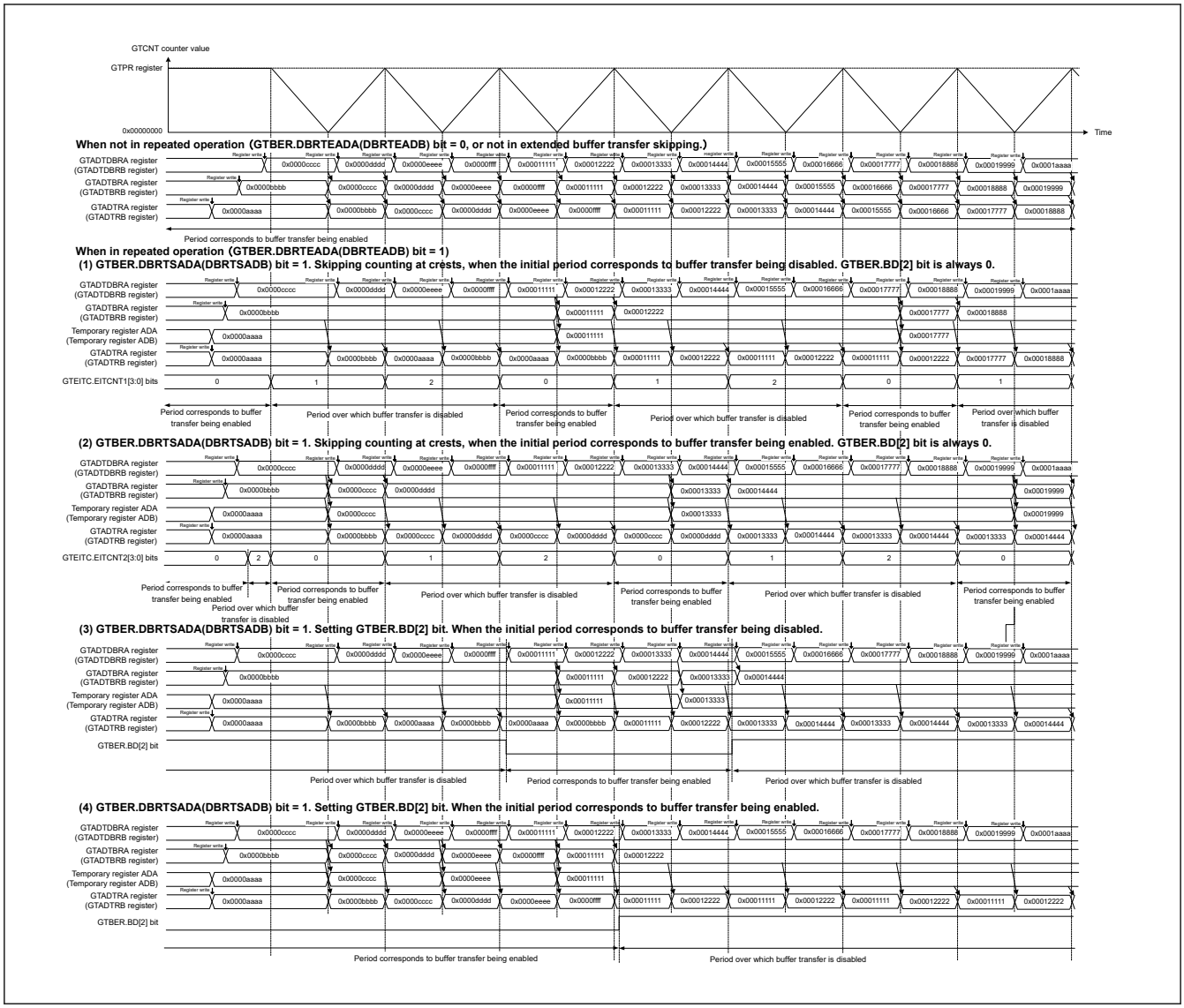


Figure 20.85 Example of repeated double-buffer operation when GTADTR buffer transfer is disabled (triangle-wave, repeating period between crests, first half of period at GTCNT counting start)

20.9.3 GTIOCm_nA and GTIOCm_nB Pin Output Negate Control ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

For protection from system failure, the output negate control that changes the GTIOCm_nA and GTIOCm_nB pin output forcibly by the output stop request from the POEG is provided.

Output protection is required when a dead-time error or the same output level on the GTIOCm_nA and GTIOCm_nB pins is detected. At that time, the detection of disabling of output is output to the group of the POEG set in the GTINTAD.GRP[1:0] bits based on the setting of the output disable detection enable bits, such as the GRPDTE, GRPABH, and GRPABL bits in the GTINTAD register. The POEG takes the logical OR of detection of the types as described and other forms of detection of the disabling of output, and outputs requests for the disabling of output to the GPT.

By setting the GTINTAD.GRP[1:0] bits, one output stop request can be selected out of four output stop requests which are input from the POEG as output stop request signal common in the GTIOCm_nA and GTIOCm_nB pins. Selected output stop request can be checked by reading the GTST.ODF flag.

The states of outputs at the time of control for their negation can be set with the GTIOR.OADF[1:0] bits for the GTIOCm_nA pin and with the GTIOR.OBDF[1:0] bits for the GTIOCm_nB pin.

Transition to the output negate condition by generating the output stop request from the POEG is performed asynchronously, while a release from the output negate condition by making the output stop request no longer satisfied is performed at the end of the cycle. It is after 3 PCLKGPTL (LLPP)/PCLKM (NONSAFETY, SAFETY) at shortest when the output negate

condition is released after the output stop request is no longer satisfied. To reliably control output negation, clear the POEG flag for which the condition of the request to disable the output is no longer satisfied after 4 PCLKGPTL (LLPP)/PCLKM (NONSAFETY, SAFETY) cycles.

To release the output stop condition during event count operation or without waiting the end of the cycle, set the OADF[1:0] bits to 00b for the GTIOCm_nA pin, and set the OBDF[1:0] bits to 00b for the GTIOCm_nB pin. Figure 20.86 shows an example of the output negate control operation for the GTIOCm_nA and GTIOCm_nB pin output.

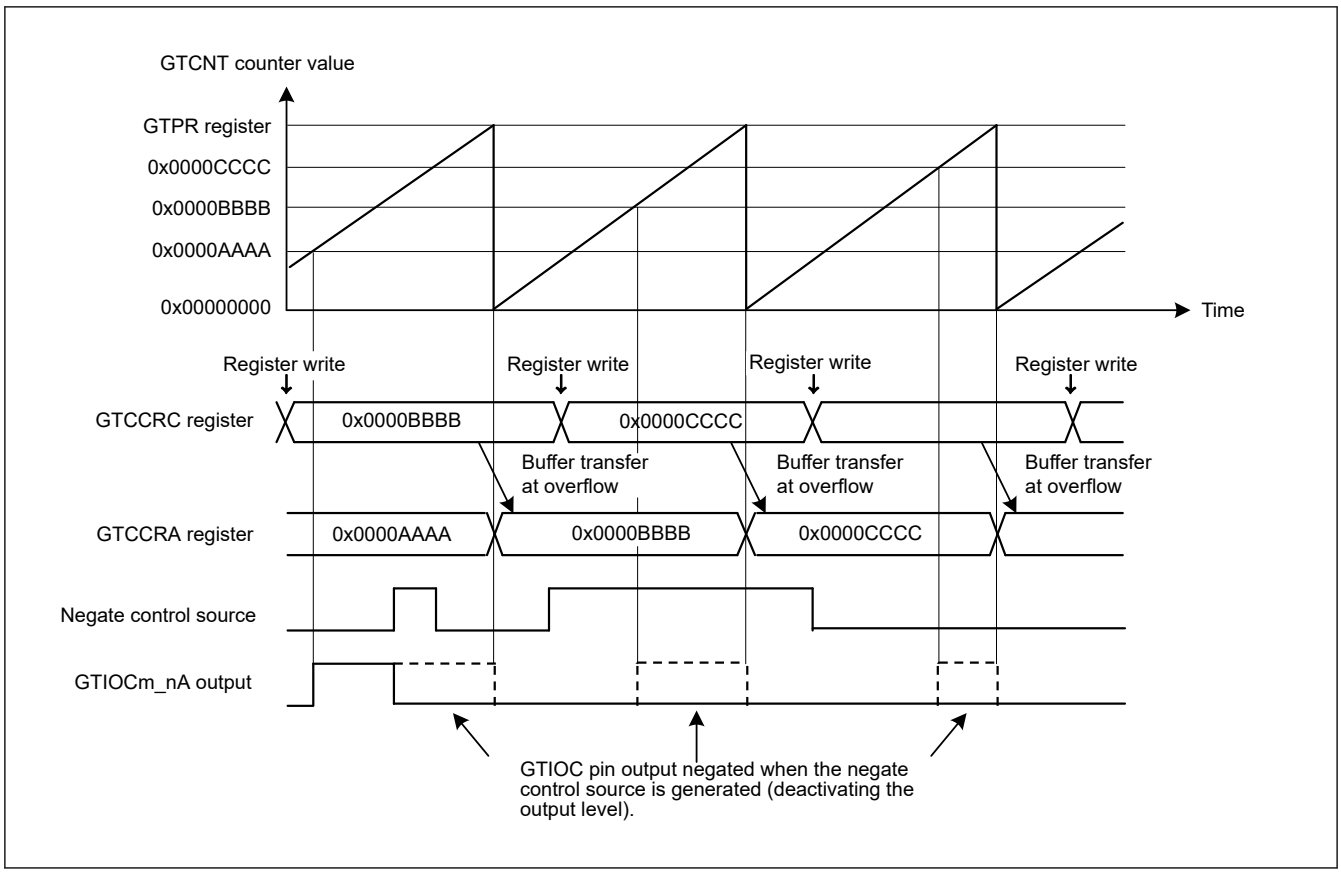


Figure 20.86 Example of operation for GTIOCm_nA and GTIOCm_nB pin output negate control (sawtooth-wave up-counting, buffer operation, active-level: high output at GTCCRA register compare match, low output at the end of the cycle, and low output at output negate) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

20.9.4 GTIOCm_nA and GTIOCm_nB Protection Function for GTIOCm_nA and GTIOCm_nB Pin Output ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

To prepare for a case when an incorrect value (0x00000000 or a value greater than or equal to the GTPR register value) is set in the GTCCRA register, the output protection function for the GTIOCm_nA and GTIOCm_nB pin output (disabling function) is activated when the automatic dead time is set (GTDTCR.TDE bit = 1) in triangle-wave PWM mode.

The status of the output protection function can be read from the GTSOS.SOS[1:0] bits.

Figure 20.87 shows the output protection function state transition.

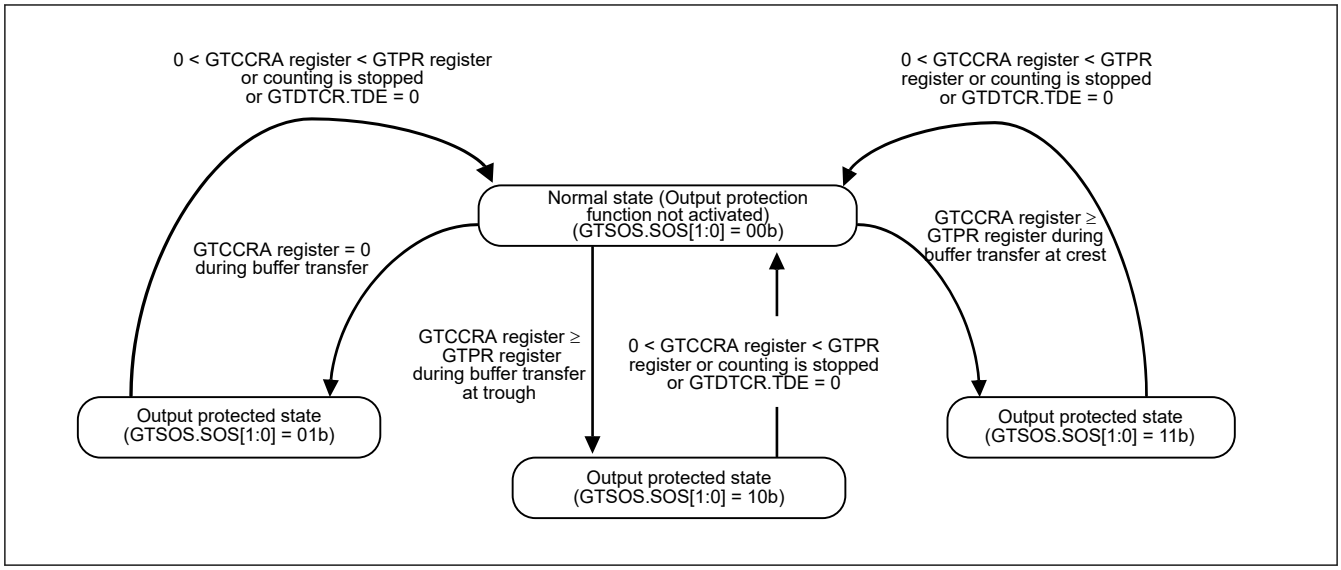
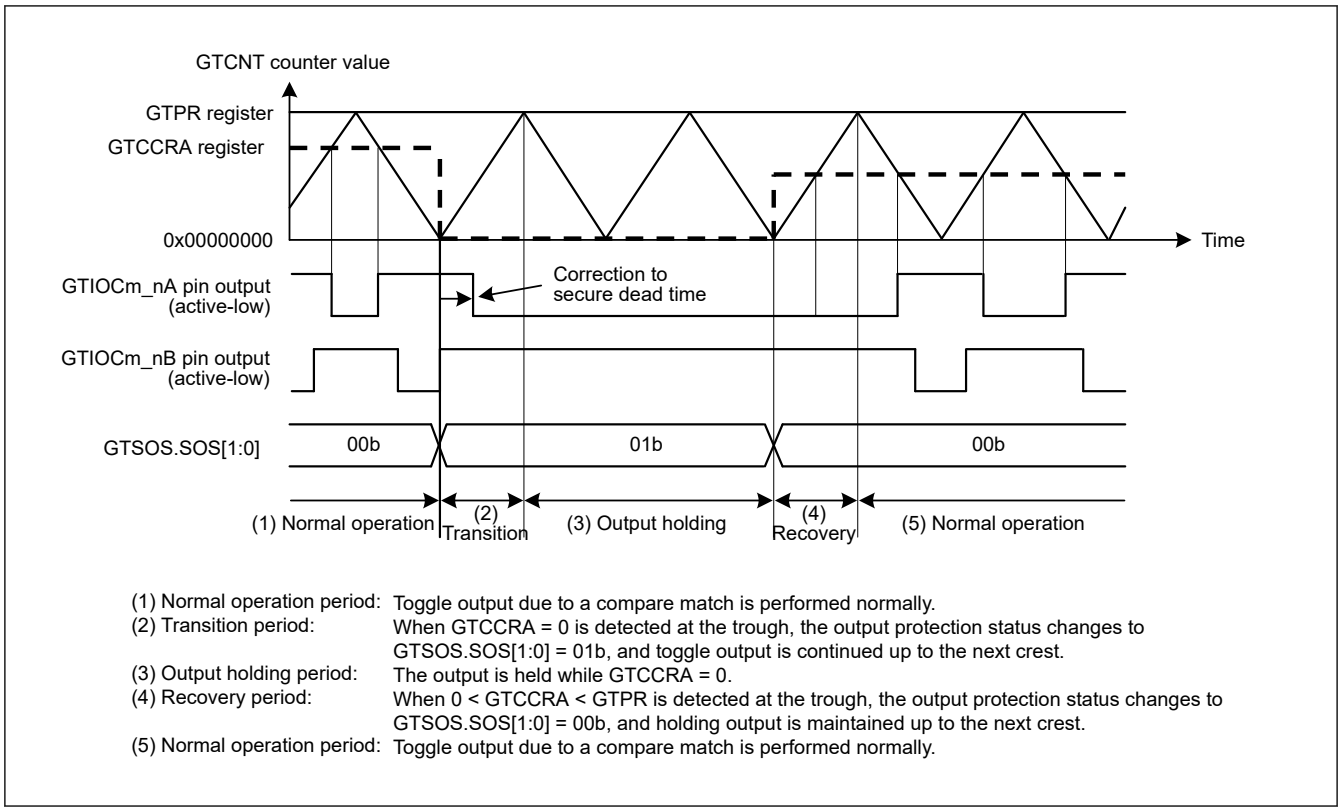


Figure 20.87 Output protection function

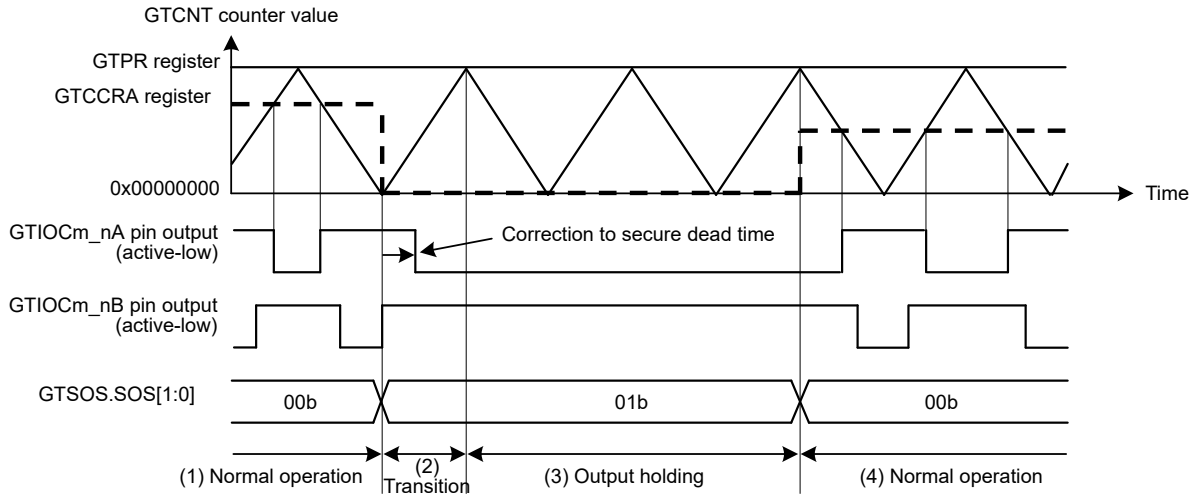
(1) Output Protection Function when the GTCCRA Register is Set to 0x00000000 during Buffer Transfer

Figure 20.88 and Figure 20.89 show examples of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at troughs, and Figure 20.90 and Figure 20.91 show examples of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at crests.



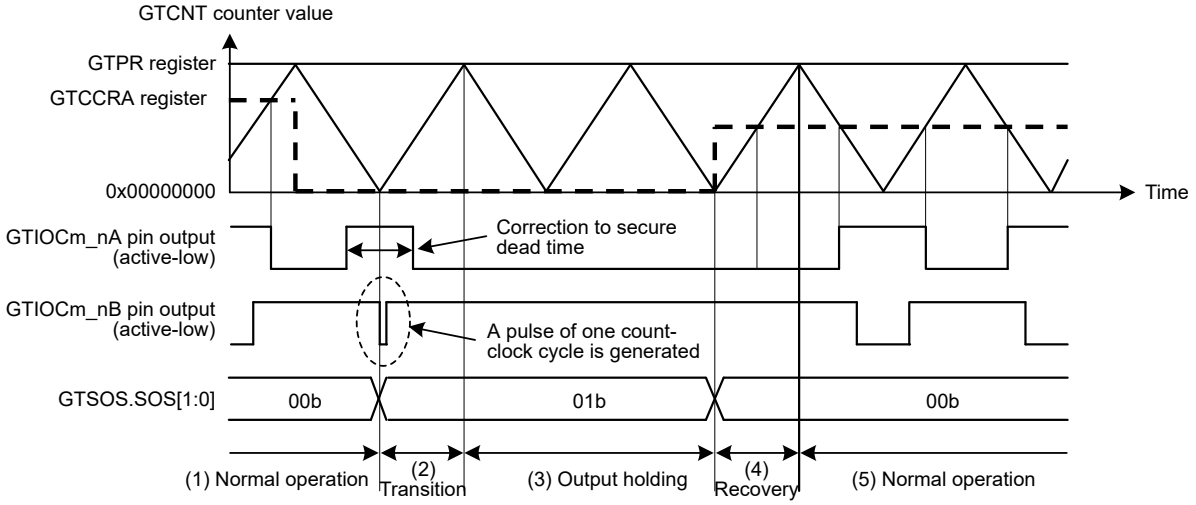
(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA = 0 is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA = 0.
(4) Recovery period: When 0 < GTCCRA < GTPR is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 20.88 Example of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at troughs (restored to 0 < GTCCRA register < GTPR register during buffer transfer at troughs, active-level: low) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA = 0 is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA = 0.
(4) Normal operation period: When 0 < GTCCRA < GTPR is detected at the crest, the output protection status changes to GTSOS.SOS[1:0] = 00b, and toggle output due to a compare match is performed normally.

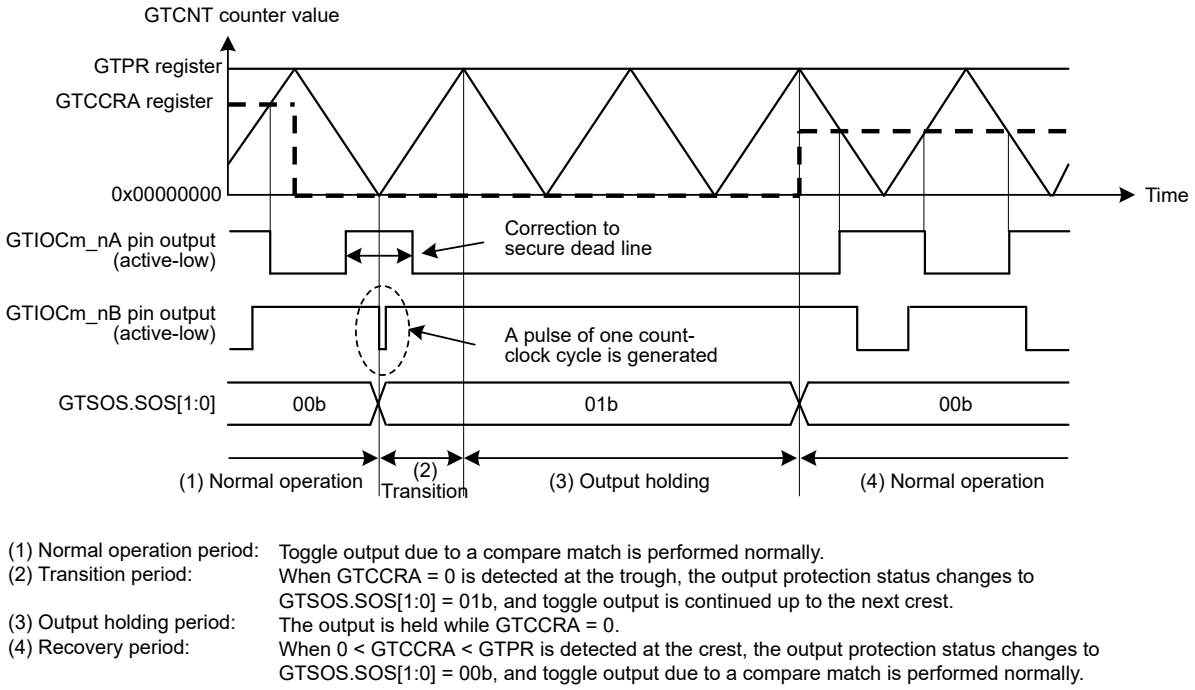
Figure 20.89 Example of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at troughs (restored to 0 < GTCCRA register < GTPR register during buffer transfer at crests, active-level: low) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA = 0 is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA = 0.
(4) Recovery period: When 0 < GTCCRA < GTPR is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 20.90 Example of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at crests (restored to 0 < GTCCRA register < GTPR register

during buffer transfer at troughs, active-level: low) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

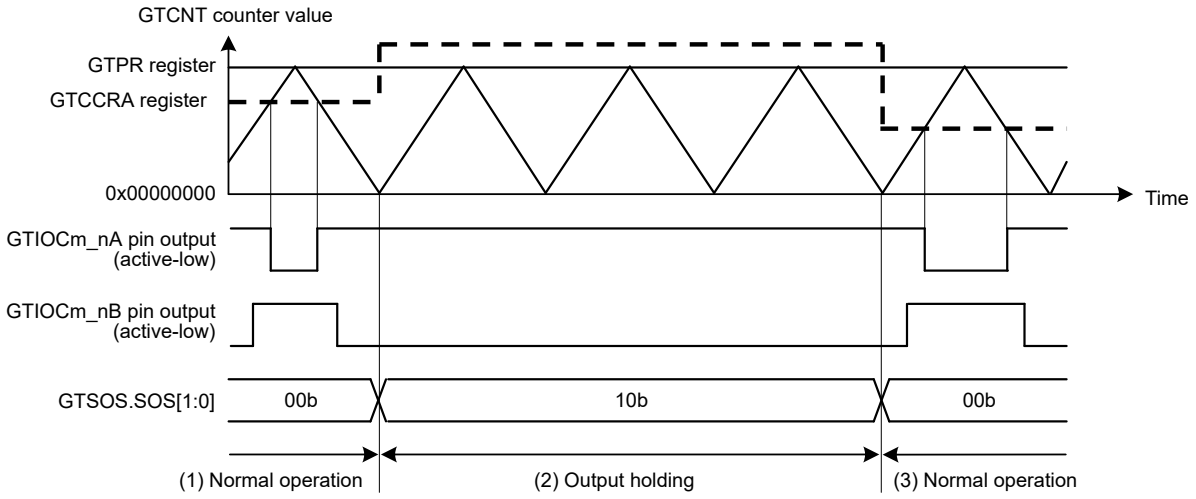


(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA = 0 is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA = 0.
(4) Recovery period: When 0 < GTCCRA < GTPR is detected at the crest, the output protection status changes to GTSOS.SOS[1:0] = 00b, and toggle output due to a compare match is performed normally.

Figure 20.91 Example of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at crests (restored to 0 < GTCCRA register < GTPR register during buffer transfer at crests, active-level: low) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

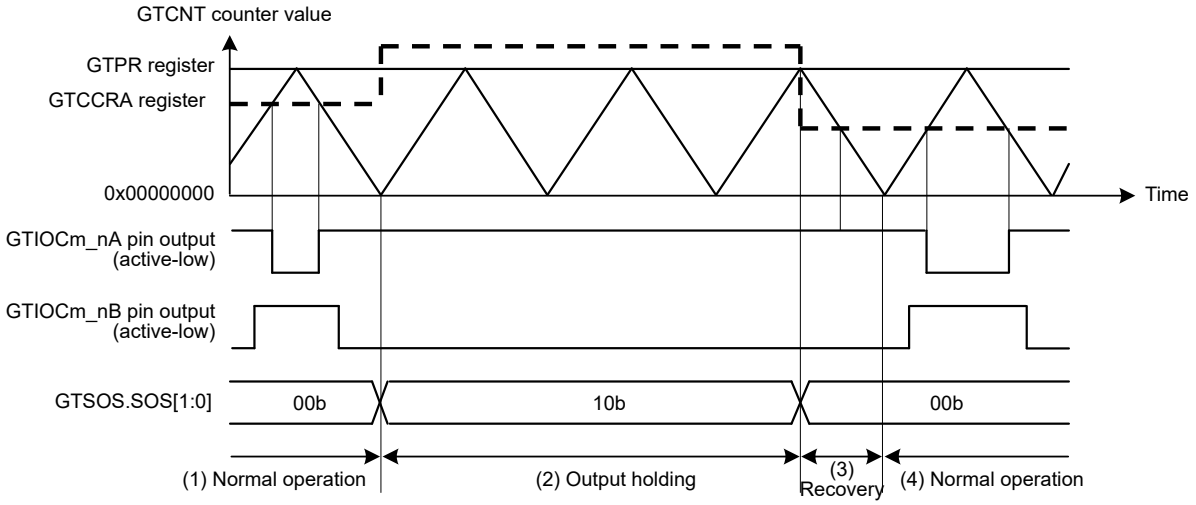
(2) Output Protection Function when GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Troughs

Figure 20.92 and Figure 20.93 show examples of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at troughs.



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA ≥ GTPR is detected at the trough, the output protection status changes to GTSOS[1:0] = 10b, and the output is held while GTCCRA ≥ GTPR.
(3) Normal operation period: When 0 < GTCCRA < GTPR is detected at the trough, the output protection status changes to GTSOS[1:0] = 00b, ant toggle output due to a compare match is performed normally.

Figure 20.92 Example of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at troughs (restored to 0 < GTCCRA register < GTPR register during buffer transfer at crests, active-level: low) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

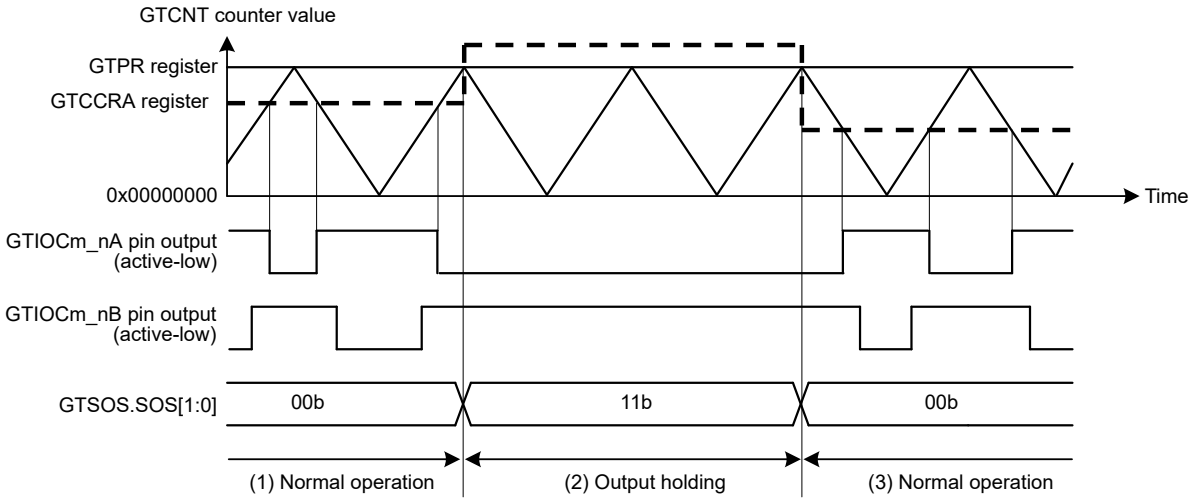


(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA ≥ GTPR is detected at the trough, the output protection status changes to GTSOS[1:0] = 10b, and the output is held while GTCCRA ≥ GTPR.
(3) Recovery period: When 0 < GTCCRA < GTPR is detected at the crest, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next trough.
(4) Normal operation period: When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to GTSOS[1:0] = 00b, and toggle output due to a compare match is performed normally.

Figure 20.93 Example of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at troughs (restored to 0 < GTCCRA register < GTPR register during buffer transfer at troughs, active-level: low) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

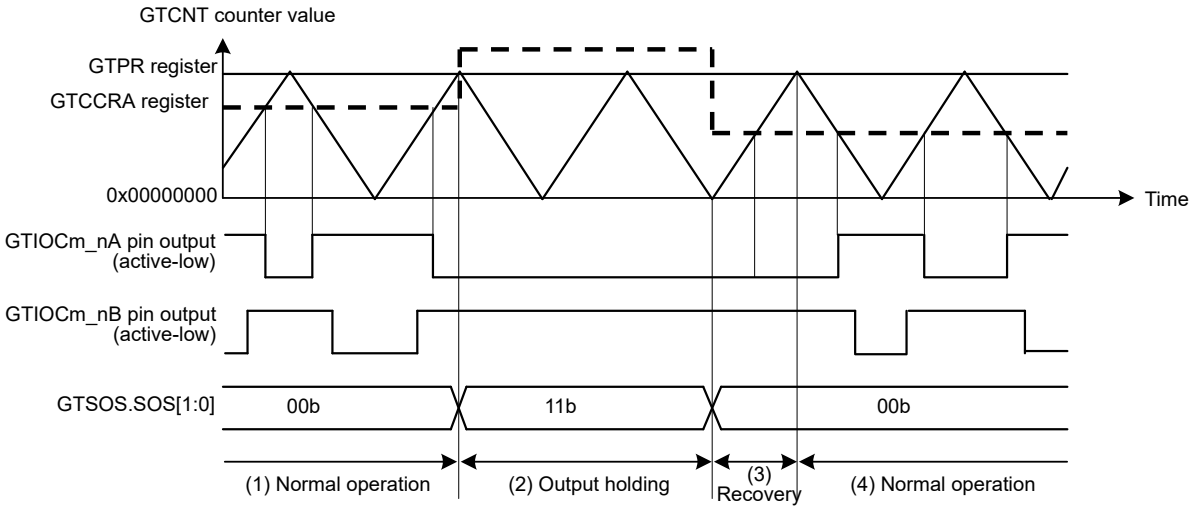
(3) Output Protection Function when GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Crests

Figure 20.94 and Figure 20.95 show examples of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at crests.



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA ≥ GTPR is detected at the crest, the output protection status changes to GTSOS[1:0] = 11b, and the output is held while GTCCRA ≥ GTPR.
(3) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 20.94 Example of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at crests (restored to 0 < GTCCRA register < GTPR register during buffer transfer at crests, active-level: low) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA ≥ GTPR Is detected at the crest, the output protection status changes to GTSOS[1:0] = 11b, and the output is held while GTCCRA ≥ GTPR.
(3) Recovery period: When 0 < GTCCRA < GTPR is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next crest.
(4) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 20.95 Example of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at crests (restored to 0 < GTCCRA register < GTPR register during buffer transfer at troughs, active-level: low) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

(4) Restricted Specification of Output Protection Function

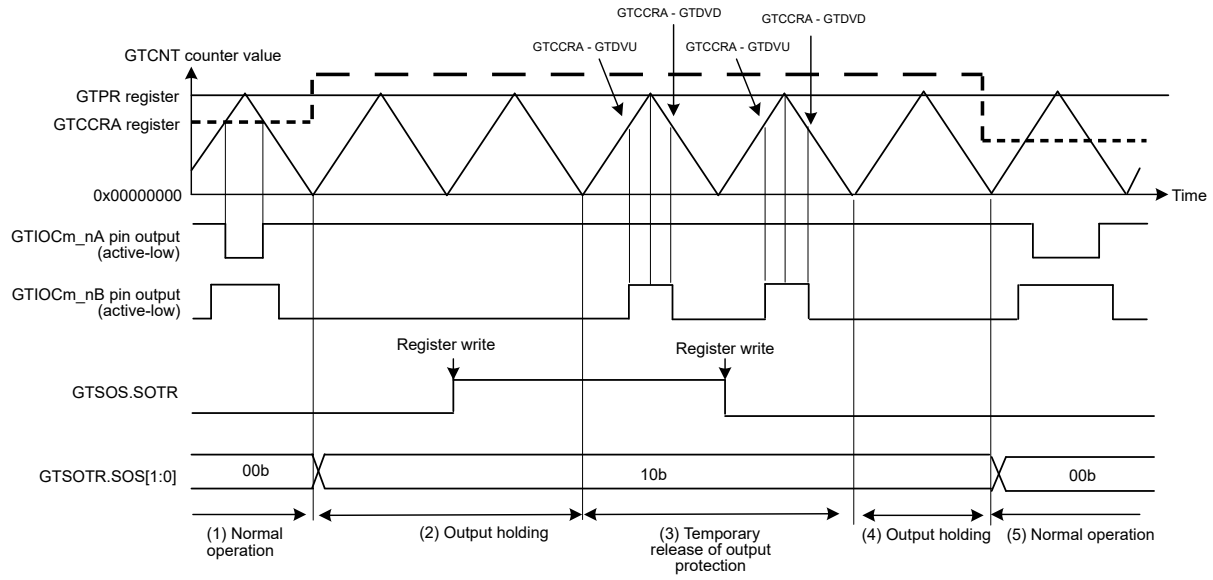
Even if an incorrect value (0x00000000 or a value greater than or equal to the GTPR register value) is set in the GTCCRA register during count operation, the output protection functions in a specific way such that one of the positive- and negative-phase outputs becomes non-active. However, the output protection does not operate normally when the GTCCRA register value at the start of count operation is greater than 0x00000000 and less than the setting value of the GTPR register.

(5) Temporary Release of Output Protection Function

When the GTSOS.SOS[1:0] bits = 10b (protected state in which GTCCRA register ≥ GTPR register occurred during transfer at trough), the protected state of the GTIOCm_nB pin output can be temporarily released by setting the GTSOTR.SOTR bit to 1. The SOS[1:0] bits retain 10b even if the output protection function is released.

When the SOTR bit is set to 0, the GTIOCm_nB pin output protection can be restarted.

Figure 20.96 shows an example of the operation of temporary release of output protection when the setting of the GTCCRA register ≥ GTPR register during buffer transfer at troughs.



(1) Normal operation period: Toggle output due to compare match is performed normally.

(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to GTSOS[1:0] = 10b, and the output is held while GTCCRA register ≥ GTPR register. Even when the GTSOTR.SOTR bit is set to 1 during the output holding period, the GTIOCm_nB pin output is held until the first trough after the setting.

(3) Temporary release of output protection period: After GTSOTR.SOTR bit is set to 1, the GTIOCm_nB pin is performed normally from the first trough after the setting. Even when the GTSOTR.SOTR bit is set to 0 during the output protection releasing period, the GTIOCm_nB pin output is held until the first trough after the setting.

(4) Output holding period: The output is held until 0 < GTCCRA register < GTPR register is detected in the trough.

(5) Normal operation period: When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to GTSOS[1:0] = 00b, and toggle output due to a compare match is performed normally.

Figure 20.96 Example of the operation of temporary release of output protection when the setting of the GTCCRA register ≥ GTPR register during buffer transfer at troughs (restored to 0 < GTCCRA register < GTPR register during buffer transfer at troughs, active-level: low) ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3))

20.10 Initialization Method of Output Pins

20.10.1 Pin Settings after Reset

The GPT registers are initialized at a reset. Initialize the GPT for output to the external pins by making the port mode settings and setting the OAE and OBE bits in the GTIOR register, and then start the counter counting.

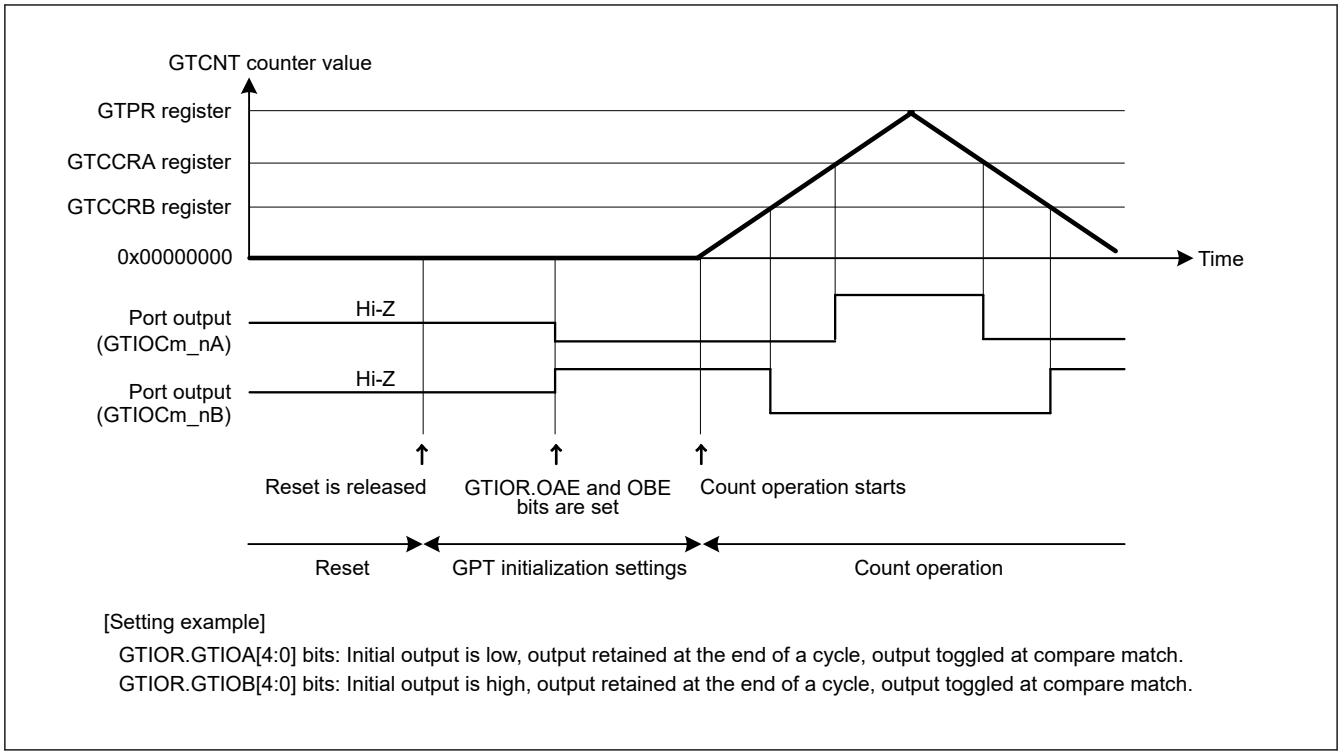


Figure 20.97 Example of pin settings after reset (n = 0 to 17)

20.10.2 Pin Initialization Due to Error during Operation

If an error occurs during GPT operation, the following four types of pin processing can be performed before pin initialization.

1. Set OAHLD and OBHLD bits in the GTIOR register to 1 and retain the outputs at count stop.
2. Set OAHLD and OBHLD bits to 0, specify arbitrary output values at OADFLT and OBDFLT bits in the GTIOR register, and output the arbitrary values at count stop.
3. Set the pin to output an arbitrary value as a general output port by setting the registers of the I/O port in advance. Set the OAE and OBE bits in the GTIOR register to 0 and the control bit in register that corresponds to the pin to allow the arbitrary values to be output from the pin set as a general output port when an error occurs.
4. Drive the output to a high impedance state using the POEG function.

When automatic dead time setting is made, set the GTDTCR.TDE bit to 0 once after counting is stopped. When counting is stopped, only the values of registers that are changed by a GPT external source change. If counting is resumed, operation carries on from where it was stopped.

If counting was stopped, registers should be initialized before counting is started.

20.11 Usage Notes

20.11.1 Module-Stop Function Setting

The Module Stop Control register can enable or disable operation of the GPT. The initial setting is to halt operation of the GPT. Register access is enabled by clearing the module-stop state. For details, see section 9, Low-Power Consumption Function.

20.11.2 Settings of the GTCCRn Register During Compare Match Operation (n = A to F)

(1) When Automatic Dead Time Setting is Made in Triangle-Wave PWM Mode

The GTCCRA register must satisfy the following conditions:

- GTCCRA register > GTDVU register

- GTCCRA register > GTDVD register
- GTCCRA register < GTPR register

When the GTCCRA register is set to 0x00000000 or a value greater than or equal to the GTPR register value during count operation, the output protection function is activated.

However, if the following conditions are not satisfied, the output protection does not operate normally.

- When the GTCCRA register value at the start of count operation is 0x00000001 or greater, and less than the setting value of the GTPR register

For details, see section 20.9.4. GTIOCm_nA and GTIOCm_nB Protection Function for GTIOCm_nA and GTIOCm_nB Pin Output ((m = 00 to 08 and n = 0 to 4), (m = 09 and n = 0 to 6), or (m = 10 and n = 0 to 3)).

(2) When Automatic Dead Time Setting is not Made in Triangle-Wave PWM Mode

Set a value greater than 0x00000000 and less than the setting value of the GTPR register in the GTCCRA register. When 0x00000000 or the same value as that of the GTPR register is set in the GTCCRA register, compare match is generated in one cycle only when [GTCCRA register = 0x00000000] or [GTCCRA register = GTPR register] is met. Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRA register, compare match does not occur.

Similarly, set a value greater than 0x00000000, and less than the setting value of the GTPR register in the GTCCRB register. When 0x00000000 or the same value as that of the GTPR register is set in the GTCCRB register, compare match is generated in one cycle only when [GTCCRB register = 0x00000000] or [GTCCRB register = GTPR register] is met.

Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRB register, compare match does not occur.

(3) When Automatic Dead Time Setting is Made in Sawtooth-Wave One-Shot Pulse Mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, correct output waveforms with secured dead time may not be obtained.

- In up-counting:

  GTCCRC register < GTCCRD register

  GTCCRC register > GTDVU register

  GTCCRD register < GTPR register − GTDVD register
- In down-counting:

  GTCCRC register > GTCCRD register

  GTCCRC register < GTPR register − GTDVU register

  GTCCRD register > GTDVD register

(4) When Automatic Dead Time Setting is not Made in Sawtooth-Wave One-Shot Pulse Mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC register < GTCCRD register < GTPR register
- In down-counting: GTPR register > GTCCRC register > GTCCRD register > 0

Similarly, GTCCRE and GTCCRF registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRE register < GTCCRF register < GTPR register
- In down-counting: GTPR register > GTCCRE register > GTCCRF register > 0

(5) In Sawtooth-Wave PWM Mode

The GTCCRA register must be set with the range of 0x00000000 < GTCCRA register < GTPR register. If GTCCRA register = 0x00000000 or GTCCRA register = GTPR register is set, a compare match occurs within the cycle only when

GTCCRA register = 0x00000000 or GTCCRA register = GTPR register is satisfied. If GTCCRA register > GTPR register is set, no compare match occurs.

Similarly, GTCCRB register must be set with the range of 0x00000000 < GTCCRB register < GTPR register. If GTCCRB register = 0x00000000 or GTCCRB register = GTPR register is set, a compare match occurs within the cycle only when GTCCRB register = 0x00000000 or GTCCRB register = GTPR register is satisfied. If GTCCRB register > GTPR register is set, no compare match occurs.

20.11.3 Setting Range of the GTCNT Counter

Set the range of the GTCNT counter within the range of 0 ≤ GTCNT counter ≤ GTPR register.

20.11.4 The GTCNT Counter Start/Stop

The GTCNT counter start/stop control by the GTCR.CST bit is synchronized with the count clock selected by the GTCR.TPCS[3:0] bits. Since the GTCNT counter starts or stops after one count clock cycle selected by the GTCR.TPCS[3:0] bits following the CST bit updating, events until the GTCNT counter actually starts are ignored, where events may be accepted and the interrupt may be generated after the CST bit becomes 0.

20.11.5 Order of Priority in Events

(1) GTCNT Counter

Order of priority in events to update the GTCNT counter is shown below.

Table 20.28 Order of priority in updating GTCNT counter

The GTCNT counter updating source	Order of priority
CPU writing (GTCNT counter writing/GTCLR register writing)	High
Clearing by a hardware source set by the GTCSR register	↑
Up-counting/down-counting by a hardware source set by the GTUPSR and GTDNSR registers	↑
Count operation	Low

In case of contention between up-counting by the GTUPSR register and down-counting by the GTDNSR register, the counter value is not updated.

When updating of the GTCNT counter conflicts with reading by the CPU, the value before updating is reflected.

(2) GTCR.CST Bit

When start/stop by a hardware source set by the GTSSR and GTPSR registers conflicts with the CPU writing (GTCR register writing/GTSTR register writing/GTSTP register writing), the CPU writing takes priority.

When there is contention between start by a hardware source set by the GTSSR register and stop by a hardware source set by the GTPSR register, the state of the CST bit does not change.

When there is contention between the CST bit updating and the CPU reading (GTCR register reading/GTSTR register reading/GTSTP register reading), data before updating is read.

(3) GTCCRm Register (m = A to F)

When there is contention between the GTCCRm register writing and an input capture/buffer transfer, the GTCCRm register writing takes priority over an input capture/buffer transfer.

When an input capture conflicts with the CPU writing or the counter updating by a hardware source, the counter value before updating is captured.

When there is contention between the GTCCRm register updating and the CPU reading, data before updating is read.

(4) GTPR Register

When there is contention between a buffer transfer and the GTPR register writing, the GTPR register writing takes priority over a buffer transfer.

When there is contention between the GTPR register updating and the CPU reading, data before updating is read.

(5) GTADTRm Register (m = A, B)

When there is contention between a buffer transfer and the GTADTRm register writing, the GTADTRm register writing takes priority over a buffer transfer.

When there is contention between the GTADTRm register updating and the CPU reading, data before updating is read.

(6) GTDVm Register (m = U, D)

When there is contention between a buffer transfer and the GTDVm register writing, the GTDVm register writing takes priority over a buffer transfer.

When there is contention between the GTDVm register updating and the CPU reading, data before updating is read.

20.11.6 Input Capture Signal Select

Input capture signals for channel 4 of GPTm unit (m = 00 to 08 (LLPP0 and LLPP1)) can be selected from GTIOCm_4A / GTIOCm_4B or GTIOCm_3A / GTIOCm_3B by setting GTIOCSEL register. When GTIOCm_3A / GTIOCm_3B are selected, same input capture signals can be used for channel 3 and channel 4 in GPTm unit. It enables phase counting and pulse width measurement of the input capture signals at the same time. Note that output signals are not changed.

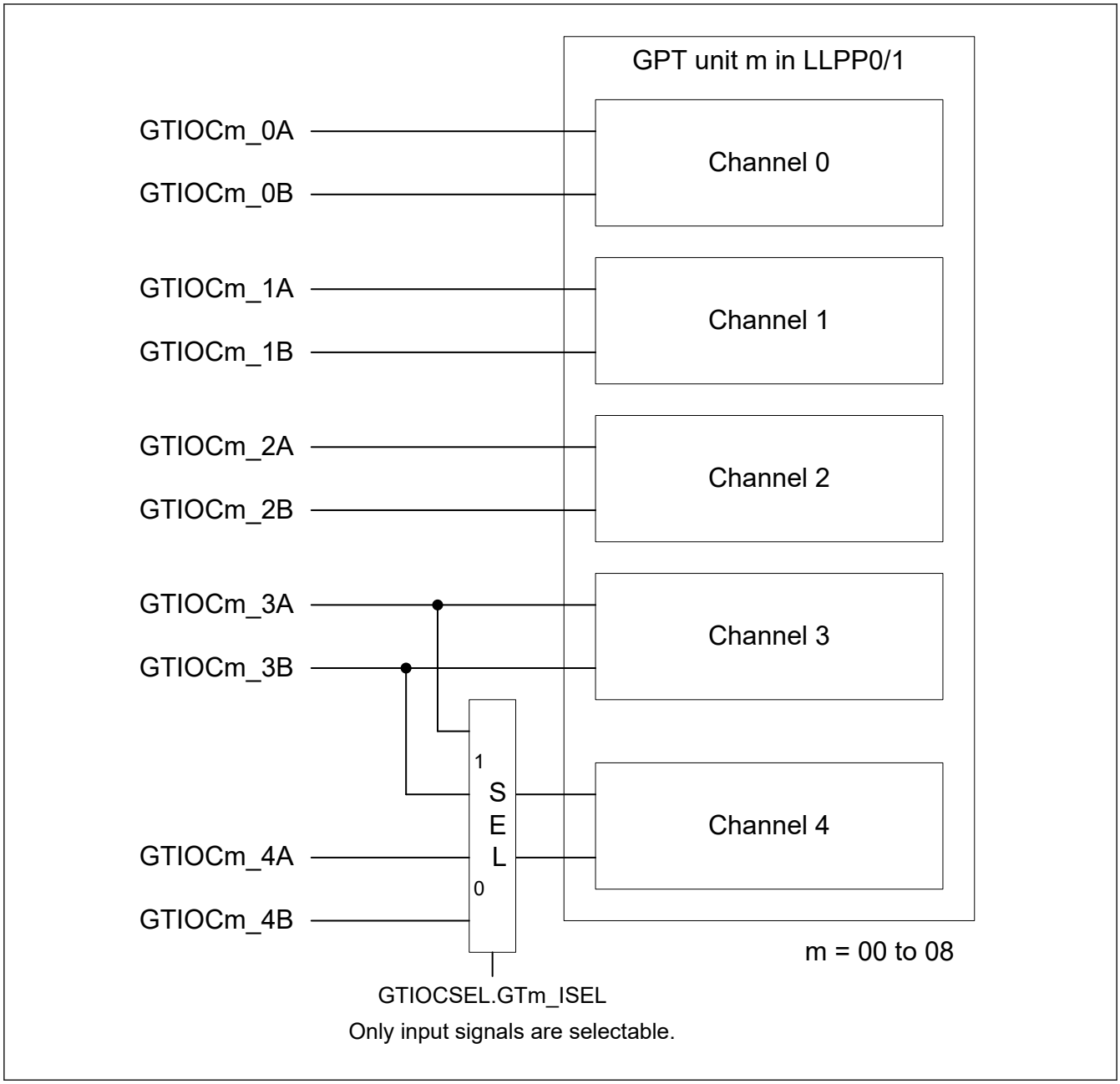


Figure 20.98 Input signal connection in unit 00 to 08

21. Port Output Enable for GPT (POEG)

21.1 Overview

The POEG issues requests to stop output from output pins of the general PWM timer (GPT). The combination of output pins of the POEG to be disabled can be specified from any channel of the GPT.

Select the method of detection for stopping the output from the list below.

- Input level detection of the GTETRGn pin (n = A to D)
- Detection from the GPT to stop output
- Detection of stopping of oscillation by the oscillation stop detection circuit for the main clock
- Register setting

The GTETRGn pin can be used for output to the external trigger input pins of the GPT.

Table 21.1 lists POEG specifications, Figure 21.1 shows a block diagram of POEG, Figure 21.4 shows POEG external trigger connections, Table 21.2 lists POEG I/O pins, and Table 21.3 lists POEG Interrupt Sources.

Table 21.1 POEG specifications

Item	Description
Number of units	Three units (LLPP (POEG0), NONSAFETY (POEG1), SAFETY (POEG2))
Number of groups	Four groups (A to D)
Output-disable control through input level detection	The GPT output pins can be disabled when a GTETRGA to GTETRGD rising edge or high level is sampled after polarity and filter selection.
Output-disable request from the GPT	• When the GTIOCA pin and the GTIOCB pin are driven to an active level simultaneously, the GPT generates an output-disable request to the POEG. Through reception of these requests, the POEG can control whether the GTIOCA and GTIOCB pins are output-disabled. • GPT output pins can be set to be disabled when the GPT output pins detect a dead time error.
Output-disable control through oscillation stop detection	The GPT output pins can be disabled when oscillation of the clock generation circuit stops.
Output-disable control by software (registers)	The GPT output pins can be disabled by modifying the register settings.
Output-disable control through DSMIF error detection	The GPT output pins can be disabled when DSMIF error 0 or 1 occurs (LLPP only).
Interrupt	• Allows output-disable control by the input level detection • Allows output-disable requests from the GPT
External trigger output to the GPT (count start, count stop, count clear, up-count, down-count, or input capture function)	The GTETRGA to GTETRGD signals can be output to the GPT after polarity and filter selection.
Noise filtering	• Three times sampling for every PCLK/1, PCLK/8, PCLK/32, or PCLK/128 can be set for any of the input pins GTETRGA to GTETRGD. (PCLK is PCLKH for LLPP and PCLKL for NONSAFETY and SAFETY). • Positive or negative polarity can be selected for any of the input pins, GTETRGA to GTETRGD. • Signal state after polarity and filter selection can be monitored.

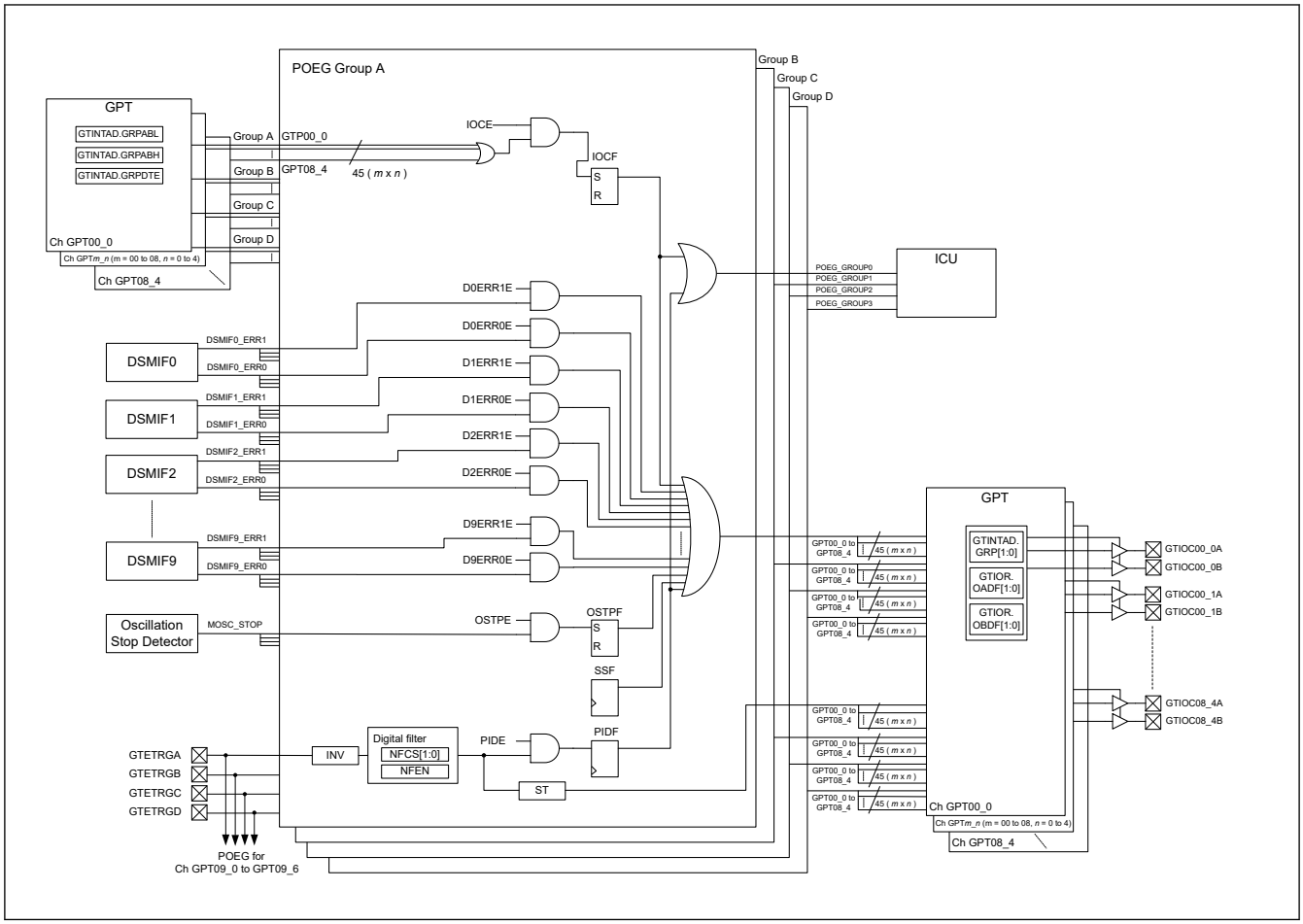


Figure 21.1 POEG0 block diagram

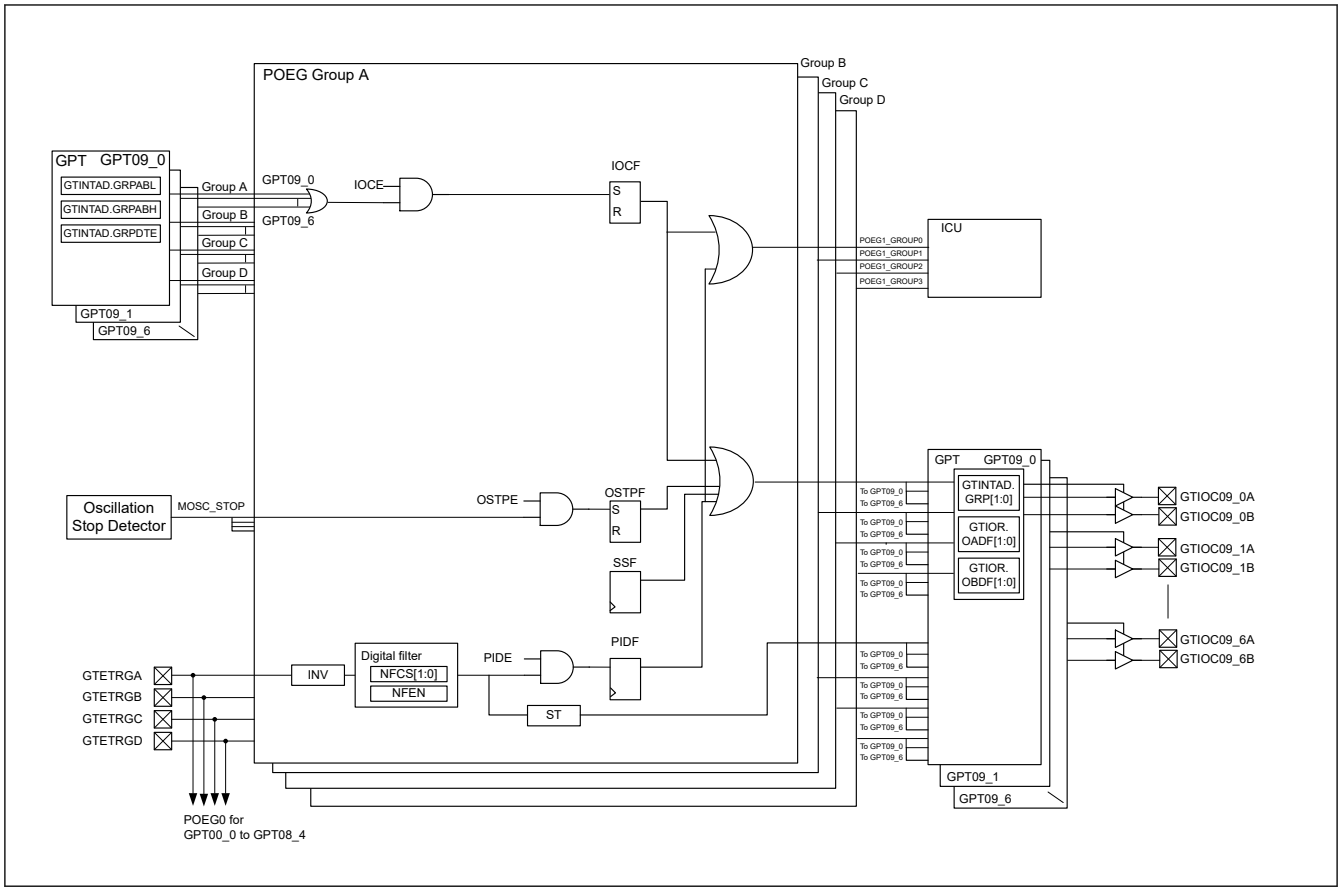


Figure 21.2 POEG1 block diagram

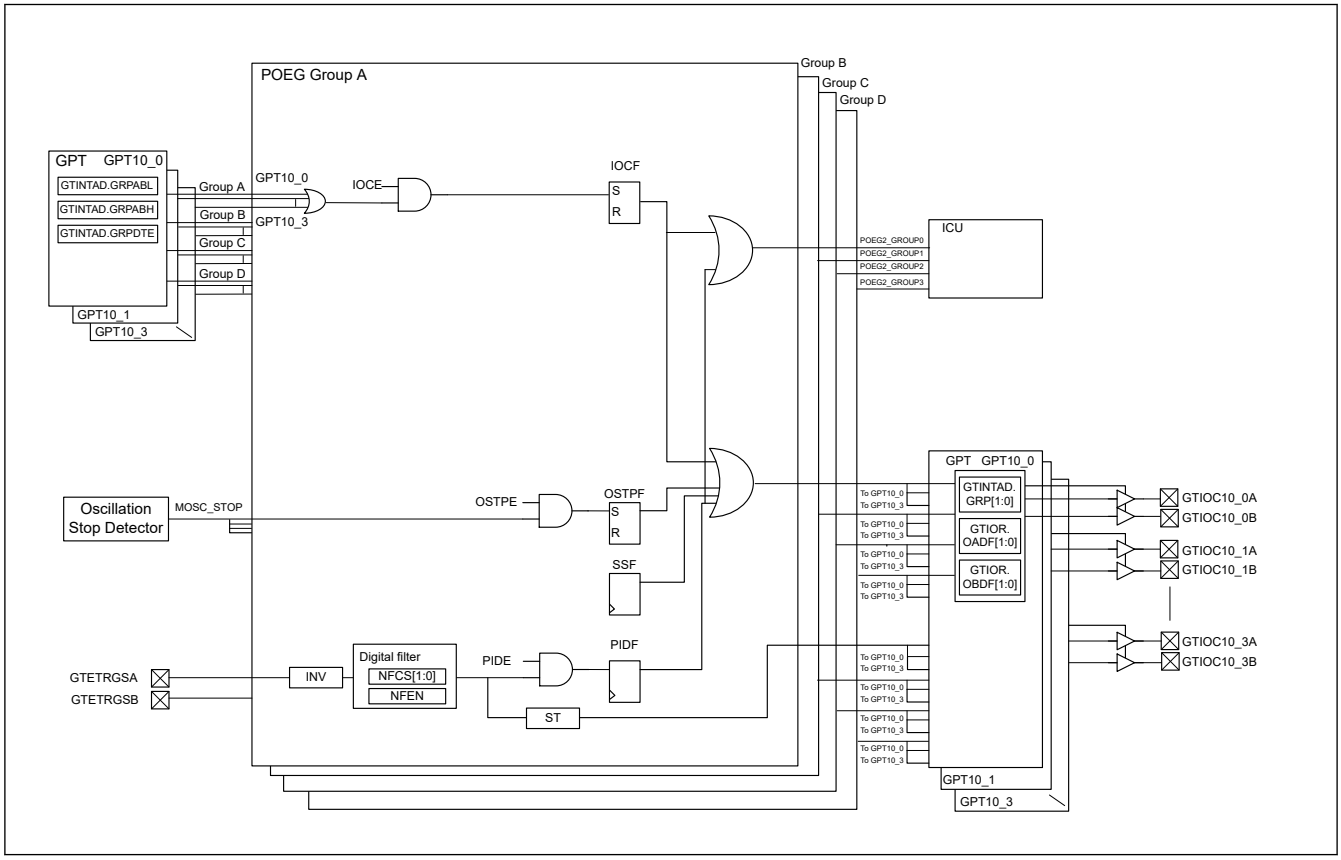


Figure 21.3 POEG2 block diagram

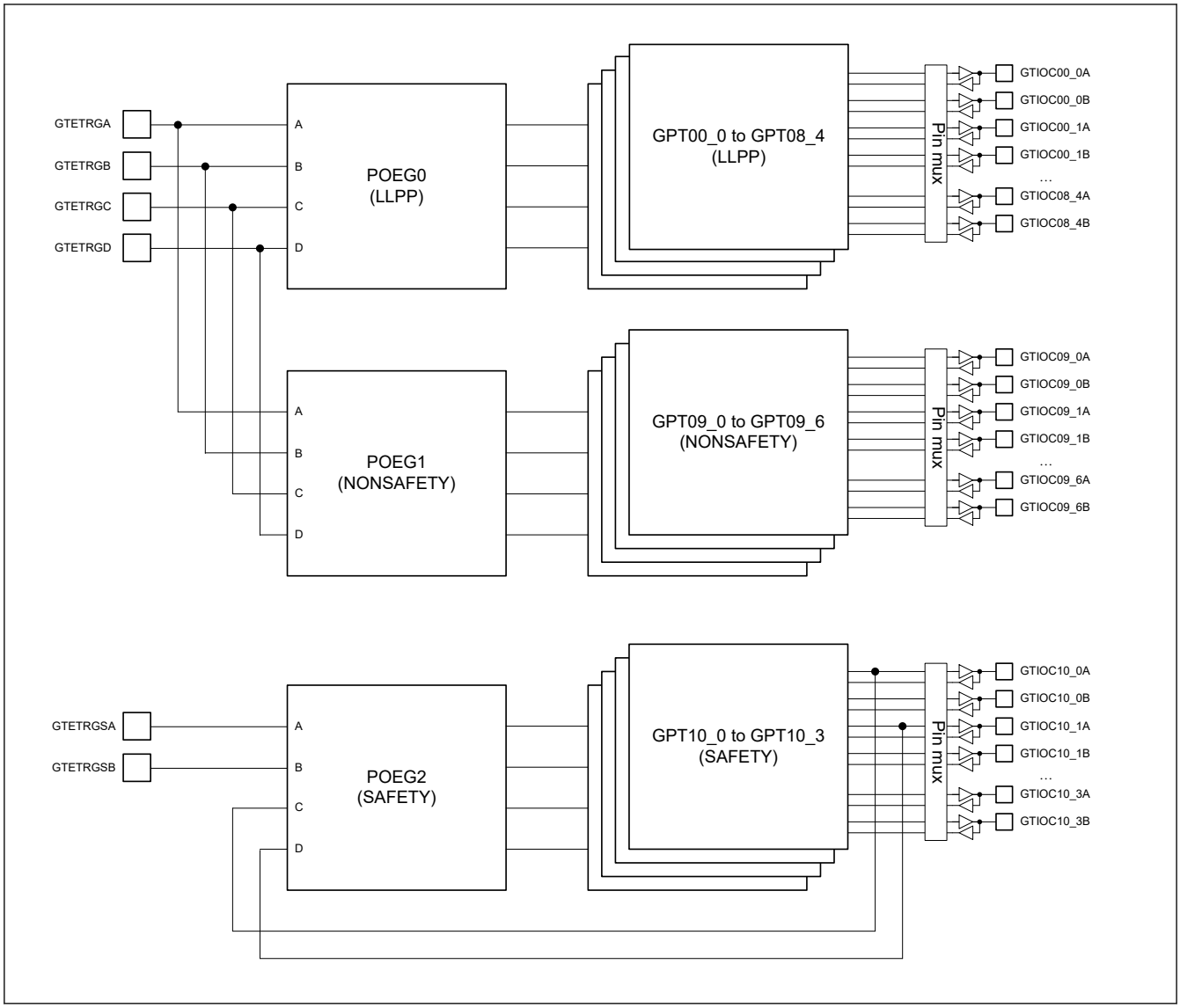


Figure 21.4 External trigger connections

Table 21.2 POEG I/O pins

Pin name	I/O	Function
GTETRGA	Input	GPT output pin output-disable request signal and GPT external trigger input pin A for channels in LLPP and NONSAFETY
GTETRGB	Input	GPT output pin output-disable request signal and GPT external trigger input pin B for channels in LLPP and NONSAFETY
GTETRGC	Input	GPT output pin output-disable request signal and GPT external trigger input pin C for channels in LLPP and NONSAFETY
GTETRGD	Input	GPT output pin output-disable request signal and GPT external trigger input pin D for channels in LLPP and NONSAFETY
GTETRGSA	Input	GPT output pin output-disable request signal and GPT external trigger input pin A for channels in SAFETY
GTETRGSB	Input	GPT output pin output-disable request signal and GPT external trigger input pin B for channels in SAFETY

Table 21.3 POEG interrupt sources (1 of 2)

Name	Interrupt Sources
POEG0_GROUP0	POEG group A interrupt for channels in LLPP
POEG0_GROUP1	POEG group B interrupt for channels in LLPP

Table 21.3 POEG interrupt sources (2 of 2)

Name	Interrupt Sources
POEG0_GROUP2	POEG group C interrupt for channels in LLPP
POEG0_GROUP3	POEG group D interrupt for channels in LLPP
POEG1_GROUP0	POEG group A interrupt for channels in NONSAFETY
POEG1_GROUP1	POEG group B interrupt for channels in NONSAFETY
POEG1_GROUP2	POEG group C interrupt for channels in NONSAFETY
POEG1_GROUP3	POEG group D interrupt for channels in NONSAFETY
POEG2_GROUP0	POEG group A interrupt for channels in SAFETY
POEG2_GROUP1	POEG group B interrupt for channels in SAFETY
POEG2_GROUP2	POEG group C interrupt for channels in SAFETY
POEG2_GROUP3	POEG group D interrupt for channels in SAFETY

21.2 Register Descriptions

21.2.1 POEG0Gn0 : POEG0 Group n Setting Register 0 (n = A to D)

Base address: POEG0 = 0x9001_9000

Offset address: 0x000 (POEG0GA0)
0x400 (POEG0GB0)
0x800 (POEG0GC0)
0xC00 (POEG0GD0)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCS[1:0]		NFEN	INV	—	—	—	—	—	—	—	—	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIDF	Port Input Detection Flag 0: The selected input level was not detected on the GTETRGn pin 1: The selected input level was detected on the GTETRGn pin	R/W*1
1	IOCF	GPT Output Stop Request Detection Flag 0: Stopping of GPT output was not detected 1: Stopping of GPT output was detected	R/W*1
2	OSTPF	Oscillation Stop Detection Flag 0: Stopping of oscillation was not detected 1: Stopping of oscillation was detected	R/W*1
3	SSF	Software Stop Flag 0: Software has not stopped output 1: Software has stopped output	R/W
4	PIDE	Port Input Detection Enable 0: Detection of input levels on the corresponding GTETRGn pin is disabled 1: Detection of input levels on the corresponding GTETRGn pin is enabled	R/W*2
5	IOCE	GPT Output Stop Request Enable 0: Detection of stopping of output from the GPT is disabled 1: Detection of stopping of output from the GPT is enabled	R/W*2

Bit	Symbol	Function	R/W
6	OSTPE	Enable Stopping Output on Stopping of Oscillation 0: Detection of stopping of oscillation is disabled 1: Detection of stopping of oscillation is enabled	R/W*2
15:7	—	These bits are read as 0. The write value should be 0.	R/W
16	ST	GTETRGn Input Status Flag 0: GTETRGn input after filtering was 0 1: GTETRGn input after filtering was 1	R
27:17	—	These bits are read as 0. The write value should be 0.	R/W
28	INV	GTETRGn Input Inverting 0: Input on the GTETRGn pin is not inverted 1: Input on the GTETRGn pin is inverted	R/W
29	NFEN	Noise filter Enable 0: Digital noise filter on the GTETRGn pin is disabled 1: Digital noise filter on the GTETRGn pin is enabled	R/W
31:30	NFCS[1:0]	Noise filter Clock Select 0 0: Samples the input level of GTETRGn pin three times per PCLKH clock 0 1: Samples the input level of GTETRGn pin three times per PCLKH/8 clock 1 0: Samples the input level of GTETRGn pin three times per PCLKH/32 clock 1 1: Samples the input level of GTETRGn pin three times per PCLKH/128 clock	R/W

Note 1. Only 0 to clear the flag can be written.
Note 2. Can write only once after reset.

The POEG0Gn0 register (n = A to D) controls requests for stopping output and an external trigger for the GPT based in response to the detection of various signals.

21.2.2 POEG0Gn1 : POEG0 Group n Setting Register 1 (n = A to D)

Base address: POEG0 = 0x9001_9000

Offset address: 0x004 (POEG0GA1)
0x404 (POEG0GB1)
0x804 (POEG0GC1)
0xC04 (POEG0GD1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	D9ERR1ST	D8ERR1ST	D7ERR1ST	D6ERR1ST	D5ERR1ST	D4ERR1ST	D3ERR1ST	D2ERR1ST	D1ERR1ST	D0ERR1ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9ERR0ST	D8ERR0ST	D7ERR0ST	D6ERR0ST	D5ERR0ST	D4ERR0ST	D3ERR0ST	D2ERR0ST	D1ERR0ST	D0ERR0ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0ERR0ST	DSMIF0 Error 0 Status 0: Indicates that DSMIF0 error 0 has not been generated 1: Indicates that DSMIF0 error 0 has been generated	R
1	D1ERR0ST	DSMIF1 Error 0 Status 0: Indicates that DSMIF1 error 0 has not been generated 1: Indicates that DSMIF1 error 0 has been generated	R
2	D2ERR0ST	DSMIF2 Error 0 Status 0: Indicates that DSMIF2 error 0 has not been generated 1: Indicates that DSMIF2 error 0 has been generated	R
3	D3ERR0ST	DSMIF3 Error 0 Status 0: Indicates that DSMIF3 error 0 has not been generated 1: Indicates that DSMIF3 error 0 has been generated	R

Bit	Symbol	Function	R/W
4	D4ERR0ST	DSMIF4 Error 0 Status 0: Indicates that DSMIF4 error 0 has not been generated 1: Indicates that DSMIF4 error 0 has been generated	R
5	D5ERR0ST	DSMIF5 Error 0 Status 0: Indicates that DSMIF5 error 0 has not been generated 1: Indicates that DSMIF5 error 0 has been generated	R
6	D6ERR0ST	DSMIF6 Error 0 Status 0: Indicates that DSMIF6 error 0 has not been generated 1: Indicates that DSMIF6 error 0 has been generated	R
7	D7ERR0ST	DSMIF7 Error 0 Status 0: Indicates that DSMIF7 error 0 has not been generated 1: Indicates that DSMIF7 error 0 has been generated	R
8	D8ERR0ST	DSMIF8 Error 0 Status 0: Indicates that DSMIF8 error 0 has not been generated 1: Indicates that DSMIF8 error 0 has been generated	R
9	D9ERR0ST	DSMIF9 Error 0 Status 0: Indicates that DSMIF9 error 0 has not been generated 1: Indicates that DSMIF9 error 0 has been generated	R
15:10	—	These bits are read as 0.	R
16	D0ERR1ST	DSMIF0 Error 1 Status 0: Indicates that DSMIF0 error 1 has not been generated 1: Indicates that DSMIF0 error 1 has been generated	R
17	D1ERR1ST	DSMIF1 Error 1 Status 0: Indicates that DSMIF1 error 1 has not been generated 1: Indicates that DSMIF1 error 1 has been generated	R
18	D2ERR1ST	DSMIF2 Error 1 Status 0: Indicates that DSMIF2 error 1 has not been generated 1: Indicates that DSMIF2 error 1 has been generated	R
19	D3ERR1ST	DSMIF3 Error 1 Status 0: Indicates that DSMIF3 error 1 has not been generated 1: Indicates that DSMIF3 error 1 has been generated	R
20	D4ERR1ST	DSMIF4 Error 1 Status 0: Indicates that DSMIF4 error 1 has not been generated 1: Indicates that DSMIF4 error 1 has been generated	R
21	D5ERR1ST	DSMIF5 Error 1 Status 0: Indicates that DSMIF5 error 1 has not been generated 1: Indicates that DSMIF5 error 1 has been generated	R
22	D6ERR1ST	DSMIF6 Error 1 Status 0: Indicates that DSMIF6 error 1 has not been generated 1: Indicates that DSMIF6 error 1 has been generated	R
23	D7ERR1ST	DSMIF7 Error 1 Status 0: Indicates that DSMIF7 error 1 has not been generated 1: Indicates that DSMIF7 error 1 has been generated	R
24	D8ERR1ST	DSMIF8 Error 1 Status 0: Indicates that DSMIF8 error 1 has not been generated 1: Indicates that DSMIF8 error 1 has been generated	R
25	D9ERR1ST	DSMIF9 Error 1 Status 0: Indicates that DSMIF9 error 1 has not been generated 1: Indicates that DSMIF9 error 1 has been generated	R
31:26	—	These bits are read as 0.	R

The POEG0Gn1 register (n = A to D) indicates DSMIF error status.

21.2.3 POEG0Gn2 : POEG0 Group n Setting Register 2 (n = A to D)

Base address: POEG0 = 0x9001_9000

Offset address: 0x008 (POEG0GA2)
0x408 (POEG0GB2)
0x808 (POEG0GC2)
0xC08 (POEG0GD2)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	D9ERR1E	D8ERR1E	D7ERR1E	D6ERR1E	D5ERR1E	D4ERR1E	D3ERR1E	D2ERR1E	D1ERR1E	D0ERR1E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D9ERR0E	D8ERR0E	D7ERR0E	D6ERR0E	D5ERR0E	D4ERR0E	D3ERR0E	D2ERR0E	D1ERR0E	D0ERR0E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0ERR0E	Enable Stopping Output on DSMIF0 error 0 detection 0: Disabled 1: Enabled	R/W*1
1	D1ERR0E	Enable Stopping Output on DSMIF1 error 0 detection 0: Disabled 1: Enabled	R/W*1
2	D2ERR0E	Enable Stopping Output on DSMIF2 error 0 detection 0: Disabled 1: Enabled	R/W*1
3	D3ERR0E	Enable Stopping Output on DSMIF3 error 0 detection 0: Disabled 1: Enabled	R/W*1
4	D4ERR0E	Enable Stopping Output on DSMIF4 error 0 detection 0: Disabled 1: Enabled	R/W*1
5	D5ERR0E	Enable Stopping Output on DSMIF5 error 0 detection 0: Disabled 1: Enabled	R/W*1
6	D6ERR0E	Enable Stopping Output on DSMIF6 error 0 detection 0: Disabled 1: Enabled	R/W*1
7	D7ERR0E	Enable Stopping Output on DSMIF7 error 0 detection 0: Disabled 1: Enabled	R/W*1
8	D8ERR0E	Enable Stopping Output on DSMIF8 error 0 detection 0: Disabled 1: Enabled	R/W*1
9	D9ERR0E	Enable Stopping Output on DSMIF9 error 0 detection 0: Disabled 1: Enabled	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	D0ERR1E	Enable Stopping Output on DSMIF0 error 1 detection 0: Disabled 1: Enabled	R/W*1
17	D1ERR1E	Enable Stopping Output on DSMIF1 error 1 detection 0: Disabled 1: Enabled	R/W*1

Bit	Symbol	Function	R/W
18	D2ERR1E	Enable Stopping Output on DSMIF2 error 1 detection 0: Disabled 1: Enabled	R/W*1
19	D3ERR1E	Enable Stopping Output on DSMIF3 error 1 detection 0: Disabled 1: Enabled	R/W*1
20	D4ERR1E	Enable Stopping Output on DSMIF4 error 1 detection 0: Disabled 1: Enabled	R/W*1
21	D5ERR1E	Enable Stopping Output on DSMIF5 error 1 detection 0: Disabled 1: Enabled	R/W*1
22	D6ERR1E	Enable Stopping Output on DSMIF6 error 1 detection 0: Disabled 1: Enabled	R/W*1
23	D7ERR1E	Enable Stopping Output on DSMIF7 error 1 detection 0: Disabled 1: Enabled	R/W*1
24	D8ERR1E	Enable Stopping Output on DSMIF8 error 1 detection 0: Disabled 1: Enabled	R/W*1
25	D9ERR1E	Enable Stopping Output on DSMIF9 error 1 detection 0: Disabled 1: Enabled	R/W*1
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can write only once after reset.

The POEG0Gn2 register (n = A to D) controls to enable stopping output on DSMIF error detection.

21.2.4 POEGmGn0 : POEGm Group n Setting Register 0 (m = 1, 2) (n = A to D)

Base address: POEG1 = 0x8008_7000 (m = 1)
POEG2 = 0x8100_A000 (m = 2)

Offset address: 0x000 (POEGmGA0)
0x400 (POEGmGB0)
0x800 (POEGmGC0)
0xC00 (POEGmGD0)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCS[1:0]		NFEN	INV	—	—	—	—	—	—	—	—	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIDF	Port Input Detection Flag 0: The selected input level was not detected on the GTETRGn pin 1: The selected input level was detected on the GTETRGn pin	R/W*1
1	IOCF	GPT Output Stop Request Detection Flag 0: Stopping of GPT output was not detected 1: Stopping of GPT output was detected	R/W*1

Bit	Symbol	Function	R/W
2	OSTPF	Oscillation Stop Detection Flag 0: Stopping of oscillation was not detected 1: Stopping of oscillation was detected	R/W*1
3	SSF	Software Stop Flag 0: Software has not stopped output 1: Software has stopped output	R/W
4	PIDE	Port Input Detection Enable 0: Detection of input levels on the corresponding GTETRGn pin is disabled 1: Detection of input levels on the corresponding GTETRGn pin is enabled	R/W*2
5	IOCE	GPT Output Stop Request Enable 0: Detection of stopping of output from the GPT is disabled 1: Detection of stopping of output from the GPT is enabled	R/W*2
6	OSTPE	Enable Stopping Output on Stopping of Oscillation 0: Detection of stopping of oscillation is disabled 1: Detection of stopping of oscillation is enabled	R/W*2
15:7	—	These bits are read as 0. The write value should be 0.	R/W
16	ST	GTETRGn Input Status Flag 0: GTETRGn input after filtering was 0 1: GTETRGn input after filtering was 1	R
27:17	—	These bits are read as 0. The write value should be 0.	R/W
28	INV	GTETRGn Input Inverting 0: Input on the GTETRGn pin is not inverted 1: Input on the GTETRGn pin is inverted	R/W
29	NFEN	Noise filter Enable 0: Digital noise filter on the GTETRGn pin is disabled 1: Digital noise filter on the GTETRGn pin is enabled	R/W
31:30	NFCS[1:0]	Noise filter Clock Select 0 0: Samples the input level of GTETRGn pin three times per PCLKL clock 0 1: Samples the input level of GTETRGn pin three times per PCLKL/8 clock 1 0: Samples the input level of GTETRGn pin three times per PCLKL/32 clock 1 1: Samples the input level of GTETRGn pin three times per PCLKL/128 clock	R/W

Note 1. Only 0 to clear the flag can be written.
Note 2. Can write only once after reset.

The POEGmGn0 register (m = 1, 2), (n = A to D) controls requests for stopping output and an external trigger for the GPT based in response to the detection of various signals.

21.3 Operation

21.3.1 Output-Disable Control Operation

If any of the following conditions is satisfied, the GTIOCxA, GTIOCxB, and the three-phase PWM output for motor control pins can be set to output-disable:

- Input level or edge detection of the GTETRGA to GTETRGD pins
  When POEGmGn0.PIDE = 1 (m = 0 to 2, n = A to D), the POEGmGn0.PIDF flag is set to 1.
- Output-disable request from the GPT
  When POEGmGn0.IOCE = 1 (m = 0 to 2, n = A to D), the POEGmGn0.IOCF flag is set to 1 if the output-disable request is enabled in GTINTAD. The GTINTAD.GRPDTE, GTINTAD.GRPABH, and GTINTAD.GRPABL settings apply to the group selected in GPT register GTINTAD.GRP[1:0] and OPSCR.GRP[1:0].
- Oscillation stop detection for the Clock Generation Circuit
  When POEGmGn0.OSTPE = 1 (m = 0 to 2, n = A to D), the POEGmGn0.OSTPF flag is set to 1.
- SSF bit setting
  When POEGmGn0.SSF (m = 0 to 2, n = A to D) is set to 1.

The state of output-disable is controlled in the GPT module. The output-disable of the GTIOCxA and GTIOCxB pins are set in the GTINTAD.GRP[1:0] bits, GTIOR.OADF[1:0] bits, and GTIOR.OBDF[1:0] bits in GPTx. The output-disable of the PWM output for motor control pins are set in the OPSCR.GRP[1:0] bits and OPSCR.GODF bit in GPT_OPS.

21.3.1.1 Pin Input Level Detection Operation

If the input conditions set in POEGmGn0.PIDE, POEGmGn0.NFCS[1:0], POEGmGn0.NFEN, and POEGmGn0.INV (m = 0 to 2, n = A to D) occur on the GTETRGA to GTETRGD pins, the GPT output pins are output-disabled.

(1) Digital Noise Filter

Figure 21.5 shows high-level detection by the digital filter. When a high level associated with the POEGmGn0.INV (m = 0 to 2, n = A to D) polarity setting is detected three times consecutively with the sampling clock selected in POEGmGn0.NFCS[1:0] and POEGmGn0.NFEN, the detected level is recognized as high, and the GPT output pins are output-disabled. If even one low level is detected during this interval, the detected level is not recognized as high. In addition, in an interval where the sampling clock is not being output, changes of the levels on the GTETRGA to GTETRGD pins are ignored.

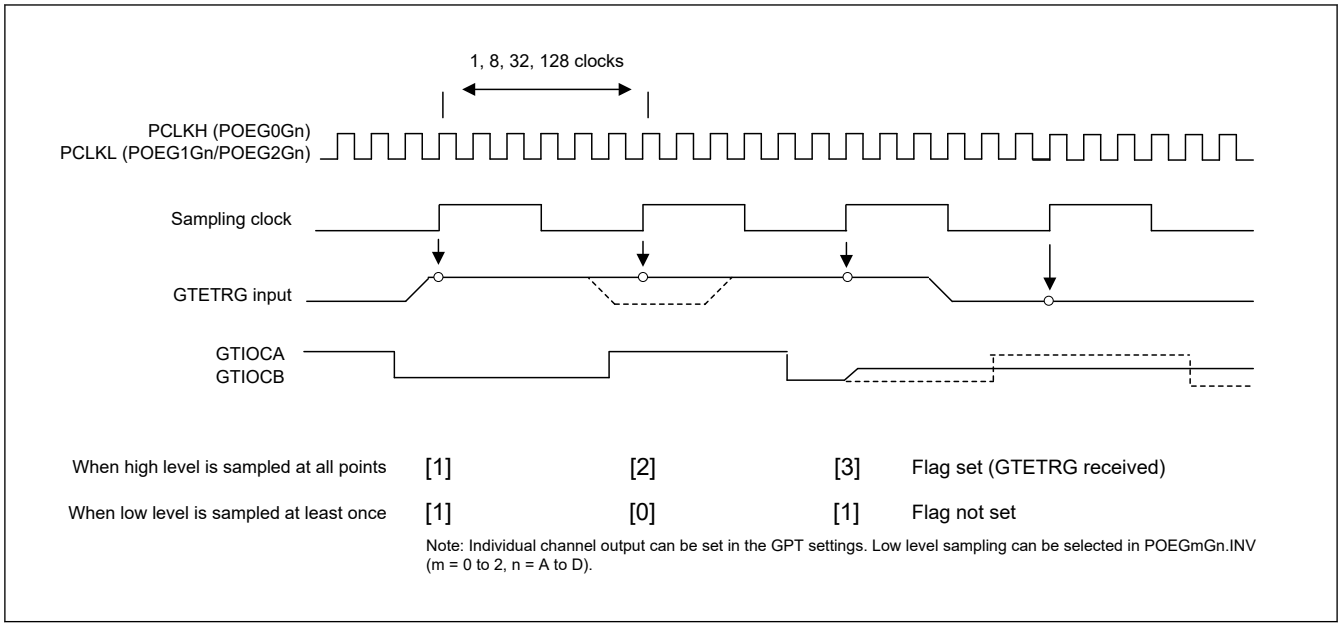


Figure 21.5 Example of digital noise filter operation

21.3.2 Output-Disable Requests from the GPT

For details on this operation, see the description of output-disable control for GTIOC pin outputs in section 20, General PWM Timer (GPT).

21.3.3 Output-Disable Control Using Detection of Stopped Oscillation

When the oscillation stop detection function in the Clock Generation Circuit detects stopped oscillation while POEGmGn0.OSTPE = 1 (m = 0 to 2, n = A to D), the GPT output pins are output-disabled for each group.

21.3.4 Output-Disable Control Using DSMIF Error Detection

When the DSMIF detects error while POEG0Gn.DmERR0E =1 or POEG0Gn.DmERR1E = 1 (m = 0, 1), the GPT output pins in the LLPP region can be output-disabled. Target DSMIF error can be selected in POEG0Gn2 register (n = A to D).

21.3.5 Output-Disable Control Using Registers

The GPT output pins can be directly controlled by writing to the software stop flag, POEGmGn0.SSF.

21.3.6 Release from Output-Disable

To release the GPT output pins placed in the output-disable state, either return them to their initial state with a reset, clear all of the following (n = A to D), or DSMIF error flag in DSMIFn.DSCESR:

- POEGmGn0.PIDF flag
- POEGmGn0.IOCF flag
- POEGmGn0.OSTPF flag
- POEGmGn0.SSF flag

Writing 0 to the POEGmGn0.PIDF (n = A to D) flag is ignored (the flag is not cleared) if the external input pins, GTETRGA to GTETRGD, are not disabled and the POEGmGn0.ST bit is not set to 0.

Writing 0 to the POEGmGn0.IOCF (n = A to D) flag is valid (the flag is cleared) only if all of the GTST.DTEF, GTST.OABHF, and GTST.OABLF flags in the GPT are set to 0.

Figure 21.6 shows the release timing for output-disable. The output-disable is released at the beginning of the next count cycle of the GPT after the flag is cleared.

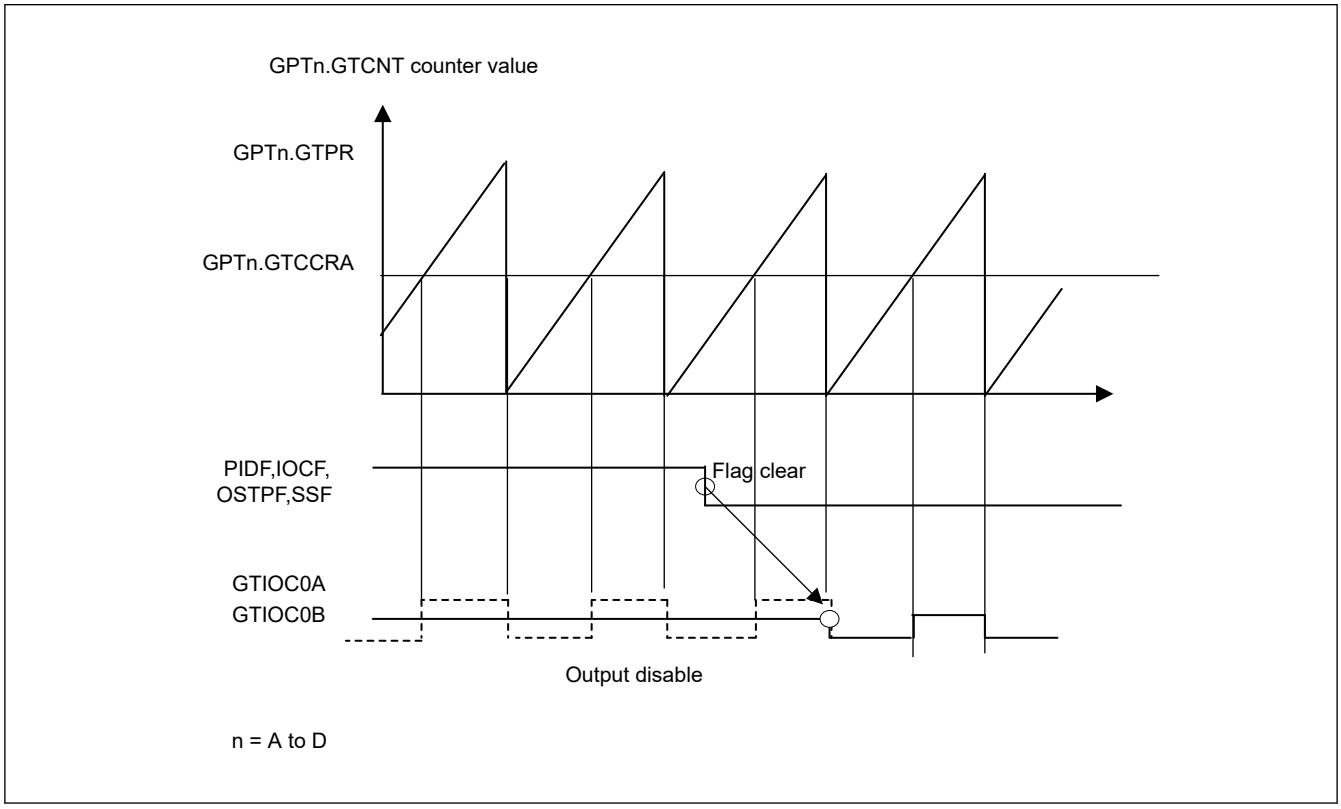


Figure 21.6 Output-disable release timing for GPT pin outputs

22. Trigonometric Function Unit (TFU)

22.1 Overview

A trigonometric Function Unit (TFU) handles the high-speed calculation for sinf, cosf, atan2f, and hypotf functions.

The PCLKH is used as the operating clock for the TFU.

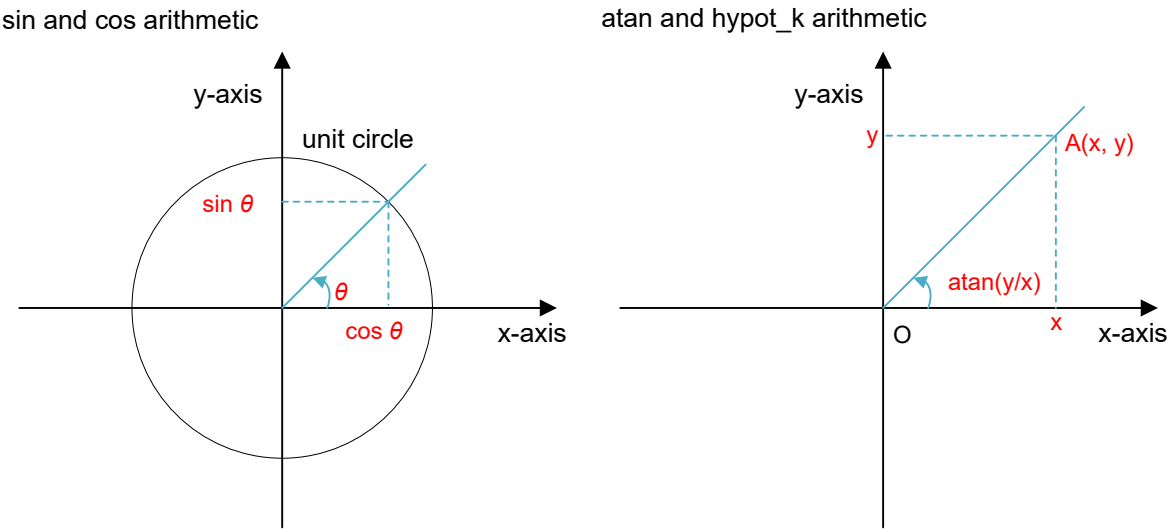
Table 22.1 lists the specifications of the TFU.

Table 22.1 TFU specifications

Item	Description			
Number of unit	2 units			
Arithmetic Processing	Calculation of sine, cosine, arctangent, and hypot_k ($\sqrt{x^2+y^2}/k$) • A sine and cosine can be simultaneously calculated. • An arctangent and hypot_k can be simultaneously calculated.			
Range and Unit of Values	**Floating-point**	**I/O**		**Range**	**Unit**
	Calculating sine	Input	Angle θ	$-\text{float\_max} \le \theta \le \text{float\_max}$ *1	radian
		Output	$\sin\theta$	$-1.0 \le \sin\theta \le 1.0$	—
	Calculating cosine	Input	Angle θ	$-\text{float\_max} \le \theta \le \text{float\_max}$ *1	radian
		Output	$\cos\theta$	$-1.0 \le \cos\theta \le 1.0$	—
	Calculating arctangent	Input	x and y coordinates	$-\text{float\_max} \le x \le \text{float\_max}$ *1 $-\text{float\_max} \le y \le \text{float\_max}$ *1	—
		Output	$\text{atan}(y/x)$	$-\pi \le \text{atan}(y/x) \le \pi$	radian
	Calculating hypot_k	Input	x and y coordinates	$-\text{float\_max} \le x \le \text{float\_max}$ *1 $-\text{float\_max} \le y \le \text{float\_max}$ *1	—
		Output	$\sqrt{x^2+y^2}/k$	$0 \le \sqrt{x^2+y^2}/k \le \infty$	—
	Fixed-point	**I/O**		**Range**	**Unit**
	Calculating sine	Input	Angle θ	$-1.0 \le \theta < 1.0$	turn
				$-4.0 \le \theta < 4.0$	radian
		Output	$\sin\theta$	$-1.0 \le \sin\theta \le 1.0$	—
	Calculating cosine	Input	Angle θ	$-1.0 \le \theta < 1.0$	turn
				$-4.0 \le \theta < 4.0$	radian
		Output	$\cos\theta$	$-1.0 \le \cos\theta \le 1.0$	—
	Calculating arctangent	Input	x and y coordinates	$-1.0 \le x < 1.0$ $-1.0 \le y < 1.0$	—
		Output	$\text{atan}(y/x)$	$-0.5 \le \text{atan}(y/x) \le 0.5$	turn
				$-\pi \le \text{atan}(y/x) \le \pi$	radian
	Calculating hypot_k	Input	x and y coordinates	$-1.0 \le x < 1.0$ $-1.0 \le y < 1.0$	—
		Output	$\sqrt{x^2+y^2}/k$	$0 \le \sqrt{x^2+y^2}/k \le \sqrt{2}/k$	—
Data Type for Processing	Single-precision floating-point, Fixed-point (two's complement)			
Number of cycles for calculation	Sine: 5 Cosine: 5 Arctangent: 14 hypot_k: 14			

Note: k is a constant. See section 22.4.1. Arithmetic Processing.

Note 1. float_max is the maximum value that can be expressed as single-precision floating-point: $(2 - 2^{-23}) \times 2^{127}$.



Note: sin and cos arithmetic:
Operation to find $\sin\theta$ and $\cos\theta$, which are the triangular ratios of the input angle values θ.
atan arithmetic:
Operation to find the angle value between the vector OA (O is the origin and A is the point indicated by the input coordinates x and y) and the positive part of the x-axis.
hypot_k arithmetic:
Operation to find a value proportional to the distance from the point A indicated by the input coordinates x and y to the origin O.

Note: By multiplying the TFU output result hypot_k by the constant k, the distance between OA ($\sqrt{x^2+y^2}$) can be obtained. for details about k, see section 22.4.1. Arithmetic Processing.

Figure 22.1 Explanation for operations

22.1.1 Precautions on Use of the Trigonometric Function Unit

This section describes precautions on use of the trigonometric function unit.

22.1.1.1 General Precautions

If another operation is started after the operation is started and before the result is read, the result of the preceding operation will be discarded.

22.2 Register Map

Table 22.2 TFU register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x9001_0004 + 0x10_0000 × m	FXSCIOC	Fixed-point Sine Cosine Input/Output Setting Register	—
0x9001_0005 + 0x10_0000 × m	FXATIOC	Fixed-point Arctangent hypot_k Input/Output Setting Register	—
0x9001_0008 + 0x10_0000 × m	TRGSTS	Trigonometric Status Register	—
0x9001_0010 + 0x10_0000 × m	FPSCDT0	Floating-point Sine Cosine Data Register 0	—
0x9001_0014 + 0x10_0000 × m	FPSCDT1	Floating-point Sine Cosine Data Register 1	—
0x9001_0018 + 0x10_0000 × m	FPATDT0	Floating-point Arctangent hypot_k Data Register 0	—
0x9001_001C + 0x10_0000 × m	FPATDT1	Floating-point Arctangent hypot_k Data Register 1	—
0x9001_0020 + 0x10_0000 × m	FXSCDT0	Fixed-point Sine Cosine Data Register 0	—
0x9001_0024 + 0x10_0000 × m	FXSCDT1	Fixed-point Sine Cosine Data Register 1	—
0x9001_0028 + 0x10_0000 × m	FXATDT0	Fixed-point Arctangent hypot_k Data Register 0	—

Table 22.2 TFU register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x9001_002C + 0x10_0000 × m	FXATDT1	Fixed-point Arctangent hypot_k Data Register 1	—
0x9001_0030 + 0x10_0000 × m	DTSR0	Data Save Restore Register 0	—
0x9001_0034 + 0x10_0000 × m	DTSR1	Data Save Restore Register 1	—

Note: m = 0, 1

Table 22.3 TFU related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	—	MSTPCRC.MSTPCRC05	SLVACCCTL7.LLPP_SL*1
1	—	MSTPCRC.MSTPCRC24	SLVACCCTL7.LLPP_SL*1

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

22.3 Register Descriptions

22.3.1 FXSCIOC : Fixed-point Sine Cosine Input/Output Setting Register

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	OF[1:0]		—	—	—	IUF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IUF	Input unit and format setting Specify the unit and format for input value of fixed-point sincos operation 0: unit, format = turn, Q1.31 1: unit, format = radian, Q3.29	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
5:4	OF[1:0]	Output format setting 0 0: Q1.31 0 1: Q2.30 1 0: Q3.29 1 1: Setting prohibited	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

IUF bit (Input unit and format setting)

The IUF bit specifies the unit and format for input value of fixed-point sincos operation.

OF[1:0] bits (Output format setting)

The OF[1:0] bits specify the format for output value of fixed-point sincos operation.

When Q1.31 is selected, +1.0 cannot be expressed accurately. If $\sin\theta$ or $\cos\theta$ is +1.0 ordinally when Q1.31, $+1.0 - 2^{-31}$ will be read out from the FXSCDT0 and FXSCDT1 registers.

This register should be set before the calculation start. If this register is changed during the calculation, the operation is not guaranteed.

22.3.2 FXATIOC : Fixed-point Arctangent hypot_k Input/Output Setting Register

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OUF	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	OUF	Output unit and format setting Specify the unit and format for output value of fixed-point arctangent operation 0: unit, format = turn, Q1.31 1: unit, format = radian, Q3.29	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

OUF bit (Output unit and format setting)

The OUF bit specifies the unit and format for output value when arctangent calculation is executed by fixed-point atanhypot_k operation. The output format of hypot_k calculation executed with arctangent calculation in parallel is Q3.29 regardless of this bit setting.

This register should be set before the calculation start. If this register is changed during the calculation, the operation is not guaranteed.

22.3.3 TRGSTS : Trigonometric Status Register

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ERRF	BSYF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BSYF	Calculation in progress flag 0: No calculating 1: Calculating	R/W*1
1	ERRF	Input error flag 0: No input error occurred 1: Input error occurred	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to this bit is invalid.

BSYF bit (Calculation in progress flag)

The BSYF bit specifies the calculation in progress flag. Writing to this bit is ignored.

[Setting condition]

- When calculation started

[Clearing condition]

- When calculation completed

ERRF bit (Input error flag)

The ERRF bit specifies the input error flag.

Do not write to this bit except error flag restoration procedure shown in section 22.4.5. Relationship Between Input and Output Values for Atan Operation.
[Setting condition]

- When input error occurred or 1 wrote to this bit. For details on input error, see Table 22.10.

[Clearing condition]

- When the next calculation started or 0 wrote to this bit

22.3.4 FPSCDT0 : Floating-point Sine Cosine Data Register 0

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x10

Bit position:	31																												0
Bit field:	FPSCDT0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FPSCDT0[31:0]	Floating-point Sine Cosine Data Register 0	R/W

For the floating-point sincos operation, the FPSCDT0 register is dedicated for the output value $\cos\theta$ of the trigonometric function unit. See Table 22.4 for detail. In terms of usage of this register, see section 22.4.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the floating-point sincos operation, do not access this register. If other data register 0 such as FPATDT0, FXSCDT0, and FXATDT0 registers is written, this register value will be uncertainty.

When the DTSR0 register is written, this register value will be the same as DTSR0 register writing value.

22.3.5 FPSCDT1 : Floating-point Sine Cosine Data Register 1

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x14

Bit position:	31																												0
Bit field:	FPSCDT1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FPSCDT1[31:0]	Floating-point Sine Cosine Data Register 1	R/W

For the floating-point sincos operation, the FPSCDT1 register is shared for the input angle value θ and the output value $\sin\theta$ of the trigonometric function unit. See Table 22.4 for detail. Writing to this register starts the floating-point sincos operation. In terms of usage of this register, see section 22.4.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the floating-point sincos operation, do not access this register. If other data register 1 such as FPATDT1, FXSCDT1, and FXATDT1 registers is written, this register value will be uncertainty.

When the DTSR1 register is written, this register value will be the same as DTSR1 register writing value.

Table 22.4 Input/Output value of FPSCDT0 and FPSCDT1

Register	Input value	Output value
FPSCDT0	—	$\cos\theta$
FPSCDT1	Angle θ	$\sin\theta$

22.3.6 FPATDT0 : Floating-point Arctangent hypot_k Data Register 0

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x18

Bit position:	31																												0
Bit field:	FPATDT0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FPATDT0[31:0]	Floating-point Arctangent hypot_k Data Register 0	R/W

For the floating-point atanhypot_k operation, the FPATDT0 register is shared for the input coordinate value x and the output value $\sqrt{x^2+y^2}/k$ of the trigonometric function unit. See Table 22.5 for detail. In terms of usage of this register, see section 22.4.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the floating-point atanhypot_k operation, do not access this register. If other data register 0 such as FPSCDT0, FXSCDT0, and FXATDT0 registers is written, this register value will be uncertainty.

When the DTSR0 register is written, this register value will be the same as DTSR0 register writing value.

22.3.7 FPATDT1 : Floating-point Arctangent hypot_k Data Register 1

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x1C

Bit position:	31																												0
Bit field:	FPATDT1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FPATDT1[31:0]	Floating-point Arctangent hypot_k Data Register 1	R/W

For the floating-point atanhypot_k operation, the FPATDT1 register is shared for the input coordinate value y and the output value atan(y/x) of the trigonometric function unit. See Table 22.5 for detail. Writing to this register starts the floating-point atanhypot_k operation. In terms of usage of this register, see section 22.4.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the floating-point atanhypot_k operation, do not access this register. If other data register 1 such as FPSCDT1, FXSCDT1, and FXATDT1 registers is written, this register value will be uncertainty.

When the DTSR1 register is written, this register value will be the same as DTSR1 register writing value.

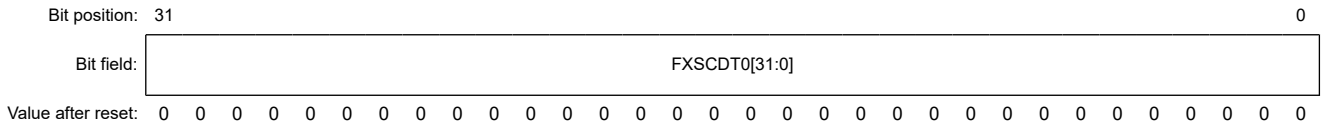
Table 22.5 Input/Output value of FPATDT0 and FPATDT1

Register	Input value	Output value
FPATDT0	Coordinates x	hypot_k
FPATDT1	Coordinates y	atan(y/x)

22.3.8 FXSCDT0 : Fixed-point Sine Cosine Data Register

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x20



Bit	Symbol	Function	R/W
31:0	FXSCDT0[31:0]	Fixed-point Sine Cosine Data Register 0	R/W

For the fixed-point sincos operation, the FXSCDT0 register is dedicated for the output value $\cos\theta$ of the trigonometric function unit. See Table 22.6 for detail. In terms of usage of this register, see section 22.4.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

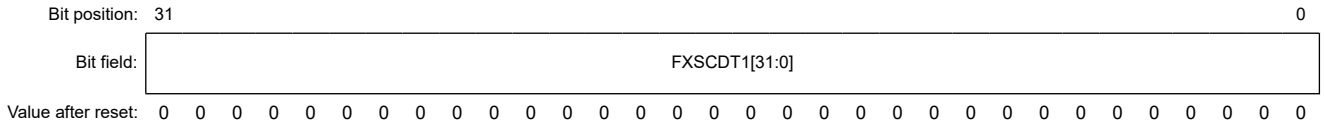
For calculations except the fixed-point sincos operation, do not access this register. If other data register 0 such as FPSCDT0, FPATDT0, and FXATDT0 registers is written, this register value will be uncertainty.

When the DTSR0 register is written, this register value will be the same as DTSR0 register writing value.

22.3.9 FXSCDT1 : Fixed-point Sine Cosine Data Register 1

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x24



Bit	Symbol	Function	R/W
31:0	FXSCDT1[31:0]	Fixed-point Sine Cosine Data Register 1	R/W

For the fixed-point sincos operation, the FXSCDT1 register is shared for the input angle value θ and the output value $\sin\theta$ of the trigonometric function unit. See Table 22.6 for detail. Writing to this register starts the fixed-point sincos operation. In terms of usage of this register, see section 22.4.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the fixed-point sincos operation, do not access this register. If other data register 1 such as FPSCDT1, FPATDT1, and FXATDT1 registers is written, this register value will be uncertainty.

When the DTSR1 register is written, this register value will be the same as DTSR1 register writing value.

Table 22.6 Input/Output value of FXSCDT0 and FXSCDT1

Register	Input value	Input unit and format	Output value	Output format
FXSCDT0	Unused	—	$\cos\theta$	Depends on FXSCIOC.OF[1:0] setting: Q1.31, Q2.30, or Q3.29
FXSCDT1	Angle θ	When FXSCIOC.IUF = 0: [turn], Q1.31 When FXSCIOC.IUF = 1: [radian], Q3.29	$\sin\theta$	Same as $\cos\theta$

When Q1.31 is selected, +1.0 cannot be expressed accurately. If $\sin\theta$ or $\cos\theta$ is +1.0 ordinally when Q1.31, +1.0 - 2^{-31} will be read out from the FXSCDT0 and FXSCDT1 registers.

22.3.10 FXATDT0 : Fixed-point Arctangent hypot_k Data Register 0

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x28

Bit	Symbol	Function	R/W
31:0	FXATDT0[31:0]	Fixed-point Arctangent hypot_k Data Register 0	R/W

For the fixed-point atanhypot_k operation, the FXATDT0 register is shared for the input coordinate value x and the output value $\sqrt{x^2+y^2}/k$ of the trigonometric function unit. See Table 22.7 for detail. In terms of usage of this register, see section 22.4.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

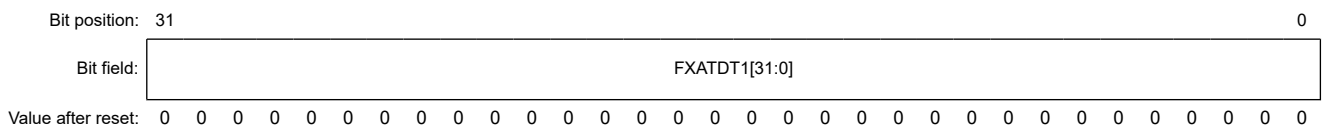
For calculations except the fixed-point atanhypot_k operation, do not access this register. If other data register 0 such as FPSCDT0, FPATDT0, and FXSCDT0 registers is written, this register value will be uncertainty.

When the DTSR0 register is written, this register value will be the same as DTSR0 register writing value.

22.3.11 FXATDT1 : Fixed-point Arctangent hypot_k Data Register 1

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x2C



Bit	Symbol	Function	R/W
31:0	FXATDT1[31:0]	Fixed-point Arctangent hypot_k Data Register 1	R/W

For the fixed-point atanhypot_k operation, the FXATDT1 register is shared for the input coordinate value y and the output value atan(y/x) of the trigonometric function unit. See Table 22.7 for detail. Writing to this register starts the fixed-point atanhypot_k operation. In terms of usage of this register, see section 22.4.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the fixed-point atanhypot_k operation, do not access this register. If other data register 1 such as FPSCDT1, FPATDT1, and FXSCDT1 registers is written, this register value will be uncertainty.

When the DTSR1 register is written, this register value will be the same as DTSR1 register writing value.

Table 22.7 Input/Output value of FXATDT0 and FXATDT1

Register	Input value	Input format	Output value	Output unit and format
FXATDT0	Coordinates x	Q1.31	hypot_k	[—], Q3.29
FXATDT1	Coordinates y	Q1.31	atan(y/x)	When FXATIOC.OUF = 0: [turn], Q1.31 When FXATIOC.OUF = 1: [radian], Q3.29

22.3.12 DTSR0 : Data Save Restore Register 0

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x30

Bit position:	31																												0
Bit field:	DTSR0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DTSR0[31:0]	Data Save Restore Register 0	R/W

The DTSR0 register is used to save and restore data in the data register 0 such as FPSCDT0, FPATDT0, FXSCDT0, and FXATDT0.

When read this register, the last written value into data register 0 or the result of the last arithmetic operation for data register 0 will be read.

When write to this register, the value for all data register 0 will be set to the written data of this register.

Writing to this register is prohibited during the operation.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

In terms of usage of this register, see section 22.4.8. Usage of Interrupt.

22.3.13 DTSR1 : Data Save Restore Register 1

Base address: TFUm = 0x9001_0000 + 0x10_0000 × m (m = 0, 1)

Offset address: 0x34

Bit position:	31																												0
Bit field:	DTSR1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DTSR1[31:0]	Data Save Restore Register 1	R/W

The DTSR1 register is used to save and restore data in the data register 1 such as FPSCDT1, FPATDT1, FXSCDT1, and FXATDT1.

When read this register, the last written value into data register 1 or the result of the last arithmetic operation for data register 1 will be read.

When write to this register, the value for all data register 1 will be set to the written data of this register.

Writing to this register is prohibited during the operation.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

In terms of usage of this register, see section 22.4.8. Usage of Interrupt.

22.4 Operation

22.4.1 Arithmetic Processing

The trigonometric function unit has two arithmetic operations, the sincos operation and the atanhypot_k operation. See Table 22.8 for detail.

Table 22.8 Arithmetic processing

Operation	Input value	Output value
sincos	Angle value θ	$\cos\theta$ and $\sin\theta$
atanhypot_k	Coordinates x and y	atan(y/x) and hypot_k

The value of scaling factor k is:

$$k = \prod_{i=0}^{\infty} \frac{1}{\sqrt{1 + 2^{-2i}}} \approx 0.607252935008881256169 4$$

22.4.2 Input and Output Value Formats

The input/output value of the TFU supports single-precision floating-point and fixed-point. For the single-precision floating-point in detail, see Table 22.4 and Table 22.9. For the fixed-point in detail, see Table 22.5. As a side of note, fixed-point input/output data is in 32-bit format, but it is performed with less than 32 bits of precision.

- Floating-point
  Support single-precision floating-point specified by the IEEE754 standard.
  - Single-precision floating-point

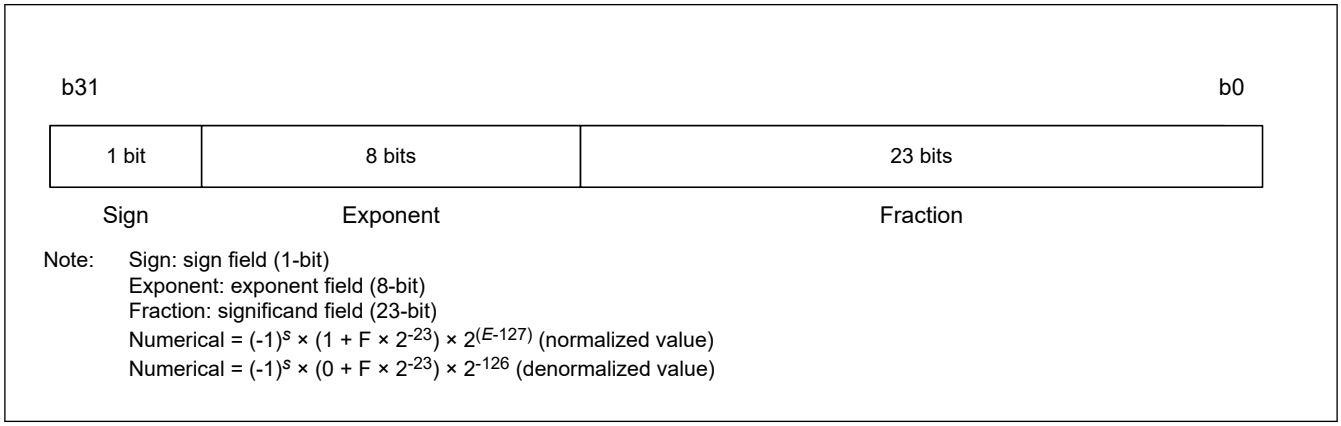


Figure 22.2 Input and output value formats

Table 22.9 Support single-precision floating-point (1 of 2)

S	E	F	Numerical
Any value	0 < E < 255	Any value	$(-1)^s \times 1.F \times 2^{(E-127)}$ (normalized value — Normal Numbers)
Any value	E = 0	F > 0	$(-1)^s \times 0.F \times 2^{-126}$ (denormalized value — Subnormal Numbers)

Table 22.9 Support single-precision floating-point (2 of 2)

S	E	F	Numerical
S = 0	E = 0	F = 0	$(-1)^0 \times 0.0$ (positive zero — +0)
S = 1	E = 0	F = 0	$(-1)^{-1} \times 0.0$ (negative zero — -0)
S = 0	E = 255	F = 0	(positive infinity — +∞)
S = 1	E = 255	F = 0	(negative infinity — -∞)
Any value	E = 255	$2^{22} > F > 0$	(non-number — SNaN: Signaling Not a Number)
Any value	E = 255	$F \geq 2^{22}$	(non-number — QNaN: Quiet Not a Number)

- Fixed-point
  Format of two's complement representation shown in Figure 22.3 is used.

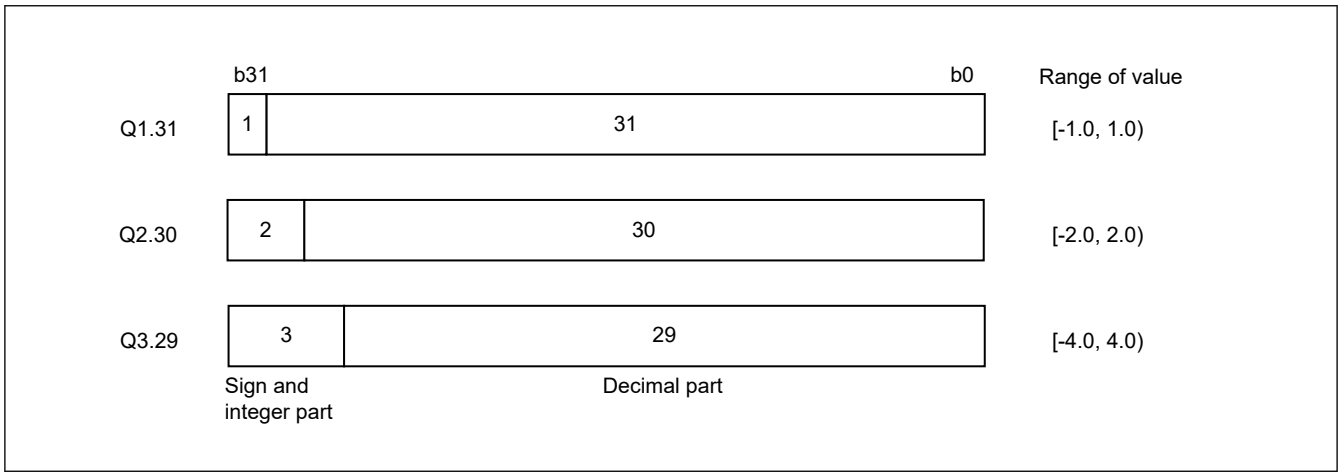


Figure 22.3 Fixed-point

22.4.3 Angle Unit

The units of angle used in the TFU are radian and turn. The radian is one rotation expressed as 0 to 2π. The turn is one rotation expressed as 0 to 1.

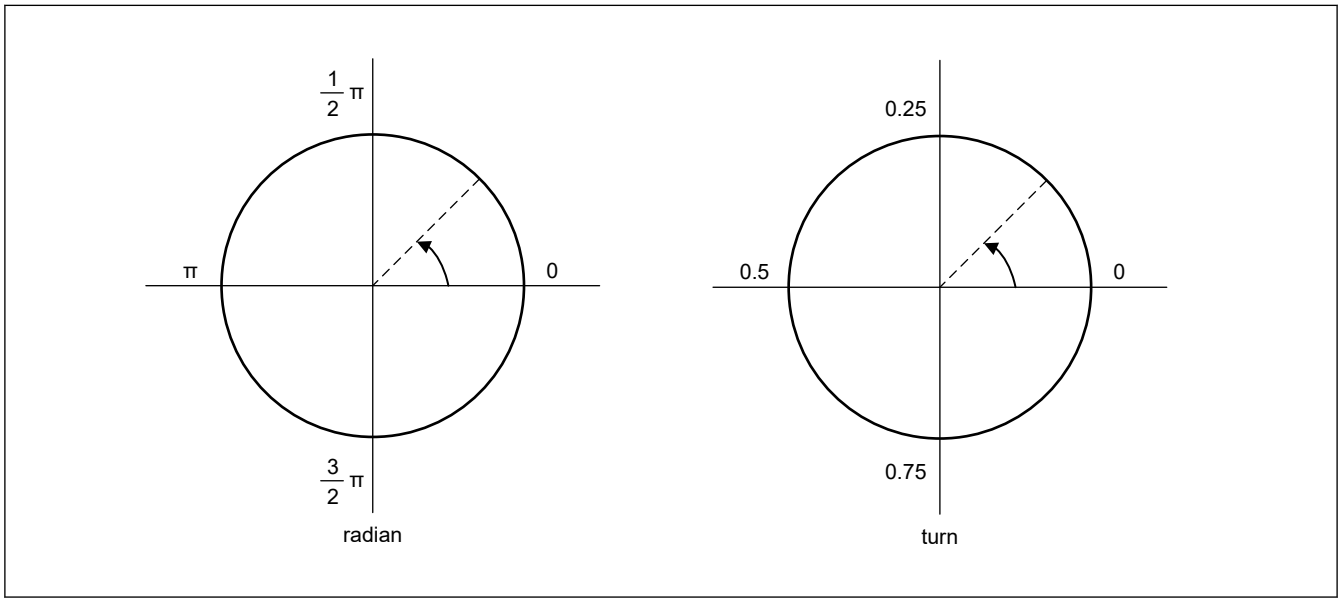


Figure 22.4 Angle unit

22.4.4 Relationship Between Input and Output Values for Sincos Operation

When the input values in the floating-point sincos operation are ±0, ±∞, SNaN (Signaling Not a Number), and QNaN (Quiet Not a Number), a fixed value is output as shown in Table 22.10.

Table 22.10 Relationship between special input value and its output value (for floating-point sincos operation)

Input (θ)	Output ($\cos\theta$)	Output ($\sin\theta$)
-∞	QNaN	QNaN
-0	+1	-0
+0	+1	+0
+∞	QNaN	QNaN
SNaN/QNaN	QNaN	QNaN

Note: The output value of QNaN is 0xFFC0_0000.

22.4.5 Relationship Between Input and Output Values for Atan Operation

When either the input values x or y in the floating-point atan operation are ±0, ±∞, SNaN (Signaling Not a Number), and QNaN (Quiet Not a Number), a fixed value is output as shown in Table 22.11.

If both input values are ±0, it is determined as an input error.

Table 22.11 Relationship between special input value and its output value (for floating-point atan operation)

		x						
		-∞	Negative value	-0	+0	Positive value	+∞	SNaN/QNaN
y	-∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	Negative value	QNaN	—	-π/2	-π/2	—	QNaN	QNaN
	-0	QNaN	-π	QNaN*1	QNaN*1	-0	QNaN	QNaN
	+0	QNaN	+π	QNaN*1	QNaN*1	+0	QNaN	QNaN
	Positive value	QNaN	—	+π/2	+π/2	—	QNaN	QNaN
	+∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	SNaN/QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN

Note 1. An input error occurs for the special input value, and the input error flag (TRGSTS.ERRF) is set.

For fixed-point atan operation, when both x and y input values are 0, it is determined as an input error. When an input error happens, output value of atan operation is 0.

22.4.6 Relationship Between Input and Output Values for hypot_k Operation

When either the input values x or y in the floating-point hypot_k operation are ±0, ±∞, SNaN (Signaling Not a Number), and QNaN (Quiet Not a Number), or when both x and y are ± 0, a fixed value is output as shown in Table 22.12.

If both input values are ±0, it is determined as an input error.

Table 22.12 Relationship between special input value and its output value (for floating-point hypot_k operation)

		x						
		-∞	**Negative value**	**-0**	**+0**	**Positive value**	**+∞**	**SNaN/QNaN**
y	**-∞**	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	Negative value	QNaN	—	—	—	—	QNaN	QNaN
	-0	QNaN	—	+0[*1]	+0[*1]	—	QNaN	QNaN
	+0	QNaN	—	+0[*1]	+0[*1]	—	QNaN	QNaN
	Positive value	QNaN	—	—	—	—	QNaN	QNaN
	+∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	SNaN/QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN

Note 1. An input error occurs for the special input value, and the input error flag (TRGSTS.ERRF) is set.

For fixed-point hypot_k operation, when both x and y input values are 0, it is determined as an input error. Because atan operation becomes error, although it is no problem for hypot_k operation. When an input error happens, output value of hypot_k is 0.

22.4.7 Procedure for Trigonometric Function Operation

Figure 22.5 shows the procedure for sincos operation. There are two procedures. The advantages and disadvantages of each are shown in Table 22.13. For floating-point sincos operation, replace SCDTn in the figure with FPSCDTn. And for fixed-point sincos operation, replace with FXSCDTn. (n = 0 or 1)

Figure 22.6 shows the procedure for atanhypot_k operation. There are two procedures. The advantages and disadvantages of each are shown in Table 22.13. For floating-point atanhypot_k operation, replace ATDTn in the figure with FPATDTn. And for fixed-point atanhypot_k operation, replace with FXATDTn. (n = 0 or 1)

Table 22.13 Advantages and disadvantages of sincos and atanhypot_k operations

Method	Advantages	Disadvantages
Procedure 1	Not occupy the bus.	Operation end determination is required by checking TRGSTS.BSYF.
Procedure 2	Operation end determination is not required by checking TRGSTS.BSYF. (reduced number of execution cycles)	Occupy the bus while waiting for the number of execution cycles when the result of the operation is read.

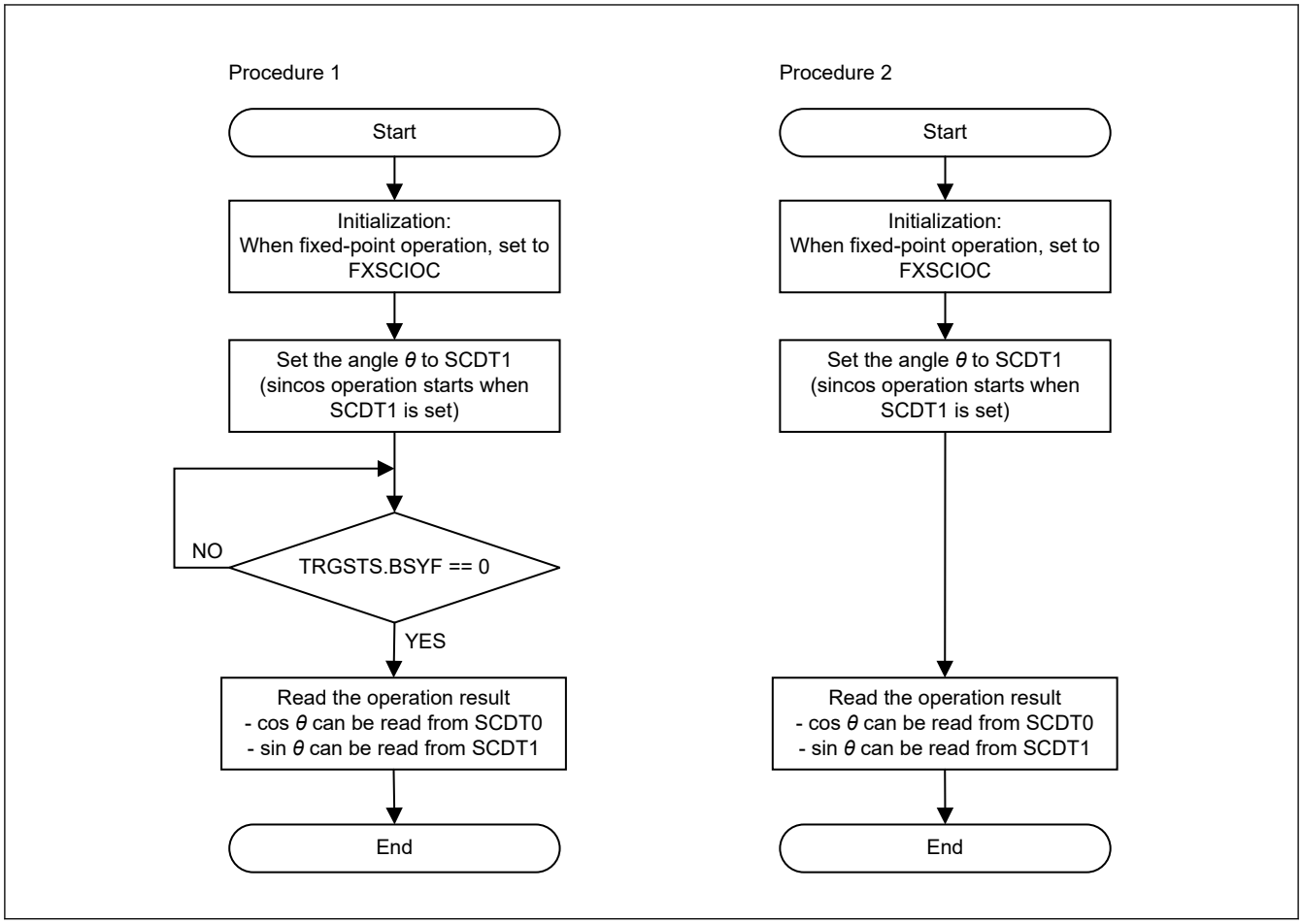


Figure 22.5 Procedure for using TFU (sincos operation)

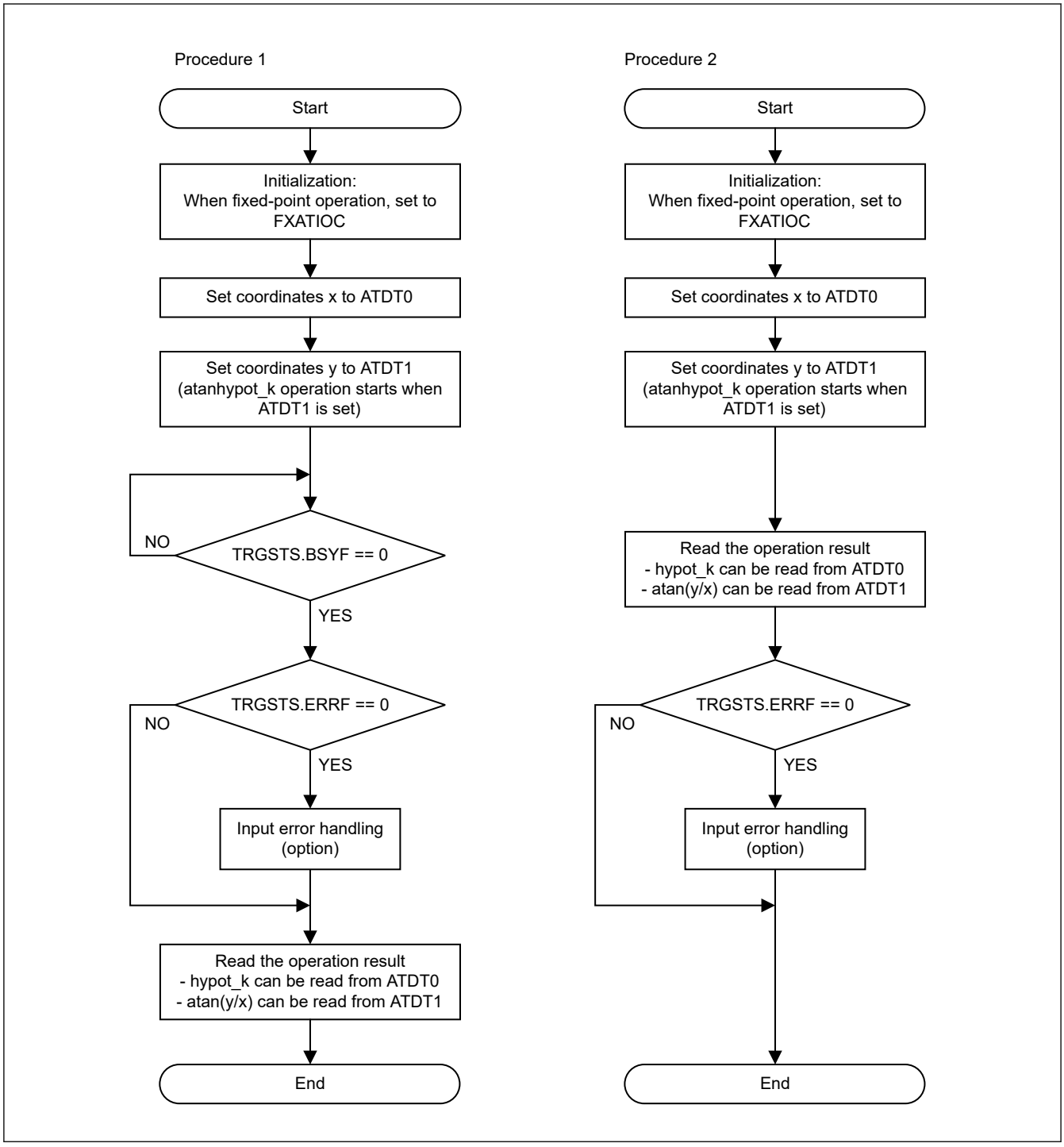


Figure 22.6 Procedure for using TFU (atanhypot_k operation)

22.4.8 Usage of Interrupt

In the middle of procedure for trigonometric function operation, when an interrupt occurs between a data register write and an error flag/data register read and trigonometric function operation is executed in the interrupt handler, malfunction occurs after returning from the interrupt handler.

Therefore, the following measures are required so that malfunction does not occur when multiple interrupts are used, and multiple interrupt handlers use TFU module.

1. Disable interrupts on the process from a data register write to an error flag and data register read
2. Save and restore an error flag and data registers in an interrupt handler

The latter measure is described as below.

Regardless of a kind of an operation in the interrupt handler and an operation being performed before the interrupt is received, data registers and error flag should be saved/restored by using DTSR0, DTSR1, and TRGSTS registers. Figure 22.7 shows the data save/restore procedure. By following this procedure, the operation before the interrupt is received can be continued without malfunction after returning from the interrupt handler.

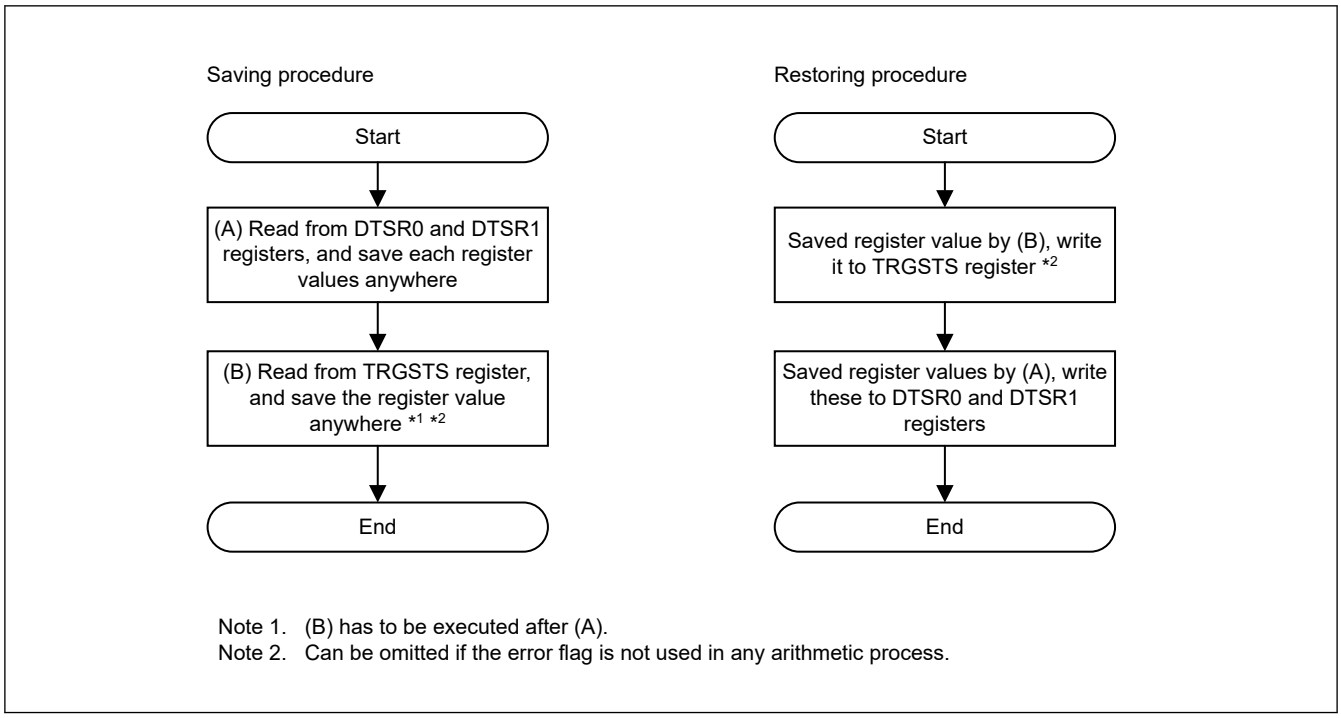


Figure 22.7 Data save/restore procedure in exception handler

When a setting will be changed in the interrupt handler and an arithmetic processing will be performed, in addition to the data registers and error flag, setting registers such as FXSCIOC and FXATIOC need to be saved and restored as well. Save and restore the setting registers as necessary. When saving the setting registers, be sure to do so after saving (A).

23. Compare Match Timer (CMT)

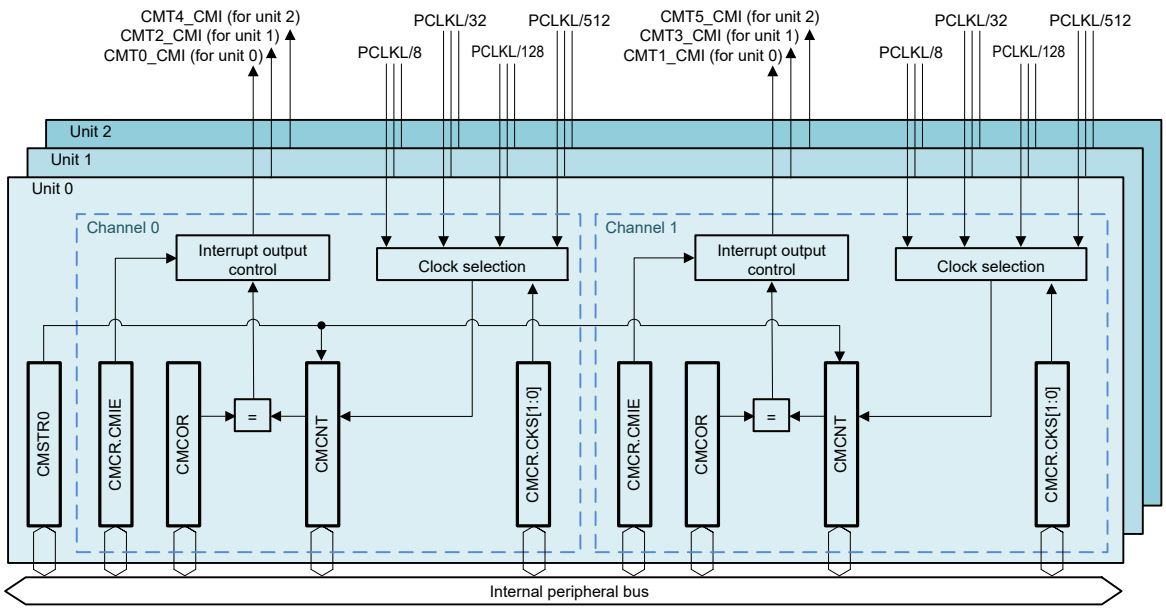
23.1 Overview

The LSI has three on-chip compare match timer (CMT) units, each consisting of a 2-channel 16-bit timer, for a total of six channels. The CMT has a 16-bit counter and can generate interrupts at set intervals.

Table 23.1 lists the specifications for the CMT and Figure 23.1 shows a block diagram of the CMT.

Table 23.1 CMT specifications

Item	Description
Number of internal channels	6 channels (2 channels × 3 units)
Timer counter (per channel)	16-bit up counter (counted according to the count enable signal output by the prescaler.) Returned to 0x0000 after compare match.
Prescaler (per channel)	9-bit counter (linked with enabling/disabling of timer counter operation.) ● Outputs four types of count clocks. The type can be selected from PCLKL/8, PCLKL/32, PCLKL/128, and PCLKL/512.
Interrupt	A compare match interrupt can be requested for each channel.
Module-stop function	Module-stop state can be set to reduce power consumption



CMSTRn: Compare match timer start register (n = 0, 2, 4)
CMCR: Compare match timer control register
CMCOR: Compare match timer constant register
CMCNT: Compare match timer counter register

Figure 23.1 CMT block diagram

23.2 Register Map

Table 23.2 CMT register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x8008_0000 + 0x200 × n, (n = 0, 2, 4)	CMSTRn	Compare Match Timer Start Register (n = 0, 2, 4)	—
0x8008_0002 + 0x200 × n, (n = 0, 2, 4) 0x8008_0008 + 0x200 × (n - 1), (n = 1, 3, 5)	CMCR	Compare Match Timer Control Register	—

Table 23.2 CMT register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x8008_0004 + 0x200 × n, (n = 0, 2, 4) 0x8008_000A + 0x200 × (n - 1), (n = 1, 3, 5)	CMCNT	Compare Match Timer Counter Register	—
0x8008_0006 + 0x200 × n, (n = 0, 2, 4) 0x8008_000C + 0x200 × (n - 1), (n = 1, 3, 5)	CMCOR	Compare Match Timer Constant Register	—

Table 23.3 CMT related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	—	MSTPCRD.MSTPCRD02	SLVACCCTL0.CMT0_SL
1	—	MSTPCRD.MSTPCRD03	SLVACCCTL0.CMT1_SL
2	—	MSTPCRD.MSTPCRD04	SLVACCCTL0.CMT2_SL

23.3 Register Descriptions

23.3.1 CMSTRn : Compare Match Timer Start Register (n = 0, 2, 4)

Base address: CMTC = 0x8008_0000

Offset address: 0x000 + 0x200 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR1	STR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STR0	CMT Channel n Count Start This bit selects starting or stopping of the CMTn.CMCNT counter operation. The corresponding prescaler starts or stops according to the setting of this bit. 0: The CMTn.CMCNT counter is stopped. 1: The CMTn.CMCNT counter is started.	R/W
1	STR1	CMT Channel n+1 Count Start This bit selects starting or stopping of the CMTn+1.CMCNT counter operation. The corresponding prescaler starts or stops according to the setting of this bit. 0: The CMTn+1.CMCNT counter is stopped. 1: The CMTn+1.CMCNT counter is started.	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

The CMSTR0 register sets starting or stopping of the CMT0.CMCNT and CMT1.CMCNT counters of unit 0.

The CMSTR2 register sets starting or stopping of the CMT2.CMCNT and CMT3.CMCNT counters of unit 1.

The CMSTR4 register sets starting or stopping of the CMT4.CMCNT and CMT5.CMCNT counters of unit 2.

23.3.2 CMCR : Compare Match Timer Control Register

Base address: CMT0 = 0x8008_0002, CMT2 = 0x8008_0402, CMT4 = 0x8008_0802
CMT1 = 0x8008_0008, CMT3 = 0x8008_0408, CMT5 = 0x8008_0808

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CMIE	—	—	—	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKS[1:0]	Clock Select 0 0: PCLKL/8 0 1: PCLKL/32 1 0: PCLKL/128 1 1: PCLKL/512	R/W
5:2	—	These bits are read as 0. The write value should be 0.	R/W
6	CMIE	Compare Match Interrupt Enable 0: Compare match interrupt (CMTn_CMI) disabled 1: Compare match interrupt (CMTn_CMI) enabled	R/W
7	—	The read value is undefined. The write value should be 0.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

The CMCR register specifies a clock used for count-up operation.

CKS[1:0] bits (Clock Select)

These bits select a clock to be input to the CMTn.CMCNT counter from the internal clocks obtained by dividing the frequency of the low-speed peripheral module clock (PCLKL) (n = 0 to 5). Setting the CMSTRm.STRn bit to 1 starts count-up operation of the corresponding CMCNT counter by using the clock selected in the CKS[1:0] bits (m = 0, 2, 4; n = 0, 1).

CMIE bit (Compare Match Interrupt Enable)

This bit selects whether to enable or disable generation of a compare match interrupt (CMTn_CMI) when the values in the CMCNT counter and in the CMCOR register match (n = 0 to 5).

23.3.3 CMCNT : Compare Match Timer Counter Register

Base address: CMT0 = 0x8008_0002, CMT2 = 0x8008_0402, CMT4 = 0x8008_0802
CMT1 = 0x8008_0008, CMT3 = 0x8008_0408, CMT5 = 0x8008_0808

Offset address: 0x002

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMCNT counter (the main unit of the compare match timer) is a readable/writable up-counter. When an internal clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 2, 4; n = 0, 1) bit is set to 1, the CMCNT counter starts counting up using the selected clock.

When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is cleared to 0x0000. At the same time, a compare match interrupt (CMTn_CMI) is generated (n = 0 to 5).

23.3.4 CMCOR : Compare Match Timer Constant Register

Base address: CMT0 = 0x8008_0002, CMT2 = 0x8008_0402, CMT4 = 0x8008_0802
CMT1 = 0x8008_0008, CMT3 = 0x8008_0408, CMT5 = 0x8008_0808

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMCOR register is a readable/writable register to set a cycle for compare match with the CMCNT counter. The cycle for compare matches is as follows:

Compare-match cycle = (setting of the CMCOR register + 1) × counter-clock cycle*1

Note 1. This is a clock cycle set by the CMCR.CKS[1:0] bits.

23.4 Operation

23.4.1 Periodic Count Operation

When an internal clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 2, 4; n = 0, 1) bit is set to 1, the CMCNT counter starts counting up using the selected clock.

When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is cleared to 0x0000, and then a compare match interrupt (CMTn_CMI) is generated (n = 0 to 5). The CMCNT counter then starts counting up again from 0x0000. Figure 23.2 shows the operation of the CMCNT counter.

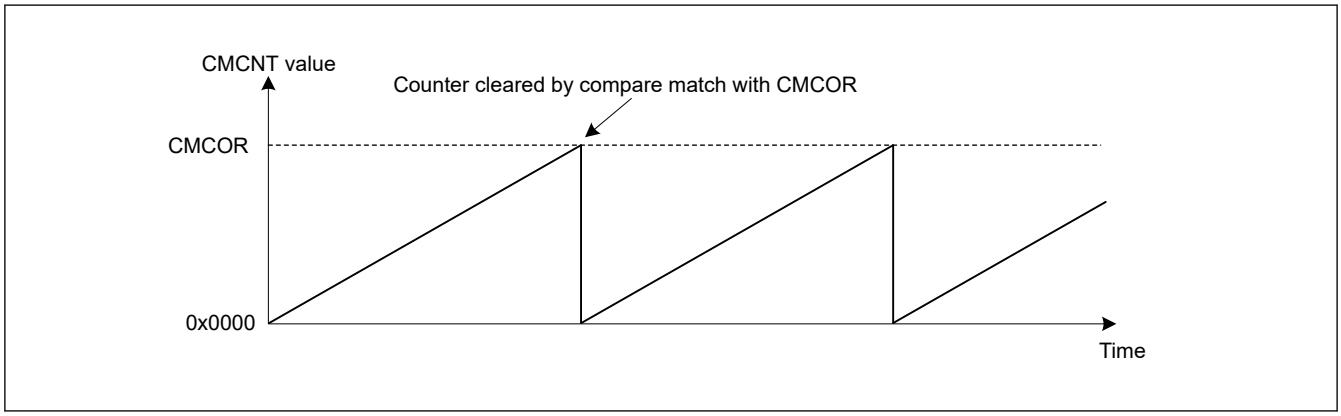


Figure 23.2 Counter operation

23.4.2 CMCNT Count Timing

As the count clock, one of four internal clocks (PCLKL/8, PCLKL/32, PCLKL/128, and PCLKL/512) obtained by dividing the low-speed peripheral module clock (PCLKL) can be selected with the CMCR.CKS[1:0] bits. Figure 23.3 shows the timing of the CMCNT counter.

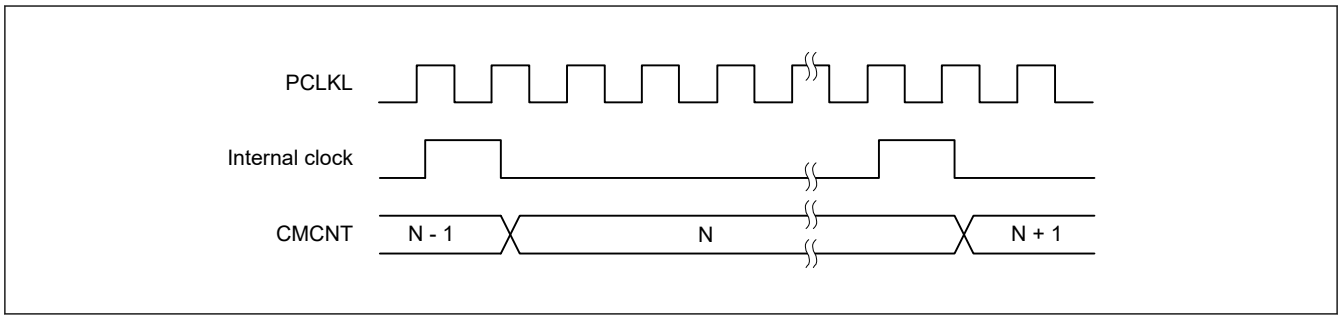


Figure 23.3 CMCNT count timing

23.5 Interrupts

23.5.1 Interrupt Sources

For each CMT channel, a compare-match interrupt (CMTn_CMI) is allocated and has a different event number. The priority of channels can be changed by the interrupt controller settings.

Table 23.4 CMT interrupt sources (1 of 2)

Name	Interrupt Sources	CPU0 GIC request	CPU1 GIC request	DMAC activation
CMT0_CMI	Compare match between the CMT0.CMCNT counter and CMT0.CMCOR register	Possible	Possible	Possible

Table 23.4 CMT interrupt sources (2 of 2)

Name	Interrupt Sources	CPU0 GIC request	CPU1 GIC request	DMAC activation
CMT1_CMI	Compare match between the CMT1.CMCNT counter and CMT1.CMCOR register	Possible	Possible	Possible
CMT2_CMI	Compare match between the CMT2.CMCNT counter and CMT2.CMCOR register	Possible	Possible	Possible
CMT3_CMI	Compare match between the CMT3.CMCNT counter and CMT3.CMCOR register	Possible	Possible	Possible
CMT4_CMI	Compare match between the CMT4.CMCNT counter and CMT4.CMCOR register	Possible	Possible	Possible
CMT5_CMI	Compare match between the CMT5.CMCNT counter and CMT5.CMCOR register	Possible	Possible	Possible

23.5.2 Timing of Compare Match Interrupt Generation

When the CMCNT counter and the CMCOR register match, a compare match interrupt (CMTn_CMI) is generated.

A compare match signal is generated at the last state in which the values match (the timing when the CMCNT counter updates the matched count value). That is, after a match between the CMCOR register and the CMCNT counter, the compare match signal is not generated until the CMCNT counter input clock is generated (n = 0 to 5).

Figure 23.4 shows the timing of a compare match interrupt.

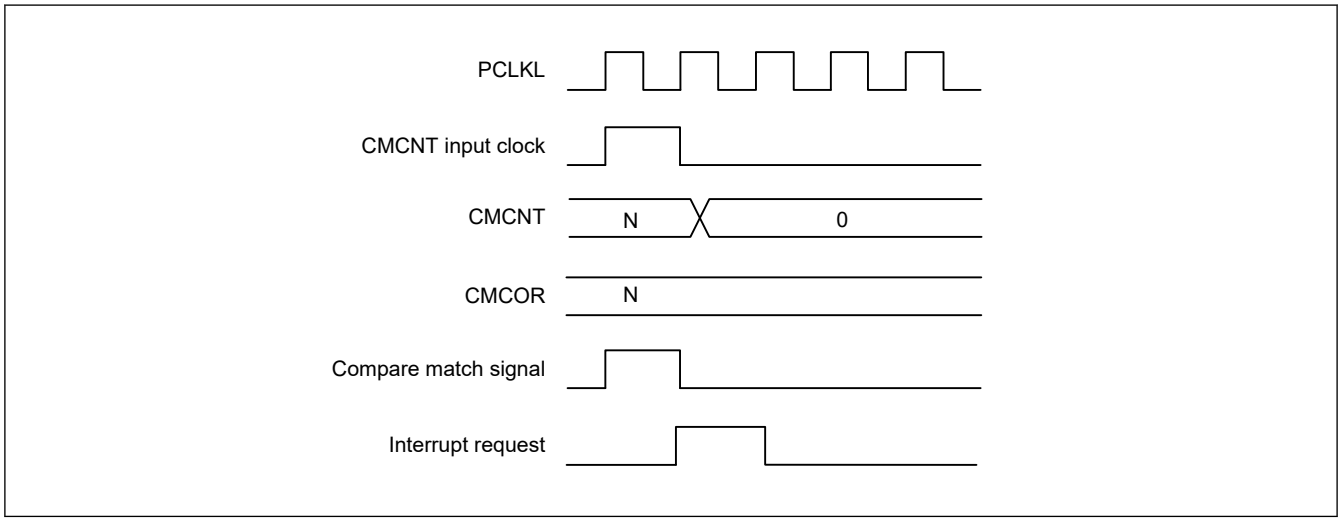


Figure 23.4 Timing a compare match interrupt is set

23.6 Usage Notes

23.6.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the Module Stop Control register. After a reset, the CMT is in the module-stop state. The registers can be accessed by canceling the module-stop state. For details, see section 9, Low-Power Consumption Function.

23.6.2 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated while writing to the CMCNT counter, clearing the CMCNT counter is given priority over the writing to it. In this case, the CMCNT counter is not written to. Figure 23.5 shows the timing to clear the CMCNT counter.

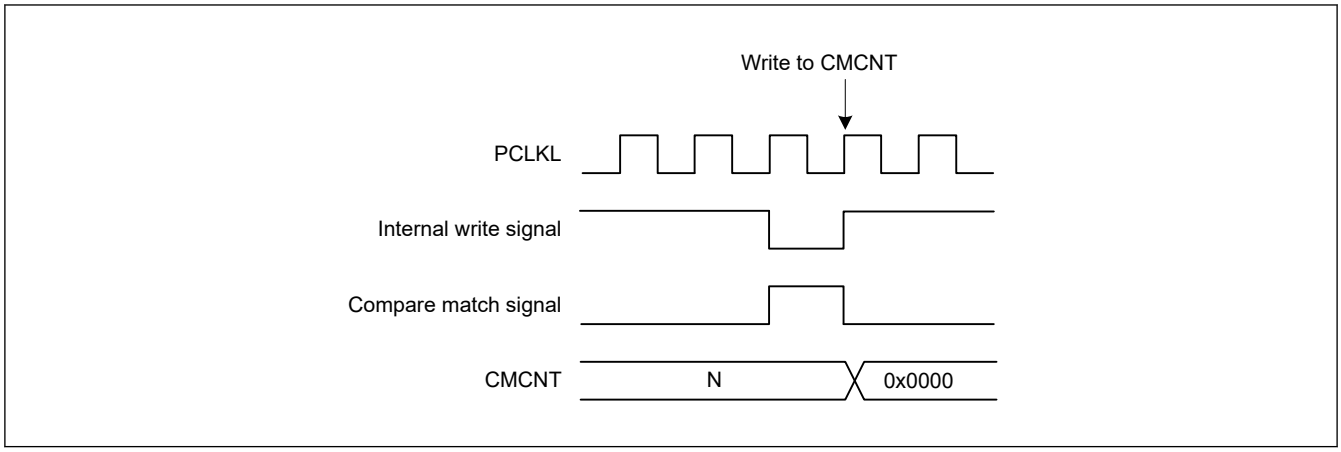


Figure 23.5 Conflict between write and compare match processes of CMCNT

23.6.3 Conflict between Write and Count-Up Processes of CMCNT

If count-up occurs during the write access to the CMCNT counter, that writing has priority over the count-up. Figure 23.6 shows the timing to write the CMCNT counter.

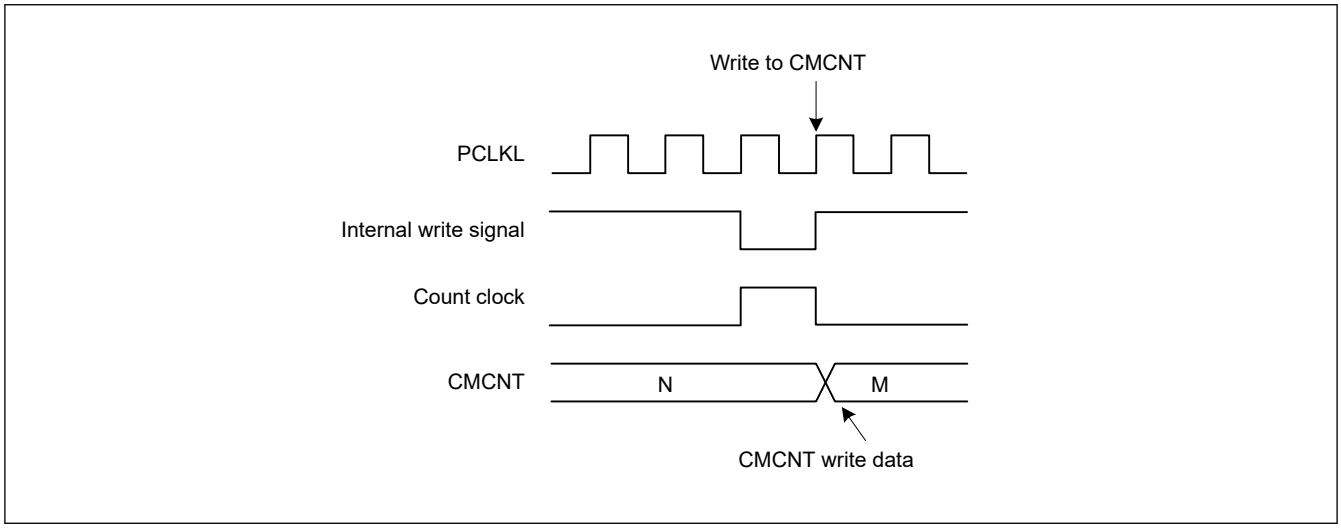


Figure 23.6 Conflict between write and count-up processes of CMCNT

24. Compare Match Timer W (CMTW)

24.1 Overview

The LSI has two channels (1 channel × 2 units) of 32-bit compare match timer W (CMTW). CMTW has a 32-bit counter and can generate interrupts each time a set period elapses.

Table 24.1 shows the specifications of the CMTW and Figure 24.1 shows a block diagram of the CMTW.

Table 24.1 CMTW specifications

Item	Description
Number of channel	Two channels (1 channel × 2 units)
Timer counter	16-bit/32-bit selectable up-counter. The counter returns to 0x00000000 after a compare match.
Prescaler	Four dividing clocks are output. Selectable from any of PCLKL/8, PCLKL/32, PCLKL/128, or PCLKL/512.
Input capture	Up to two input capture input signals available per unit
Output compare	Up to two output compare output signals available per unit
Compare match	One compare match available (no output compare output pin used)
Interrupt	Compare match interrupt Input capture 0 and 1 interrupts per unit Output compare 0 and 1 interrupts per unit
Module-stop function	Module-stop state can be set to reduce power consumption

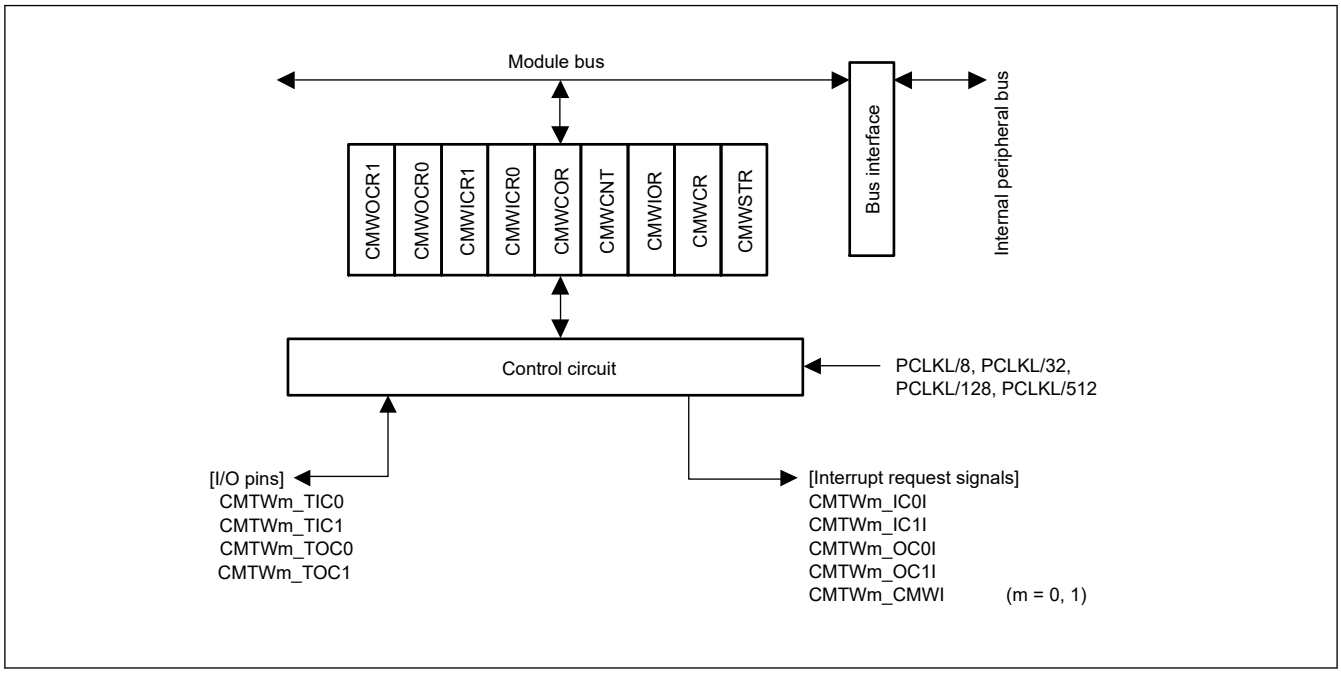


Figure 24.1 Block diagram of CMTW (units 0 to 1)

Table 24.2 shows the CMTW pin configuration.

Table 24.2 Input/Output pins of CMTW

Unit	Pin name	I/O	Description
CMTWm (m = 0, 1)	CMTWm_TIC0	Input	CMTWm input capture input 0
	CMTWm_TIC1	Input	CMTWm input capture input 1
	CMTWm_TOC0	Output	CMTWm output compare output 0
	CMTWm_TOC1	Output	CMTWm output compare output 1

Note: For convention, pin names within this chapter are written without "CMTWm_" (m = 0, 1) notation except this table and the block diagram.

24.2 Register Map

Table 24.3 CMTW register map

Address	Register symbol	Register name	Write protection
0x8008_1000 + 0x400 × m	CMWSTR	Timer Start Register	—
0x8008_1004 + 0x400 × m	CMWCR	Timer Control Register	—
0x8008_1008 + 0x400 × m	CMWIOR	Timer I/O Control Register	—
0x8008_1010 + 0x400 × m	CMWCNT	Timer Counter	—
0x8008_1014 + 0x400 × m	CMWCOR	Compare Match Constant Register	—
0x8008_1018 + 0x400 × m + 0x004 × n	CMWICRn	Input Capture Registers (n = 0, 1)	—
0x8008_1020 + 0x400 × m + 0x004 × n	CMWOCRn	Output Compare Registers (n = 0, 1)	—

Note: m = 0, 1

Table 24.4 CMTW related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	—	MSTPCRD.MSTPCRD05	SLVACCCTL0.CMTW0_SL
1	—	MSTPCRD.MSTPCRD06	SLVACCCTL0.CMTW1_SL

24.3 Register Descriptions

24.3.1 CMWSTR : Timer Start Register

Base address: CMTWm = 0x8008_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STR	Counter Start 0: The CMWCNT counter stops counting. The value immediately before a stop of counting is retained and counting is stopped. 1: The CMWCNT counter starts counting.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

The CMWSTR register is used to start or stop the CMWCNT counter.

STR bit (Counter Start)

Specifies whether the timer counter operates or is stopped. The relevant prescaler operates or is stopped according to the settings of STR bit.

24.3.2 CMWCR : Timer Control Register

Base address: CMTWm = 0x8008_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CCLR[2:0]			—	—	—	CMS	—	OC1IE	OC0IE	IC1IE	IC0IE	CMWIE	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKS[1:0]	Clock Select 0 0: PCLKL/8 0 1: PCLKL/32 1 0: PCLKL/128 1 1: PCLKL/512	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CMWIE	Compare Match Interrupt Enable 0: Disables a compare match interrupt request (CMTWm_CMWI). 1: Enables a compare match interrupt request (CMTWm_CMWI).	R/W
4	IC0IE	Input Capture 0 Interrupt Enable 0: Disables an interrupt request by the input capture 0 (CMTWm_IC0I). 1: Enables an interrupt request by the input capture 0 (CMTWm_IC0I).	R/W
5	IC1IE	Input Capture 1 Interrupt Enable 0: Disables an interrupt request by the input capture 1 (CMTWm_IC1I). 1: Enables an interrupt request by the input capture 1 (CMTWm_IC1I).	R/W
6	OC0IE	Output Compare 0 Interrupt Enable 0: Disables an interrupt request by the output compare 0 (CMTWm_OC0I). 1: Enables an interrupt request by the output compare 0 (CMTWm_OC0I).	R/W
7	OC1IE	Output Compare 1 Interrupt Enable 0: Disables an interrupt request by the output compare 1 (CMTWm_OC1I). 1: Enables an interrupt request by the output compare 1 (CMTWm_OC1I).	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	CMS	Timer Counter Size 0: 32 bits 1: 16 bits	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
15:13	CCLR[2:0]	Counter Clear 0 0 0: The CMWCNT counter is cleared by CMWCOR register compare match. 0 0 1: The CMWCNT counter is not cleared. 0 1 0: The CMWCNT counter is not cleared. 0 1 1: The CMWCNT counter is not cleared. 1 0 0: The CMWCNT counter is cleared by CMWICR0 register input capture. 1 0 1: The CMWCNT counter is cleared by CMWICR1 register input capture. 1 1 0: The CMWCNT counter is cleared by CMWOCR0 register compare match. 1 1 1: The CMWCNT counter is cleared by CMWOCR1 register compare match.	R/W

The CMWCR register selects the counter clearing source and the counter input clock, and enables or disables interrupts.

The CMWCR register should be set while the timer counter (CMWCNT) operation is stopped.

CKS[1:0] bits (Clock Select)

Select the clock to be input to the CMWCNT counter among four internal clocks obtained by dividing the peripheral clock (PCLKL). When the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting up based on the clock selected with the CMWCR.CKS[1:0] bits.

CMWIE bit (Compare Match Interrupt Enable)

Enables or disables compare match interrupt (CMTWm_CMWI) request generation when the CMWCNT counter and CMWCOR register values match.

IC0IE bit (Input Capture 0 Interrupt Enable)

Enables or disables input capture interrupt 0 (CMWm_IC0I) request generation when input capture is generated in the CMWICR0 register.

IC1IE bit (Input Capture 1 Interrupt Enable)

Enables or disables input capture interrupt 1 (CMWm_IC1I) request generation when input capture is generated in the CMWICR1 register.

OC0IE bit (Output Compare 0 Interrupt Enable)

Enables or disables compare match interrupt 0 (CMWm_OC0I) request generation when the CMWCNT counter and CMWOCR0 register values match.

OC1IE bit (Output Compare 1 Interrupt Enable)

Enables or disables compare match interrupt 1 (CMWm_OC1I) request generation when the CMWCNT counter and CMWOCR1 register values match.

CMS bit (Timer Counter Size)

Selects either 16 or 32 bits as the size of the timer counter (CMWCNT). The size selected with the CMS bit is valid in the compare match constant register (CMWCOR), input capture registers (CMWICR0 and CMWICR1), and output compare registers (CMWOCR0 and CMWOCR1).

CCLR[2:0] bits (Counter Clear)

Selects the CMWCNT counter clearing source.

24.3.3 CMWIOR : Timer I/O Control Register

Base address: CMTWm = 0x8008_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMWE	—	OC1E	OC0E	OC1[1:0]		OC0[1:0]		—	—	IC1E	IC0E	IC1[1:0]		IC0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	IC0[1:0]	Input Capture Control 0 0 0: Input capture at the rising edge on the TIC0 pin. 0 1: Input capture at the falling edge on the TIC0 pin. 1 0: Input capture at both edges on the TIC0 pin. 1 1: Setting prohibited	R/W
3:2	IC1[1:0]	Input Capture Control 1 0 0: Input capture at the rising edge on the TIC1 pin. 0 1: Input capture at the falling edge on the TIC1 pin. 1 0: Input capture at both edges on the TIC1 pin. 1 1: Setting prohibited	R/W
4	IC0E	Input Capture Enable 0 0: Disables the input capture operation of the CMWICR0 register. 1: Enables the input capture operation of the CMWICR0 register.	R/W
5	IC1E	Input Capture Enable 1 0: Disables the input capture operation of the CMWICR1 register. 1: Enables the input capture operation of the CMWICR1 register.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
9:8	OC0[1:0]	Output Compare Control 0 0 0: Retains the output value.*1 0 1: Initially outputs 0 and toggles the output value upon compare match. 1 0: Initially outputs 1 and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
11:10	OC1[1:0]	Output Compare Control 1 0 0: Retains the output value.*1 0 1: Initially outputs 0 and toggles the output value upon compare match. 1 0: Initially outputs 1 and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
12	OC0E	Compare Match Enable 0 0: Disables the compare match operation using the CMWOCR0 register. 1: Enables the compare match operation using the CMWOCR0 register.	R/W
13	OC1E	Compare Match Enable 1 0: Disables the compare match operation using the CMWOCR1 register. 1: Enables the compare match operation using the CMWOCR1 register.	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	CMWE	Compare Match Enable 0: Disables the compare match operation using the CMWCOR register. 1: Enables the compare match operation using the CMWCOR register.	R/W

Note 1. After reset, 0 is output until the CMWIOR register is set.

The CMWIOR register controls the CMWCOR, CMWICR0, CMWICR1, CMWOCR0, and CMWOCR1 registers.

CMWIOR should be set while the timer counter (CMWCNT) operation is stopped.

IC0[1:0] bits (Input Capture Control 0)

Selects the input capture operation of the CMWICR0 register.

IC1[1:0] bits (Input Capture Control 1)

Selects the input capture operation of the CMWICR1 register.

IC0E bit (Input Capture Enable 0)

Enables or disables the input capture operation of the CMWICR0 register.

IC1E bit (Input Capture Enable 1)

Enables or disables the input capture operation of the CMWICR1 register.

OC0[1:0] bits (Output Compare Control 0)

Sets the output compare operation using the CMWOCR0 register.

OC1[1:0] bits (Output Compare Control 1)

Sets the output compare operation using the CMWOCR1 register.

OC0E bit (Compare Match Enable 0)

Enables or disables the compare match operation using the CMWOCR0 register.

OC1E bit (Compare Match Enable 1)

Enables or disables the compare match operation using the CMWOCR1 register.

CMWE bit (Compare Match Enable)

Enables or disables the compare match operation using the CMWCOR register.

24.3.4 CMWCNT : Timer Counter

Base address: CMTWm = 0x8008_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x010

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Timer Counter	R/W

The CMWCNT counter is used as a readable/writable up-counter.

Before starting counter operation, the timer control register (CMWCR) should be set. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in the CMWCNT counter are valid. Since access to this register is in 32-bit units, when writing, write 32-bit values to it with 0x0000 in the higher-order bits. If a value other than 0x0000 is set in the higher-order bits, a value greater than 0x0000FFFF may be read when this register is read.

When the STR bit is set to 1, the CMWCNT counter starts counting. When the STR bit is set to 0, the CMWCNT counter retains the value immediately before a stop of counting and stops counting.

24.3.5 CMWCOR : Compare Match Constant Register

Base address: CMTWm = 0x8008_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x014

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The CMWCOR register is a readable/writable register that specifies the time up to a compare match between the timer counter (CMWCNT) value and CMWCOR value. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bit 15 to bit 0 in this register are valid. Since access to this register is in 32-bit units, when writing, write 32-bit values to it with 0x0000 in the higher-order bits.	R/W

The cycle for compare matches is as follows:

Compare-match cycle = (setting of the CMWCOR register + 1) × counter-clock cycle[*1]

Note 1. This is a clock cycle set by the CMWCR.CKS[1:0] bits.

24.3.6 CMWICRn : Input Capture Registers (n = 0, 1)

Base address: CMTWm = 0x8008_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x018 + 0x004 × n

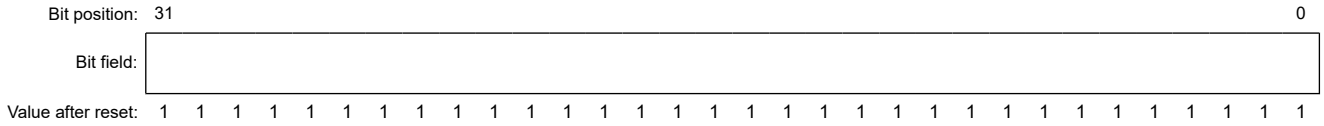
Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	The CMWICR0 and CMWICR1 registers are read-only registers in which the CMWCNT counter value is stored when an input capture is generated. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in these registers are valid. Writing to these registers is invalid.	R

24.3.7 CMWOCRn : Output Compare Registers (n = 0, 1)

Base address: CMTWm = 0x8008_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x020 + 0x004 × n



Bit	Symbol	Function	R/W
31:0	n/a	The CMWOCR0 and CMWOCR1 registers are readable/writable registers that set the value to be compared when an output compare is generated. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 of these registers become valid. Since access to this register is in 32-bit units, when writing, write 32-bit values to it with 0x0000 in the higher-order bits.	R/W

24.4 Operation

When the CMWCR register is set and then the STR bit in CMWCR is set to 1, the CMTW starts counter operation.

Setting the CMWSTR.STR bit to 0 enables the CMWCNT counter to retain the value immediately before a stop of counting and stop counting. Setting CMWIOR register enables using the compare match function, input capture input function, and output compare output function.

24.4.1 Period Counting Operation

When the internal clock is selected by using the CMWCNT.CKS[1:0] bits and the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting up cycles of the selected clock. When the CCLR[2:0] bits in CMWCR are set so that CMWCNT should be cleared by a specific counter clearing source and the counter clearing source is generated, the CMWCNT counter is cleared to 0x00000000 and continues incrementing. When the CCLR[2:0] bits are set so that CMWCNT should not be cleared by any specific counter clearing source, the CMWCNT counter is cleared to 0x00000000 only when an overflow is generated (0xFFFFFFFF → 0x00000000 (when the counter size is 32 bits) or 0x0000FFFF → 0x00000000 (when the counter size is 16 bits)) and continues incrementing.

24.4.2 Compare Match Function

When the values of the CMWCNT counter and CMWCOR register match, a compare match interrupt (CMTWm_CMWI) is generated. The CMWCNT counter operates as follows according to the setting of CMWCR.CCLR[1:0].

1. When CMWCR.CCLR[2:0] = 000b
   When the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter is cleared to 0x00000000. The CMWCNT counter then restarts counting up from 0x00000000.
2. When CMWCR.CCLR[2:0] ≠ 000b
   Even when the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter is not cleared to 0x00000000 but continues counting up until the clearing condition set in CMWCR.CCLR[1:0] is satisfied or the value of the counter reaches 0xFFFFFFFF (when the size of the counter is 32 bits) or 0x0000FFFF (when the size of the counter is 16 bits).
   The CMWCNT counter is then cleared to 0x00000000 and restarts counting up from 0x00000000.

Figure 24.2 shows an example of procedure for setting compare match operation.

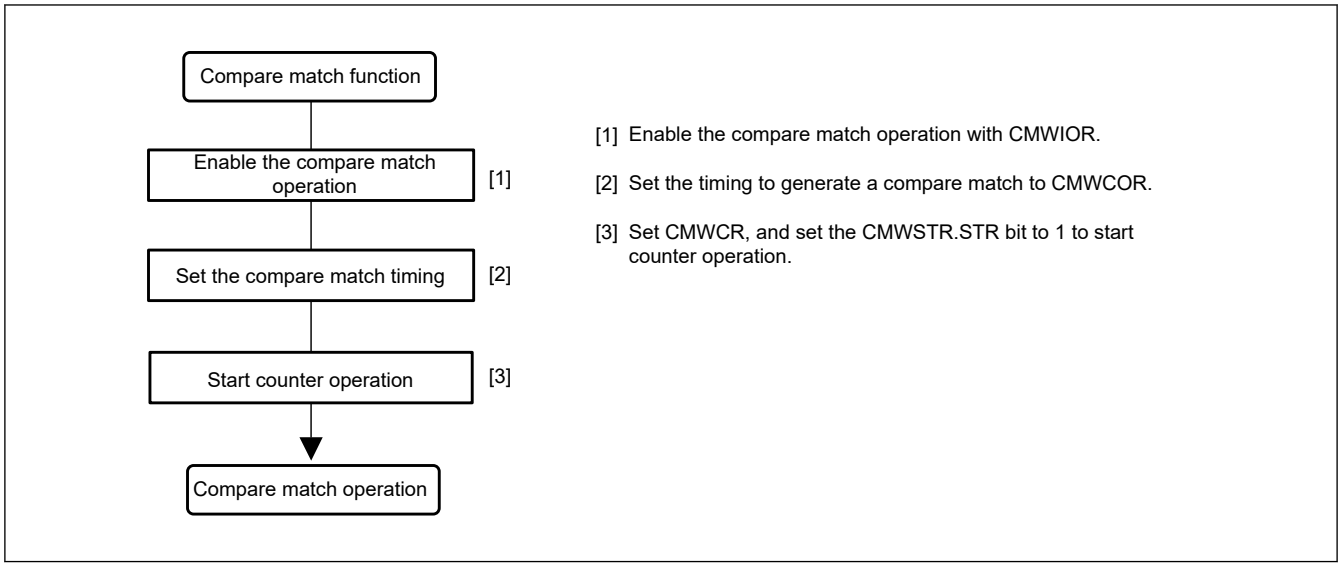


Figure 24.2 Procedure for setting compare match operation

Figure 24.3 shows an example when compare match with CMWCOR is set as a counter clearing source.

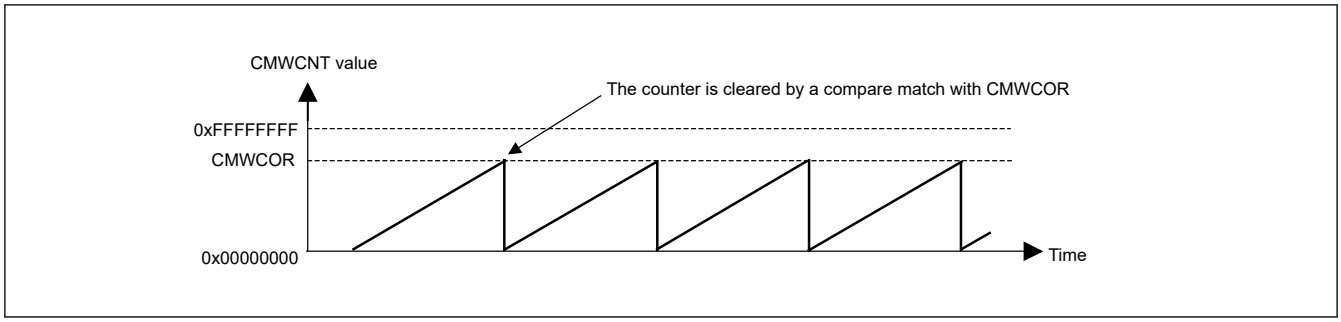


Figure 24.3 Example of compare match operation

Figure 24.4 shows an example when CMWCOR is set to 0xFFFFFFFF and detected as overflow.

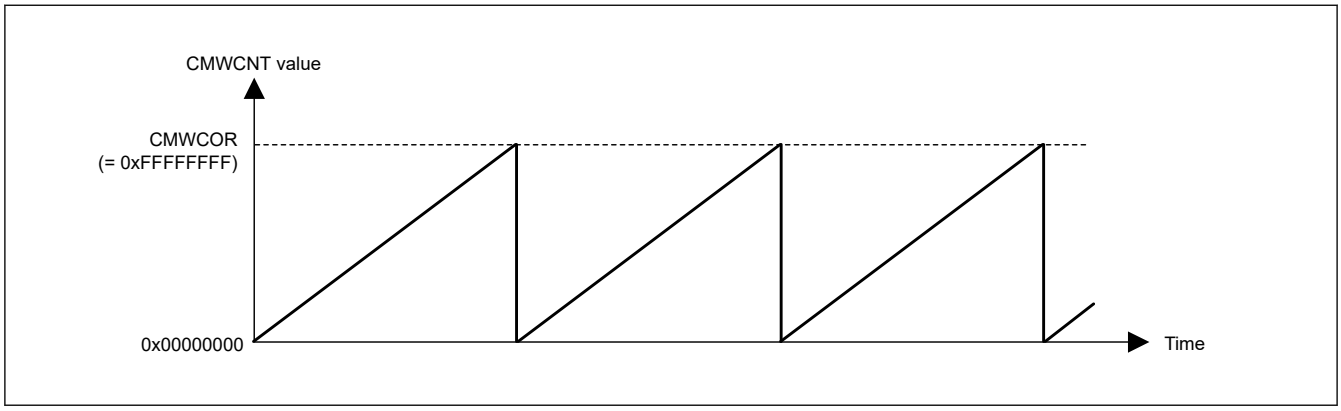


Figure 24.4 Example of compare match operation (overflow detected)

24.4.3 Output Compare Function

Using the output compare function, toggle output from the relevant output pins can be provided. When the CMWCNT counter value matches either of the values of CMWOCR0 or CMWOCR1 register, the output compare interrupt (CMTWm_OC0I or CMTWm_OC1I (m = 0, 1)) is generated. The CMWCNT counter operates as follows according to the setting of CMWCR.CCLR[2:0].

1. When CMWCR.CCLR[2:0] = 110b
   When the values of the CMWCNT counter and CMWOCR0 register match, the CMWCNT counter is cleared to 0x00000000. The CMWCNT counter then restarts counting up from 0x00000000.

2. When CMWCR.CCLR[2:0] = 111b
   When the values of the CMWCNT counter and CMWOCR1 register match, the CMWCNT counter is cleared to 0x00000000. The CMWCNT counter then restarts counting up from 0x00000000.
3. When CMWCR.CCLR[2:0] = others except 110b and 111b
   Even when the values of the CMWCNT counter and CMWOCR0 or CMWOCR1 register match, the CMWCNT counter is not cleared to 0x00000000 but continues counting up until the clearing condition set in CMWCR.CCLR[2:0] is satisfied or the value of the counter reaches 0xFFFFFFFF (when the size of the counter is 32 bits) or 0x0000FFFF (when the size of the counter is 16 bits). The CMWCNT counter is then cleared to 0x00000000 and restarts counting up from 0x00000000.

Figure 24.5 shows an example of procedure for setting output compare operation.

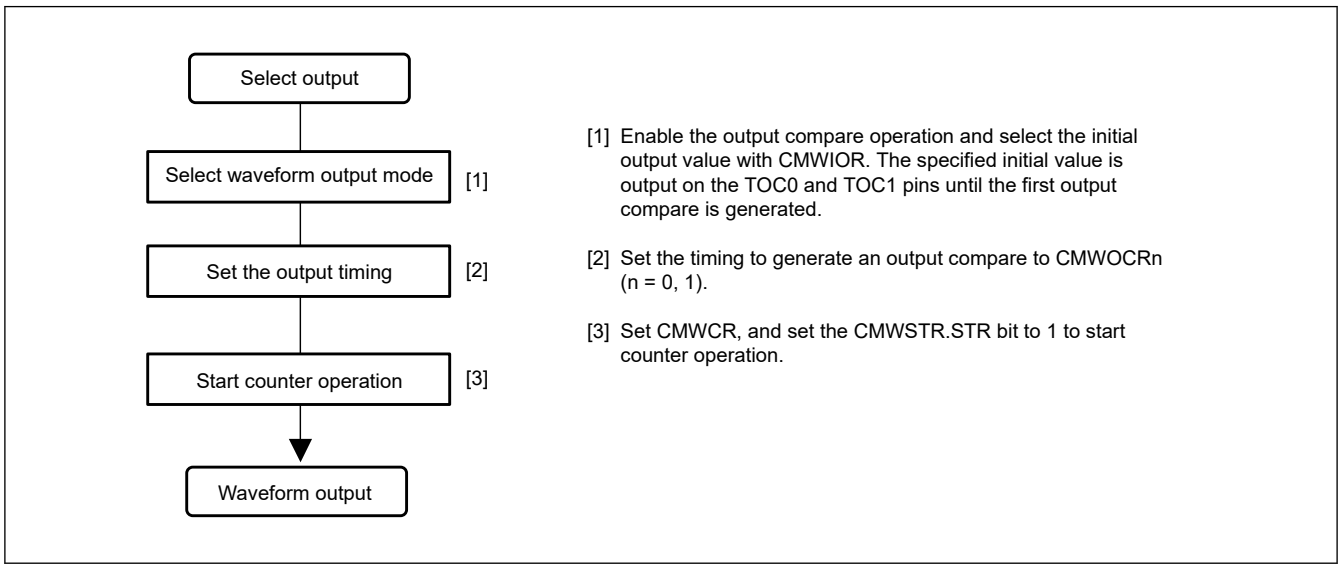


Figure 24.5 Procedure for setting output compare operation

Figure 24.6 shows an example when the counter is cleared upon compare match with CMWOCR1 register and toggle outputs are provided from the TOC0 and TOC1 pins.

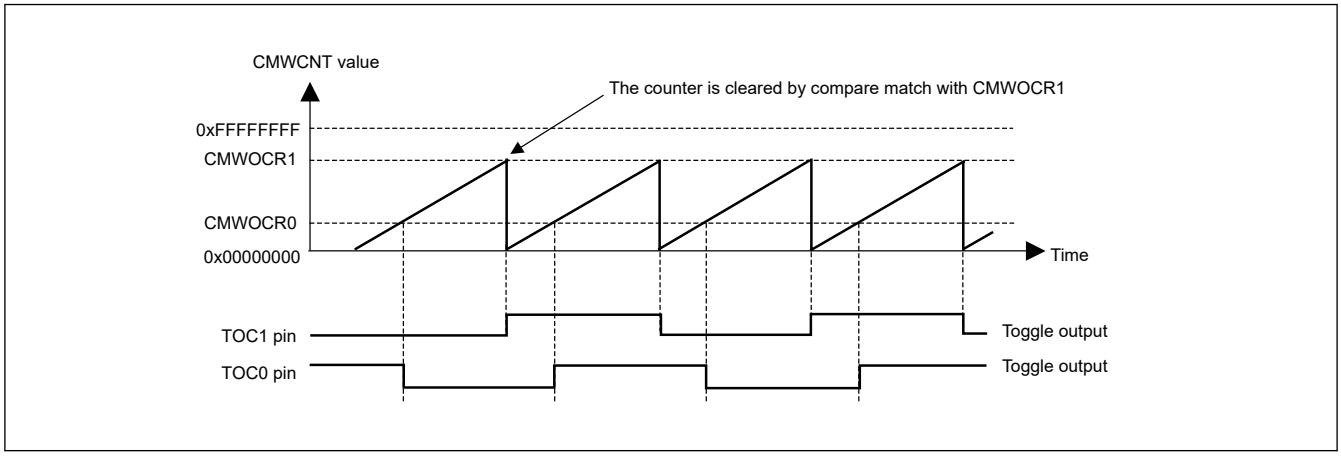


Figure 24.6 Example of output compare operation

24.4.4 Input Capture Function

Through detecting the edge on the TIC0 and TIC1 pin input, the CMWCNT counter value can be transferred to CMWICR0 and CMWICR1 registers, respectively. The edges to be detected can be selected from among the rising edge alone, falling edge alone, and both the rising and falling edges. When the CMWCNT counter value is transferred to CMWICR0 or CMWICR1 register using the input capture function, an input capture interrupt (CMTWm_IC0I or CMTWm_IC1I (m = 0, 1)) is generated. The CMWCNT counter operates as follows according to the setting of CMWCR.CCLR[2:0].

1. When CMWCR.CCLR[2:0] = 100b
   When the CMWCNT counter value is transferred to CMWICR0 using the input capture operation, the CMWCNT counter is cleared to 0x00000000.
   The CMWCNT counter then restarts counting up from 0x00000000.
2. When CMWCR.CCLR[2:0] = 101b
   When the CMWCNT counter value is transferred to CMWICR1 using the input capture operation, the CMWCNT counter is cleared to 0x00000000.
   The CMWCNT counter then restarts counting up from 0x0000000.
3. When CMWCR.CCLR[2:0] = others except 110b and 111b
   Even when the CMWCNT counter value is transferred to CMWICR0 or CMWICR1 using the input capture operation, the CMWCNT counter is not cleared to 0x00000000 but continues counting up until the clearing condition set in CMWCR.CCLR[2:0] is satisfied or the value of the counter reaches 0xFFFFFFFF (when the size of the counter is 32 bits) or 0x0000FFFF (when the size of the counter is 16 bits). The CMWCNT counter is then cleared to 0x00000000 and restarts counting up from 0x00000000.

Figure 24.7 shows an example of procedure for setting input capture operation.

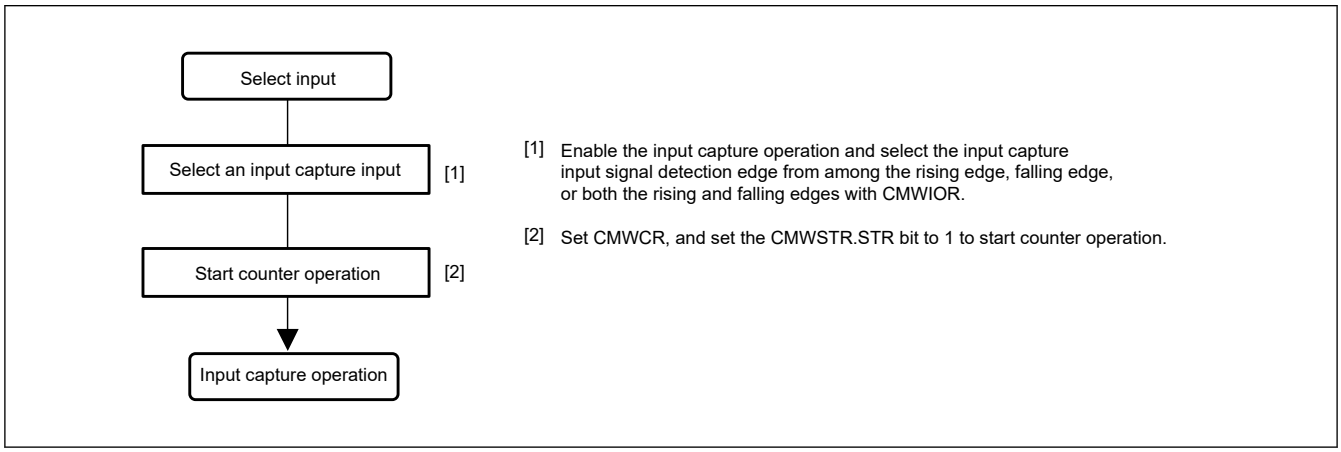


Figure 24.7 Procedure for setting input capture operation

Figure 24.8 shows an example in which both the rising and falling edges are selected for the TIC0 pin input capture input edge and the falling edge for the TIC1 pin, and the CMWCNT counter is cleared by a CMWICR1 register input capture.

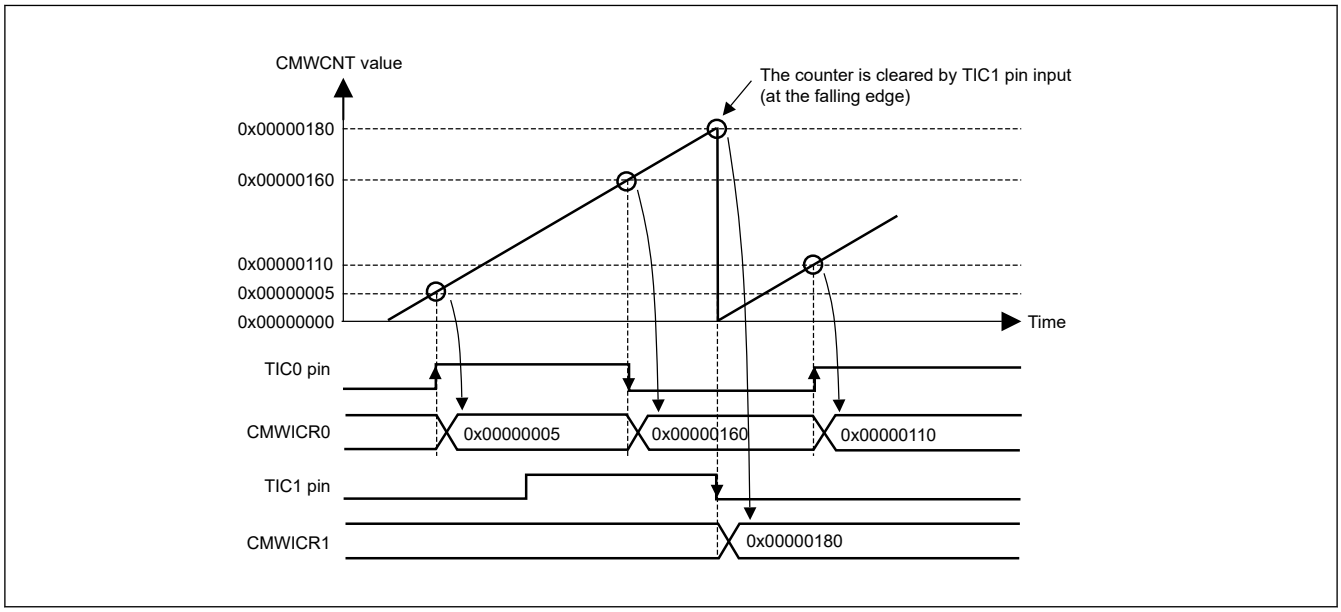


Figure 24.8 Example of input capture operation

24.4.5 Counter Size

With the CMTW, either 16 or 32 bits can be selected as the counter size by using the CMWCR.CMS bit. When the counter is used as a 32-bit counter, set the CMWCOR, CMWOCR0, or CMWOCR1 register to the desired values in 32-bit units. In reading, all 32 bits of CMWICR0 and CMWICR1 are valid. When the counter is used as a 16-bit counter, a 32-bit value should be set to the CMWCOR register with 0x0000 in the higher-order bits. Similarly, a 32-bit value should be set to CMWOCR0 and CMWOCR1 registers with 0x0000 in the higher-order bits.

A 32-bit value with 0x0000 in the higher-order bits is read from CMWICR0 and CMWICR1 registers.

24.4.6 Count Timing Based on CMWCNT

One of four clocks (PCLKL/8, PCLKL/32, PCLKL/128, and PCLKL/512) obtained by dividing the peripheral clock (PCLKL) can be selected with the CMWCR.CKS[1:0] bits. Figure 24.9 shows the timing.

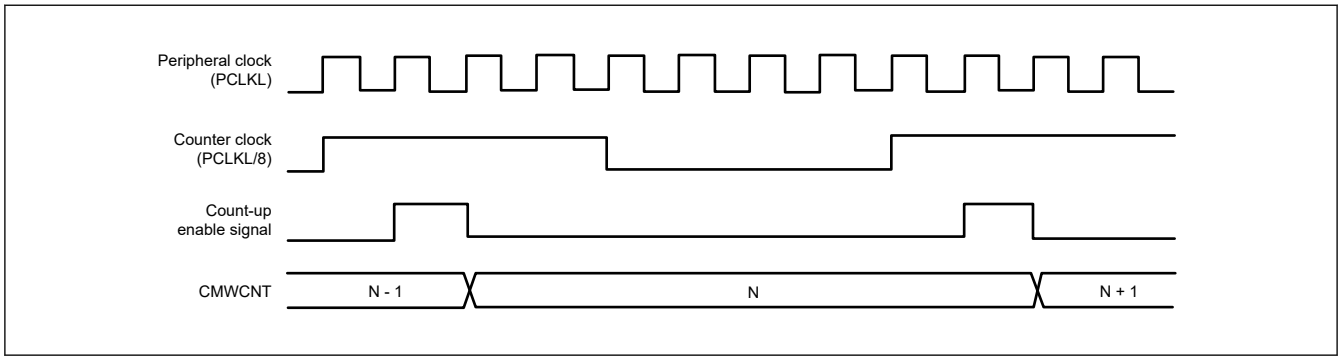


Figure 24.9 Count timing (PCLKL/8)

24.4.7 Output Compare Output Timing

A compare match signal is generated in the last state in which the CMWOCRn (n = 0, 1) register and CMWCNT counter values match (the CMWCNT counter value is updated immediately after the state). That is, the compare match signal is not generated if the CMWCNT counter clock is not input after a match between the CMWOCRn register and CMWCNT counter values. When a compare match signal is generated, the output compare output pin (TOC) changes in accord with the setting of the OC0[1:0] or OC1[1:0] bits in the CMWIOR register. Figure 24.10 shows output compare output timing.

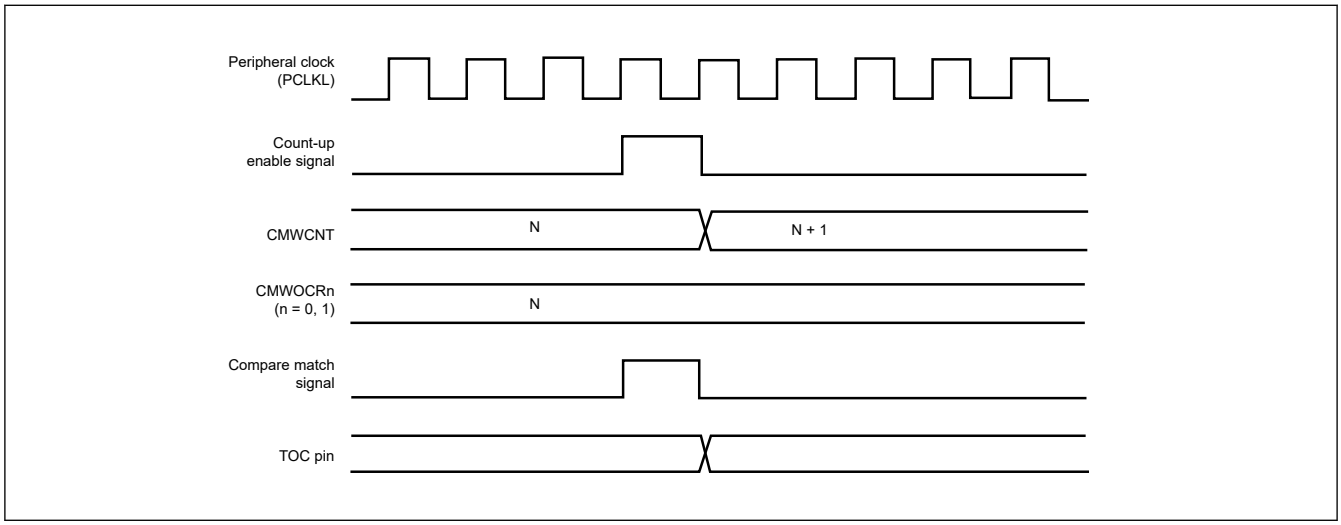


Figure 24.10 Output compare output timing

24.4.8 Input Capture Signal Timing

Figure 24.11 shows the input capture timing.

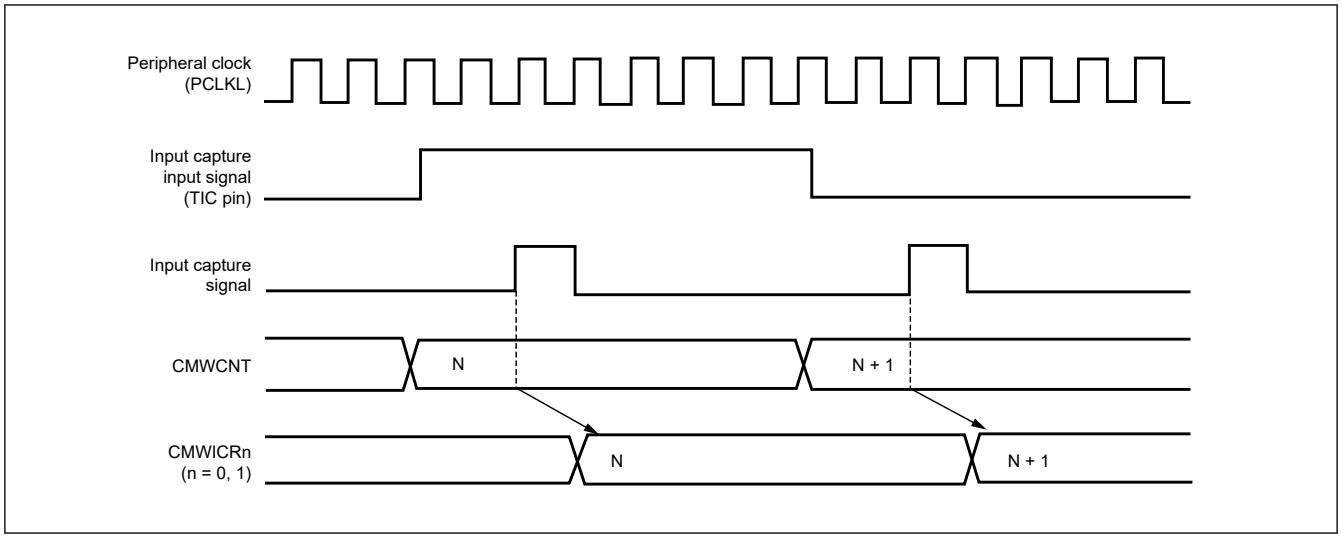


Figure 24.11 Input capture input signal timing

24.5 Interrupts

24.5.1 CMTW Interrupt Sources and DMAC Transfer Requests

The CMTW has five interrupt sources: two input capture interrupt requests (CMTWm_IC0I and CMTWm_IC1I), two output compare interrupt requests (CMTWm_OC0I and CMTWm_OC1I), and a compare match interrupt request (CMTWm_CMWI) (m = 0, 1).

Table 24.5 shows the interrupt sources and priority. The interrupt sources can be enabled or disabled using the IC0IE, IC1IE, OC0IE, OC1IE, and CMWIE bits in CMWCR and are separately issued to the interrupt controller.

Table 24.5 CMTW interrupt sources

Interrupt source	Interrupt	Interrupt enable bit	CPU0 GIC request	CPU1 GIC request	DMAC activation	Priority
CMTWm_CMWI	Interrupt caused by compare match	CMWIE	Possible	Possible	Possible	High
CMTWm_IC0I	Interrupt caused by input capture 0	IC0IE	Possible	Possible	Possible	↑
CMTWm_IC1I	Interrupt caused by input capture 1	IC1IE	Possible	Possible	Possible	↑
CMTWm_OC0I	Interrupt caused by output compare 0	OC0IE	Possible	Possible	Possible	↑
CMTWm_OC1I	Interrupt caused by output compare 1	OC1IE	Possible	Possible	Possible	Low

24.5.2 Timing of Compare Match Interrupt Generation

When the values of the CMWCNT counter and CMWCOR register match, a compare match interrupt (CMTWm_CMWI (m = 0, 1)) is generated. The compare match signal is generated at the end of the cycle where the values matched (i.e. when the CMWCNT counter is updated from the matching counter value). The compare match signal, therefore, is not generated until a further cycle of the input clock (PCLKL/8, PCLKL/32, PCLKL/128, or PCLKL/512) for the CMWCNT counter arrives after the values of the CMWCNT counter and CMWCOR register have matched. Figure 24.12 shows the timing of compare match interrupt generation.

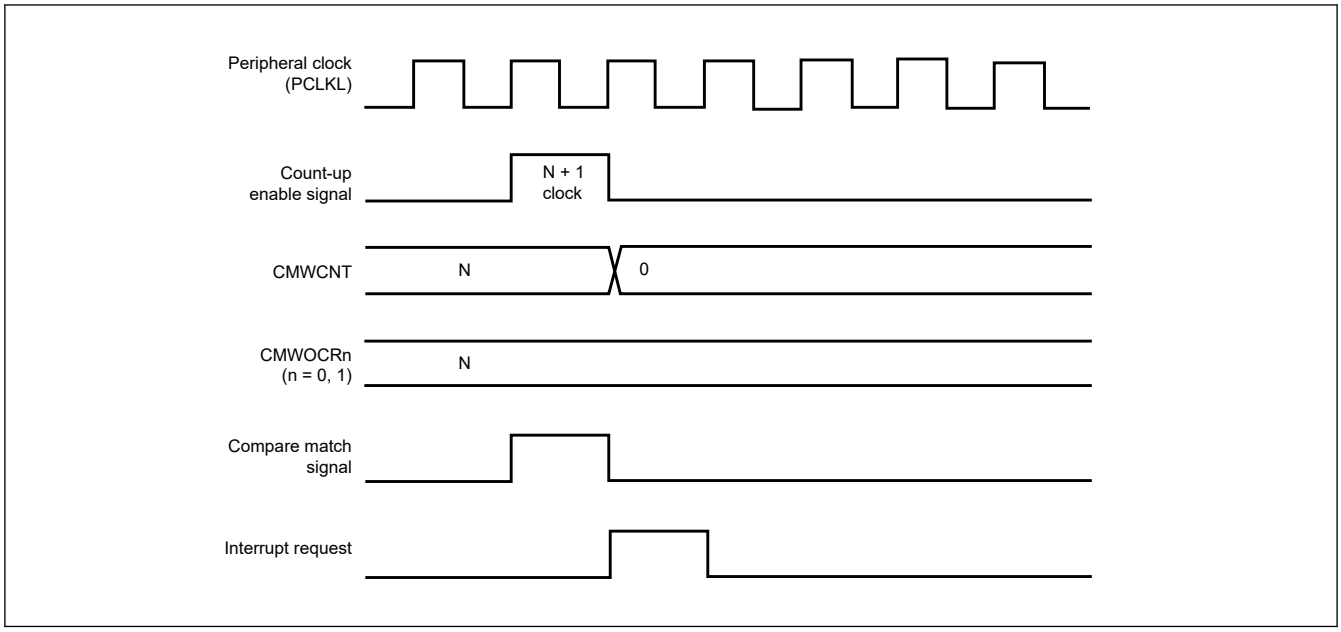


Figure 24.12 Timing of compare match interrupt generation

24.5.3 Timing of Output Compare Interrupt Generation

Figure 24.13 shows the timing of output compare interrupt generation.

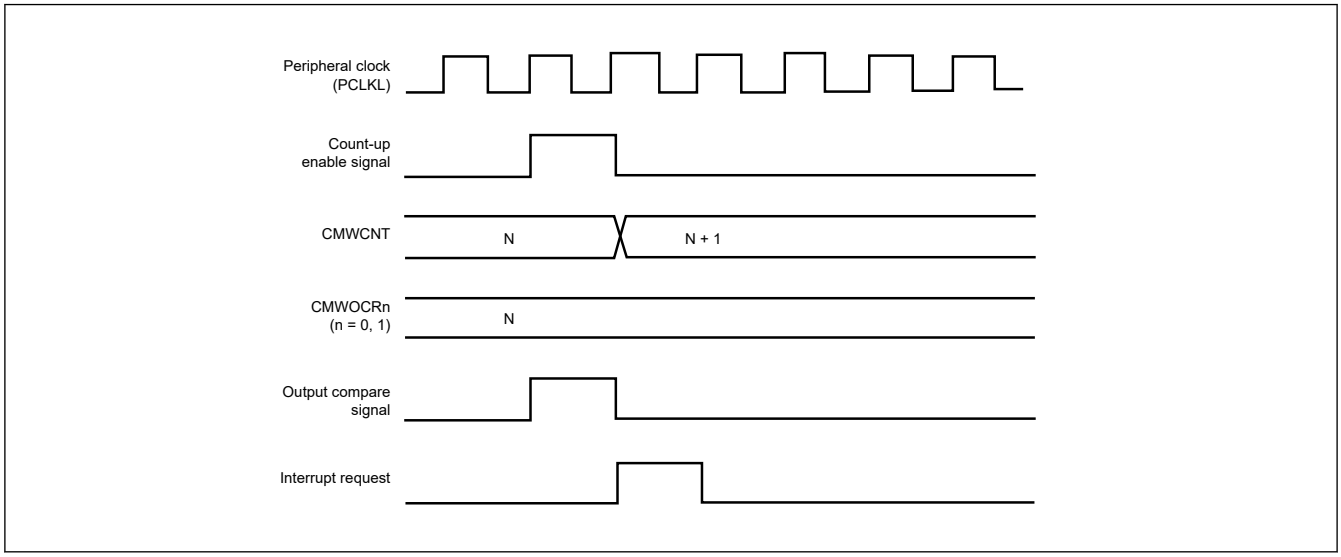


Figure 24.13 Timing of output compare interrupt generation

24.5.4 Timing of Input Capture Interrupt Generation

Figure 24.14 shows the timing of input capture interrupt generation.

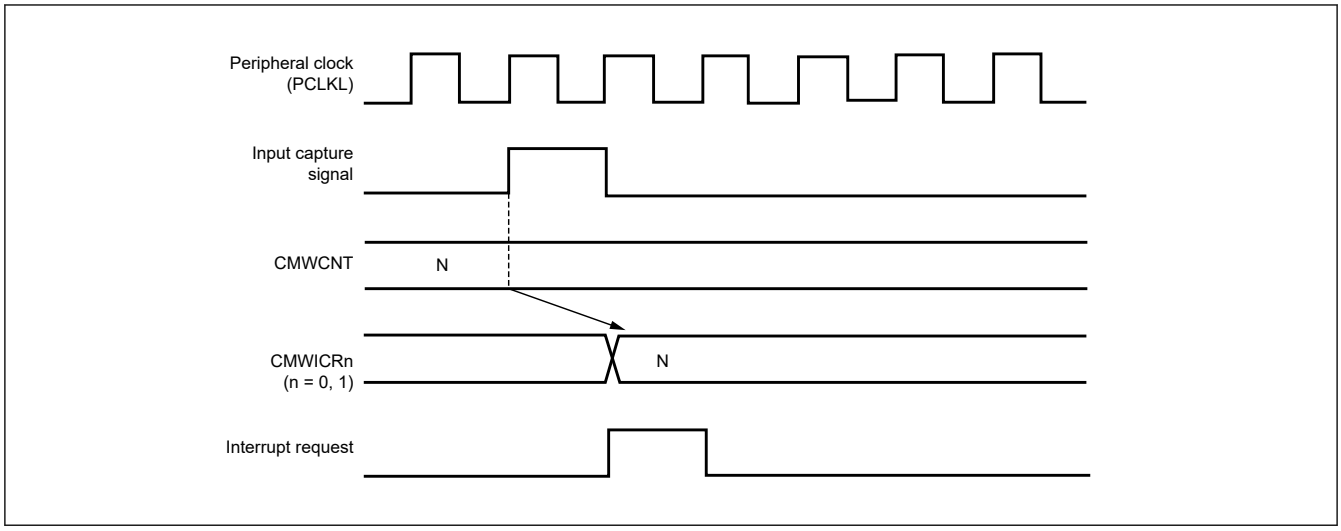


Figure 24.14 Timing of input capture interrupt generation

24.6 Usage Notes

24.6.1 Module-Stop Function

The CMTW operation can be enabled and disabled using the Module-Stop Control Register (MSTPCRD). With the initial value, the CMTW is in module-stop state. Register access is enabled by release from the module-stop state. For details, see section 9, Low-Power Consumption Function.

24.6.2 Contention Between CMWCNT Counter Writing and Compare Match

If the compare match signal is generated during CPU writing to the CMWCNT counter, the compare match interrupt request is output but the counter is not cleared since the CPU writing to the counter is given priority.

Figure 24.15 shows the timing of contention between CMWCNT counter writing and compare match.

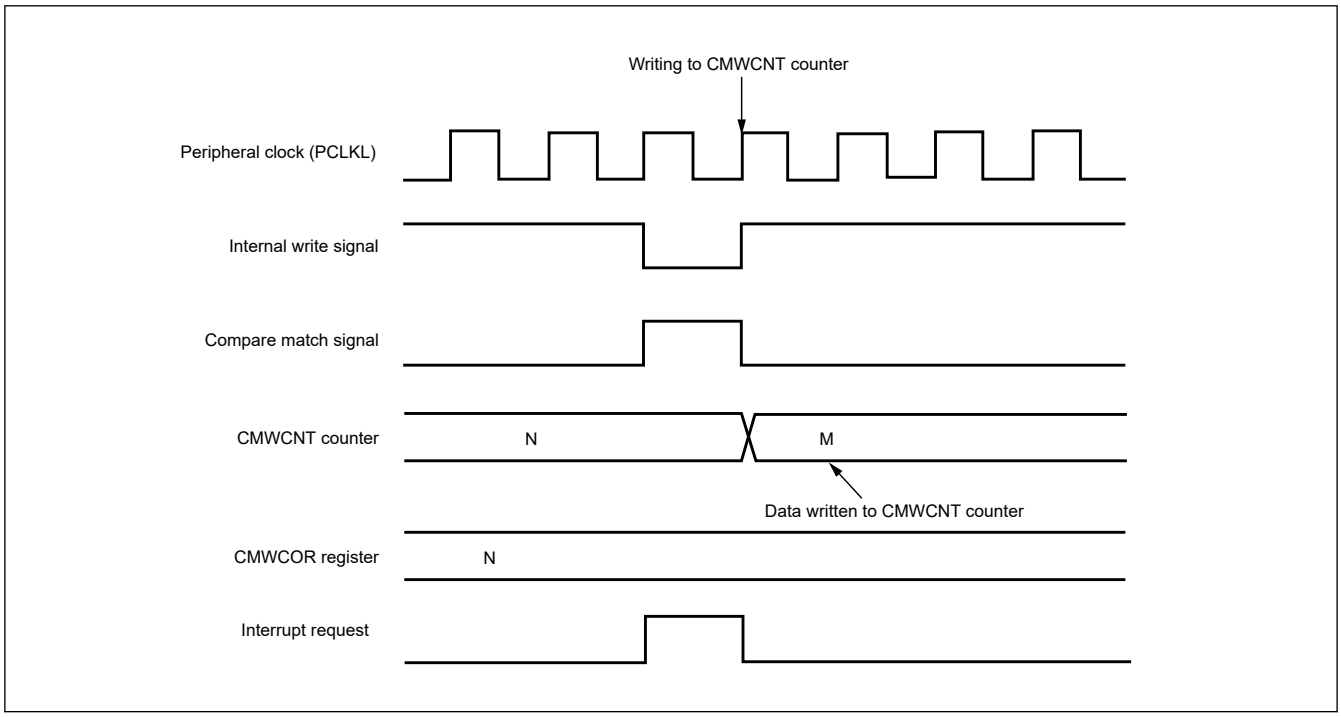


Figure 24.15 Contention between CMWCNT counter writing and compare match

24.6.3 Contention Between CMWCNT Counter Writing and Incrementing or Clearing

In case of contention between incrementation or clearing of the CMWCNT counter and CPU writing to the counter, the counter is not actually incremented or cleared since the CPU writing to the CMWCNT counter is given priority.

Figure 24.16 shows the timing in the case of contention between writing to the CMWCNT counter and incrementation or clearing.

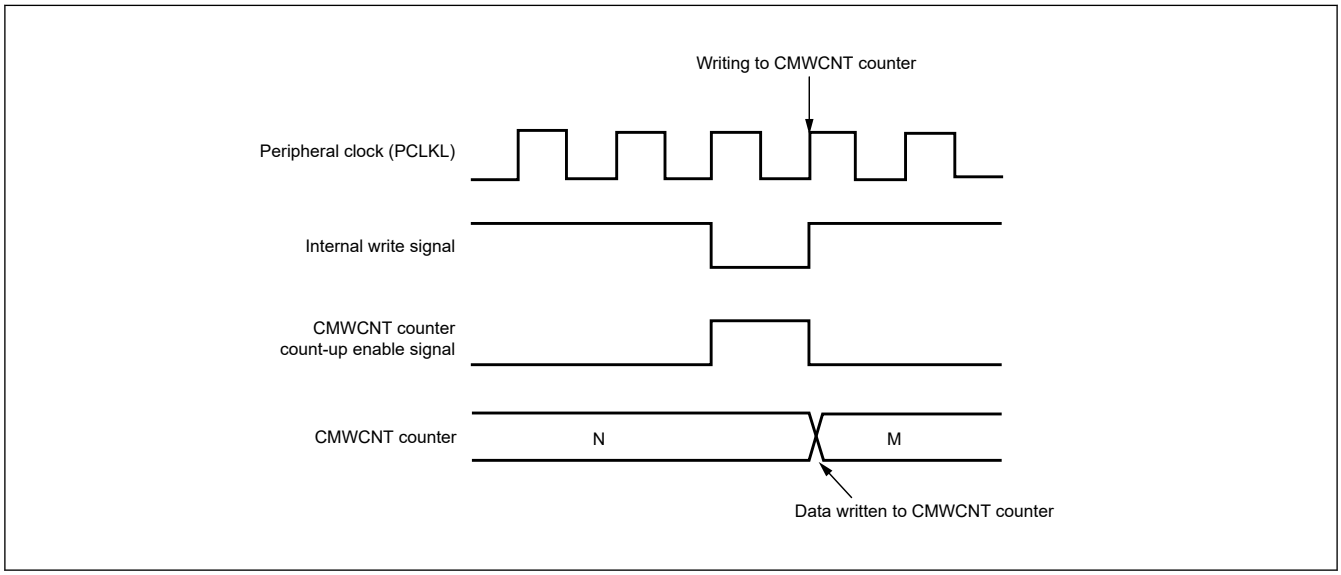


Figure 24.16 Contention between CMWCNT counter writing and incrementing

24.6.4 Contention Between CMWCOR Register Writing and Compare Match

If the compare match is generated during CPU writing to the CMWCOR register, the CPU writing to the CMWCOR register proceeds and also the compare match signal is output.

Figure 24.17 shows the timing of contention between CMWCOR register writing and compare match.

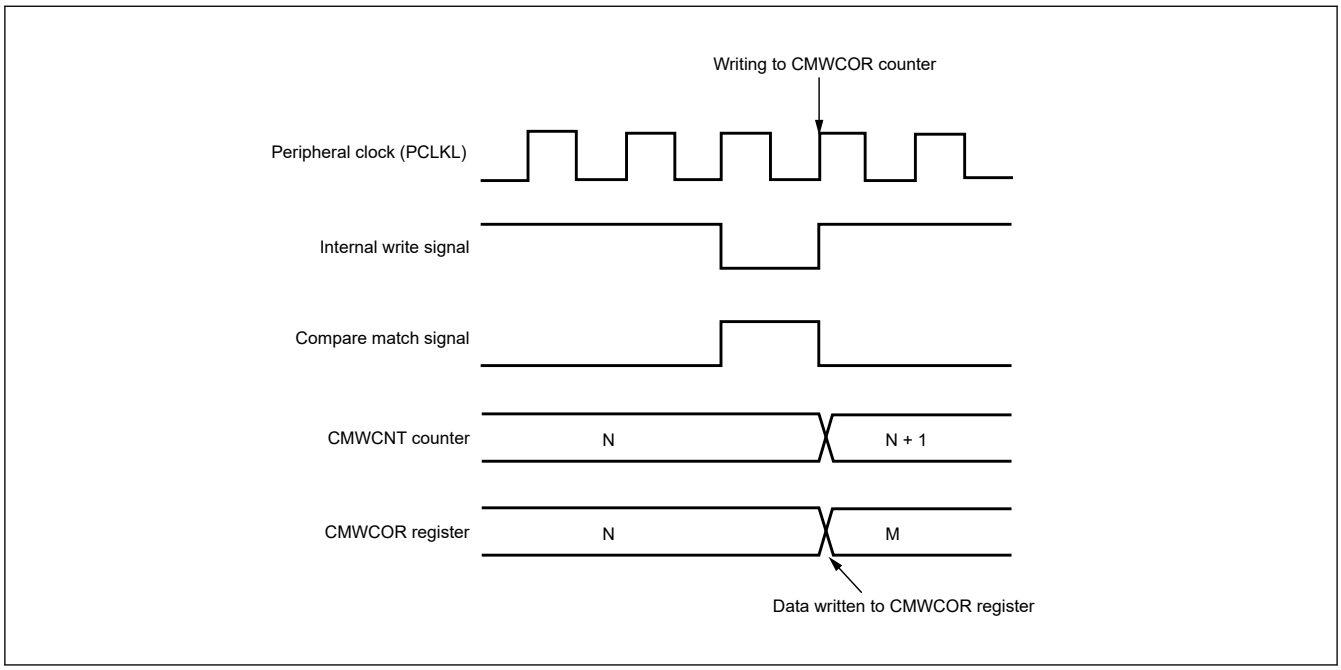


Figure 24.17 Contention between CMWCOR register writing and compare match

24.6.5 Contention Between CMWOCRn Register Writing and Compare Match

If the compare match is generated during CPU writing to the CMWOCRn register, the CPU writing to the CMWOCRn register proceeds and also the compare match signal is output (n = 0, 1).

Figure 24.18 shows the timing of contention between CMWOCRn register writing and compare match.

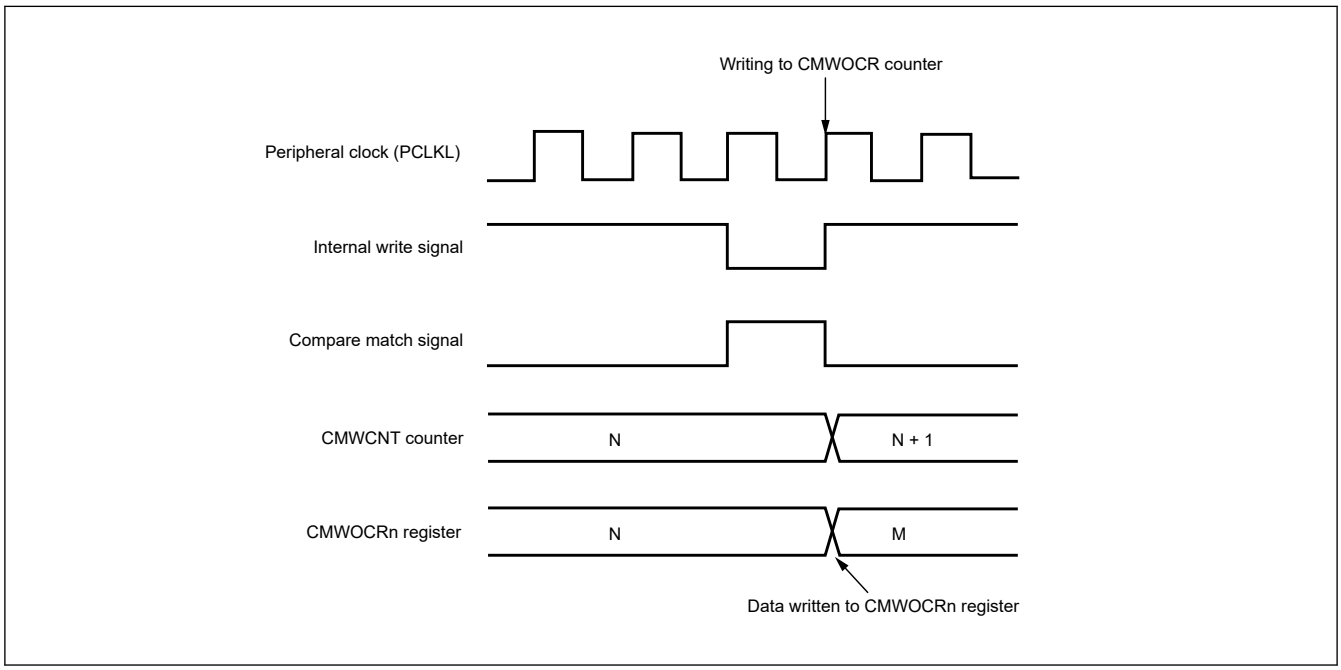


Figure 24.18 Contention between CMWOCRn register writing and compare match

24.6.6 Contention Between CMWCNT Counter Reading and Incrementing or Clearing

If the CMWCNT counter incrementing or clearing process occurs at the same time that the data of the CMWCNT counter is read, the value having been in the CMWCNT counter before incremented or cleared is read.

Figure 24.19 shows the timing of contention between the CMWCNT counter reading and incrementing.

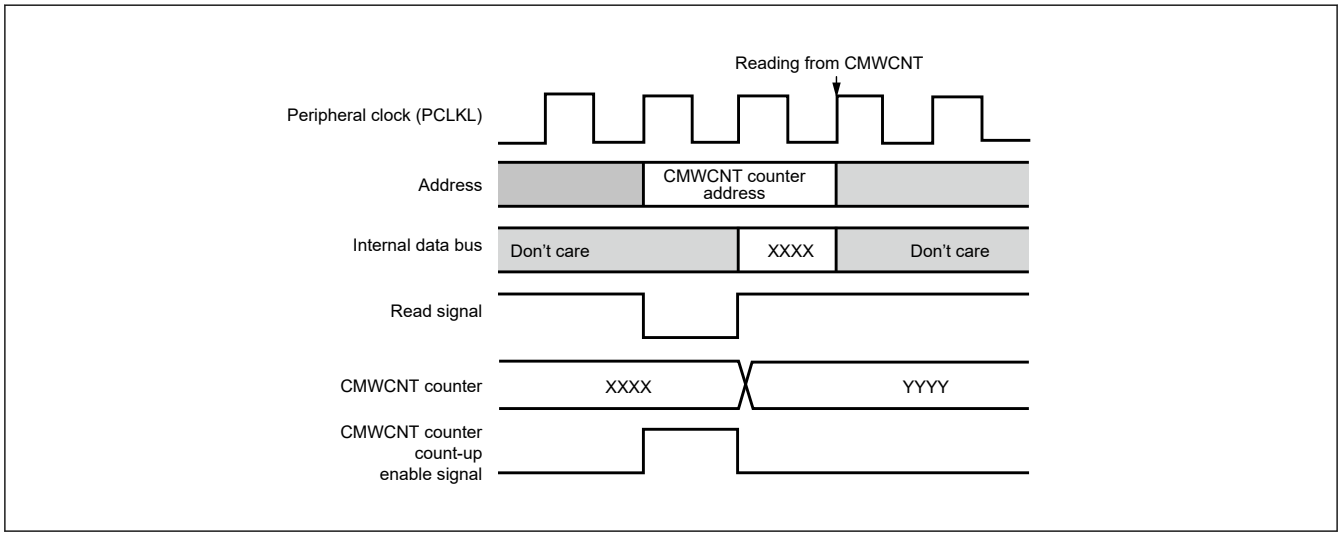


Figure 24.19 Contention between CMWCNT counter reading and incrementing (when the data reading and incrementing process occur simultaneously)

24.6.7 Contention Between CMWICRn Register Reading and Input Capture

If the input capture signal is generated at the same time that the data of CMWICRn register is read, the value having been in CMWICRn register before updated by input capture transfer is read (n = 0, 1).

Figure 24.20 shows the timing of contention between CMWICRn register reading and input capture.

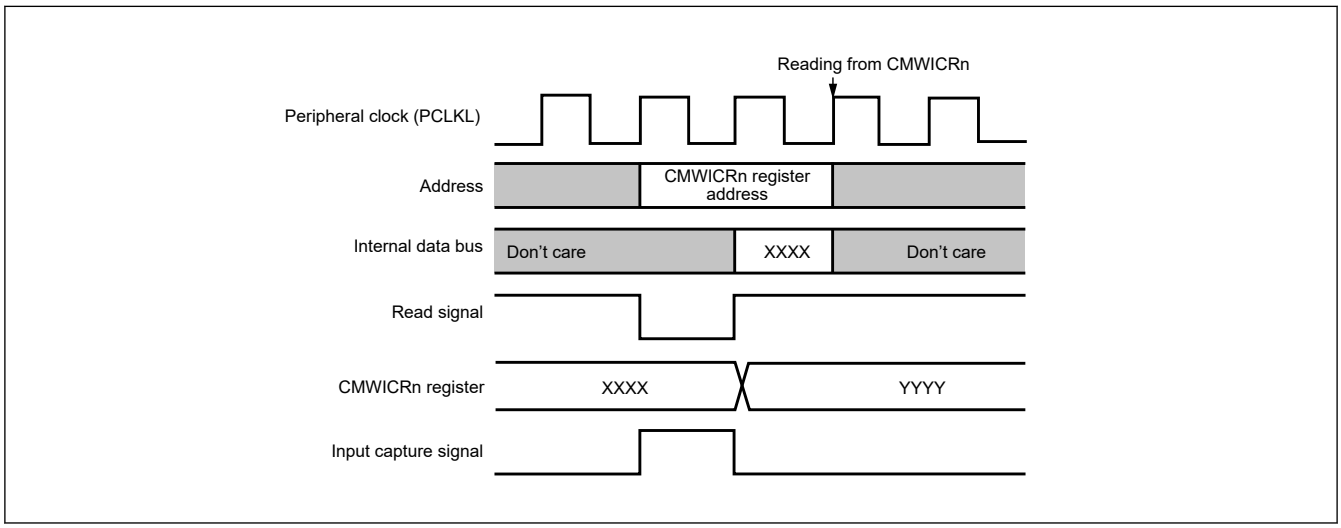


Figure 24.20 Contention between CMWICRn register reading and input capture (when the input capture signal and read signal are generated simultaneously)

Table 24.6 Summary of contention between operations due to the access to registers and changes to the counter's state

Register access	CMWCNT state	CMWICR0/1 state	Operation to be performed
Writing to CMWCNT	Compare match	—	Output of compare match interrupt request / Writing to CMWCNT
	Counting up	—	Writing to CMWCNT
Writing to CMWCOR	Compare match	—	Compare match
Writing to CMWOCR0	Output compare 0	—	Output compare 0
Writing to CMWOCR1	Output compare 1	—	Output compare 1
Reading from CMWCNT	Counting up	—	Counting up and reading of the previous value
Reading from CMWICR0	—	Input capture 0	Input capture 0 and reading of the value before transfer
Reading from CMWICR1	—	Input capture 1	Input capture 1 and reading of the value before transfer

25. Watchdog Timer (WDT)

25.1 Overview

The watchdog timer (WDT) is a 14-bit down-counter and can be used to inform error events to ICU when the counter underflows because the system has run out of control and is unable to refresh the WDT. The refresh-permitted period can be set to refresh the counter and to detect when the system runs out of control.

Table 25.1 lists the specifications of the WDT and Figure 25.1 shows a block diagram of the WDT.

Table 25.1 WDT specifications

Parameter	Specifications
Number of units	6 units (2 units for Cortex-R52 and 4 units for Cortex-A55)
Count source	Peripheral clock (PCLKL)
Clock division ratio	Divide by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	Register start mode: Counting is started by refresh on a write to the WDTRR register
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error occurs • Cortex-R52 and Cortex-A55 debug state
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
WDT error events	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the WDTSR register.
Output signal (internal signal)	Interrupt request output

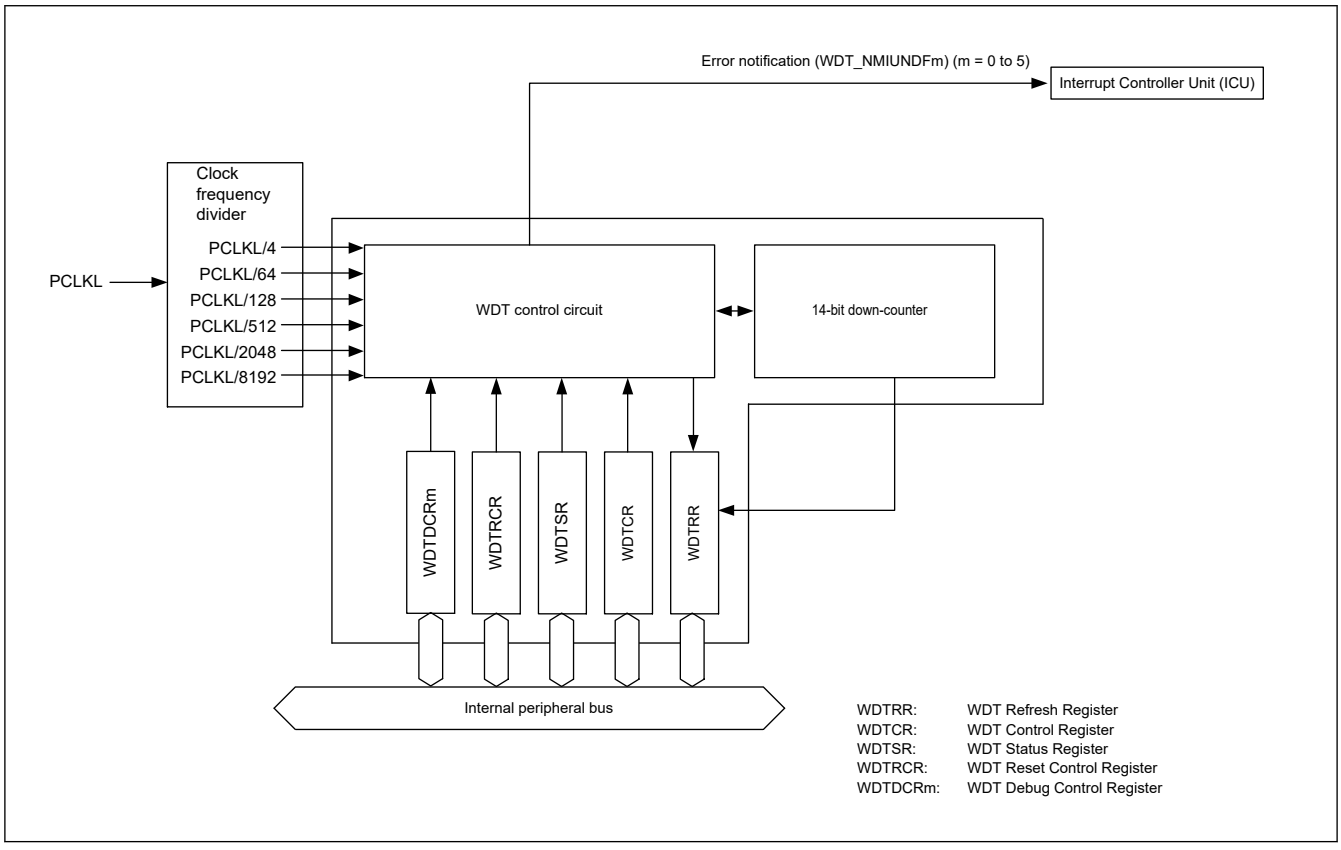


Figure 25.1 WDT block diagram

Table 25.2 WDT interrupt sources

Name	Interrupt sources
WDT_NMIUNDF0	Down-counter underflow/Refresh error for Cortex-R52 CPU0
WDT_NMIUNDF1	Down-counter underflow/Refresh error for Cortex-R52 CPU1
WDT_NMIUNDF2	Down-counter underflow/Refresh error for Cortex-A55 Core0
WDT_NMIUNDF3	Down-counter underflow/Refresh error for Cortex-A55 Core1
WDT_NMIUNDF4	Down-counter underflow/Refresh error for Cortex-A55 Core2
WDT_NMIUNDF5	Down-counter underflow/Refresh error for Cortex-A55 Core3

25.2 Register Map

Table 25.3 WDT register map

Address	Register symbol	Register name	Write protection
0x8008_2000 + 0x0400 × m	WDTRR	WDT Refresh Register	—
0x8008_2002 + 0x0400 × m	WDTCR	WDT Control Register	—
0x8008_2004 + 0x0400 × m	WDTSR	WDT Status Register	—
0x8008_2006 + 0x0400 × m	WDTRCR	WDT Reset Control Register	—
0x8129_5100 + 0x0004 × m	WDTDCRm	WDT Debug Control Register	—

Table 25.4 WDT related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0 (for Cortex-R52 CPU0)	SWRCPU0	—	SLVACCCTL0.WDT0_SL
1 (for Cortex-R52 CPU1)	SWRCPU1	—	SLVACCCTL0.WDT1_SL
2 (for Cortex-A55 Core0)	SWR550	—	SLVACCCTL0.WDT2_SL
3 (for Cortex-A55 Core1)	SWR551	—	SLVACCCTL0.WDT3_SL
4 (for Cortex-A55 Core2)	SWR552	—	SLVACCCTL0.WDT4_SL
5 (for Cortex-A55 Core3)	SWR553	—	SLVACCCTL0.WDT5_SL

25.3 Register Descriptions

25.3.1 WDTRR : WDT Refresh Register

Base address: WDTm = 0x8008_2000 + 0x0400 × m (m = 0 to 5)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	n/a	The down-counter is refreshed by writing 0x00 and then 0xFF to this register.	R/W

WDTRR register refreshes the down-counter of the WDT.

The down-counter of the WDT is refreshed by writing 0x00 and then 0xFF to WDTRR (refresh operation) within the refresh-permitted period. After being refreshed, the down-counter starts counting down from the value specified with the WDTCR.TOPS[1:0] bits.

When 0x00 is written, the read value is 0x00, when a value other than 0x00 is written, the read value is 0xFF. For details of the refresh operation, see section 25.4.3. Refresh Operation.

25.3.2 WDTCR : WDT Control Register

Base address: WDTm = 0x8008_2000 + 0x0400 × m (m = 0 to 5)

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Function	R/W
1:0	TOPS[1:0]	Timeout Period Selection 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	CKS[3:0]	Clock Division Ratio Selection 0x1: PCLKL/4 0x4: PCLKL/64 0xF: PCLKL/128 0x6: PCLKL/512 0x7: PCLKL/2048 0x8: PCLKL/8192 Others: Setting prohibited.	R/W
9:8	RPES[1:0]	Window End Position Selection 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RPSS[1:0]	Window Start Position Selection 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

The WDTCR register allows the selecting of a timeout period before the down-counter underflows, a clock division ratio, and a window start/end position for refreshing. There are some restrictions for writing to this register. For details, see section 25.4.2. Control over Writing to the WDTCR and WDTRCR Registers.

TOPS[1:0] bits (Timeout Period Selection)

These bits select a timeout period (period before the down-counter underflows) from among 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of PCLKL cycles) before the counter underflows.

Table 25.5 lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKL cycles.

CKS[3:0] bits (Clock Division Ratio Selection)

These bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from among the peripheral module clock (PCLKL) divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, a count period between 4,096 and 134,217,728 cycles of the PCLKL clock can be selected for the WDT.

RPES[1:0] bits (Window End Position Selection)

These bits allow selection of the window end position of the down counter from among 75%, 50%, 25%, and 0% of the counting period. The selected window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is valid.

The counter values of the window start and end positions specified by the RPES[1:0] and RPSS[1:0] bit settings vary according to the TOPS[1:0] bit setting.

Table 25.6 lists the counter values of the window start and end positions corresponding to the TOPS[1:0] bit setting.

RPSS[1:0] bits (Window Start Position Selection)

These bits allow the selection of the window start position of the down counter from among 100%, 75%, 50%, and 25% of the counting period (assuming that the count start position as 100% and underflow occurrence as 0%). The period from the window start position to the window end position is the refresh-permitted period, any other periods are the refresh-prohibited periods.

Table 25.6 lists the counter values for the window start and end positions and Figure 25.2 shows the relationship between the settings of the RPSS[1:0] and RPES[1:0] bits, including the refresh-permitted and refresh-prohibited periods.

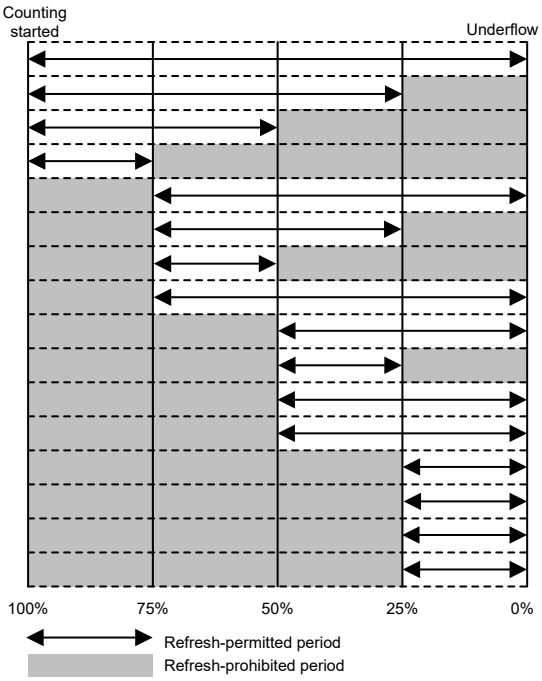
Table 25.5 Timeout period settings

CKS[3:0] bits				TOPS[1:0] bits		Clock division ratio	Timeout period (number of cycles)	No. of Peripheral clock (PCLKL) cycles @Count clock = PCLKL
b7	b6	b5	b4	b1	b0			
0	0	0	1	0	0	Count clock/4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	Count clock/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	Count clock/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	Count clock/512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	Count clock/2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	Count clock/8192	1024	8388608
				0	1		4096	33554432
				1	0		8192	67108864
				1	1		16384	134217728

Table 25.6 Relationship between timeout period and window start and end counter values

TOPS[1:0] bits		Timeout period		Window start and end counter value			
b1	b0	Cycles	Counter value	100%	75%	50%	25%
0	0	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
0	1	4096	0x0FFF	0x0FFF	0x0BFF	0x07FF	0x03FF
1	0	8192	0x1FFF	0x1FFF	0x17FF	0x0FFF	0x07FF
1	1	16384	0x3FFF	0x3FFF	0x2FFF	0x1FFF	0x0FFF

RPSS[1:0] bits		RPES[1:0] bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75



Note: If window end setting ≥ window start setting, the window end setting is set to 0%.

Figure 25.2 RPSS[1:0] and RPES[1:0] bit settings and the refresh-permitted period

25.3.3 WDTSR : WDT Status Register

Base address: WDTm = 0x8008_2000 + 0x0400 × m (m = 0 to 5)

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	REFEF	UNDF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-Counter Value Value counted by the down-counter	R
14	UNDFF	Underflow Flag 0: No underflow occurred 1: Underflow occurred	R/W*1

Bit	Symbol	Function	R/W
15	REFEF	Refresh Error Flag 0: No refresh error occurred 1: Refresh error occurred	R/W*1

Note 1. Only 0 can be written to clear the flag.

The WDTSR register displays the counter value of the down-counter and the states of underflow and refresh error occurrences.

CNTVAL[13:0] bits (Down-Counter Value)

Read these bits to confirm the value of the down-counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF flag (Underflow Flag)

Read this flag to confirm whether an underflow has occurred in the down-counter.

The value 1 indicates that the down-counter has underflowed. The value 0 indicates that the down-counter has not underflowed.

Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N + 1) PCLKL cycles. In addition, clearing of the flag is ignored for (N + 1) PCLKL cycles after an underflow. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0x1, N = 4
- When WDTCR.CKS[3:0] = 0x4, N = 64
- When WDTCR.CKS[3:0] = 0xF, N = 128
- When WDTCR.CKS[3:0] = 0x7, N = 2048
- When WDTCR.CKS[3:0] = 0x8, N = 8192

REFEF flag (Refresh Error Flag)

Read this flag to confirm whether a refresh error (performing a refresh operation during a refresh-prohibited period) has occurred.

The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.

Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N + 1) PCLKL cycles. In addition, clearing of the flag is ignored for (N + 1) PCLKL cycles after a refresh error. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0x1, N = 4
- When WDTCR.CKS[3:0] = 0x4, N = 64
- When WDTCR.CKS[3:0] = 0xF, N = 128
- When WDTCR.CKS[3:0] = 0x6, N = 512
- When WDTCR.CKS[3:0] = 0x7, N = 2048
- When WDTCR.CKS[3:0] = 0x8, N = 8192

25.3.4 WDTRCR : WDT Reset Control Register

Base address: WDTm = 0x8008_2000 + 0x0400 × m (m = 0 to 5)

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RSTIRQS	—						
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	RSTIRQS	Reset Interrupt Request Selection 0: Error notification to ICU is permitted 1: Error notification to ICU is not performed	R/W

The WDTRCR register controls whether to send an error notification when underflow occurs in the down-counter of WDT.

There are some restrictions for writing to the WDTRCR register. For details, see section 25.4.2. Control over Writing to the WDTCR and WDTRCR Registers.

RSTIRQS bit (Reset Interrupt Request Selection)

The RSTIRQS bit specifies whether an error notification should be sent to the ICU module when an underflow or refresh error occurs in the down-counter. Set this bit to 0 to enable error notification from the WDT module.

25.3.5 WDTDCRm : WDT Debug Control Register m (m = 0 to 5)

Base address: WDT_DBG = 0x8129_5100

Offset address: 0x00 + 0x04 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTSTOPMASK
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTSTOPCTRL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WDTSTOPCTRL	WDT Count Stop Control 0: WDT count is enabled, and interrupt is enabled. 1: WDT count is stopped, and interrupt is disabled.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
16	WDTSTOPMASK	WDT Debug Break Mask 0: haltdbg is not masked 1: haltdbg is masked	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The WDTDCRm is used to stop WDT count and to mask haltdbg from CoreSight.

25.4 Operation

25.4.1 Count Operation in Start Mode

Counting starts by a refresh operation of the WDT Refresh Register (WDTRR) when the WDT Control Register (WDTCR) and WDT Reset Control Register (WDTRCR) are set.

25.4.1.1 Register Setting

After the reset state is released, set the clock division ratio, window start and end positions, and timeout period in the WDTCR register. The value specified by the WDTCR.TOPS[1:0] bits is then set in the down-counter by a refresh operation to start counting down.

Thereafter, the value in the counter is reset at each refresh operation and count-down continues if the program runs normally and the counter is refreshed in the refresh-permitted period.

Figure 25.3 shows an example of operation under the following conditions:

- Reset Interrupt Request bit (RSTIRQS): 0b (error notification to ICU is permitted)
- Window Start Position Selection bits (RPSS[1:0]): 10b (75%)
- Window End Position Selection bits (RPES[1:0]): 10b (25%)

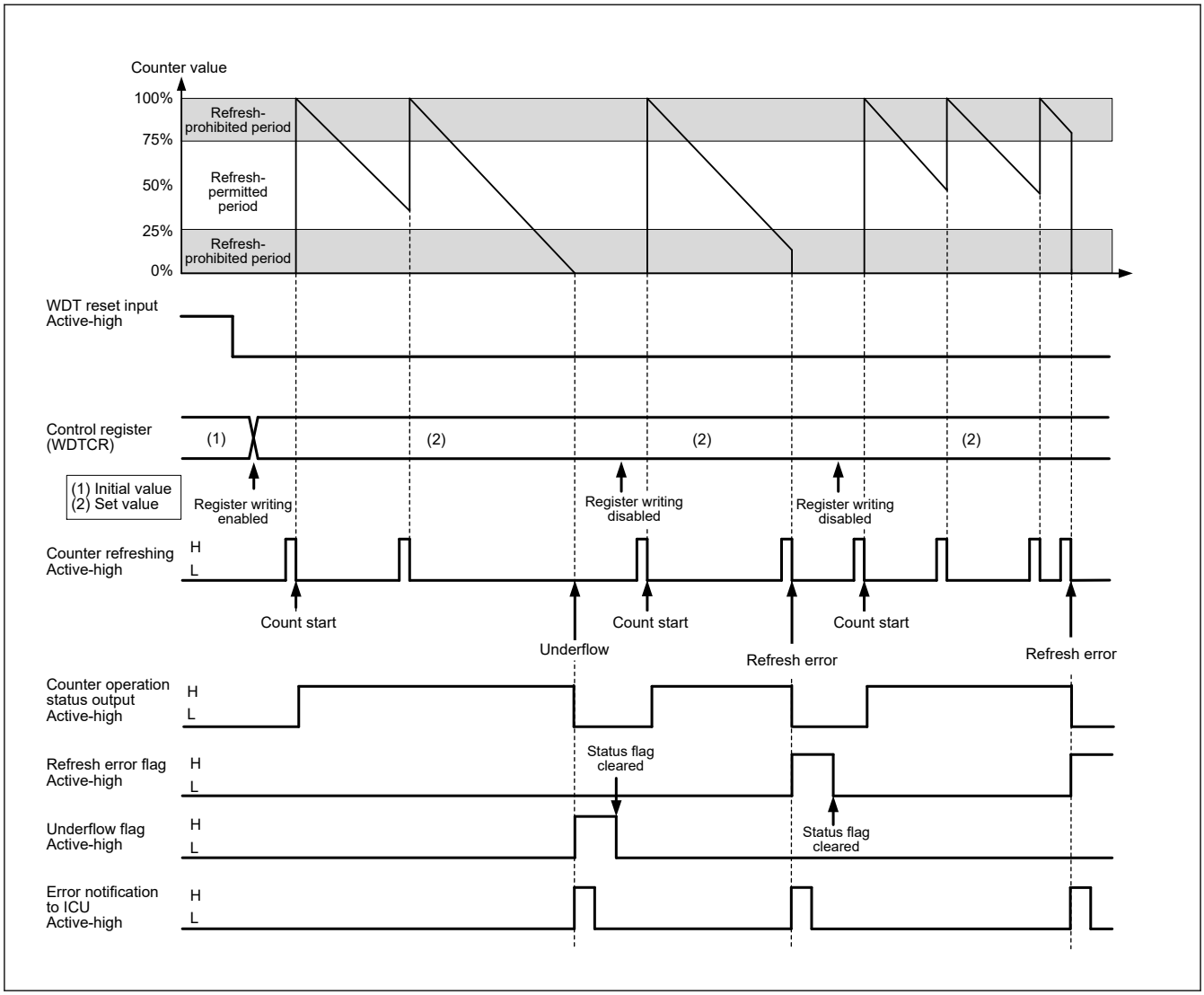


Figure 25.3 Operation example in register start mode

25.4.2 Control over Writing to the WDTCR and WDTRCR Registers

Writing to the WDT Control Register (WDTCR) is only possible once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or by writing to the WDTCR register, the protection signal in the WDT becomes 1 to protect the WDTCR register against subsequent attempts at writing.

Writing to the WDT Reset Control Register (WDTRCR) is also controlled similarly.

This protection is released by the reset source of the WDT. Any other reset sources cannot release the protection. Figure 25.4 shows control waveforms produced in response to writing to the WDTCR register.

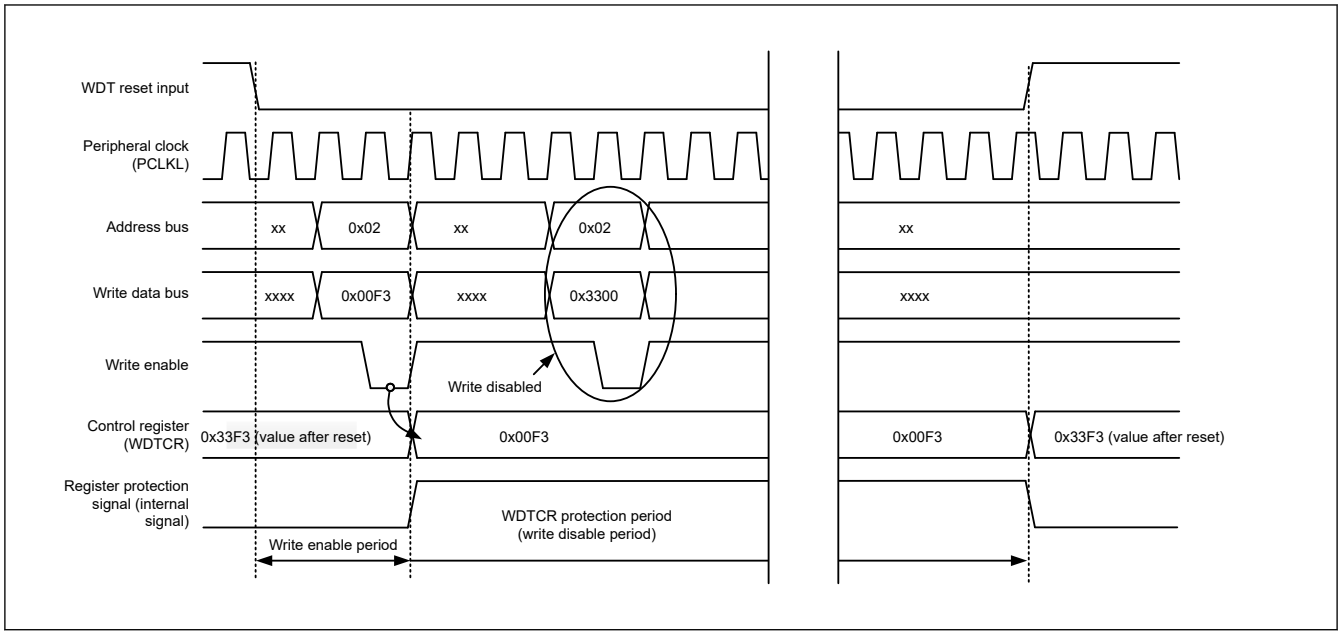


Figure 25.4 Control waveforms produced in response to writing to the WDTCR register

25.4.3 Refresh Operation

To refresh the down-counter and to start the down-counter operation (count start due to refreshing), write the values 0x00 and then 0xFF to the WDT Refresh Register (WDTRR). If a value other than 0xFF is written after 0x00, the down-counter is not refreshed. To perform refreshing after an invalid write, write 0x00 and then 0xFF again to the WDTRR register.

If 0x00 is written twice in succession, writing 0xFF after the second 0x00 refreshes the down-counter because the 0x00 → 0xFF condition is satisfied. The write sequence of 0x00 (n-1th) → 0x00 (nth) → 0xFF also satisfies the refreshing condition and thus refreshes the down-counter.

Even if a register other than WDTRR is accessed or WDTRR is read between writing 0x00 and writing 0xFF to WDTRR, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 0x00 → 0xFF
- 0x00 (n - 1th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from the WDTRR register → 0xFF

[Sample sequences of writing that are not valid for refreshing the counter]

- 0x23 (a value other than 0x00) → 0xFF
- 0x00 → 0x54 (a value other than 0xFF)
- 0x00→ 0xAA (0x00 and a value other than 0xFF) → 0xFF

If 0xFF is written to the WDTRR register during the refresh-permitted period after 0x00 is written outside of the period, write operation is acknowledged and down-counter is refreshed. Whether writing is made within the refresh-permitted period is determined by when 0xFF is written.

After 0xFF is written to the WDT Refresh Register (WDTRR), refreshing the down-counter requires up to 4 counting cycles. The number of peripheral clock (PCLKL) cycles in a single counting cycle differs depending on the setting of the Clock Division Ratio Selection bits (WDTCR.CKS[3:0]).

Therefore, writing 0xFF to the WDTRR register should be completed within 4 count cycles before the refresh-permitted period end position or before the down-counter underflows. Confirm the down-counter value with the down-counter value bits (WDTSR.CNTVAL[13:0]).

[Sample of refresh operation timing]

- When the window start position is 0x1FFF, if 0x00 is written to the WDTRR register before 0x1FFF (for example, 0x2002), the down-counter can be refreshed by writing 0xFF to the WDTRR register after the value of WDTSR.CNTVAL[13:0] becomes 0x1FFF.
- When the window end position is 0x1FFF, if the value of WDTSR.CNTVAL[13:0] is 0x2003 (4 counts before 0x1FFF) or more immediately after 0x00 and 0xFF are written to the WDTRR register, the down-counter will be refreshed.
- If the refresh-permitted period goes down to 0x0000, refreshing is possible until just before underflow. In this case, if the value of WDTSR.CNTVAL[13:0] is 0x0003 (4 count cycles before underflow) or more immediately after 0x00 and 0xFF are written to the WDTRR register, underflow does not occur and the down-counter is refreshed.

Figure 25.5 shows WDT refresh-operation waveforms when the clock division ratio is PCLKL/64.

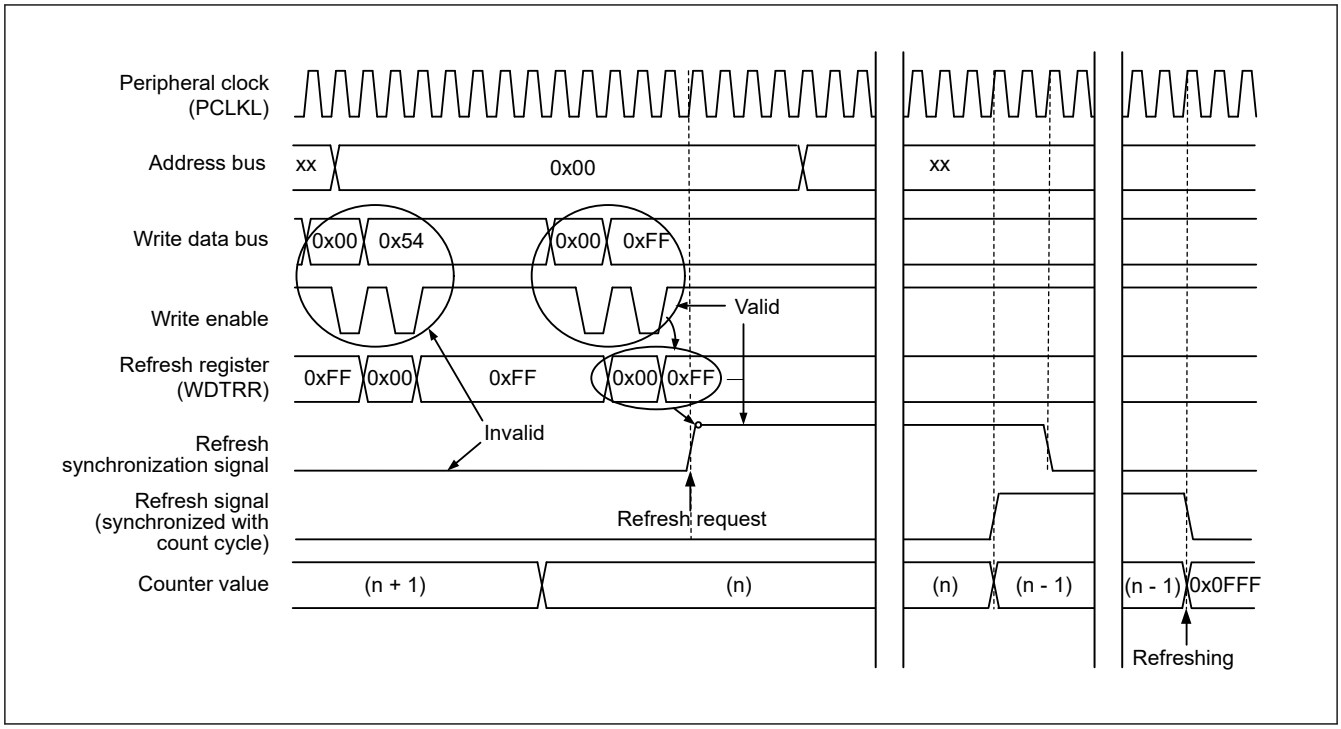


Figure 25.5 WDT refresh operation waveforms (WDTCR.CKS[3:0] = 0x4, WDTCR.TOPS[1:0] = 01b)

25.4.4 Status Flag

The Refresh Error flag (WDTSR.REFEF) and the Underflow flag (WDTSR.UNDFF) retain the error causes after error notifications are output to the Interrupt Controller (ICU).

Occurrence state of error notifications to ICU can be confirmed by reading the WDTSR.REFEF flag or the WDTSR.UNDFF flag after reset is released or on occurrence of an error notification to ICU.

To clear these flags, write 0. Writing 1 is ignored.

If these flags are not cleared, there is no effect on the operation. On occurrence of the next error notification to ICU, the previous error notification is automatically cleared and a new notification to ICU is written.

25.4.5 Count Stop Control

WDT count can be stopped by debug event (haltdbg) from CoreSight through TSGCNT HALTREQ or RESTARTREQ to prevent unintentional WDT overflow during debug operation, or by WDTSTOPCTRL bit in WDTDCRm register during normal operation. Assert or de-assert of haltdbg needs to be controlled by connecting stop/re-start event of CPU (e.g. cross halt trigger event etc.) to TSGCNT HALTREQ or RESTARTREQ via CTI. Stop control by haltdbg is maskable by setting WDTSTOPMASK bit in WDTDCRm register. When setting 1 to WDTSTOPMASK bit in WDTDCRm register, WDT continues to count even during debug operation.

26. Real Time Clock (RTC)

26.1 Overview

The RTC module can count a 100-year calendar from 2000 to 2099 and automatically adjust dates for leap years. The RTC uses an internal clock acquired by dividing the main clock oscillator or PLL output by a prescaler. Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second is counted. No external oscillator is required but battery backup is not supported.

Table 26.1 lists the RTC specifications and Figure 26.1 shows a block diagram. Table 26.2 lists the RTC I/O pin and Table 26.3 lists the interrupt sources.

Table 26.1 RTC specifications

Parameter	Specifications
Number of units	1 unit
Count mode	Calendar count mode
Count source	Main clock oscillator divided by 128 (195.3125 kHz)
Clock and calendar functions	• Year, month, day of the week, day, hour, minute, second counters and sub-counters, counting up to 99 years are supported • Year, month, day of the week, day, hour, minute, and second counters: BCD code display • 12 hours/24 hours mode switching function • Automatic adjustment function for leap years • Start/stop function • 1 Hz pulse output
Interrupts	• Alarm interrupt (RTC_ALM) As an alarm interrupt condition, day-of-week, hour and minute can be selected • 1 second interrupt (RTC_1S) Interrupt can be generated every 1 second • Fixed interval interrupt (RTC_PRD) 0.25 second, 0.5 second, 1 second, 1 minute, 1 hour, 1 day, or 1 month can be selected as an interrupt period
Module-stop function	Module-stop state can be set to reduce power consumption.

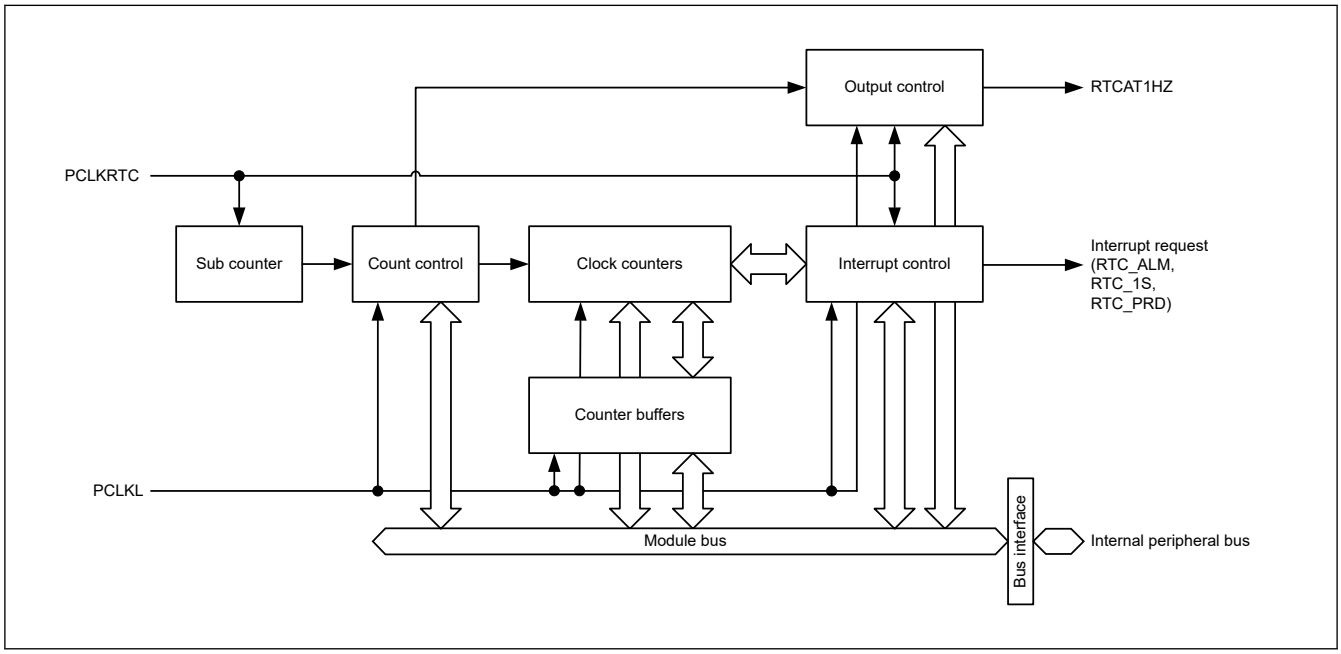


Figure 26.1 RTC block diagram

Table 26.2 RTC I/O pins

Pin name	I/O	Function
RTCAT1HZ	Output	RTC 1 Hz output

Table 26.3 RTC interrupt sources

Name	Interrupt sources
RTC_ALM	Alarm interrupt
RTC_1S	1 second interrupt
RTC_PRD	Fixed interval interrupt

26.2 Register Map

Table 26.4 RTC register map

Address	Register symbol	Register name	Write protect cancel
0x8100_9000	RTCA0CTL0	RTC Control Register 0	—
0x8100_9004	RTCA0CTL1	RTC Control Register 1	—
0x8100_9008	RTCA0CTL2	RTC Control Register 2	—
0x8100_900C	RTCA0SUBC	RTC Sub Count Register	—
0x8100_9010	RTCA0SRBU	RTC Sub Count Register Read Buffer	—
0x8100_9014	RTCA0SEC	RTC Sec Count Buffer Register	—
0x8100_9018	RTCA0MIN	RTC Min Count Buffer Register	—
0x8100_901C	RTCA0HOUR	RTC Hour Count Buffer Register	—
0x8100_9020	RTCA0WEEK	RTC Week Count Buffer Register	—
0x8100_9024	RTCA0DAY	RTC Day Count Buffer Register	—
0x8100_9028	RTCA0MONTH	RTC Month Count Buffer Register	—
0x8100_902C	RTCA0YEAR	RTC Year Count Buffer Register	—
0x8100_9030	RTCA0TIME	RTC Time Set Register	—
0x8100_9034	RTCA0CAL	RTC Calendar Set Register	—
0x8100_903C	RTCA0SCMP	RTC Sub Count Compare Register	—
0x8100_9040	RTCA0ALM	RTC Alarm Min Set Register	—
0x8100_9044	RTCA0ALH	RTC Alarm Hour Set Register	—
0x8100_9048	RTCA0ALW	RTC Alarm Week Set Register	—
0x8100_904C	RTCA0SECC	RTC Second Count Register	—
0x8100_9050	RTCA0MINC	RTC Minute Count Register	—
0x8100_9054	RTCA0HOURC	RTC Hour Count Register	—
0x8100_9058	RTCA0WEEKC	RTC Week Count Register	—
0x8100_905C	RTCA0DAYC	RTC Day Count Register	—
0x8100_9060	RTCA0MONC	RTC Month Count Register	—
0x8100_9064	RTCA0YEARC	RTC Year Count Register	—
0x8100_9068	RTCA0TIMEC	RTC Time Count Register	—
0x8100_906C	RTCA0CALC	RTC Calendar Count Register	—

Table 26.5 RTC related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	—	MSTPCRG.MSTPCRG05	SLVACCCTL2.RTC_SL

26.3 Register Descriptions

26.3.1 RTCA0CTL0 : RTC Control Register 0

Base address: RTC = 0x8100_9000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	RTCA0CE	RTCA0CEST	RTCA0AMPM	RTCA0SLSB	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	RTCA0SLSB	RTCA0SCMP enable/disable setting Set this bit to 1, see the RTCA0SCMP register description. When setting this bit, be sure to follow the flow described in section 26.4.1.1. Initial Setting. It is prohibited to change the setting of this bit while the clock counter operation is enabled (RTCA0CE = 1). If the setting of this bit is changed during clock counter operation, the operation of RTC Controller cannot be guaranteed. When changing the setting of this bit, first perform initialization according to section 26.4.1.5. Initialization of RTC While Clock Counter Operation is Enabled, and then perform the setting of this bit again according to the flow described in section 26.4.1.1. Initial Setting.	R/W
5	RTCA0AMPM	RTCA0HOUR, RTCA0ALH display format selection bit 0: 12-hour display 1: 24-hour display	R/W
6	RTCA0CEST	RTC Controller Enable Status 0: Clock counter operation stopped status 1: Clock counter operation enabled status (the clock counter counts up)	R
7	RTCA0CE	RTC Controller Enable Bit Setting this bit to 0 initializes the internal circuit of RTC Controller other than the control registers*1. Moreover, interrupts (RTC_ALM, RTC_1S, RTC_PRD) are cleared. 0: Clock counter operation stopped 1: Clock counter operation enabled (the clock counter counts up)	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Status flags RTCA0WST, RTCA0RSST, and RTCA0WSST, which cannot be written by the user, are cleared.

26.3.2 RTCA0CTL1 : RTC Control Register 1

Base address: RTC = 0x8100_9000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	RTCA01HZE	RTCA0ALME	RTCA01SE	RTCA0CT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RTCA0CT[2:0]	Fixed interval interrupt (RTC_PRD) output setting bit See section 26.4.1.7. Changing the Setting of Fixed Interval Interrupt During Clock Counter Operation for changing the setting of this field. RTCA0CT[2]/RTCA0CT[1]/RTCA0CT[0] 0 0 0: Fixed interval interrupt output stop 0 0 1: Interrupt once every 0.25 seconds (in synchronization with second count-up) 0 1 0: Interrupt once every 0.5 seconds (in synchronization with second count-up) 0 1 1: Interrupt once every 1 second (in synchronization with second count-up) 1 0 0: Interrupt once every 1 minute (every 1 minute 00 seconds) 1 0 1: Interrupt once every 1 hour (every 1 hour 00 minutes 00 seconds) 1 1 0: Interrupt once every 1 day (every 1 day 00 hours 00 minutes 00 seconds) 1 1 1: Interrupt once every 1 month (every 1 month first day 00 hours 00 minutes 00 seconds a.m.)	R/W
3	RTCA01SE	1 second interrupt (RTC_1S) output enable bit 0: Stop interrupt output at every 1 second 1: Enable interrupt output at every 1 second	R/W
4	RTCA0ALME	Alarm interrupt (RTC_ALM) output enable bit 0: Disable 1: Enable	R/W
5	RTCA01HZE	This bit enables/disables 1 Hz pulse output (RTCAT1HZ). 0: Disable 1 Hz pulse output (RTCAT1HZ fixed to 0) 1: Enable 1 Hz pulse output	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

26.3.3 RTCA0CTL2 : RTC Control Register 2

Base address: RTC = 0x8100_9000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RTCA0WSST	RTCA0RSST	RTCA0RSUB	RTCA0WST	RTCA0WAIT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RTCA0WAIT	RTC Controller Counter Wait Control This bit controls the count-up operation of the counters. Be sure to write 1 to this bit when reading and writing counter values while the clock counter operation is enabled (RTCA0CE = 1). When this bit is 1, on occurrence of RTCA0SUBC overflow, the overflow information is held internally, and after 0 is written to this bit, RTCA0SEC is counted up. However, when the value of the second counter is written while RTCA0WAIT = 1, the overflow information that was held is discarded. 0: Counter operation 1: Count-up operation of RTCA0SEC, RTCA0MIN, RTCA0HOUR, RTCA0WEEK, RTCA0DAY, RTCA0MONTH, RTCA0YEAR is stopped (count-up operation of second, minute, hour, week, day, month, and year count registers is stopped)	R/W
1	RTCA0WST	RTC Controller Counter Wait Status When 1 is written to RTCA0WAIT, the value of this bit becomes 1 when the count-up operation of RTCA0SEC, RTCA0MIN, RTCA0HOUR, RTCA0WEEK, RTCA0DAY, RTCA0MONTH, RTCA0YEAR has stopped completely. Before reading and writing clock counters (RTCA0SEC, RTCA0MIN, RTCA0HOUR, RTCA0WEEK, RTCA0DAY, RTCA0MONTH, RTCA0YEAR) during clock counter operation (RTCA0CE = 1), check that the value of this register is 1. Also, when write to a clock counter (RTCA0SEC, RTCA0MIN, RTCA0HOUR, RTCA0WEEK, RTCA0DAY, RTCA0MONTH, RTCA0YEAR) occurred, the value written to RTCA0WAIT is reflected only when the write operation has completed (status hold). 0: Count-up status of RTCA0SECC, RTCA0MINC, RTCA0HOURC, RTCA0WEEKC, RTCA0DAYC, RTCA0MONC, RTCA0YEARC 1: Count-up stopped status of RTCA0SECC, RTCA0MINC, RTCA0HOURC, RTCA0WEEKC, RTCA0DAYC, RTCA0MONC, RTCA0YEARC (count-up operation of second, minute, hour, week, day, month, and year count registers is stopped)	R
2	RTCA0RSUB	RTCA0SUBC Data Transfer Control This is a control bit for transferring the value of RTCA0SUBC of RTCA0SRBU. Use it when reading the value of RTCA0SRBU while the clock counter operation is enabled (RTCA0CE = 1). When 1 is written to this bit, the value of RTCA0SUBC is loaded to RTCA0SRBU in synchronization with PCLKRTC. For the usage method, see section 26.4.1.4. Reading RTCA0SRBU While Clock Counter Operation is Enabled. 0: RTCA0SRBU data hold 1: Transfer of value of RTCA0SUBC to RTCA0SRBU	R/W
3	RTCA0RSST	RTCA0SRBU Transfer Status When 1 is written to RTCA0RSUB, the value of this bit becomes 1 when the value of RTCA0SUBC is loaded to RTCA0SRBU. Before reading RTCA0SRBU during clock counter operation (RTCA0CE = 1), check that the value of this register is 1. 0: RTCA0SRBU data hold status 1: Transfer of the value of RTCA0SUBC to RTCA0SRBU completed	R
4	RTCA0WSST	RTCA0SCMP Write Status This bit is set to 1 when RTCA0SCMP is written to while the clock counter operation is enabled (RTCA0CE = 1). It is then cleared to 0 on completion of RTCA0SCMP write (RTCA0SUBC overflow). Do not write to RTCA0SCMP while the value of this bit is 1 because write to RTCA0SCMP is in progress. 0: RTCA0SCMP write completed 1: RTCA0SCMP write in progress	R
31:5	—	These bits are read as 0. The write value should be 0.	R/W

26.3.4 RTCA0SUBC : RTC Sub Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x0C

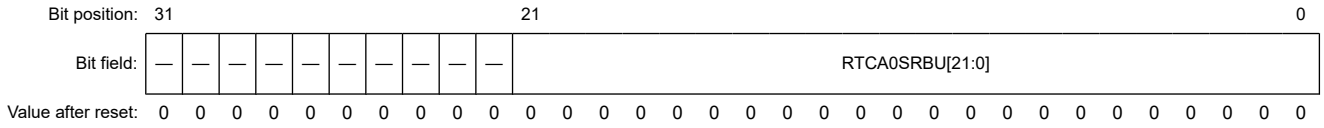
Bit position:	31										21																				0
Bit field:	—	—	—	—	—	—	—	—	—	—	RTCA0SUBC[21:0]																				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
21:0	RTCA0SUBC[21:0]	Register that counts the 1 second reference time This register is cleared on reset or on write to the RTCA0SEC (RTC Sec Count Buffer register) or RTCA0TIME (RTC Time Set register), and arbitrary values cannot be written to this register. Count-up occurs on clock input when the clock counter operation is enabled (RTCA0CE = 1). When this register is read while the clock counter operation is enabled (RTCA0CE = 1), the register value, which changes in synchronization with PCLKRTC, is read asynchronously, so that the read value is not guaranteed. To check the value of this register while the clock counter operation is enabled (RTCA0CE = 1), use the RTCA0SRBU register. The count-up operation starts from 0x0, and on matching the value set to RTCA0SCMP, an overflow signal is output to RTCA0SEC (Second Count Buffer register), and this register is cleared. [Example] Operation using 195.3 kHz clock 195312 - 1 = 195311 (decimal code) = 0x2FAEF is set to RTCA0SCMP. Counting up is performed each time a 195.3 kHz clock is input, taking on values from 0x0 to 0x2FAEF. 1 second is counted when the value changes from 0x2FAEF to 0x0.	R
31:22	—	These bits are read as 0.	R

26.3.5 RTCA0SRBU : RTC Sub Count Register Read Buffer

Base address: RTC = 0x8100_9000

Offset address: 0x10

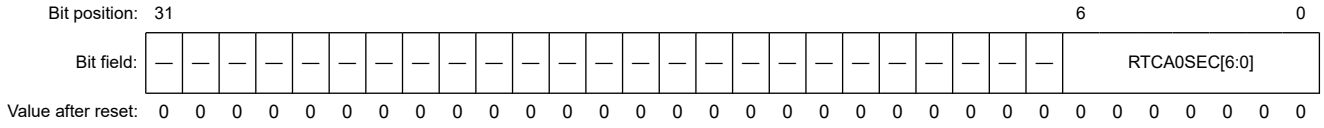


Bit	Symbol	Function	R/W
21:0	RTCA0SRBU[21:0]	Read buffer register of RTCA0SUBC This register is cleared when reset, at which time arbitrary values cannot be written. When 1 is written to the RTCA0RSUB bit while the clock counter operation is enabled (RTCA0CE = 1), the value of RTCA0SUBC is loaded to this register in synchronization with PCLKRTC clock. Be sure to perform RTCA0SRBU read according to the flow described in section 26.4.1.4. Reading RTCA0SRBU While Clock Counter Operation is Enabled.	R
31:22	—	These bits are read as 0.	R

26.3.6 RTCA0SEC : RTC Sec Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x14



Bit	Symbol	Function	R/W
6:0	RTCA0SEC[6:0]	Buffer register to read/write RTC Second Count register (RTCA0SECC). This register is used when Second Count register is read and written. When setting this register, a decimal value of 00 to 59 must be set in BCD code.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

26.3.7 RTCA0MIN : RTC Min Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x18

Bit position:	31																							6						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0MIN[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	RTCA0MIN[6:0]	Buffer register to read/write RTC Minute Count register (RTCA0MINC). This register is used when Minute Count register is read and written. When setting this register, a decimal value of 00 to 59 must be set in BCD code.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

26.3.8 RTCA0HOUR : RTC Hour Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x1C

Bit position:	31																								5					0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0HOUR[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Bit	Symbol	Function	R/W
5:0	RTCA0HOUR[5:0]	Buffer register to read/write RTC Hour Count register (RTCA0HOURC). This register is used when Hour Count register is read and written. When setting this register, a decimal value of 01 to 12 or 21 to 32 must be set when RTCA0AMPM = 0, or a decimal value of 0 to 23 when RTCA0AMPM = 1, in BCD code in both cases. Display of RTCA0HOUR is 12-hour display when the RTCA0AMPM bit is set to 0, and 24-hour display when it is 1. In the case of 12-hour display, am/pm is indicated by the 5th bit of RTCA0HOUR. During am, RTCA0HOUR[5] = 0, and during pm, RTCA0HOUR[5] = 1.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

12-hour display		24-hour display	
Time	RTCA0HOUR display	Time	RTCA0HOUR display
0 am	0x12	0	0x00
1 am	0x01	1	0x01
⋮	⋮	⋮	⋮
9 am	0x09	9	0x09
10 am	0x10	10	0x10
11 am	0x11	11	0x11
0 pm	0x32	12	0x12
1 pm	0x21	13	0x13
⋮	⋮	⋮	⋮
9 pm	0x29	21	0x21
10 pm	0x30	22	0x22
11 pm	0x31	23	0x23

26.3.9 RTCA0WEEK : RTC Week Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x20

Bit position:	31																										2	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0WEEK[2:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RTCA0WEEK[2:0]	Buffer register to read/write RTC Week Count register (RTCA0WEEKC). This register is used when Week Count register is read and written. When setting this register, a decimal value of 00 to 06 must be set in BCD code. There is no particular correspondence between the value of RTCA0WEEK and the day of the week. Set the correspondence according to the application to be used. [Example] 0 → Sunday, 1 → Monday, …, 6 → Saturday	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

26.3.10 RTCA0DAY : RTC Day Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x24

Bit position:	31																									5	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0DAY[5:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
5:0	RTCA0DAY[5:0]	Buffer register to read/write RTC Day Count register (RTCA0DAYC). This register is used when Day Count register is read and written. When setting this register, a decimal value of 01 to 31 must be set in BCD code. ● 01 to 31 (January, March, May, July, August, October, December) ● 01 to 30 (April, June, September, November) ● 01 to 29 (February, leap year) ● 01 to 28 (February, non-leap year)	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

26.3.11 RTCA0MONTH : RTC Month Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x28

Bit position:	31																									4	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0MONTH[4:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
4:0	RTCA0MONTH[4:0]	Buffer register to read/write RTC Month Count register (RTCA0MONC). This register is used when Month Count register is read and written. When setting this register, a decimal value of 01 to 12 must be set in BCD code.	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

26.3.12 RTCA0YEAR : RTC Year Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x2C

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0YEAR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RTCA0YEAR[7:0]	Buffer register to read/write RTC Year Count register (RTCA0YEARC). This register is used when Year Count register is read and written. When setting this register, a decimal value of 00 to 99 must be set in BCD code.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

26.3.13 RTCA0TIME : RTC Time Set Register

Base address: RTC = 0x8100_9000

Offset address: 0x30

Bit position:	31								23							16	15							8	7						0
Bit field:	—	—	—	—	—	—	—	—	RTCA0HOUR[7:0]								RTCA0MIN[7:0]								RTCA0SEC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RTCA0SEC[7:0]	See RTCA0SEC register	R/W
15:8	RTCA0MIN[7:0]	See RTCA0MIN register	R/W
23:16	RTCA0HOUR[7:0]	See RTCA0HOUR register	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The RTCA0TIME register is a register for accessing the RTCA0HOUR, RTCA0MIN, and RTCA0SEC registers simultaneously. By using this register, the RTCA0HOUR, RTCA0MIN, and RTCA0SEC registers can be read or written simultaneously.

26.3.14 RTCA0CAL : RTC Calendar Set Register

Base address: RTC = 0x8100_9000

Offset address: 0x34

Bit position:	31							24	23							16	15							8	7						0
Bit field:	RTCA0YEAR[7:0]								RTCA0MONTH[7:0]								RTCA0DAY[7:0]								RTCA0WEEK[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RTCA0WEEK[7:0]	See RTCA0WEEK register	R/W
15:8	RTCA0DAY[7:0]	See RTCA0DAY register	R/W
23:16	RTCA0MONTH[7:0]	See RTCA0MONTH register	R/W
31:24	RTCA0YEAR[7:0]	See RTCA0YEAR register	R/W

The RTCA0CAL register is a register for accessing the RTCA0YEAR, RTCA0MONTH, RTCA0DAY, and RTCA0WEEK registers simultaneously. By using this register, the RTCA0YEAR, RTCA0MONTH, RTCA0DAY, and RTCA0WEEK registers can be read or written simultaneously.

26.3.15 RTCA0SCMP : RTC Sub Count Compare Register

Base address: RTC = 0x8100_9000

Offset address: 0x3C

Bit position:	31										21																				0
Bit field:	—	—	—	—	—	—	—	—	—	—	RTCA0SCMP[21:0]																					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
21:0	RTCA0SCMP[21:0]	Register that sets the compare value of RTCA0SUBC (sub-counter). The setting of this register is valid when RTCA0SLSB = 1. When RTCA0SLSB = 1 is set, RTCA0SUBC starts counting up from 0, and when the value set to this register is matched, an overflow signal is output to RTCA0SEC (Second Count Buffer register), and the register is cleared. Set the value for this register according to the frequency of the selected input clock. Accurate one second counting can be performed to PCLKRTC through the value set to this register. [Example] Setting of 195,312 (decimal) - 1 = 195,311 (decimal) = 0x2FAEF to RTCA0SCMP When performing write to this register during clock counter operation (RTCA0CE = 1), be sure to check that RTCA0WSST = 0. When writing to this register during clock counter operation (RTCA0CE = 1), write ends at the timing of RTCA0SUBC overflow. Be sure to perform RTCA0SCMP write according to the flows described in section 26.4.1.1. Initial Setting and section 26.4.1.6. Writing to RTCA0SCMP During Clock Counter Operation.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

26.3.16 RTCA0ALM : RTC Alarm Min Set Register

Base address: RTC = 0x8100_9000

Offset address: 0x40

Bit position:	31																								6						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0ALM[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	RTCA0ALM[6:0]	RTCA0ALM is a register that performs the minute setting for the alarm interrupt. Set a decimal value of 00 to 59 in BCD code (alarm detection is not performed if an out-of-range value is set.)	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

26.3.17 RTCA0ALH : RTC Alarm Hour Set Register

Base address: RTC = 0x8100_9000

Offset address: 0x44

Bit position:	31																								5		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0ALH[5:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Bit	Symbol	Function	R/W
5:0	RTCA0ALH[5:0]	RTCA0ALH is a register that performs the hour setting for the alarm interrupt. When RTCA0AMPM = 0, set a decimal value of 01 to 12 or 21 to 32, and when RTCA0AMPM = 1, set a decimal value of 00 to 23, in BCD code in both cases (alarm detection is not performed if an out-of-range value is set.)	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

26.3.18 RTCA0ALW : RTC Alarm Week Set Register

Base address: RTC = 0x8100_9000

Offset address: 0x48

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RTCA0ALW6	RTCA0ALW5	RTCA0ALW4	RTCA0ALW3	RTCA0ALW2	RTCA0ALW1	RTCA0ALW0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RTCA0ALW0	Alarm interrupt day of the week setting bit 0 0: Disable alarm 1: Enable alarm	R/W
1	RTCA0ALW1	Alarm interrupt day of the week setting bit 1 0: Disable alarm 1: Enable alarm	R/W
2	RTCA0ALW2	Alarm interrupt day of the week setting bit 2 0: Disable alarm 1: Enable alarm	R/W
3	RTCA0ALW3	Alarm interrupt day of the week setting bit 3 0: Disable alarm 1: Enable alarm	R/W
4	RTCA0ALW4	Alarm interrupt day of the week setting bit 4 0: Disable alarm 1: Enable alarm	R/W
5	RTCA0ALW5	Alarm interrupt day of the week setting bit 5 0: Disable alarm 1: Enable alarm	R/W
6	RTCA0ALW6	Alarm interrupt day of the week setting bit 6 0: Disable alarm 1: Enable alarm	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The RTCA0ALW register enables the alarm corresponding to RTCA0WEEK value. For example, if RTCA0WEEK = 0 is Sunday, RTCA0ALW0 bit functions as Sunday alarm. If RTCA0WEEK = 6 is Saturday, RTCA0ALW6 bit functions as Saturday alarm.

26.3.19 RTCA0SECC : RTC Second Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x4C

Bit position:	31																								6						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0SECC[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	RTCA0SECC[6:0]	Counts up the seconds This register counts from 00 to 59, and when its value changes from 59 to 00 in BCD code, an overflow signal is output to RTCA0MINC.	R
31:7	—	These bits are read as 0.	R

26.3.20 RTCA0MINC : RTC Minute Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x50

Bit position:	31																								6						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0MINC[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	RTCA0MINC[6:0]	Counts up the minutes This register counts from 00 to 59, and when its value changes from 59 to 00 in BCD code, an overflow signal is output to RTCA0HOURC.	R
31:7	—	These bits are read as 0.	R

26.3.21 RTCA0HOURC : RTC Hour Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x54

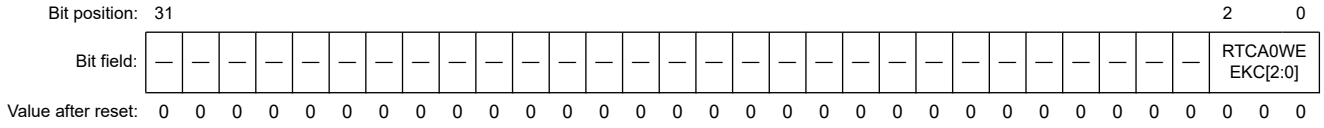
Bit position:	31																									5					0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0HOURC[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Bit	Symbol	Function	R/W
5:0	RTCA0HOURC[5:0]	Counts up the hours This register counts from 00 to 23, or 01 to 12, or 21 to 32 in BCD code depending on the setting of RTCA0AMPM, and when its value changes from 23 to 00 or from 32 to 01, an overflow signal is output to RTCA0DAY and RTCA0WEEK.	R
31:6	—	These bits are read as 0.	R

26.3.22 RTCA0WEEKC : RTC Week Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x58

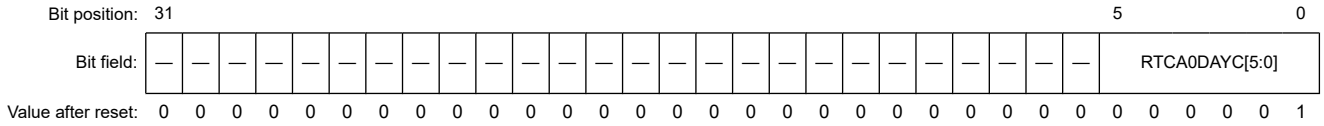


Bit	Symbol	Function	R/W
2:0	RTCA0WEEKC[2:0]	Counts up the weeks This register counts up on occurrence of RTCA0HOURC overflow when the clock counter operation is enabled (RTCA0CE = 1).	R
31:3	—	These bits are read as 0.	R

26.3.23 RTCA0DAYC : RTC Day Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x5C

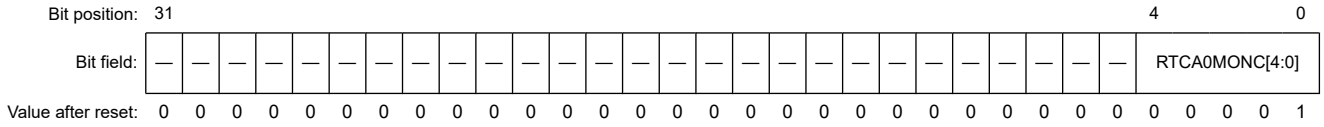


Bit	Symbol	Function	R/W
5:0	RTCA0DAYC[5:0]	Counts up the days This register counts up on occurrence of RTCA0HOURC overflow when the clock counter operation is enabled (RTCA0CE = 1). This register counts according to the value of RTCA0MONC, and on occurrence of an overflow, an overflow signal is output to RTCA0MONC.	R
31:6	—	These bits are read as 0.	R

26.3.24 RTCA0MONC : RTC Month Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x60

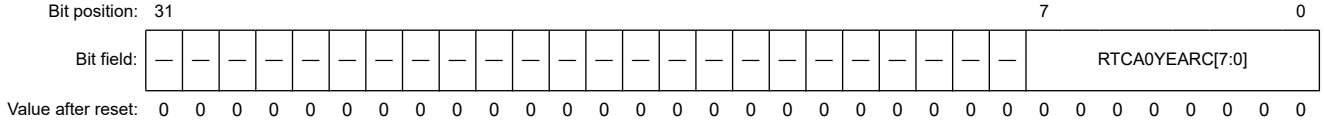


Bit	Symbol	Function	R/W
4:0	RTCA0MONC[4:0]	Counts up the months This register counts up on occurrence of RTCA0DAYC overflow while the clock counter operation is enabled (RTCA0CE = 1). This register counts from 01 to 12 in BCD code, and when its value changes from 12 to 01, an overflow signal is output to RTCA0YEARC.	R
31:5	—	These bits are read as 0.	R

26.3.25 RTCA0YEARC : RTC Year Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x64

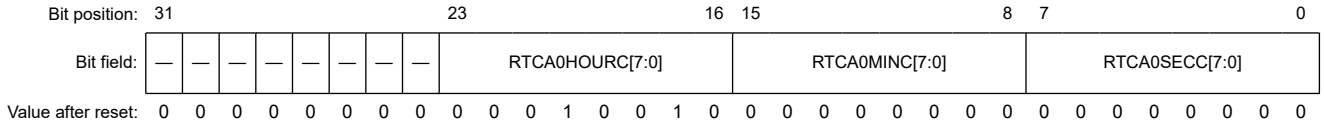


Bit	Symbol	Function	R/W
7:0	RTCA0YEARC[7:0]	Counts up the years This register counts up on occurrence of RTCA0MONC overflow while the clock counter operation is enabled (RTCA0CE = 1). This register counts from 00 to 99 in BCD code.	R
31:8	—	These bits are read as 0.	R

26.3.26 RTCA0TIMEC : RTC Time Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x68



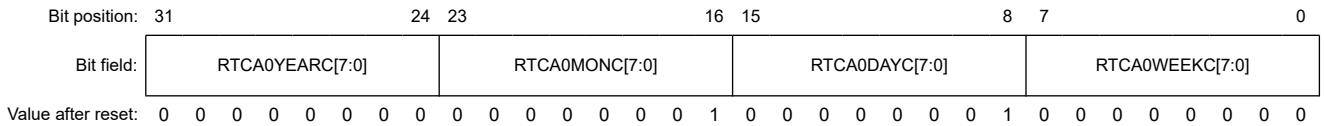
Bit	Symbol	Function	R/W
7:0	RTCA0SECC[7:0]	See RTCA0SECC register	R
15:8	RTCA0MINC[7:0]	See RTCA0MINC register	R
23:16	RTCA0HOURC[7:0]	See RTCA0HOURC register	R
31:24	—	These bits are read as 0.	R

This register is for reading the RTCA0HOURC, RTCA0MINC, and RTCA0SECC simultaneously.

26.3.27 RTCA0CALC : RTC Calendar Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x6C



Bit	Symbol	Function	R/W
7:0	RTCA0WEEKC[7:0]	See RTCA0WEEKC register	R
15:8	RTCA0DAYC[7:0]	See RTCA0DAYC register	R
23:16	RTCA0MONC[7:0]	See RTCA0MONC register	R
31:24	RTCA0YEARC[7:0]	See RTCA0YEARC register	R

The RTCA0CALC register is for reading the RTCA0YEARC, RTCA0MONC, RTCA0DAYC, and RTCA0WEEKC simultaneously.

26.4 Operation

26.4.1 Programming RTC

26.4.1.1 Initial Setting

Perform initial setting of RTC according to the flow in Figure 26.2.

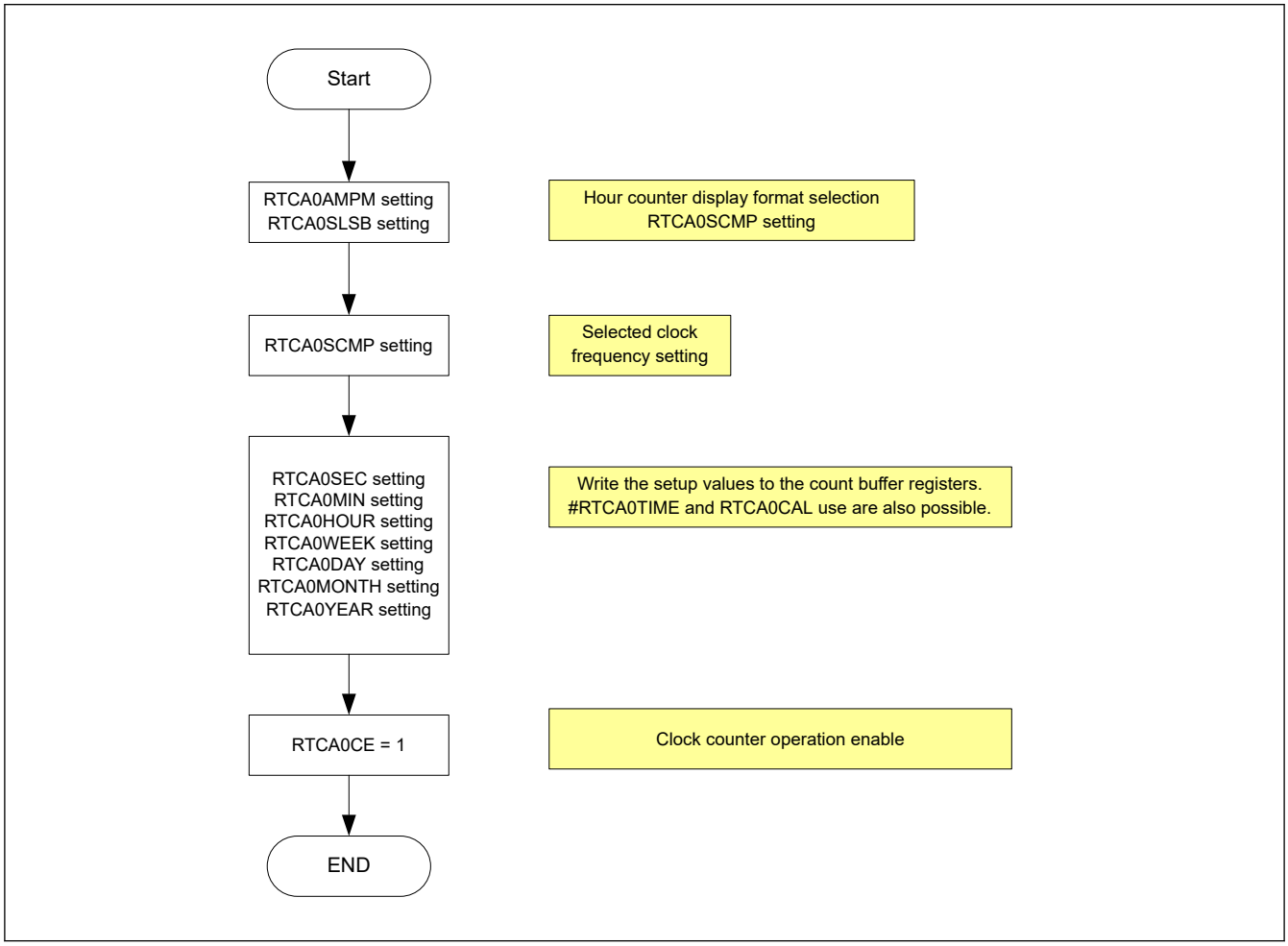


Figure 26.2 Setting RTC when clock counter is disabled

The internal clock counter operates in synchronization with PCLKRTC, and two PCLKRTC cycles are required until completion of the initial value setting procedure described in Figure 26.2. Therefore, PCLKL must be continuously supplied until completion of the initial value setting procedure. To stop PCLKL supply after the initial value setting procedure, check first that RTCA0CEST = 1 (clock counter operation enabled status).

26.4.1.2 Writing to Clock Counters While Clock Counter Operation is Enabled

Be sure to write to the RTCA0SEC, RTCA0MIN, RTCA0HOUR, RTCA0WEEK, RTCA0DAY, RTCA0MONTH, and RTCA0YEAR counters while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 26.3.

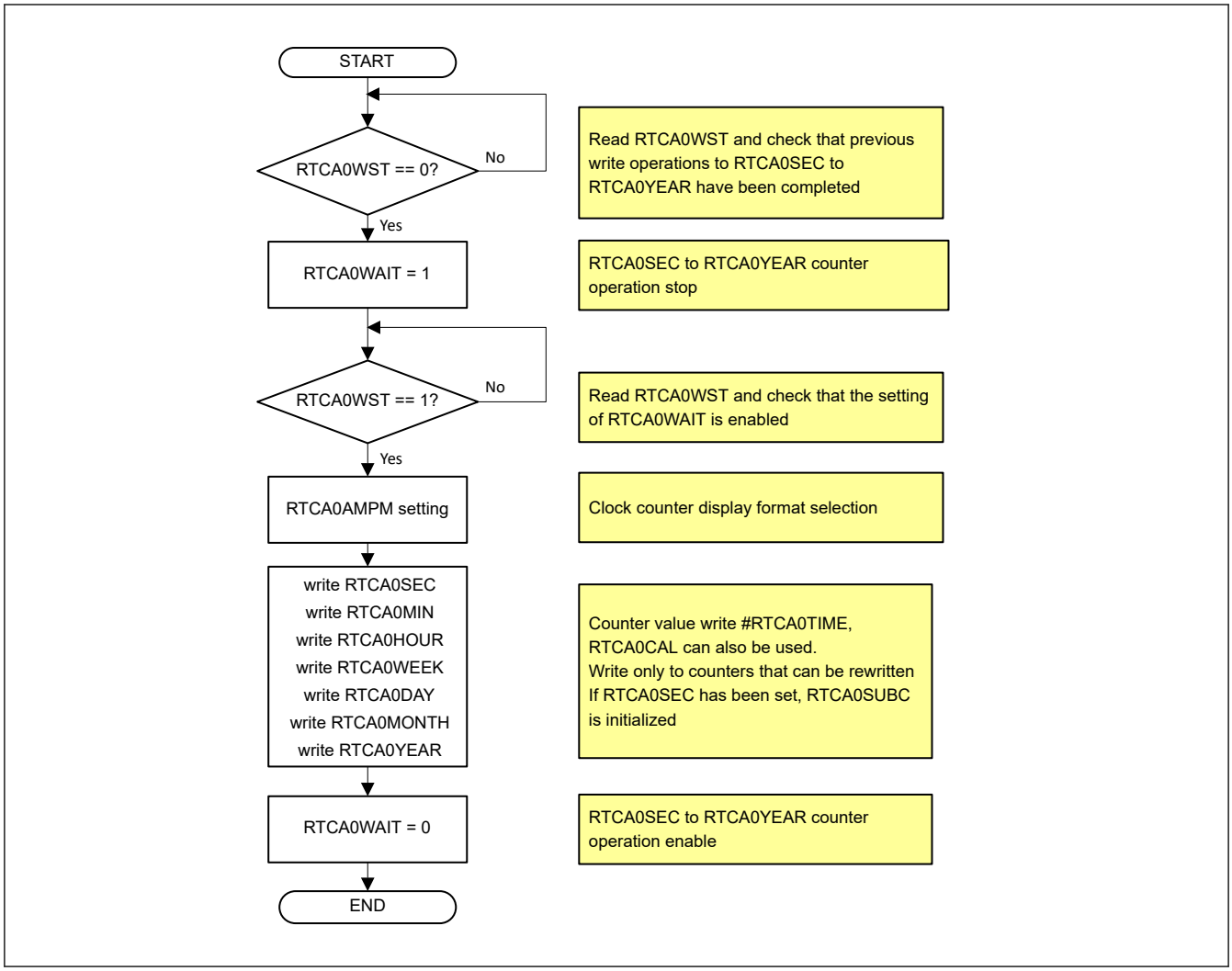


Figure 26.3 Setting RTC when clock counter is enabled

It is prohibited to write two or more times to the same register in the flow described in Figure 26.3. For example, it is prohibited to write twice to the second counter. End the programming flow within 1 second.

When RTCA0WAIT = 1 is set, the operation of RTCA0SEC (second counter buffer) stops. If RTCA0SUBC overflow occurs while RTCA0WAIT = 1, one overflow is held internally. However, if overflow occurs two or more times, the overflow count cannot be held.

The internal clock counter operates in synchronization with PCLKRTC. Two PCLKRTC cycles are required until completion of the above-described initial value setting procedure. Therefore, PCLKL must be continuously supplied until completion of the initial value setting procedure.

To stop PCLKL supply after the initial value setting procedure, check first that RTCA0WST = 0 (clock counter count-up status).

26.4.1.3 Reading Clock Counters While Clock Counter Operation is Enabled

There are two methods to read clock counters while clock counter operation is enabled (RTCA0CE = 1):

- Using Count Buffer Registers
- Reading Count Registers

The advantages and disadvantages of the two methods are mentioned in the next table.

Method	Advantage	Disadvantage
(1) Using count buffer registers	It is unnecessary to read clock counter several times such as in method (2), because clock counters are read synchronously.	Program wait*1 occurs from setting RTCA0WAIT = 1 to completion of data transfer.
(2) Reading count registers	Program wait does not occur.	When the reading of counter and overflow of RTCA0SUBC competed with each other, the clock counters are read several times again*2.

Note 1. The maximum of program wait is 3 clock cycles of PCLKL + 2 clock cycles of PCLKRTC.
Note 2. When reading of counter and overflow of sub-counter competed with each other, unknown value might be read.

(1) Using Count Buffer Registers

This is how to read clock counter value by using buffer registers. The value of clock count registers are transferred to count buffer registers by writing 1 in RTCA0WAIT. The count buffer registers are read after that. In this method, program wait occurs from setting RTCA0WAIT = 1 to completion of data transfer.

Reading clock counters while clock counter operation is enabled (RTCA0CE = 1)

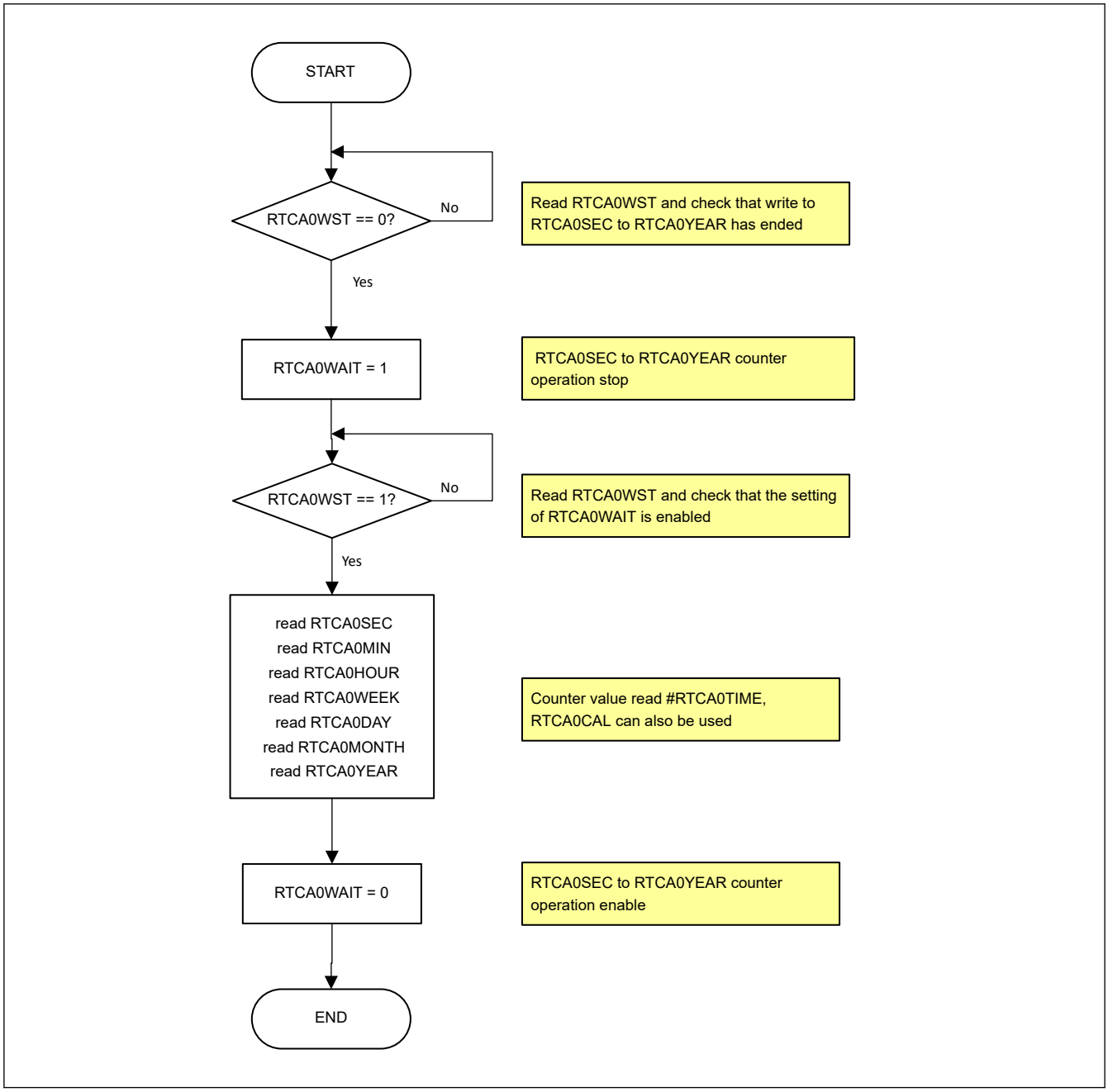


Figure 26.4 Reading RTC using count buffer registers

End this flow within 1 second.

When RTCA0WAIT = 1 is set, the operation of RTCA0SEC (second counter buffer) stops. If RTCA0SUBC overflow occurs while RTCA0WAIT = 1, one overflow is held internally. However, if overflow occurs two or more times, the overflow count cannot be held.

(2) Reading Count Registers

This is how to read clock counter immediately. RTCA0SECC (RTC Second Count Register) is read twice in the beginning and in the end. This is because it confirms whether overflow of sub-counter does not occur while counter is read. After that, the first read value is compared with the second read value. If the first read value is the same as the second read value, it is determined that overflow of sub-counter did not occur during counter read flow. If the first read value is not the same as the second read value, it is determined that overflow of sub-counter occurred during counter read flow, and it begins to read the clock counter again.

Reading clock counters while clock counter operation is enabled (RTCA0CE = 1)

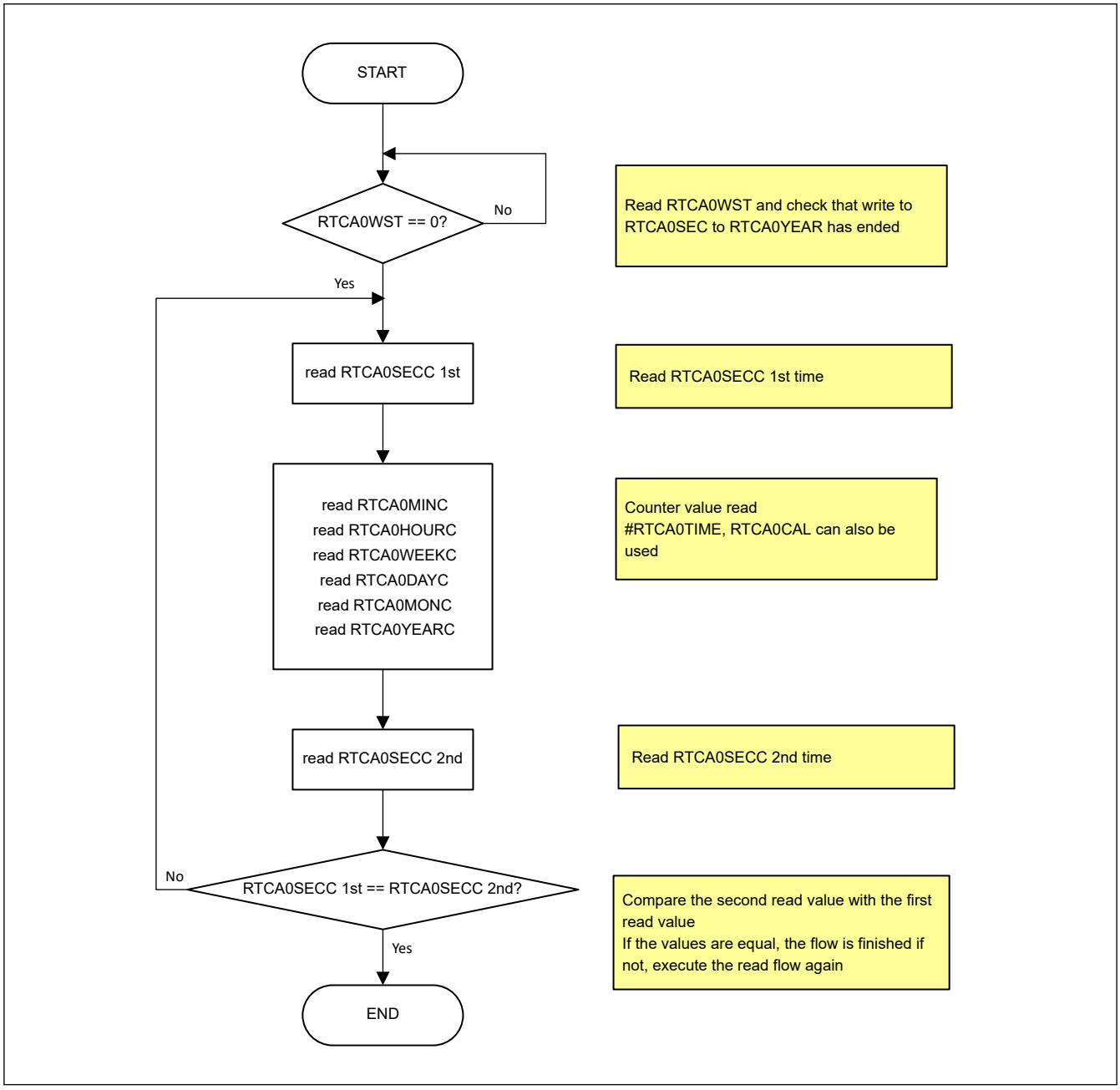


Figure 26.5 Reading RTC using the counting registers

End this flow within 1 second.

When it is not finished within one second, read clock counter value cannot be assured.

26.4.1.4 Reading RTCA0SRBU While Clock Counter Operation is Enabled

Be sure to read RTCA0SRBU while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 26.6.

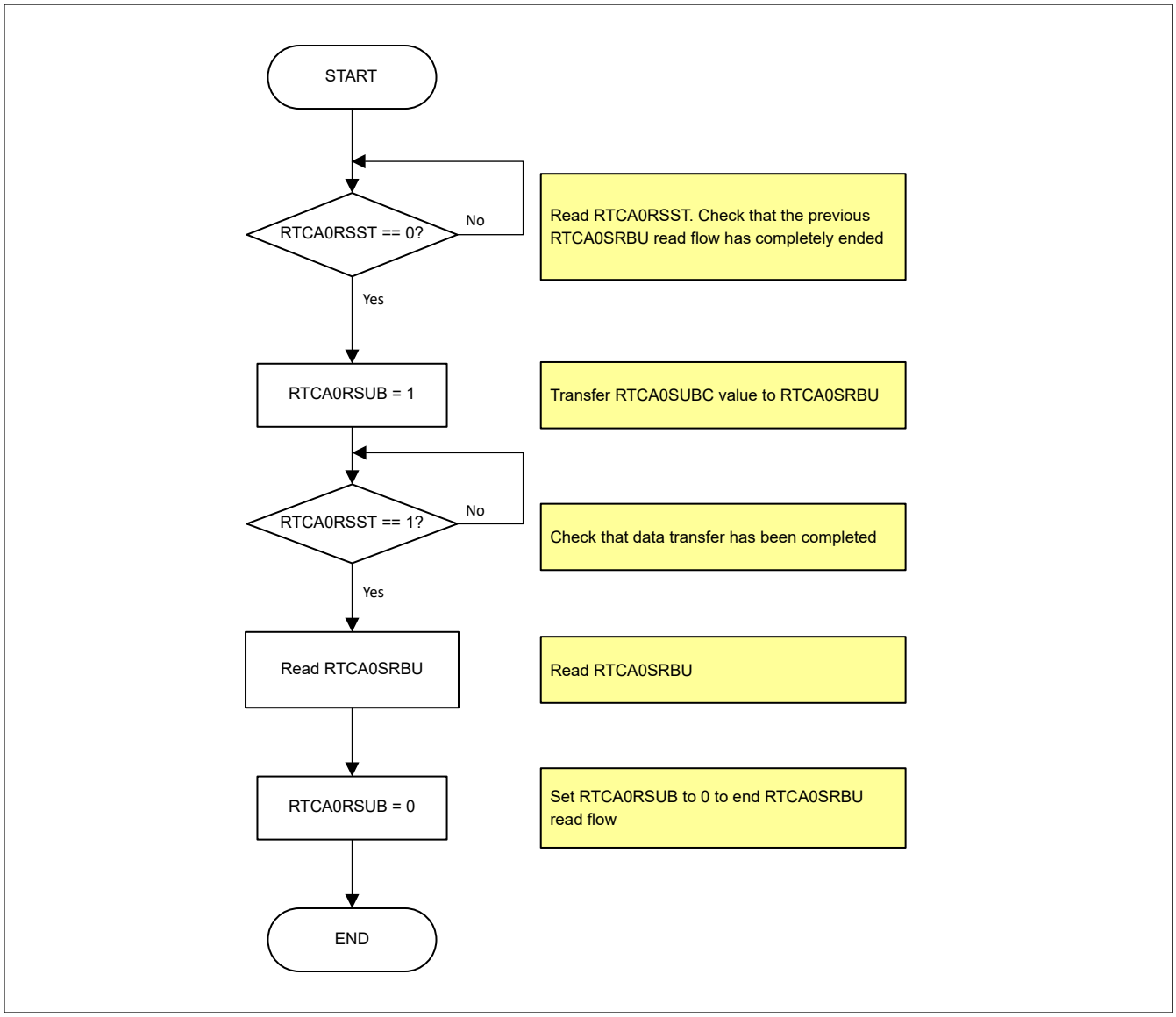


Figure 26.6 Reading RTCA0SRBU

26.4.1.5 Initialization of RTC While Clock Counter Operation is Enabled

Be sure to perform initialization of the RTC controller while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 26.7. To restart the RTC controller operation at the end of the flow, implement the flow as described in section 26.4.1.1. Initial Setting.

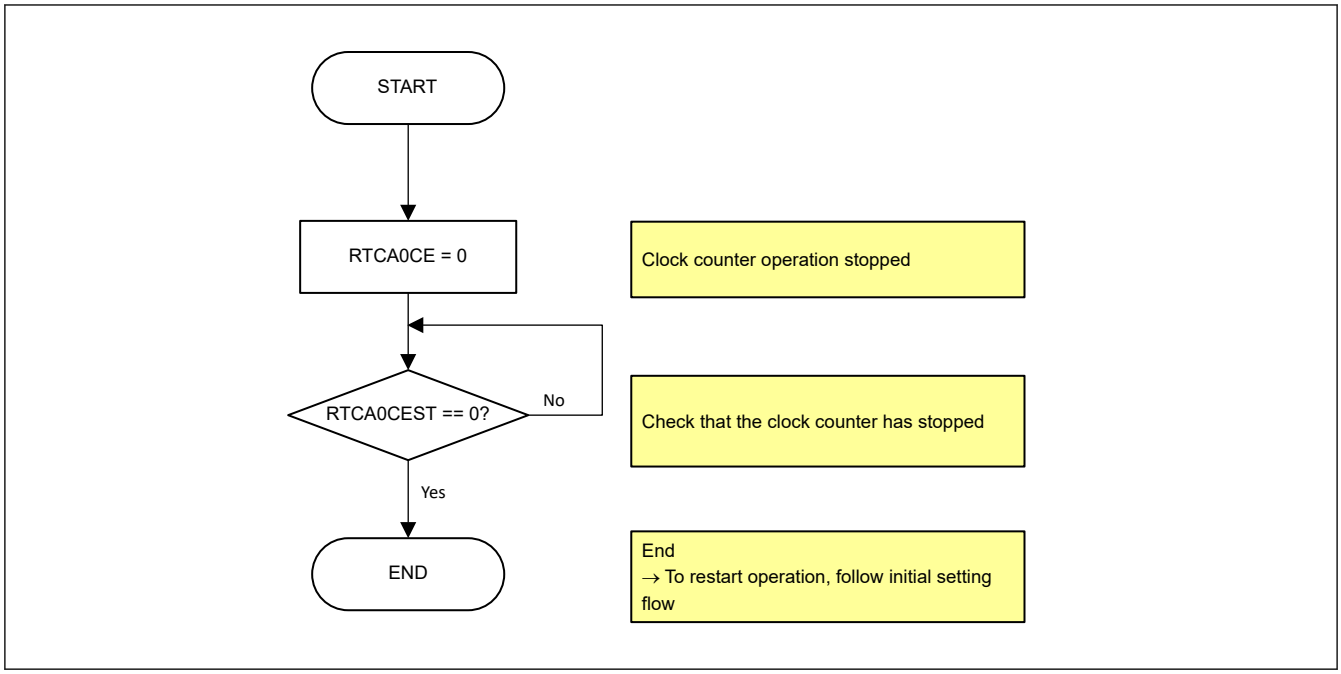


Figure 26.7 Initialization of RTC while clock counter operation is enabled

26.4.1.6 Writing to RTCA0SCMP During Clock Counter Operation

Be sure to write to RTCA0SCMP while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 26.8.

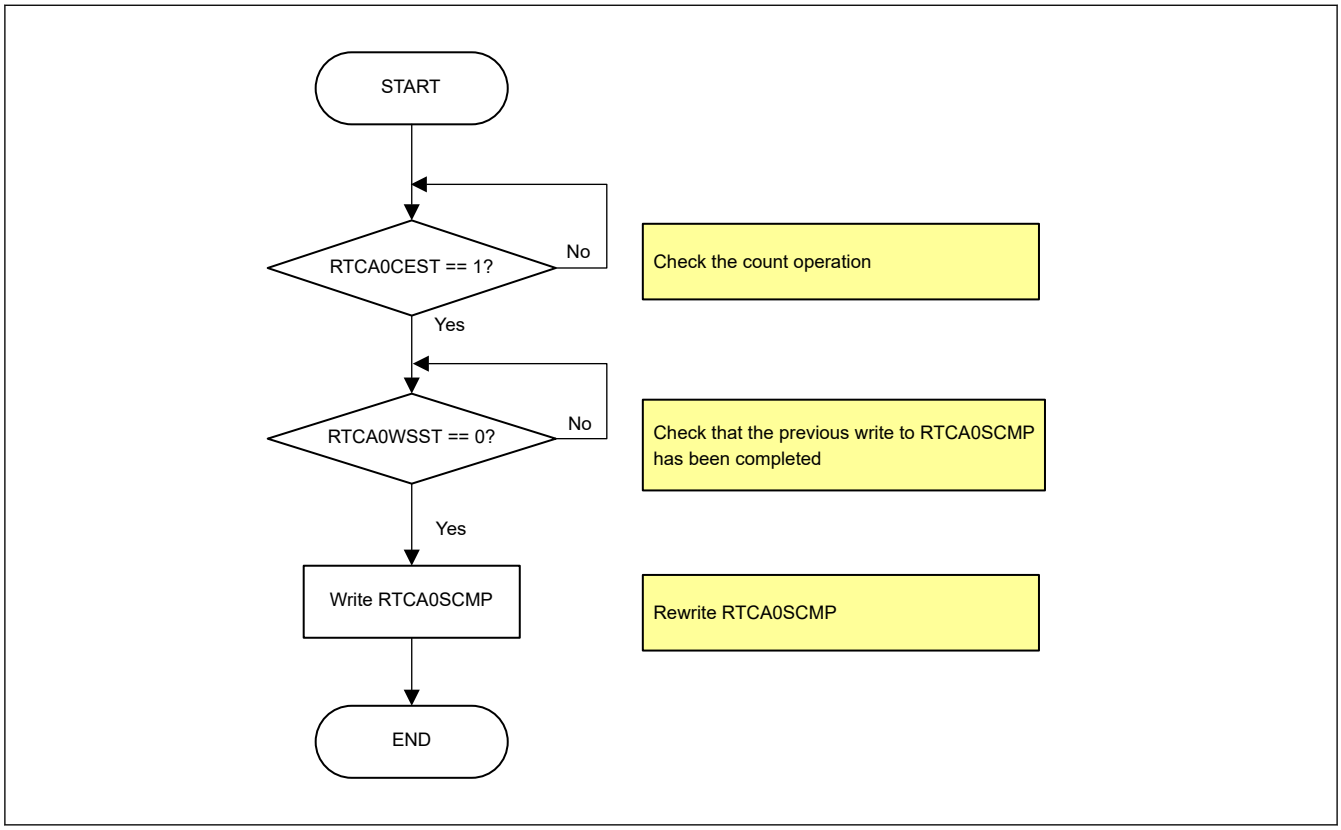


Figure 26.8 Writing to RTCA0SCMP

RTCA0WSST is set to 1 when RTCA0SCMP is written to while the clock counter operation is enabled (RTCA0CE = 1), and on completion of RTCA0SCMP write (RTCA0SUBC overflow), RTCA0WSST is cleared to 0. Because RTCA0WSST displays 1 for up to 1 second for this reason, caution is required when polling RTCA0WSST.

26.4.1.7 Changing the Setting of Fixed Interval Interrupt During Clock Counter Operation

Be sure to change the fixed interval interrupt setting while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 26.9.

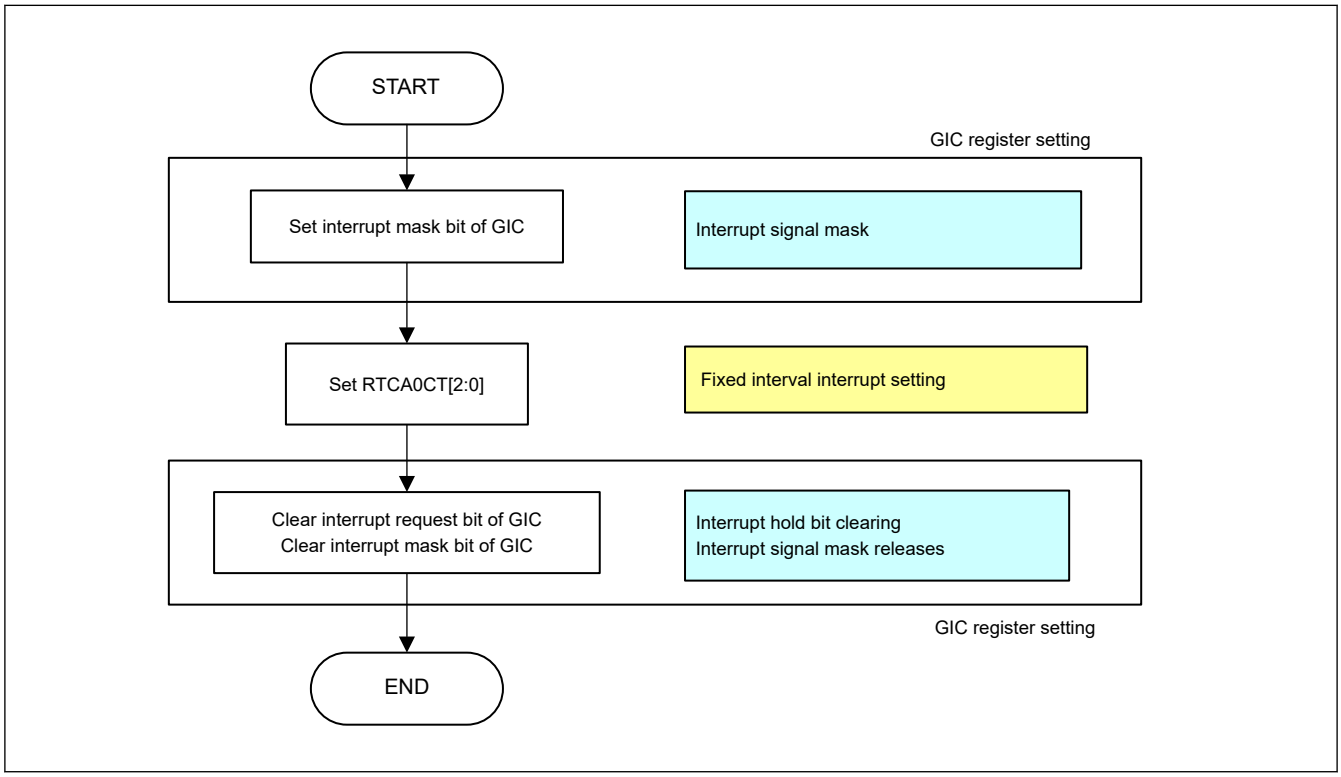


Figure 26.9 Setting of fixed interval interrupt during clock counter operation

26.4.1.8 Changing Alarm Setting During Clock Counter Operation

Be sure to change the alarm interrupt setting while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 26.10.

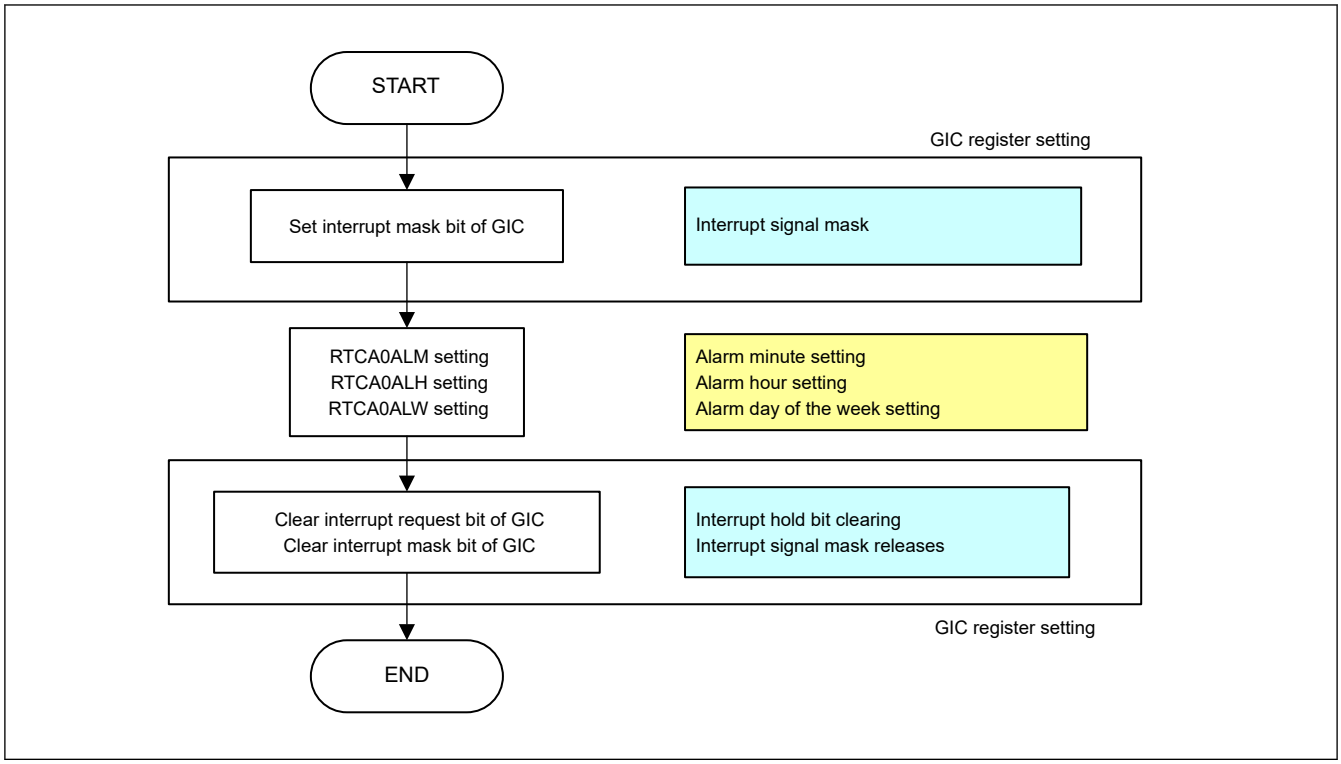


Figure 26.10 Changing the alarm setting during clock counter operation

27. Ethernet Subsystem

27.1 Overview

Ethernet Subsystem (Ethernet_SS) consists of Ethernet MAC (GMAC), Ethernet Switch (ETHSW), EtherCAT Slave Controller (ESC), and RGMII/RMII converters. Four MAC-PHY interfaces supporting RGMII, MII, and RMII are shared with these functions.

Table 27.1 lists the specification of the Ethernet Subsystem and Figure 27.1 shows a block diagram of the Ethernet Subsystem.

Table 27.1 Ethernet Subsystem specifications

Parameter	Description
Ethernet function	• Gigabit Ethernet MAC (GMAC0 to GMAC2) • Ethernet Switch supporting TSN (ETHSW) • EtherCAT Slave Controller (ESC)
MAC-PHY interface	Supported interface by RGMII/RMII converter • RGMII (2.5 MHz/25 MHz/125 MHz) • RMII (50 MHz) • MII (2.5 MHz/25 MHz)
Clock output for PHY	• 25 MHz reference clock output • 50 MHz reference clock output for RMII
Event link (input)	• GMAC PTP Timer capture • EtherCAT DC capture (LATCH0/1)
Event link (output)	• Ethernet Switch PTP Timer pulse event • Ethernet Switch TDMA scheduler event • EtherCAT DC event (SYNC0/1)
Module-stop function	Module-stop state can be set to reduce power consumption each module

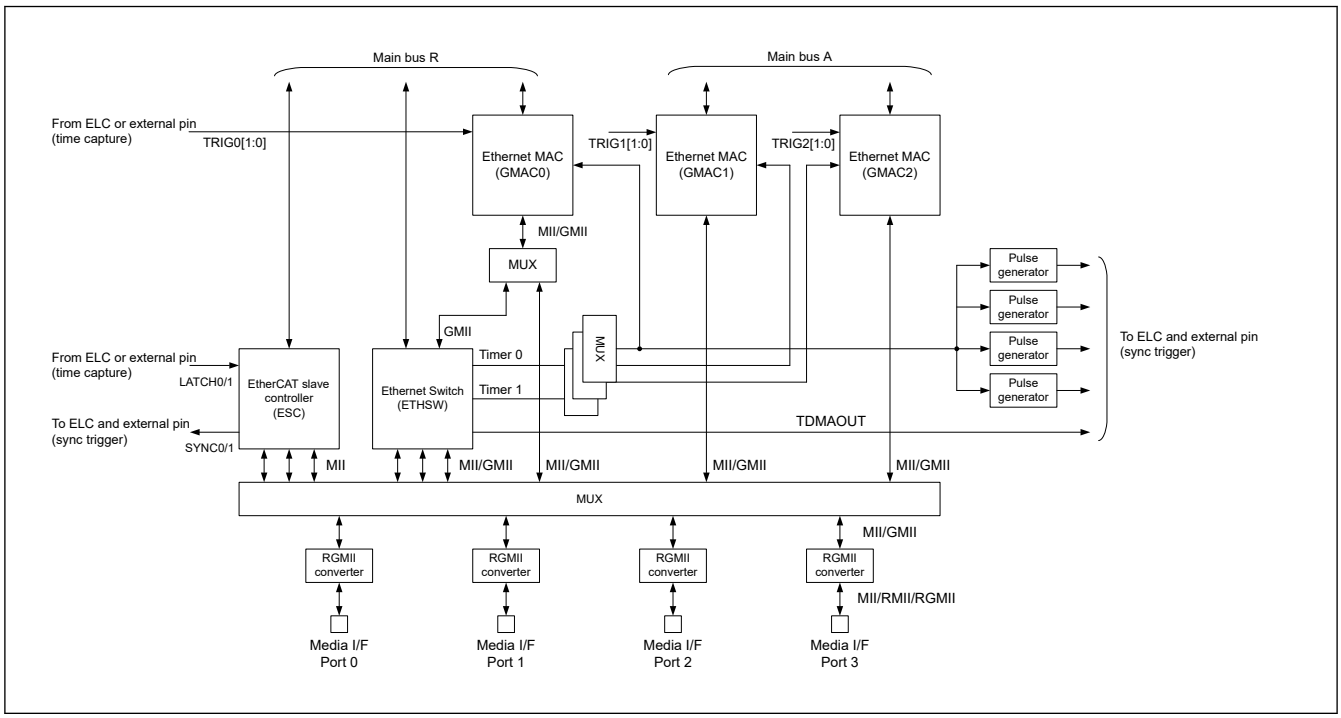


Figure 27.1 Block diagram of Ethernet Subsystem

Table 27.2 Ethernet Subsystem I/O pins (common PHY interface only)

Channel	Pin Name	I/O	Function
ETHn	ETHn_TXCLK	I/O	TX clock input pin (MII mode)/TX clock output pin (RGMII mode)
	ETHn_TXD0	Output	TX data 0 output pin (RGMII, RMII, and MII modes)
	ETHn_TXD1	Output	TX data 1 output pin (RGMII, RMII, and MII modes)
	ETHn_TXD2	Output	TX data 2 output pin (RGMII and MII modes)
	ETHn_TXD3	Output	TX data 3 output pin (RGMII and MII modes)
	ETHn_TXEN	Output	TX data enable pins (RMII and MII modes) TX data enable/TX data error (TX_CTL) pin (RGMII mode)
	ETHn_TXER	Output	TX data error output pin (MII mode)
	ETHn_RXCLK	Input	RX clock input pin (RGMII, RMII, and MII modes)
	ETHn_RXD0	Input	RX data 0 input pin (RGMII, RMII, and MII modes)
	ETHn_RXD1	Input	RX data 1 input pin (RGMII, RMII, and MII modes)
	ETHn_RXD2	Input	RX data 2 input pin (RGMII and MII modes)
	ETHn_RXD3	Input	RX data 3 input pin (RGMII and MII modes)
	ETHn_RXDV	Input	RX data valid pins (MII mode) Carrier sense/RX data valid (CRS_DV) pin (RMII mode) RX data valid/RX error (RX_CTL) pin (RGMII mode)
	ETHn_RXER	Input	RX data error pins (RMII and MII modes)
	ETHn_CRS	Input	Carrier sense pins (MII mode)
	ETHn_COL	Input	Collision detection pins (MII mode)

Note: n = 0 to 3

Table 27.3 Combination of MAC-PHY interface

Pin name	I/O	MII mode	RMII mode (REF_CLK input)	RMII mode (REF_CLK output)	RGMII mode
ETHn_TXCLK	I/O	TX_CLK (Input)	—	—	TXC (Output)
ETHn_TXD0	Output	TXD0	TXD0	TXD0	TXD0
ETHn_TXD1	Output	TXD1	TXD1	TXD1	TXD1
ETHn_TXD2	Output	TXD2	—	—	TXD2
ETHn_TXD3	Output	TXD3	—	—	TXD3
ETHn_TXEN	Output	TX_EN	TX_EN	TX_EN	TX_CTL
ETHn_TXER	Output	TX_ER	—	—	—
ETHn_RXCLK	Input	RX_CLK	REF_CLK	—	RXC
ETHn_RXD0	Input	RXD0	RXD0	RXD0	RXD0
ETHn_RXD1	Input	RXD1	RXD1	RXD1	RXD1
ETHn_RXD2	Input	RXD2	—	—	RXD2
ETHn_RXD3	Input	RXD3	—	—	RXD3
ETHn_RXDV	Input	RX_DV	CRS_DV	CRS_DV	RX_CTL
ETHn_RXER	Input	RX_ER	RX_ER (option)	RX_ER (option)	—
ETHn_CRS	Input	CRS	—	—	—
ETHn_COL	Input	COL	—	—	—
ETHn_REFCLK	Output	(PHY reference clock)			
RMIIn_REFCLK	Output	—	—	REF_CLK	—

Note: n = 0 to 3

Table 27.4 MAC-PHY interface mode

Combination	ETH0	ETH1	ETH2	ETH3
No.0*1	ETHSW Port 0	ETHSW Port 1	ETHSW Port 2	GMAC1
No.1	ESC Port 0	ESC Port 1	GMAC2	GMAC1
No.2*1	ESC Port 0	ESC Port 1	ETHSW Port 2	GMAC1
No.3	ESC Port 0	ESC Port 1	ESC Port 2	GMAC1
No.4*1 *2	ETHSW Port 0	ESC Port 1	ESC Port 2	GMAC1
No.5*1 *2	ETHSW Port 0	ESC Port 1	ETHSW Port 2	GMAC1
No.6*1	ETHSW Port 0	ETHSW Port 1	GMAC2	GMAC1
No.7	GMAC0	GMAC1	GMAC2	—

Note: This is the pin connection inside the Ethernet Subsystem. For example, if GMAC2 in ETH2 is not required for No.1, pins which are assigned to ETH2 can be used as another peripheral by pin mux. GMAC2 can be put to the module stop state (clock is stopped).
Note 1. In case using ETHSW, GMAC0 is used as management port.
Note 2. ETHSW Port 1 is connected to ESC Port 0 internally.

27.2 Register Map

Table 27.5 ETHSS register map

Address	Register symbol	Register name	Write protect
0x8011_0000	PRCMD	Ethernet Protect Register	—
0x8011_0008	MODCTRL	Mode Control Register	PRCMD
0x8011_000C	PTPMCTRL	PTP Mode Control Register	PRCMD
0x8011_0014	PHYLNK	Ethernet PHY Link Mode Register	PRCMD
0x8011_0100 + 0x04 × n	CONVCTRLn	RGMII/RMII Converter n Control Register (n = 0 to 3)	—
0x8011_0114	CONVRST	RGMII/RMII Converter Reset Control Register	PRCMD
0x8011_0304	SWCTRL	Switch Core Control Register	PRCMD
0x8011_0308	SWDUPC	Switch Core Duplex Mode Register	PRCMD

Table 27.6 ETHSS related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
All registers	MRCTLE.MRTCLE05	MSTPCRE.MSTPCRE03	SLVACCCTL6.ETHSS_SL

27.3 Register Descriptions

27.3.1 PRCMD : Ethernet Protect Register

Base address: ETHSS = 0x8011_0000

Offset address: 0x000

Bit position:	31																												0
Bit field:	PRCMD[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PRCMD[31:0]	Permit write access to protected registers For detail, see Figure 27.2. 0x0000 00A5: DLE to STATE1 at IDLE 0x0000 0001: STATE1 to STATE2 at STATE1 0x0000 FFFE: STATE2 to STATE3 at STATE2 0x0000 0001: STATE3 to Release Protect at STATE3 0x0000 0000: Clear at Release Protect	R/W

The PRCMD register is used to permit write access to protected registers. The PRCMD register mitigates the risk when application system carries out unusual operations by software runaways.

Before setting the PRCMD register to 0x00000001, protected registers are denied write access (no error occurs and the registers are not updated by write data).

[Setting procedure]

1. Writing 0x000000A5 to PRCMD register
2. Writing 0x00000001 to PRCMD register
3. Writing 0x0000FFFE to PRCMD register
4. Writing 0x00000001 to PRCMD register

[Clearing condition]

- Writing 0x00000000 to PRCMD register

Figure 27.2 shows an ethernet protect command sequence.

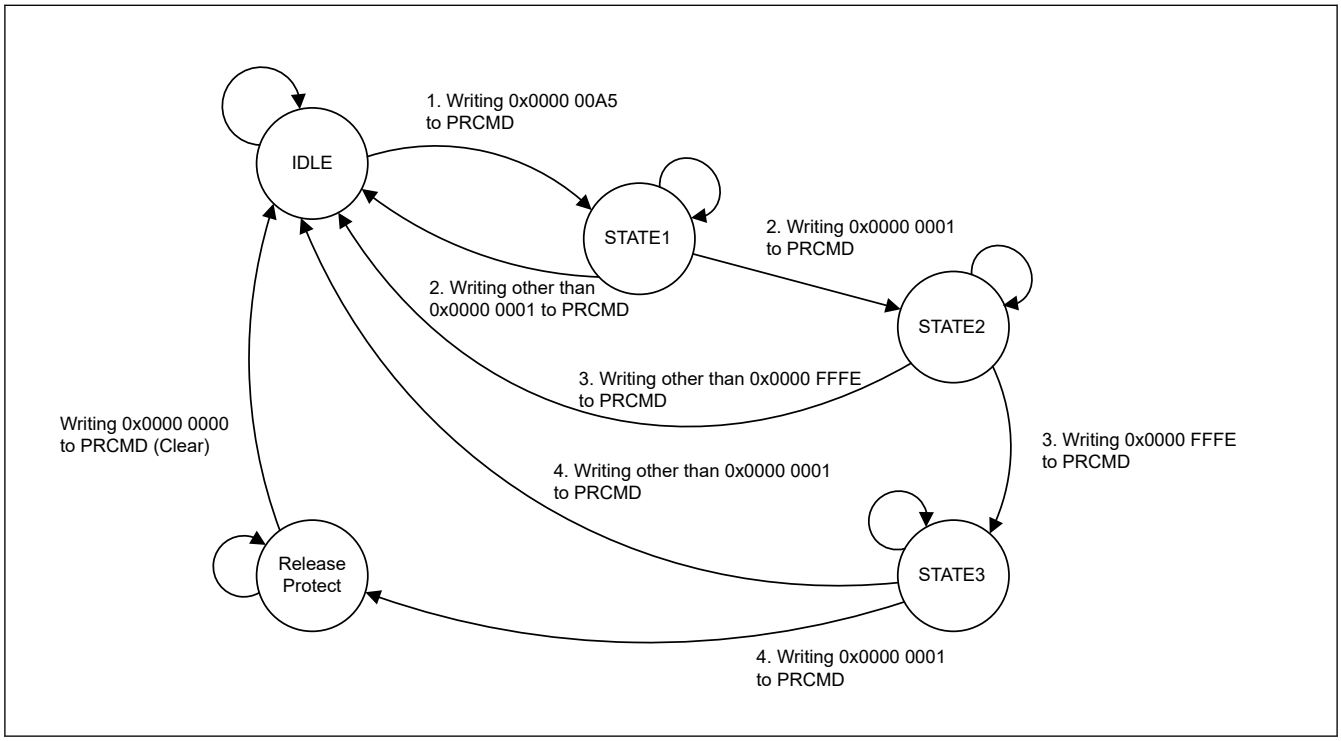


Figure 27.2 Ethernet protect command sequence

27.3.2 MODCTRL : Mode Control Register

Base address: ETHSS = 0x8011_0000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SW_MODE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SW_MODE[2:0]	Selects the function of the Media interface of the MAC to be used Media I/F connection	R/W

SW_MODE[2:0]	Media I/F Port 0	Port 1	Port 2	Port 3
000b*1	ETHSW Port 0	ETHSW Port 1	ETHSW Port 2	GMAC1
001b	ESC Port 0	ESC Port 1	GMAC2	GMAC1
010b*1	ESC Port 0	ESC Port 1	ETHSW Port 2	GMAC1
011b	ESC Port 0	ESC Port 1	ESC Port 2	GMAC1
100b*1 *2	ETHSW Port 0	ESC Port 1	ESC Port 2	GMAC1
101b*1 *2	ETHSW Port 0	ESC Port 1	ETHSW Port 2	GMAC1
110b*1	ETHSW Port 0	ETHSW Port 1	GMAC2	GMAC1
111b	GMAC0	GMAC1	GMAC2	—

Internal Connection among GMAC, ETHSW, and ESC

SW_MODE [2:0]	GMAC0	GMAC1	GMAC2	ETHSW Port 0	ETHSW Port 1	ETHSW Port 2	ETHSW Management Port (MPort)	ESC Port 0	ESC Port 1	ESC Port 2
000b	ETHSW MPort*1	ETH3	—	ETH0	ETH1	ETH2	GMAC0 *1	—	—	—
001b	—	ETH3	ETH2	—	—	—	—	ETH0	ETH1	—
010b	ETHSW MPort*1	ETH3	—	—	—	ETH2	GMAC0 *1	ETH0	ETH1	—
011b	—	ETH3	—	—	—	—	—	ETH0	ETH1	ETH2
100b	ETHSW MPort*1	ETH3	—	ETH0	ESC Port 0*2	—	GMAC0 *1	ETHSW Port 1*2	ETH1	ETH2
101b	ETHSW MPort*1	ETH3	—	ETH0	ESC Port 0*2	ETH2	GMAC0 *1	ETHSW Port 1*2	ETH1	—
110b	ETHSW MPort*1	ETH3	ETH2	ETH0	ETH1	—	GMAC0 *1	—	—	—
111b	ETH0	ETH1	ETH2	—	—	—	—	—	—	—

Note 1. Inter-module connection between GMAC0 and ETHSW MPort
Note 2. Inter-module connection between ETHSW Port 1 and ESC Port 0

Bit	Symbol	Function	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The MODCTRL register is used to switch the operation control of Switch mode. See Figure 27.1.

27.3.3 PTPMCTRL : PTP Mode Control Register

Base address: ETHSS = 0x8011_0000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PTP_PLS_RSTn
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	PTP_MODE2	PTP_MODE1	PTP_MODE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PTP_MODE0	Select the unit number of PTP Timer for GMAC0 and Pulse Generator (unit 0 - 3) 0: Ethernet Switch Timer 0 1: Ethernet Switch Timer 1	R/W
1	PTP_MODE1	Select the unit number of PTP Timer for GMAC1 0: Ethernet Switch Timer 0 1: Ethernet Switch Timer 1	R/W
2	PTP_MODE2	Select the unit number of PTP Timer for GMAC2 0: Ethernet Switch Timer 0 1: Ethernet Switch Timer 1	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
16	PTP_PLS_RSTn	Reset control for Pulse Generator (unit 0 - 3) 0: Assert RESET 1: Release RESET	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The PTPMCTRL register is used to select the unit number of PTP timer.

27.3.4 PHYLNK : Ethernet PHY Link Mode Register

Base address: ETHSS = 0x8011_0000

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CATLNK[2:0]			—	SWLINK[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SWLINK[2:0]	Specify the active level of the ETHSW_PHYLINKn signal using the Ethernet switch interface SWLINK[n] control ETHSW_PHYLINKn signal (n = 0 to 2) If not used, set the initial value. 0: Active-High PHYLINK signal 1: Active-Low PHYLINK signal	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6:4	CATLNK[2:0]	Specify the active level of the ESC_PHYLINKn signal using the EtherCAT interface CATLNK[n] control ESC_PHYLINKn signal (n = 0 to 2) If not used, set the initial value. 0: Active-Low PHYLINK signal 1: Active-High PHYLINK signal	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The PHYLNK register is used to switch the polarity of the Ethernet Switch and EtherCAT Link.

27.3.5 CONVCTRLn : RGMII/RMII Converter n Control Register (n = 0 to 3)

Base address: ETHSS = 0x8011_0000

Offset address: 0x100 + 0x004 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RGMII_SPEED[1:0]		RGMII_DUPLEX	RGMII_LINK	—	RMII_CRS_MODE	RMII_RX_ER_EN	FULLD	—	—	—	CONV_MODE[4:0]				
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	CONV_MODE[4:0]	Converter operation mode **CONV_MODE[4:0] / Description** x0000b: MII mode (through mode) 00100b: RMII mode 10 Mbps REF_CLK input 00101b: RMII mode 100 Mbps REF_CLK input 10100b: RMII mode 10 Mbps REF_CLK output 10101b: RMII mode 100 Mbps REF_CLK output x1000b: RGMII mode 10 Mbps x1001b: RGMII mode 100 Mbps x1010b: RGMII mode 1000 Mbps Others: Setting prohibited	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	FULLD	Duplex mode (for RMII/RGMII mode) 0: Half-duplex 1: Full-duplex	R/W
9	RMII_RX_ER_EN	ETHn_RXER pin input control (only for RMII mode) 0: RX_ER is always 0 1: RX_ER signal	R/W
10	RMII_CRS_MODE	CRS detection mode (only for RMII mode) 0: CRS | TX_EN 1: CRS | TX_EN | RX_DV	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	RGMII_LINK	Indicates link status (only for RGMII mode) 0: Down 1: Up	R

Bit	Symbol	Function	R/W
13	RGMII_DUPLEX	Indicates duplex status (only for RGMII mode) 0: Half-duplex 1: Full-duplex	R
15:14	RGMII_SPEED[1:0]	Indicates link speed (only for RGMII mode) 0 0: 2.5 MHz 0 1: 25 MHz 1 0: 125 MHz 1 1: Reserved	R
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The CONVCTRLn register is used to select the mode for RGMII/RMII.

Table 27.7 Ethernet interface support mode

Ethernet interface mode	Module			
		ETHSW		
	GMAC	Store and forward / cut through	Hub mode	EtherCAT (Port 0 / 1 / 2)
MII mode, 10 Mbps, Half-duplex	Supported	Supported	Not supported	Not supported
MII mode, 10 Mbps, Full-duplex	Supported	Supported	Not supported	Not supported
MII mode, 100 Mbps, Half-duplex	Supported	Supported	Supported	Not supported
MII mode, 100 Mbps, Full-duplex	Supported	Supported	Not supported	Supported*1
RMII mode, 10 Mbps, Half-duplex, Reference clock input	Supported	Supported	Not supported	Not supported
RMII mode, 10 Mbps, Full-duplex, Reference clock input	Supported	Supported	Not supported	Not supported
RMII mode, 100 Mbps, Half-duplex, Reference clock input	Supported	Supported	Supported	Not supported
RMII mode, 100 Mbps, Full-duplex, Reference clock input	Supported	Supported	Not supported	Supported*2
RMII mode, 10 Mbps, Half-duplex, Reference clock output	Supported	Supported	Not supported	Not supported
RMII mode, 10 Mbps, Full-duplex, Reference clock output	Supported	Supported	Not supported	Not supported
RMII mode, 100 Mbps, Half-duplex, Reference clock output	Supported	Supported	Supported	Not supported
RMII mode, 100 Mbps, Full-duplex, Reference clock output	Supported	Supported	Not supported	Supported*3
RGMII mode, 10 Mbps, Half-duplex	Supported	Supported	Not supported	Not supported
RGMII mode, 10 Mbps, Full-duplex	Supported	Supported	Not supported	Not supported
RGMII mode, 100 Mbps, Half-duplex	Supported	Supported	Supported	Not supported
RGMII mode, 100 Mbps, Full-duplex	Supported	Supported	Not supported	Supported
RGMII mode, 1000 Mbps, Half-duplex	Supported	Not supported	Not supported	Not supported
RGMII mode, 1000 Mbps, Full-duplex	Supported	Supported	Not supported	Not supported

Note 1. When using EtherCAT in MII mode with 100 Mbps and Full-duplex, ETHn_REFCLK (n = 0 to 2) output needs to be used for the reference clock of the external Ethernet PHY. And the PHY must synchronize the ETHn_TXCLK (n = 0 to 2) output with the ETHn_REFCLK.

Note 2. When using EtherCAT in RMII mode with 100 Mbps, Full-duplex, and Reference clock input, RMIIn_REFCLK (n = 0 to 2) output needs to be used for the reference clock of the external Ethernet PHY. And the PHY must synchronize the ETHn_RXCLK (n = 0 to 2) output with the RMIIn_REFCLK.

Note 3. When using EtherCAT in RMII mode with 100 Mpbs, Full-duplex, and Reference clock output, RMIIn_REFCLK (n = 0 to 2) output needs to be used for the reference clock of the external Ethernet PHY.

27.3.6 CONVRST : RGMII/RMII Converter Reset Control Register

Base address: ETHSS = 0x8011_0000

Offset address: 0x114

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PHYIR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PHYIR[3:0]	Release reset of RGMII/RMII Converter. Bit 0 = port 0, bit 1= port 1, bit 2 = port 2, bit 3 = port 3. 0: Reset RGMII/RMII Converter 1: Activate RGMII/RMII Converter	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The CONVRST register is used to control the reset for RGMII/RMII Converter.

27.3.7 SWCTRL : Switch Core Control Register

Base address: ETHSS = 0x8011_0000

Offset address: 0x304

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STRAP_HUB_ENB	STRAP_SX_ENB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	SET1000[2:0]			—	SET10[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SET10[2:0]	Port control to select use of 10 Mbps. Bit 0 = port 0, bit 1 = port 1, bit 2 = port 2. 0: Not 10 Mbps 1: 10 Mbps	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	SET1000[2:0]	Port control to select use of 1000 Mbps. Bit 0 = port 0, bit 1 = port 1, bit 2 = port 2. 0: Not 1000 Mbps 1: 1000 Mbps	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
16	STRAP_SX_ENB	Initialize switch after reset (set during module reset of ETHSW) Wiring both STRAP_SX_ENB and STRAP_HUB_ENB to 1 enables all ports including the management port. 0: Switch stay uninitialized 1: Switch initialized (line port)	R/W

Bit	Symbol	Function	R/W
17	STRAP_HUB_ENB	Initialize switch port 0 and 1 (set during module reset of ETHSW) Wiring both STRAP_SX_ENB and STRAP_HUB_ENB to 1 enables all ports including the management port. When STRAP_SX_ENB is 0: 0: Port 0 and 1 stay uninitialized 1: Port 0 and 1 initialized	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The SWCTRL register is used to control the Ethernet switch.

27.3.8 SWDUPC : Switch Core Duplex Mode Register

Base address: ETHSS = 0x8011_0000

Offset address: 0x308

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PHY_DUPLEX[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	PHY_DUPLEX[2:0]	Configure the MAC of each port for full-duplex or half-duplex operation. Bit 0 = port 0, bit 1 = port 1, bit 2 = port 2. This bit is ignored if HD_ENA bit of COMMAND_CONFIG_P[n] register is 0. It is recommended that these bits are set to 0. Use HD_ENA bit of COMMAND_CONFIG_Pn register instead. 0: Half-duplex 1: Full-duplex	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The SWDUPC register is used to switch the core duplex mode.

27.4 Operation

27.4.1 Ethernet Initializing Operation

For initializing operation, complete the following flowchart.

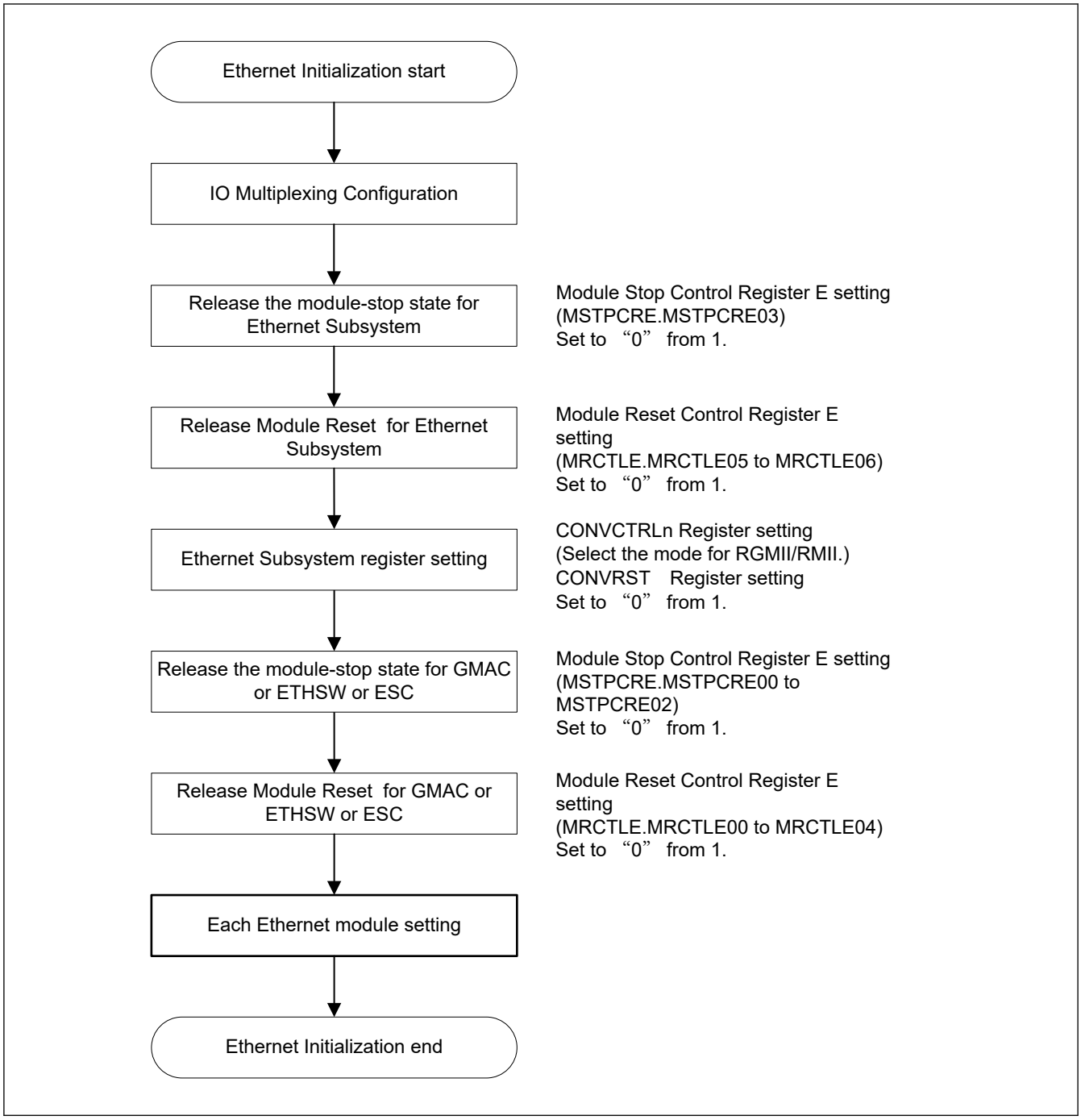


Figure 27.3 Ethernet initializing flowchart

27.5 Usage Notes

When MODCTRL.SW_MODE[2:0] is 100b or 101b (inter-module connection between ETHSW Port 1 and ESC Port 0), PHY Link signals for ETHSW Port 1 and ESC Port 0 should be handled as bellows,

- Do not select ESC_PHYLINK0 and ETHSW_PHYLINK1 as multiplexed pin configuration for any ports.
- Set PHYLINK.CATLNK[0] and PHYLINK.SWLINK[1] to the inverted value of reset value.

28. Ethernet MAC (GMAC)

28.1 Overview

The LSI includes three Ethernet GMAC.

Table 28.1 GMAC specifications

Item	Description
Number of units	• 3 Units
Features	• Compliance with the following standards: – IEEE 802.3-2008 for Ethernet MAC, Gigabit Media Independent Interface (GMII) & Media Independent Interface (MII) – IEEE 1588-2008 v2 standard for precision networked clock synchronization, IEEE 1588-2008 v2 is compliant with Power IEEE-C37.238 profile – IEEE 802.3-az, version D2.0 for Energy Efficient Ethernet (EEE) – Time Sensitive Networking • Enhancement to Scheduled Traffic (EST, 802.1Qbv) • Frame Preemption (FPE, 802.1Qbu / 802.3br) • Time-Based Scheduling • Support for 10/100/1000 Mbps data transfer rates • Support for both half-duplex and full-duplex operation • Programmable frame length to support both standard and "jumbo" ethernet frames with size up to 16 KB – Jumbo mode support in cut through mode only (not implemented in store & forward due to TX & RX fifo size) • 32 MAC Address registers for the Address Filter block • Variety of flexible addresses filtering modes are supported – Perfect (DA) address filters with masks for each byte – SA address comparison check with masks for each byte – 256-bit Hash filter for multicast and unicast (DA) addresses – Option to pass all multicast addressed frames – Promiscuous mode support to pass all frames without any filtering for network monitoring – Passes all incoming packets (as per filter) with a status report. – VLAN Filtering – L3 and L4 Filtering • Multi Queues Support up to 8 TX and 8 RX queues • Native DMA with simple independent channels Transmit and Receive engines – 8 RX channels, FIFO size 8 KB for receive channel – 8 TX channels, FIFO size 8 KB for transmit channel – Dynamic Mapping enables routing of the specific DMA channel by receive filter each packet – DMA implements dual buffer (ring) or linked list (chained) descriptor chaining • Advanced IEEE 1588-2002 & 2008 ethernet frame time stamping support – Timer module implemented – IEEE1588 Time base information, with reference clock of 125 MHz (same sources as GMII) – IEEE1588 external snapshot (Two signals) – PTP timestamp offload function – One-step timestamp • Programmable CRC generation and checking • TCP/IP offload / ARP offload features • Support for RMON statistics • Station Management Block, MDIO interface • Ethernet Energy Efficiency compliant with IEEE 802.3az-2010 – Support for wake up on LAN on magic packet and packet filtering – Support Energy Efficiency feature (LPI Mode) – Wake-up capability
Event link (input)	• PTP Timer capture
Module-stop function	• Module-stop state can be set to reduce power consumption.

Figure 28.1 is a block diagram of the Ethernet interface.

Gray modules and registers are described in section 27, Ethernet Subsystem, section 29, Ethernet Switch (ETHSW), and section 30, EtherCAT Slave Controller (ESC). See the related sections regarding the other registers and modules which are needed to operate the Ethernet MAC.

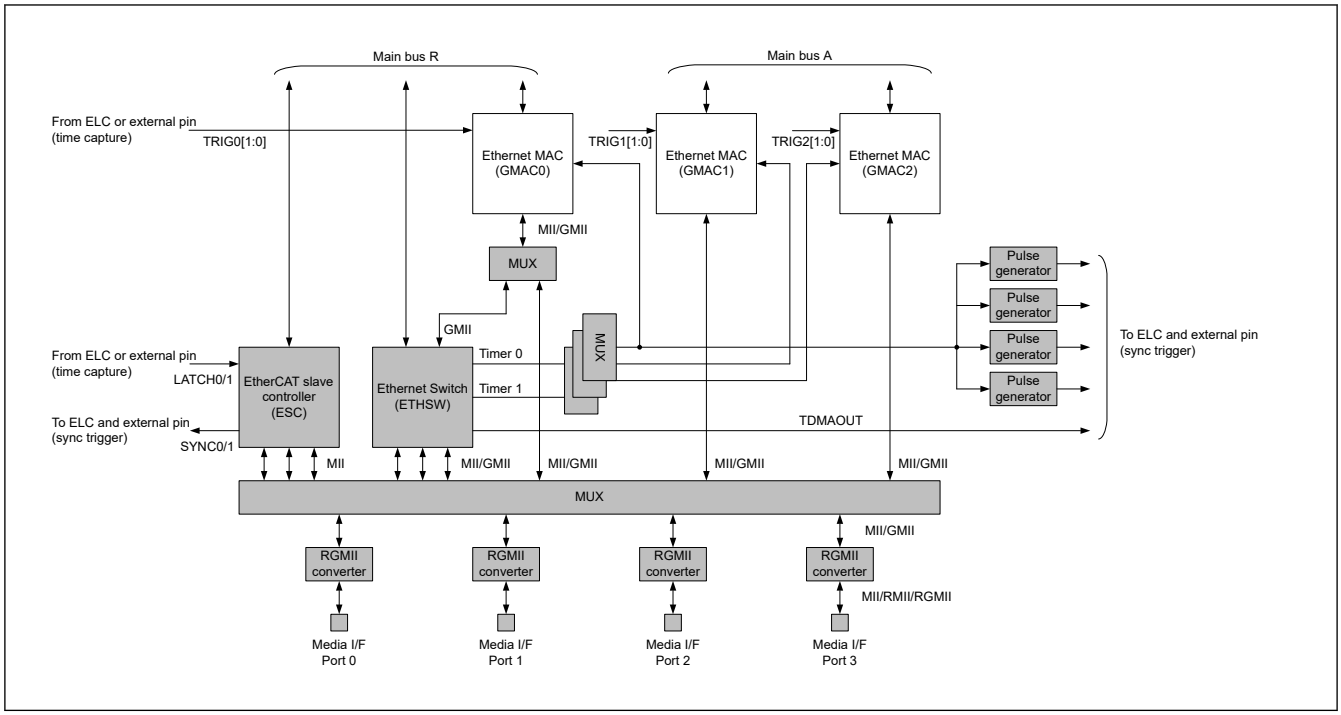


Figure 28.1 Block diagram of the Ethernet interface

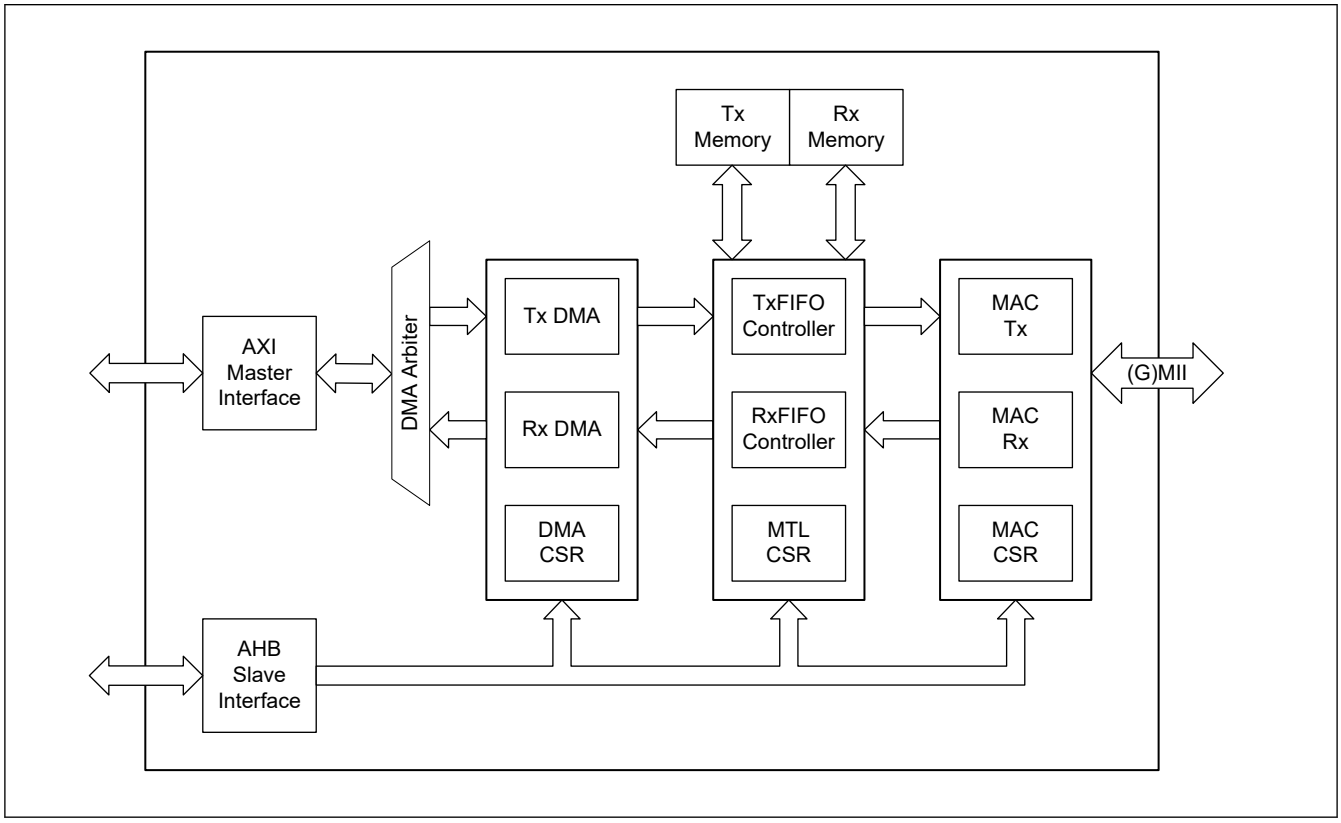


Figure 28.2 Block diagram of GMAC

Table 28.2 Input/output pins of GMAC (1 of 2)

Pin name	I/O	Function
GMACn_PTPTRG0	Input	PTP timer trigger external input 0
GMACn_PTPTRG1	Input	PTP timer trigger external input 1

Table 28.2 Input/output pins of GMAC (2 of 2)

Pin name	I/O	Function
GMACn_MDC	Output	Management data clock output pin
GMACn_MDIO	I/O	Management data I/O pin

Note: n = 0 to 2

28.2 Register Map

Table 28.3 GMAC register map (1 of 11)

Address	Register Symbol	Register Name	Write Protection
0x8011_0400	GMACTRGSEL	GMAC PTP Trigger Select register	PRCMD
0x8010_0000 (m = 0) 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Configuration	MAC Configuration Register	—
0x8010_0004 (m = 0) 0x9200_0004 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Ext_Configuration	MAC Extended Configuration Register	—
0x8010_0008 (m = 0) 0x9200_0008 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Packet_Filter	MAC Packet Filter register	—
0x8010_000C (m = 0) 0x9200_000C + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Watchdog_Timeout	Watchdog Timeout register	—
0x8010_0010 + 0x04 × n (m = 0) 0x9200_0010 + 0x04 × n + 0x1_0000 × (m - 1 + 0x04 × n) (m = 1, 2)	MAC_HASH_TABLE_REGn	Hash Table Register n (n = 0 to 7)	—
0x8010_0050 (m = 0) 0x9200_0050 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_VLAN_Tag_Ctrl	VLAN Tag Control Register	—
0x8010_0054 (m = 0) 0x9200_0054 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_VLAN_Tag_Data	VLAN Tag Data Register	—
0x8010_0058 (m = 0) 0x9200_0058 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_VLAN_Hash_Table	VLAN Hash Table Register	—
0x8010_0070 (m = 0) 0x9200_0070 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Q0_Tx_Flow_Ctrl	Flow Control register	—
0x8010_0090 (m = 0) 0x9200_0090 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Rx_Flow_Ctrl	Receive Flow Control register	—
0x8010_0094 (m = 0) 0x9200_0094 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_RxQ_Ctrl4	Receive Queue Control 4 register	—
0x8010_00A0 (m = 0) 0x9200_00A0 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_RxQ_Ctrl0	Receive Queue Control 0 register	—
0x8010_00A4 (m = 0) 0x9200_00A4 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_RxQ_Ctrl1	Receive Queue Control 1 register	—
0x8010_00A8 (m = 0) 0x9200_00A8 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_RxQ_Ctrl2	Receive Queue Control 2 register	—
0x8010_00AC (m = 0) 0x9200_00AC + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_RxQ_Ctrl3	Receive Queue Control 3 register	—
0x8010_00B0 (m = 0) 0x9200_00B0 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Interrupt_Status	MAC Interrupt Status Register	—
0x8010_00B4 (m = 0) 0x9200_00B4 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Interrupt_Enable	MAC Interrupt Enable Register	—
0x8010_00B8 (m = 0) 0x9200_00B8 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Rx_Tx_Status	Receive and Transmit Status Register	—
0x8010_00C0 (m = 0) 0x9200_00C0 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_PMT_Control_Status	PMT Control and Status Register	—
0x8010_00C4 (m = 0) 0x9200_00C4 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_RWK_Packet_Filter	Remote Wakeup Filter Register	—

Table 28.3 GMAC register map (2 of 11)

Address	Register Symbol	Register Name	Write Protection
0x8010_00D0 (m = 0) 0x9200_00D0 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_LPI_Control_Status	LPI Control and Status Register	—
0x8010_00D4 (m = 0) 0x9200_00D4 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_LPI_Timers_Control	LPI Timers Control Register	—
0x8010_00D8 (m = 0) 0x9200_00D8 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_LPI_Entry_Timer	LPI entry timer Register	—
0x8010_00DC (m = 0) 0x9200_00DC + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_1US_Tic_Counter	LP1 1us Tick Counter Register	—
0x8010_0110 (m = 0) 0x9200_0110 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Version	Version Register	—
0x8010_0114 (m = 0) 0x9200_0114 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Debug	MAC Debug Register	—
0x8010_011C (m = 0) 0x9200_011C + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_HW_Feature0	MAC Hardware Feature 0 Register	—
0x8010_0120 (m = 0) 0x9200_0120 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_HW_Feature1	MAC Hardware Feature 1 Register	—
0x8010_0124 (m = 0) 0x9200_0124 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_HW_Feature2	MAC Hardware Feature 2 Register	—
0x8010_0128 (m = 0) 0x9200_0128 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_HW_Feature3	MAC Hardware Feature 3 Register	—
0x8010_0200 (m = 0) 0x9200_0200 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_MDIO_Address	MDIO Address Register	—
0x8010_0204 (m = 0) 0x9200_0204 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_MDIO_Data	MDIO Data Register	—
0x8010_0210 (m = 0) 0x9200_0210 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_ARP_Address	ARP Address Register	—
0x8010_0230 (m = 0) 0x9200_0230 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_CSR_SW_Ctrl	CSR Software Control Register	—
0x8010_0234 (m = 0) 0x9200_0234 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_FPE_CTRL_STS	Frame Preemption Control and Status Register	—
0x8010_0238 (m = 0) 0x9200_0238 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Ext_Cfg1	Split Mode Control Register	—
0x8010_0300 (m = 0) 0x9200_0300 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_ADDRESS0_HIGH	MAC Address 0 High register	—
0x8010_0304 (m = 0) 0x9200_0304 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_ADDRESS0_LOW	MAC Address 0 Low register	—
0x8010_0300 + 0x08 × n (m = 0) 0x9200_0300 + 0x08 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_ADDRESSn_HIGH	MAC Address n High register (n = 1 to 31)	—
0x8010_0304 + 0x08 × n (m = 0) 0x9200_0304 + 0x08 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_ADDRESSn_LOW	MAC Address n Low register (n = 1 to 31)	—
0x8010_0700 (m = 0) 0x9200_0700 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_Control	MMC Control Register	—
0x8010_0704 (m = 0) 0x9200_0704 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_Rx_Interrupt	MMC Receive Interrupt Register	—
0x8010_0708 (m = 0) 0x9200_0708 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_Tx_Interrupt	MMC Transmit Interrupt Register	—
0x8010_070C (m = 0) 0x9200_070C + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_Rx_Interrupt_Mask	MMC Receive Interrupt Mask Register	—

Table 28.3 GMAC register map (3 of 11)

Address	Register Symbol	Register Name	Write Protection
0x8010_0710 (m = 0) 0x9200_0710 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_Tx_Interrupt_Mask	MMC Transmit Interrupt Mask Register	—
0x8010_0714 (m = 0) 0x9200_0714 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Octet_Count_Good_Bad	Transmit Octet Count for Good and Bad 64 Byte Packets	—
0x8010_0718 (m = 0) 0x9200_0718 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Packet_Count_Good_Bad	Transmit Packet Count for Good and Bad Packets	—
0x8010_071C (m = 0) 0x9200_071C + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Broadcast_Packets_Good	Transmit Packet Count for Good Broadcast Packets	—
0x8010_0720 (m = 0) 0x9200_0720 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Multicast_Packets_Good	Transmit Packet Count for Good Multicast Packets	—
0x8010_0724 (m = 0) 0x9200_0724 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_64Octets_Packets_Good_Bad	Transmit Octet Count for Good and Bad 64 Byte Packets	—
0x8010_0728 (m = 0) 0x9200_0728 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_65To127Octets_Packets_Good_Bad	Transmit Octet Count for Good and Bad 65 to 127 Bytes Packets	—
0x8010_072C (m = 0) 0x9200_072C + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_128To255Octets_Packets_Good_Bad	Transmit Octet Count for Good and Bad 128 to 255 Bytes Packets	—
0x8010_0730 (m = 0) 0x9200_0730 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_256To511Octets_Packets_Good_Bad	Transmit Octet Count for Good and Bad 256 to 511 Bytes Packets	—
0x8010_0734 (m = 0) 0x9200_0734 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_512To1023Octets_Packets_Good_Bad	Transmit Octet Count for Good and Bad 512 to 1023 Bytes Packets	—
0x8010_0738 (m = 0) 0x9200_0738 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_1024ToMaxOctets_Packets_Good_Bad	Transmit Octet Count for Good and Bad 1024 to Maxsize Bytes Packets	—
0x8010_073C (m = 0) 0x9200_073C + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Unicast_Packets_Good_Bad	Transmit Packet Count for Good and Bad Unicast Packets	—
0x8010_0740 (m = 0) 0x9200_0740 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Multicast_Packets_Good_Bad	Transmit Packet Count for Good and Bad Multicast Packets	—
0x8010_0744 (m = 0) 0x9200_0744 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Broadcast_Packets_Good_Bad	Transmit Packet Count for Good and Bad Broadcast Packets	—
0x8010_0748 (m = 0) 0x9200_0748 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Underflow_Error_Packets	Transmit Packet Count for Underflow Error Packets	—
0x8010_074C (m = 0) 0x9200_074C + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Single_Collision_Good_Packets	Transmit Packet Count for Packets Transmitted after Single Collision	—
0x8010_0750 (m = 0) 0x9200_0750 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Multiple_Collision_Good_Packets	Transmit Packet Count for Packets Transmitted after Multiple Collision	—
0x8010_0754 (m = 0) 0x9200_0754 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Deferred_Packets	Transmit Packet Count for Deferred Packets	—
0x8010_0758 (m = 0) 0x9200_0758 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Late_Collision_Packets	Transmit Packet Count for Late Collision Error Packets	—
0x8010_075C (m = 0) 0x9200_075C + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Excessive_Collision_Packets	Transmit Packet Count for Excessive Collision Error Packets	—
0x8010_0760 (m = 0) 0x9200_0760 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Carrier_Error_Packets	Transmit Packet Count for Carrier Sense Error Packets	—
0x8010_0764 (m = 0) 0x9200_0764 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Octet_Count_Good	Transmit Octet Count for Good Packets	—
0x8010_0768 (m = 0) 0x9200_0768 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Packet_Count_Good	Transmit Packet Count for Good Packets	—
0x8010_076C (m = 0) 0x9200_076C + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Excessive_Deferral_Error	Transmit Packet Count for Excessive Deferral Error Packets	—
0x8010_0770 (m = 0) 0x9200_0770 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_Pause_Packets	Transmit Packet Count for Good PAUSE Packets	—

Table 28.3 GMAC register map (4 of 11)

Address	Register Symbol	Register Name	Write Protection
0x8010_0774 (m = 0) 0x9200_0774 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_VLAN_Packets_Good	Transmit Packet Count for Good VLAN Packets	—
0x8010_0778 (m = 0) 0x9200_0778 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_OSize_Packets_Good	Transmit Packet Count for Good Oversize Packets	—
0x8010_0780 (m = 0) 0x9200_0780 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Packets_Count_Good_Bad	Receive Packet Count for Good and Bad Packets	—
0x8010_0784 (m = 0) 0x9200_0784 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Octet_Count_Good_Bad	Receive Octet Count for Good and Bad Packets	—
0x8010_0788 (m = 0) 0x9200_0788 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Octet_Count_Good	Receive Octet Count for Good Packets	—
0x8010_078C (m = 0) 0x9200_078C + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Broadcast_Packets_Good	Receive Packet Count for Good Broadcast Packets	—
0x8010_0790 (m = 0) 0x9200_0790 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Multicast_Packets_Good	Receive Packet Count for Good Multicast Packets	—
0x8010_0794 (m = 0) 0x9200_0794 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_CRC_Error_Packets	Receive Packet Count for CRC Error Packets	—
0x8010_0798 (m = 0) 0x9200_0798 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Alignment_Error_Packets	Receive Packet Count for Alignment Error Packets	—
0x8010_079C (m = 0) 0x9200_079C + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Runt_Error_Packets	Receive Packet Count for Runt Error Packets	—
0x8010_07A0 (m = 0) 0x9200_07A0 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Jabber_Error_Packets	Receive Packet Count for Jabber Error Packets	—
0x8010_07A4 (m = 0) 0x9200_07A4 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Undersize_Packets_Good	Receive Packet Count for Undersize Packets	—
0x8010_07A8 (m = 0) 0x9200_07A8 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Oversize_Packets_Good	Receive Packet Count for Oversize Packets	—
0x8010_07AC (m = 0) 0x9200_07AC + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_64Octets_Packets_Good_Bad	Receive Packet Count for Good and Bad 64 Byte Packets	—
0x8010_07B0 (m = 0) 0x9200_07B0 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_65To127Octets_Packets_Good_Bad	Receive Packet Count for Good and Bad 65 to 127 Bytes Packets	—
0x8010_07B4 (m = 0) 0x9200_07B4 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_128To255Octets_Packets_Good_Bad	Receive Packet Count for Good and Bad 128 to 255 Bytes Packets	—
0x8010_07B8 (m = 0) 0x9200_07B8 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_256To511Octets_Packets_Good_Bad	Receive Packet Count for Good and Bad 256 to 511 Bytes Packets	—
0x8010_07BC (m = 0) 0x9200_07BC + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_512To1023Octets_Packets_Good_Bad	Receive Packet Count for Good and Bad 512 to 1,023 Bytes Packets	—
0x8010_07C0 (m = 0) 0x9200_07C0 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_1024ToMaxOctets_Packets_Good_Bad	Receive Packet Count for Good and Bad 1,024 to Maxsize Bytes Packets	—
0x8010_07C4 (m = 0) 0x9200_07C4 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Unicast_Packets_Good	Receive Packet Count for Good Unicast Packets	—
0x8010_07C8 (m = 0) 0x9200_07C8 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Length_Error_Packets	Receive Packet Count for Length Error Packets	—
0x8010_07CC (m = 0) 0x9200_07CC + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Out_Of_Range_Type_Packets	Receive Packet Count for Out of Range Packets	—
0x8010_07D0 (m = 0) 0x9200_07D0 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Pause_Packets	Receive Packet Count for PAUSE Packets	—
0x8010_07D4 (m = 0) 0x9200_07D4 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_FIFO_Overflow_Packets	Receive Packet Count for FIFO Overflow Packets	—
0x8010_07D8 (m = 0) 0x9200_07D8 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_VLAN_Packets_Good_Bad	Receive Packet Count for Good and Bad VLAN Packets	—

Table 28.3 GMAC register map (5 of 11)

Address	Register Symbol	Register Name	Write Protection
0x8010_07DC (m = 0) 0x9200_07DC + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Watchdog_Error_Packets	Receive Packet Count for Watchdog Error Packets	—
0x8010_07E0 (m = 0) 0x9200_07E0 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Receive_Error_Packets	Receive Packet Count for Receive Error Packets	—
0x8010_07E4 (m = 0) 0x9200_07E4 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_Control_Packets_Good	Receive Packet Count for Good Control Packets	—
0x8010_07EC (m = 0) 0x9200_07EC + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_LPI_USEC_Cntr	Tx LPI Microseconds Counter	—
0x8010_07F0 (m = 0) 0x9200_07F0 + 0x1_0000 × (m - 1) (m = 1, 2)	Tx_LPI_Tran_Cntr	Tx LPI Transition Counter	—
0x8010_07F4 (m = 0) 0x9200_07F4 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_LPI_USEC_Cntr	Rx LPI Microseconds Counter	—
0x8010_07F8 (m = 0) 0x9200_07F8 + 0x1_0000 × (m - 1) (m = 1, 2)	Rx_LPI_Tran_Cntr	Rx LPI Transition Counter	—
0x8010_0800 (m = 0) 0x9200_0800 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_IPC_Rx_Interrupt_Mask	MMC Receive Checksum Off load Interrupt Mask Register	—
0x8010_0808 (m = 0) 0x9200_0808 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_IPC_Rx_Interrupt	MMC Receive Checksum Off load Interrupt Register	—
0x8010_0810 (m = 0) 0x9200_0810 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv4_Good_Packets	RxIPv4 Good Packets Counter	—
0x8010_0814 (m = 0) 0x9200_0814 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv4_Header_Error_Packets	RxIPv4 Header Error Packets Counter	—
0x8010_0818 (m = 0) 0x9200_0818 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv4_No_Payload_Packets	RxIPv4 Payload Packets Counter	—
0x8010_081C (m = 0) 0x9200_081C + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv4_Fragmented_Packets	RxIPv4 Fragmented Packets Counter	—
0x8010_0820 (m = 0) 0x9200_0820 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv4_UDP_Checksum_Disabled_Packets	RxIPv4 UDP Checksum Disabled Packets Counter	—
0x8010_0824 (m = 0) 0x9200_0824 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv6_Good_Packets	RxIPv6 Good Packets Counter	—
0x8010_0828 (m = 0) 0x9200_0828 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv6_Header_Error_Packets	RxIPv6 Header Error Packets Counter	—
0x8010_082C (m = 0) 0x9200_082C + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv6_No_Payload_Packets	RxIPv6 No Payload Packets Counter	—
0x8010_0830 (m = 0) 0x9200_0830 + 0x1_0000 × (m - 1) (m = 1, 2)	RxUDP_Good_Packets	RxUDP Good Packets Counter	—
0x8010_0834 (m = 0) 0x9200_0834 + 0x1_0000 × (m - 1) (m = 1, 2)	RxUDP_Error_Packets	RxUDP Error Packets Counter	—
0x8010_0838 (m = 0) 0x9200_0838 + 0x1_0000 × (m - 1) (m = 1, 2)	RxTCP_Good_Packets	RxTCP Good Packets Counter	—
0x8010_083C (m = 0) 0x9200_083C + 0x1_0000 × (m - 1) (m = 1, 2)	RxTCP_Error_Packets	RxTCP Error Packets Counter	—
0x8010_0840 (m = 0) 0x9200_0840 + 0x1_0000 × (m - 1) (m = 1, 2)	RxICMP_Good_Packets	RxICMP Good Packets Counter	—
0x8010_0844 (m = 0) 0x9200_0844 + 0x1_0000 × (m - 1) (m = 1, 2)	RxICMP_Error_Packets	RxICMP Error Packets Counter	—
0x8010_0850 (m = 0) 0x9200_0850 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv4_Good_Octets	RxIPv4 Good Octets Counter	—
0x8010_0854 (m = 0) 0x9200_0854 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv4_Header_Error_Octets	RxIPv4 Header Error Octets Counter	—

Table 28.3 GMAC register map (6 of 11)

Address	Register Symbol	Register Name	Write Protection
0x8010_0858 (m = 0) 0x9200_0858 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv4_No_Payload_Octets	RxIPv4 No Payload Octets Counter	—
0x8010_085C (m = 0) 0x9200_085C + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv4_Fragmented_Octets	RxIPv4 Fragmented Octets Counter	—
0x8010_0860 (m = 0) 0x9200_0860 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv4_UDP_Checksum_Disable_Octets	RxIPv4 UDP Checksum Disable Octets Counter	—
0x8010_0864 (m = 0) 0x9200_0864 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv6_Good_Octets	RxIPv6 Good Octets Counter	—
0x8010_0868 (m = 0) 0x9200_0868 + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv6_Header_Error_Octets	RxIPv6 Header Error Octets Counter	—
0x8010_086C (m = 0) 0x9200_086C + 0x1_0000 × (m - 1) (m = 1, 2)	RxIPv6_No_Payload_Octets	RxIPv6 No Payload Octets Counter	—
0x8010_0870 (m = 0) 0x9200_0870 + 0x1_0000 × (m - 1) (m = 1, 2)	RxUDP_Good_Octets	RxUDP Good Octets Counter	—
0x8010_0874 (m = 0) 0x9200_0874 + 0x1_0000 × (m - 1) (m = 1, 2)	RxUDP_Error_Octets	RxUDP Error Octets Counter	—
0x8010_0878 (m = 0) 0x9200_0878 + 0x1_0000 × (m - 1) (m = 1, 2)	RxTCP_Good_Octets	RxTCP Good Octets Counter	—
0x8010_087C (m = 0) 0x9200_087C + 0x1_0000 × (m - 1) (m = 1, 2)	RxTCP_Error_Octets	RxTCP Error Octets Counter	—
0x8010_0880 (m = 0) 0x9200_0880 + 0x1_0000 × (m - 1) (m = 1, 2)	RxICMP_Good_Octets	RxICMP Good Octets Counter	—
0x8010_0884 (m = 0) 0x9200_0884 + 0x1_0000 × (m - 1) (m = 1, 2)	RxICMP_Error_Octets	RxICMP Error Octets Counter	—
0x8010_08A0 (m = 0) 0x9200_08A0 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_FPE_Tx_Interrupt	MMC FPE Transmit Interrupt Register	—
0x8010_08A4 (m = 0) 0x9200_08A4 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_FPE_Tx_Interrupt_Mask	MMC FPE Transmit Interrupt Mask Register	—
0x8010_08A8 (m = 0) 0x9200_08A8 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_Tx_FPE_Fragment_Cntr	Tx FPE Fragment Counter	—
0x8010_08AC (m = 0) 0x9200_08AC + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_Tx_Hold_Req_Cntr	Tx Hold Request Counter	—
0x8010_08C0 (m = 0) 0x9200_08C0 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_FPE_Rx_Interrupt	MMC FPE Receive Interrupt Register	—
0x8010_08C4 (m = 0) 0x9200_08C4 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_FPE_Rx_Interrupt_Mask	MMC FPE Receive Interrupt Mask Register	—
0x8010_08C8 (m = 0) 0x9200_08C8 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_Rx_Packet_Assembly_Err_Cntr	Rx Packet Assembly Error Counter	—
0x8010_08CC (m = 0) 0x9200_08CC + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_Rx_Packet_SMD_Err_Cntr	Rx Packet SMD Error Counter	—
0x8010_08D0 (m = 0) 0x9200_08D0 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_Rx_Packet_Assembly_OK_Cntr	Rx Packet Assembly OK Counter	—
0x8010_08D4 (m = 0) 0x9200_08D4 + 0x1_0000 × (m - 1) (m = 1, 2)	MMC_Rx_FPE_Fragment_Cntr	Rx FPE Fragment Counter	—
0x8010_0900 + 0x30 × n (m = 0) 0x9200_0900 + 0x30 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_L3_L4_CONTROLn	MAC Layer 3 and Layer 4 Control Register (n = 0 to 7)	—
0x8010_0904 + 0x30 × n (m = 0) 0x9200_0904 + 0x30 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_LAYER4_ADDRESSn	MAC Layer 4 Address n Register (n = 0 to 7)	—

Table 28.3 GMAC register map (7 of 11)

Address	Register Symbol	Register Name	Write Protection
0x8010_0910 + 0x30 × n (m = 0) 0x9200_0910 + 0x30 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_LAYER3_ADDR0_REG n	MAC Layer 3 Address 0 Register n (n = 0 to 7)	—
0x8010_0914 + 0x30 × n (m = 0) 0x9200_0914 + 0x30 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_LAYER3_ADDR1_REG n	MAC Layer 3 Address 1 Register n (n = 0 to 7)	—
0x8010_0918 + 0x30 × n (m = 0) 0x9200_0918 + 0x30 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_LAYER3_ADDR2_REG n	MAC Layer 3 Address 2 Register n (n = 0 to 7)	—
0x8010_091C + 0x30 × n (m = 0) 0x9200_091C + 0x30 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_LAYER3_ADDR3_REG n	MAC Layer 3 Address 3 Register n (n = 0 to 7)	—
0x8010_0A70 (m = 0) 0x9200_0A70 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Indir_Access_Ctrl	MAC Indirect Access Control and Status Register	—
0x8010_0A74 (m = 0) 0x9200_0A74 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Indir_Access_Data	MAC Indirect Access Data Register	—
0x8010_0B00 (m = 0) 0x9200_0B00 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Timestamp_Control	Timestamp Control Register	—
0x8010_0B08 (m = 0) 0x9200_0B08 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_System_Time_Seconds	Timestamp Seconds Register	—
0x8010_0B0C (m = 0) 0x9200_0B0C + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_System_Time_Nanoseconds	Timestamp Nanoseconds Register	—
0x8010_0B20 (m = 0) 0x9200_0B20 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Timestamp_Status	Timestamp Status Register	—
0x8010_0B30 (m = 0) 0x9200_0B30 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Tx_Timestamp_Status_Nanoseconds	Transmit Timestamp Status Nanoseconds Register	—
0x8010_0B34 (m = 0) 0x9200_0B34 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Tx_Timestamp_Status_Seconds	Transmit Timestamp Status Seconds Register	—
0x8010_0B40 (m = 0) 0x9200_0B40 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Auxiliary_Control	Auxiliary Timestamp Control Register	—
0x8010_0B48 (m = 0) 0x9200_0B48 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Auxiliary_Timestamp_Nanoseconds	Auxiliary Timestamp Nanoseconds Register	—
0x8010_0B4C (m = 0) 0x9200_0B4C + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Auxiliary_Timestamp_Seconds	Auxiliary Timestamp Seconds Register	—
0x8010_0B50 (m = 0) 0x9200_0B50 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Timestamp_Ingress_Asym_Corr	Timestamp Ingress Asymmetry Correction Register	—
0x8010_0B54 (m = 0) 0x9200_0B54 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Timestamp_Egress_Asym_Corr	Timestamp Egress Asymmetry Correction Register	—
0x8010_0B58 (m = 0) 0x9200_0B58 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Timestamp_Ingress_Corr_Nanosecond	Timestamp Ingress Correction Nanosecond Register	—
0x8010_0B5C (m = 0) 0x9200_0B5C + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Timestamp_Egress_Corr_Nanosecond	Timestamp Egress Correction Nanosecond Register	—
0x8010_0B60 (m = 0) 0x9200_0B60 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Timestamp_Ingress_Corr_Subnanosec	Timestamp Ingress Correction Subnanosecond Register	—
0x8010_0B64 (m = 0) 0x9200_0B64 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Timestamp_Egress_Corr_Subnanosec	Timestamp Egress Correction Subnanosecond Register	—
0x8010_0B68 (m = 0) 0x9200_0B68 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Timestamp_Ingress_Latency	Timestamp Ingress Latency Register	—
0x8010_0B6C (m = 0) 0x9200_0B6C + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Timestamp_Egress_Latency	Timestamp Egress Latency Register	—
0x8010_0BC0 (m = 0) 0x9200_0BC0 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_PTO_Control	PTP Offload Engine Control Register	—

Table 28.3 GMAC register map (8 of 11)

Address	Register Symbol	Register Name	Write Protection
0x8010_0BC4 (m = 0) 0x9200_0BC4 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Source_Port_Identity0	PTP Source Port Identity 0 Register	—
0x8010_0BC8 (m = 0) 0x9200_0BC8 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Source_Port_Identity1	PTP Source Port Identity 1 Register	—
0x8010_0BCC (m = 0) 0x9200_0BCC + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Source_Port_Identity2	PTP Source Port Identity 2 Register	—
0x8010_0BD0 (m = 0) 0x9200_0BD0 + 0x1_0000 × (m - 1) (m = 1, 2)	MAC_Log_Message_Interval	PTP Log Message Interval Register	—
0x8010_0C00 (m = 0) 0x9200_0C00 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_Operation_Mode	MTL Operation Mode Register	—
0x8010_0C20 (m = 0) 0x9200_0C20 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_Interrupt_Status	MTL Interrupt Status Register	—
0x8010_0C30 (m = 0) 0x9200_0C30 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_RxQ_DMA_Map0	Receive Queue and DMA Channel Mapping 0 Register	—
0x8010_0C34 (m = 0) 0x9200_0C34 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_RxQ_DMA_Map1	Receive Queue and DMA Channel Mapping 1 Register	—
0x8010_0C40 (m = 0) 0x9200_0C40 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TBS_CTRL	Time Based Scheduling Control Register	—
0x8010_0C50 (m = 0) 0x9200_0C50 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_EST_Control	EST Control Register	—
0x8010_0C54 (m = 0) 0x9200_0C54 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_EST_Ext_Control	EST Extended Control Register	—
0x8010_0C58 (m = 0) 0x9200_0C58 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_EST_Status	EST Status Register	—
0x8010_0C60 (m = 0) 0x9200_0C60 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_EST_Sch_Error	EST Scheduing Error Register	—
0x8010_0C64 (m = 0) 0x9200_0C64 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_EST_Frm_Size_Error	EST Frame Size Error Register	—
0x8010_0C68 (m = 0) 0x9200_0C68 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_EST_Frm_Size_Capture	EST Frame Size Error Capture Register	—
0x8010_0C70 (m = 0) 0x9200_0C70 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_EST_Intr_Enable	EST Interrupt Enable Register	—
0x8010_0C80 (m = 0) 0x9200_0C80 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_EST_GCL_Control	EST GCL Control Register	—
0x8010_0C84 (m = 0) 0x9200_0C84 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_EST_GCL_Data	EST GCL Data Register	—
0x8010_0C90 (m = 0) 0x9200_0C90 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_FPE_CTRL_STS	FPE Control and Status Register	—
0x8010_0C94 (m = 0) 0x9200_0C94 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_FPE_Advance	FPE Advance Time Register	—
0x8010_0D00 (m = 0) 0x9200_0D00 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQ0_OPERATION_MODE	Queue 0 Transmit Operation Mode Register	—
0x8010_0D04 (m = 0) 0x9200_0D04 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQ0_UNDERFLOW	Queue 0 Underflow Counter Register	—
0x8010_0D08 (m = 0) 0x9200_0D08 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQ0_DEBUG	Queue 0 Transmit Debug Register	—
0x8010_0D14 (m = 0) 0x9200_0D14 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQ0_ETS_STATUS	Queue 0 ETS Status Register	—
0x8010_0D18 (m = 0) 0x9200_0D18 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQ0_QUANTUM_WEIGHT	Queue 0 Quantum or Weights Register	—

Table 28.3 GMAC register map (9 of 11)

Address	Register Symbol	Register Name	Write Protection
0x8010_0D2C (m = 0) 0x9200_0D2C + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_Q0_INTERRUPT_CONTROL_STATUS	Queue 0 Interrupt Control and Status Register	—
0x8010_0D30 (m = 0) 0x9200_0D30 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_RXQ0_OPERATION_MODE	Queue 0 Receive Operation Mode Register	—
0x8010_0D34 (m = 0) 0x9200_0D34 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT	Queue 0 Missed Packet and Overflow Counter Register	—
0x8010_0D38 (m = 0) 0x9200_0D38 + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_RXQ0_DEBUG	Queue 0 Receive Debug Register	—
0x8010_0D3C (m = 0) 0x9200_0D3C + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_RXQ0_CONTROL	Queue 0 Receive Control Register	—
0x8010_0D00 + 0x40 × n (m = 0) 0x9200_0D00 + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQn_OPERATIOn_MODE	Queue n Transmit Operation Mode Register (n = 1 to 7)	—
0x8010_0D04 + 0x40 × n (m = 0) 0x9200_0D04 + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQn_UNDERFLOW	Queue n Underflow Counter Register (n = 1 to 7)	—
0x8010_0D08 + 0x40 × n (m = 0) 0x9200_0D08 + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQn_DEBUG	Queue n Transmit Debug Register (n = 1 to 7)	—
0x8010_0D10 + 0x40 × n (m = 0) 0x9200_0D10 + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQn_ETS_CONTROL	Queue n ETS Control Register (n = 1 to 7)	—
0x8010_0D14 + 0x40 × n (m = 0) 0x9200_0D14 + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQn_ETS_STATUS	Queue n ETS Status Register (n = 1 to 7)	—
0x8010_0D18 + 0x40 × n (m = 0) 0x9200_0D18 + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQn_QUANTUM_WEIGHT	Queue n Quantum or Weights Register (n = 1 to 7)	—
0x8010_0D1C + 0x40 × n (m = 0) 0x9200_0D1C + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQn_SENDSLOPECREDIT	Queue n CBS SendSlope Credit Register (n = 1 to 7)	—
0x8010_0D20 + 0x40 × n (m = 0) 0x9200_0D20 + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQn_HICREDIT	Queue n CBS Hi Credit Register (n = 1 to 7)	—
0x8010_0D24 + 0x40 × n (m = 0) 0x9200_0D24 + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_TXQn_LOCREDIT	Queue n CBS Lo Credit Register (n = 1 to 7)	—
0x8010_0D2C + 0x40 × n (m = 0) 0x9200_0D2C + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_Qn_INTERRUPT_CONTROL_STATUS	Queue n Interrupt Control and Status Register (n = 1 to 7)	—
0x8010_0D30 + 0x40 × n (m = 0) 0x9200_0D30 + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_RXQn_OPERATIOn_MODE	Queue n Receive Operation Mode Register (n = 1 to 7)	—
0x8010_0D34 + 0x40 × n (m = 0) 0x9200_0D34 + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_RXQn_MISSED_PACKET_OVERFLOW_CNT	Queue n Missed Packet and Overflow Counter Register (n = 1 to 7)	—
0x8010_0D38 + 0x40 × n (m = 0) 0x9200_0D38 + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_RXQn_DEBUG	Queue n Receive Debug Register (n = 1 to 7)	—
0x8010_0D3C + 0x40 × n (m = 0) 0x9200_0D3C + 0x40 × n + 0x1_0000 × (m - 1) (m = 1, 2)	MTL_RXQn_CONTROL	Queue n Receive Control Register (n = 1 to 7)	—
0x8010_1000 (m = 0) 0x9200_1000 + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_Mode	Bus Mode Register	—

Table 28.3 GMAC register map (10 of 11)

Address	Register Symbol	Register Name	Write Protection
0x8010_1004 (m = 0) 0x9200_1004 + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_SysBus_Mode	System Bus Mode Register	—
0x8010_1008 (m = 0) 0x9200_1008 + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_Interrupt_Status	DMA Interrupt Status Register	—
0x8010_100C (m = 0) 0x9200_100C + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_Debug_Status0	DMA Debug Status 0 Register	—
0x8010_1010 (m = 0) 0x9200_1010 + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_Debug_Status1	DMA Debug Status 1 Register	—
0x8010_1014 (m = 0) 0x9200_1014 + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_Debug_Status2	DMA Debug Status 2 Register	—
0x8010_1040 (m = 0) 0x9200_1040 + 0x1_0000 × (m - 1) (m = 1, 2)	AXI_LPI_Entry_Interval	AXI LPI Entry Interval Register	—
0x8010_1050 (m = 0) 0x9200_1050 + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_TBS_CTRL0	DMA TBS Attributes Control 0 Register	—
0x8010_1054 (m = 0) 0x9200_1054 + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_TBS_CTRL1	DMA TBS Attributes Control 1 Register	—
0x8010_1058 (m = 0) 0x9200_1058 + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_TBS_CTRL2	DMA TBS Attributes Control 2 Register	—
0x8010_105C (m = 0) 0x9200_105C + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_TBS_CTRL3	DMA TBS Attributes Control 3 Register	—
0x8010_1100 + 0x80 × n (m = 0) 0x9200_1100 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_CONTROL	DMA Channel n Control Register (n = 0 to 7)	—
0x8010_1104 + 0x80 × n (m = 0) 0x9200_1104 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_TX_CONTROL	DMA Channel n Transmit Control Register (n = 0 to 7)	—
0x8010_1108 + 0x80 × n (m = 0) 0x9200_1108 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_RX_CONTROL	DMA Channel n Receive Control Register (n = 0 to 7)	—
0x8010_1114 + 0x80 × n (m = 0) 0x9200_1114 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_TXDESC_LIST_ADDRESS	DMA Channel n Tx Descriptor List Address Register (n = 0 to 7)	—
0x8010_111C + 0x80 × n (m = 0) 0x9200_111C + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_RXDESC_LIST_ADDRESS	DMA Channel n Rx Descriptor List Address Register (n = 0 to 7)	—
0x8010_1120 + 0x80 × n (m = 0) 0x9200_1120 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_TXDESC_TAIL_POINTER	DMA Channel n Tx Descriptor Tail Pointer Register (n = 0 to 7)	—
0x8010_1128 + 0x80 × n (m = 0) 0x9200_1128 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_RXDESC_TAIL_POINTER	DMA Channel n Rx Descriptor Tail Pointer Register (n = 0 to 7)	—
0x8010_112C + 0x80 × n (m = 0) 0x9200_112C + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_TXDESC_RING_LENGTH	DMA Channel n Tx Descriptor Ring Length Register (n = 0 to 7)	—
0x8010_1130 + 0x80 × n (m = 0) 0x9200_1130 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_RX_CONTROL2	DMA Channel n Receive Control 2 Register (n = 0 to 7)	—
0x8010_1134 + 0x80 × n (m = 0) 0x9200_1134 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_INTERRUPT_ENABLE	DMA Channel n Interrupt Enable Register (n = 0 to 7)	—
0x8010_1138 + 0x80 × n (m = 0) 0x9200_1138 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_RX_INTERRUPT_WATCHDOG_TIMER	DMA Channel n Receive Interrupt Watchdog Timer Register (n = 0 to 7)	—

Table 28.3 GMAC register map (11 of 11)

Address	Register Symbol	Register Name	Write Protection
0x8010_113C + 0x80 × n (m = 0) 0x9200_113C + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_SLOT_FUNCTION_CONTROL_STATUS	DMA Channel n Slot Function Control and Status Register (n = 0 to 7)	—
0x8010_1144 + 0x80 × n (m = 0) 0x9200_1144 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_CURRENT_APP_TXDESC	DMA Channel n Current Application Transmit Descriptor Register (n = 0 to 7)	—
0x8010_114C + 0x80 × n (m = 0) 0x9200_114C + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_CURRENT_APP_RXDESC	DMA Channel n Current Application Receive Descriptor Register (n = 0 to 7)	—
0x8010_1154 + 0x80 × n (m = 0) 0x9200_1154 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_CURRENT_APP_TXBUFFER	DMA Channel n Current Application Transmit Buffer Address Register (n = 0 to 7)	—
0x8010_115C + 0x80 × n (m = 0) 0x9200_115C + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_CURRENT_APP_RXBUFFER	DMA Channel n Current Application Receive Buffer Address Register (n = 0 to 7)	—
0x8010_1160 + 0x80 × n (m = 0) 0x9200_1160 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_STATUS	DMA Channel n Status Register (n = 0 to 7)	—
0x8010_1164 + 0x80 × n (m = 0) 0x9200_1164 + 0x80 × n + 0x1_0000 × (m - 1) (m = 1, 2)	DMA_CHn_MISS_FRAME_CNT	DMA Channel n Miss Packet Counter (n = 0 to 7)	—

Table 28.4 GMAC related system control register

Register / Unit	Module Reset- Control Register	Module Stop Control Register	Slave Access Control Register
GMACTRGSEL	MRCTLE.MRTCLE05	MSTPCRE.MSTPCRE03	SLVACCCTL6.ETHSS_SL
All other registers for Unit 0	MRCTLE.MRCTLE[01:00]	MSTPCRE.MSTPCRE00	SLVACCCTL6.GMAC0_SL
All other registers for Unit 1	MRCTLE.MRCTLE[17:16]	MSTPCRE.MSTPCRE16	SLVACCCTL4.GMAC1_SL
All other registers for Unit 2	MRCTLE.MRCTLE[19:18]	MSTPCRE.MSTPCRE17	SLVACCCTL4.GMAC2_SL

28.3 Register Descriptions

28.3.1 GMACTRGSEL : GMAC PTP Trigger Select register

Base address: GMACC = 0x8011_0400

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	G2TRGSEL1	G2TRGSEL0	G1TRGSEL1	G1TRGSEL0	G0TRGSEL1	G0TRGSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	G0TRGSEL0	Select PTP Timestamp Trigger 0 for GMAC0 0: Select GMAC0_PTPTRG0 trigger from ELC 1: Select external GMAC0_PTPTRG0 trigger input pin	R/W

Bit	Symbol	Function	R/W
1	G0TRGSEL1	Select PTP Timestamp Trigger 1 for GMAC0 0: Select GMAC0_PTPTRG1 trigger from ELC 1: Select external GMAC0_PTPTRG1 trigger input pin	R/W
2	G1TRGSEL0	Select PTP Timestamp Trigger 0 for GMAC1 0: Select GMAC1_PTPTRG0 trigger from ELC 1: Select external GMAC1_PTPTRG0 trigger input pin	R/W
3	G1TRGSEL1	Select PTP Timestamp Trigger 1 for GMAC1 0: Select GMAC1_PTPTRG1 trigger from ELC 1: Select external GMAC1_PTPTRG1 trigger input pin	R/W
4	G2TRGSEL0	Select PTP Timestamp Trigger 0 for GMAC2 0: Select GMAC2_PTPTRG0 trigger from ELC 1: Select external GMAC2_PTPTRG0 trigger input pin	R/W
5	G2TRGSEL1	Select PTP Timestamp Trigger 1 for GMAC2 0: Select GMAC2_PTPTRG1 trigger from ELC 1: Select external GMAC2_PTPTRG1 trigger input pin	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written when protection is unlocked by a specific sequence using the Ethernet Protect Register (PRCMD). For the protection unlock procedure, see section 27.3.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register.

The GMACTRGSEL register is used to select GMAC PTP Timestamp Trigger.

28.3.2 MAC_Configuration : MAC Configuration Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ARPEN	—	—	—	IPC	IPG[2:0]			GPSLCE	S2KP	CST	ACS	WD	BE	JD	JE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PS	FES	DM	LM	ECRSFD	DO	DCRS	DR	—	BL[1:0]		DC	PRELEN[1:0]		TE	RE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RE	Receiver Enable When this bit is set, the Rx state machine of the MAC is enabled for receiving packets from the GMII or MII interface. When this bit is reset, the MAC Rx state machine is disabled after it completes the reception of the current packet. The Rx state machine does not receive any more packets from the GMII or MII interface. 0: Receiver is disabled 1: Receiver is enabled	R/W
1	TE	Transmitter Enable When this bit is set, the Tx state machine of the MAC is enabled for transmission on the GMII or MII interface. When this bit is reset, the MAC Tx state machine is disabled after it completes the transmission of the current packet. The Tx state machine does not transmit any more packets. 0: Transmitter is disabled 1: Transmitter is enabled	R/W

Bit	Symbol	Function	R/W
3:2	PRELEN[1:0]	Preamble Length for Transmit packets These bits control the number of preamble bytes that are added to the beginning of every Tx packet. The preamble reduction occurs only when the MAC is operating in the full-duplex mode. 0 0: 7 bytes of preamble 0 1: 5 bytes of preamble 1 0: 3 bytes of preamble 1 1: Reserved	R/W
4	DC	Deferral Check When this bit is set, the deferral check function is enabled in the MAC. The MAC issues a Packet Abort status, along with the excessive deferral error bit set in the Tx packet status, when the Tx state machine is deferred for more than 24,288 bit times in 10 or 100 Mbps mode. If the MAC is configured for 1000 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but it is prevented because of an active carrier sense signal (CRS) on GMII or MII. The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and the CRS signal becomes inactive, the transmitter transmits and collision happens. Because of collision, the transmitter needs to back off and then defer again after back off completion. In such a scenario, the deferral timer is reset to 0, and it is restarted. When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive. This bit is applicable only in the half-duplex mode. 0: Deferral check function is disabled 1: Deferral check function is enabled	R/W
6:5	BL[1:0]	Back-Off Limit The back-off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps; 512 bit times for 10/100 Mbps) for which the MAC waits before rescheduling a transmission attempt during retries after a collision. n = retransmission attempt. The random integer r takes the value in the range $0 \leq r < 2^k$. This bit is applicable only in the half-duplex mode. 0 0: k = min(n, 10) 0 1: k = min(n, 8) 1 0: k = min(n, 4) 1 1: k = min(n, 1)	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	DR	Disable Retry When this bit is set, the MAC attempts only one transmission. When a collision occurs on the GMII or MII interface, the MAC ignores the current packet transmission and reports a Packet Abort with excessive collision error in the Tx packet status. When this bit is reset, the MAC retries based on the settings of the BL field. This bit is applicable only in the half-duplex mode. 0: Enable Retry 1: Disable Retry	R/W
9	DCRS	Disable Carrier Sense During Transmission When this bit is set, the MAC transmitter ignores the (G)MII CRS signal during packet transmission in the half-duplex mode. As a result, no errors are generated because of Loss of Carrier or No Carrier during transmission. When this bit is reset, the MAC transmitter generates errors because of Carrier Sense. The MAC can even abort the transmission. 0: Enable Carrier Sense During Transmission 1: Disable Carrier Sense During Transmission	R/W
10	DO	Disable Receive Own When this bit is set, the MAC disables the reception of packets when the TXEN is asserted in the half-duplex mode. When this bit is reset, the MAC receives all packets given by the PHY. This bit is not applicable in the full-duplex mode. 0: Enable Receive Own 1: Disable Receive Own	R/W

Bit	Symbol	Function	R/W
11	ECRSFD	Enable Carrier Sense Before Transmission in Full-Duplex Mode When this bit is set, the MAC transmitter checks the CRS signal before packet transmission in the full-duplex mode. The MAC starts the transmission only when the CRS signal is low. When this bit is reset, the MAC transmitter ignores the status of the CRS signal. 0: ECRSFD is disabled 1: ECRSFD is enabled	R/W
12	LM	Loopback Mode When this bit is set, the MAC operates in the loopback mode at GMII or MII. The (G)MII Rx clock input RX_CLK is required for the loopback to work properly. This is because the Tx clock is not internally looped back. 0: Loopback is disabled 1: Loopback is enabled	R/W
13	DM	Duplex Mode When this bit is set, the MAC operates in the full-duplex mode in which it can transmit and receive simultaneously. 0: Half-duplex mode 1: Full-duplex mode	R/W
14	FES	Speed This bit selects the speed mode. 0: 10 Mbps when PS bit is 1 and 1 Gbps when PS bit is 0 1: 100 Mbps when PS bit is 1	R/W
15	PS	Port Select This bit selects the Ethernet line speed. This bit, along with Bit 14, selects the exact line speed. 0: For 1000 Mbps operations 1: For 10 or 100 Mbps operations	R/W
16	JE	Jumbo Packet Enable When this bit is set, the MAC allows jumbo packets of 9,018 bytes (9,022 bytes for VLAN tagged packets) without reporting a giant packet error in the Rx packet status. 0: Jumbo packet is disabled 1: Jumbo packet is enabled	R/W
17	JD	Jabber Disable When this bit is set, the MAC disables the jabber timer on the transmitter. The MAC can transfer packets of up to 16,383 bytes. When this bit is reset, if the application sends more than 2,048 bytes of data (10,240 if JE is set high) during transmission, the MAC does not send rest of the bytes in that packet. 0: Jabber is enabled 1: Jabber is disabled	R/W
18	BE	Packet Burst Enable When this bit is set, the MAC allows packet bursting during transmission in the GMII half-duplex mode. 0: Packet Burst is disabled 1: Packet Burst is enabled	R/W
19	WD	Watchdog Disable When this bit is set, the MAC disables the watchdog timer on the receiver. The MAC can receive packets of up to 16,383 bytes. When this bit is reset, the MAC does not allow more than 2,048 bytes (10,240 if JE is set high) of the packet being received. The MAC cuts off any bytes received after 2,048 bytes. 0: Watchdog is enabled 1: Watchdog is disabled	R/W
20	ACS	Automatic Pad or CRC Stripping[*1] 0: Automatic Pad or CRC Stripping is disabled. MAC passes all incoming packets to the application, without any modification. 1: Automatic Pad or CRC Stripping is enabled. MAC strips the pad or the FCS field of the incoming Ethernet Length packets. For the Ethernet Type packets, MAC transfers the packets to the application without stripping the Pad or FCS field.	R/W

Bit	Symbol	Function	R/W
21	CST	CRC stripping for Type packets[*1] 0: CRC stripping for Type packets is disabled 1: CRC stripping for Type packets is enabled. MAC strips and drops the last four bytes (FCS) of all Ether type packets, as indicated by the Length/Type field of the packet, before forwarding the packet to the application.	R/W
22	S2KP	IEEE 802.3as Support for 2K Packets When this bit is set, the MAC considers all packets with up to 2,000 bytes length as normal packets. When the JE bit is not set, the MAC considers all received packets of size more than 2K bytes as Giant packets. When this bit is reset and the JE bit is not set, the MAC considers all received packets of size more than 1,518 bytes (1,522 bytes for tagged) as giant packets. For more information about how the setting of this bit and the JE bit impact the Giant packet status, see the Table, Giant Packet Status based on S2KP and JE Bits.[*2] 0: Support up to 2K packet is disabled 1: Support up to 2K packet is Enabled	R/W
23	GPSLCE	Giant Packet Size Limit Control Enable When this bit is set, the MAC considers the value in GPSL field in MAC_Ext_Configuration register to declare a received packet as Giant packet. This field must be programmed to more than 1,518 bytes. Otherwise, the MAC considers 1,518 bytes as giant packet limit. When this bit is reset, the MAC considers a received packet as Giant packet when its size is greater than 1,518 bytes (1522 bytes for tagged packet). The watchdog timeout limit, Jumbo Packet Enable and 2K Packet Enable have higher precedence over this bit, that is the MAC considers a received packet as Giant packet when its size is greater than 9,018 bytes (9,022 bytes for tagged packet) with Jumbo Packet Enabled and greater than 2,000 bytes with 2K Packet Enabled. The watchdog timeout, if enabled, terminates the received packet when watchdog limit is reached. Therefore, the programmed giant packet limit should be less than the watchdog limit to get the giant packet status. 0: Giant Packet Size Limit Control is disabled 1: Giant Packet Size Limit Control is enabled	R/W
26:24	IPG[2:0]	Inter-Packet Gap These bits control the minimum IPG between packets during transmission. This range of minimum IPG is valid in full-duplex mode. In the half-duplex mode, the minimum IPG can be configured only for 64-bit times (IPG = 100). Lower values are not considered. When a JAM pattern is being transmitted because of back pressure activation, the MAC does not consider the minimum IPG. This function (IPG less than 96 bit times) is valid only when EIPGEN bit in MAC_Ext_Configuration register is reset. When EIPGEN is set, then the minimum IPG (greater than 96 bit times) is controlled as per the description given in EIPG field in MAC_Ext_Configuration register. 0 0 0: 96 bit times IPG 0 0 1: 88 bit times IPG 0 1 0: 80 bit times IPG 0 1 1: 72 bit times IPG 1 0 0: 64 bit times IPG 1 0 1: 56 bit times IPG 1 1 0: 48 bit times IPG 1 1 1: 40 bit times IPG	R/W
27	IPC	Checksum Offload When set, this bit enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled. The Layer 3 and Layer 4 Packet Filter and Enable Split Header features automatically selects the IPC Full Checksum Offload Engine on the Receive side. When any of these features are enabled, you must set the IPC bit. 0: IP header/payload checksum checking is disabled 1: IP header/payload checksum checking is enabled	R/W
30:28	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31	ARPEN	ARP Offload Enable When this bit is set, the MAC can recognize an incoming ARP request packet and schedules the ARP packet for transmission. It forwards the ARP packet to the application and also indicate the events in the Rx Status. When this bit is reset, the MAC receiver does not recognize any ARP packet and indicates them as Type frame in the Rx Status. This bit is available only when the Enable IPv4 ARP Offload is selected. 0: ARP Offload is disabled 1: ARP Offload is enabled	R/W

Note 1. For more details about impact of CST and ACS bit settings on the packet length, see "Packet Length based on the CST and ACS Bits" table in the Architecture chapter.
Note 2. When the JE bit is set, setting this bit has no effect on the giant packet status.

The MAC_Configuration register establishes the operating mode of the MAC.

28.3.3 MAC_Ext_Configuration : MAC Extended Configuration Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	APDIM	EIPG[4:0]					EIPGEN	—	HDSMS[2:0]			PDC	USP	SPEN	DCRCC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	GPSL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	GPSL[13:0]	Giant Packet Size Limit If the received packet size is greater than the value programmed in this field in units of bytes, the MAC declares the received packet as Giant packet. The value programmed in this field must be greater than or equal to 1,518 bytes. Any other programmed value is considered as 1,518 bytes. For VLAN tagged packets, the MAC adds 4 bytes to the programmed value. When the Enable Double VLAN Processing option is selected, the MAC adds 8 bytes to the programmed value for double VLAN tagged packets. The value in this field is applicable when the GPSLCE bit is set in MAC_Configuration register.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	DCRCC	Disable CRC Checking for Received Packets When this bit is set, the MAC receiver does not check the CRC field in the received packets. When this bit is reset, the MAC receiver always checks the CRC field in the received packets. 0: CRC Checking is enabled 1: CRC Checking is disabled	R/W
17	SPEN	Slow Protocol Detection Enable 0: Slow Protocol Detection is disabled. MAC forwards all the error-free Slow Protocol packets to the application. The MAC considers such packets as normal Type packets. 1: Slow Protocol Detection is enabled. MAC processes the Slow Protocol packets (Ether Type = 0x8809) and provides the Slow Protocol Sub-Type and Code fields in the Rx status.	R/W

Bit	Symbol	Function	R/W
18	USP	Unicast Slow Protocol Packet Detect When this bit is set, the MAC detects the Slow Protocol packets with unicast address of the station specified in the MAC_ADDRESS0_High and MAC_ADDRESS0_Low registers. The MAC also detects the Slow Protocol packets with the Slow Protocols multicast address (01-80-C2-00-00-02). When this bit is reset, the MAC detects only Slow Protocol packets with the Slow Protocol multicast address specified in the IEEE 802.3-2015, Section 5. 0: Unicast Slow Protocol Packet Detection is disabled 1: Unicast Slow Protocol Packet Detection is enabled	R/W
19	PDC	Packet Duplication Control 0: Packet Duplication Control is disabled. MAC interprets the DSC field of the MAC_ADDRESSn_High_register as a binary value. 1: Packet Duplication Control is enabled. MAC interprets the DCS field of the MAC_ADDRESSn_High register as a one-hot value; each bit of the DCS field corresponds to a Rx DMA channel.	R/W
22:20	HDSMS[2:0]	Maximum Size for Splitting the Header Data These bits indicate the maximum header size allowed for splitting the header data in the received packet. 0 0 0: Maximum Size for Splitting the Header Data is 64 bytes 0 0 1: Maximum Size for Splitting the Header Data is 128 bytes 0 1 0: Maximum Size for Splitting the Header Data is 256 bytes 0 1 1: Maximum Size for Splitting the Header Data is 512 bytes 1 0 0: Maximum Size for Splitting the Header Data is 1024 bytes Others: Reserved	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
24	EIPGEN	Extended Inter-Packet Gap Enable*1 0: Extended Inter-Packet Gap is disabled. MAC ignores EIPG field. MAC uses the IPG field of the MAC_Configuration register as the minimum IPG less than or equal to 96-bit times, in steps of 8-bit times. 1: Extended Inter-Packet Gap is enabled. MAC uses the EIPG field for the extended inter-packet gap. For more details of how the MAC uses the EIPG field, see the description of the EIPG field.	R/W
29:25	EIPG[4:0]	Extended Inter-Packet Gap The value in this field is applicable when the EIPGEN bit of this register is 1. The 5 bits of this field are the most significant bits; IPG field of the MAC_Configuration register are the least significant 3 bits. Together, these 8 bits represent the value of the minimum IPG greater than 96-bit times, in steps of 8-bit times. {EIPG, IPG} 0x00: 104 bit times 0x01: 112 bit times 0x02: 120 bit times ⋮ 0xFF: 2144 bit times	R/W
30	APDIM	ARP Packet Drop if IP Address Mismatch When this bit is set, Packet for which Target Protocol Address does not match IPv4 address is dropped in the MTL layer. When this bit is reset, when target Protocol Address does not match, packet is forwarded to MTL. 0: mux select to drop the arp packet if target protocol address mismatches IPv4 address disabled 1: mux select to drop the arp packet if target protocol address mismatches IPv4 address enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. Use this feature only when operating in full-duplex mode. If you enable this feature in half-duplex mode, you might see undesirable effects on back-pressure function and frame transmission.

The MAC_Ext_Configuration Register establishes the operating mode of the MAC.

28.3.4 MAC_Packet_Filter : MAC Packet Filter register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RA	—	—	—	—	—	—	—	—	—	DNTU	IPFE	—	—	—	VTFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HPF	SAF	SAIF	PCF[1:0]		DBF	PM	DAIF	HMC	HUC	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PR	Promiscuous Mode 0: Promiscuous Mode is disabled. MAC transfers to the application only the packets that pass destination or source address filtering. 1: Promiscuous Mode is enabled. MAC transfers all incoming packets to the application regardless of the destination or source address filtering status. MAC clears the SA or DA Filter Fail status bits of the MAC Rx status word. For more details about MAC Rx status word, see "Receive Status Word at the MAC Interface" table in the Architecture chapter.	R/W
1	HUC	Hash Unicast 0: Hash Unicast is disabled. MAC performs a perfect filtering of destination address for unicast packets, that is, it compares the DA field with the values you programmed in the MAC_ADDRESSn_High and MAC_ADDRESSn_Low registers. 1: Hash Unicast is enabled. MAC performs the destination address filtering of unicast packets according to the hash table.	R/W
2	HMC	Hash Multicast 0: Hash Multicast is disabled. MAC performs perfect filtering of destination address for multicast packets, that is, it compares the DA field of the packet with the values programmed in the MAC_ADDRESSn_High and MAC_ADDRESSn_Low registers. 1: Hash Multicast is enabled. MAC performs the destination address filtering of the received multicast packets according to the hash table.	R/W
3	DAIF	DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast packets. When this bit is reset, normal filtering of packets is performed. 0: DA Inverse Filtering is disabled 1: DA Inverse Filtering is enabled	R/W
4	PM	Pass All Multicast When this bit is set, it indicates that all the received packets with a multicast destination address (first bit in the destination address field is '1') are passed. When this bit is reset, filtering of multicast packet depends on HMC bit. 0: Pass All Multicast is disabled 1: Pass All Multicast is enabled	R/W
5	DBF	Disable Broadcast Packets When this bit is set, the AFM module blocks all the incoming broadcast packets. In addition, it overrides all other filter settings. When this bit is reset, the AFM module passes all received broadcast packets. 0: Enable Broadcast Packets 1: Disable Broadcast Packets	R/W

Bit	Symbol	Function	R/W
7:6	PCF[1:0]	Pass Control Packets These bits control the forwarding of all control packets (including unicast and multicast Pause packets). 0 0: MAC filters all control packets from reaching the application 0 1: MAC forwards all control packets except Pause packets to the application even if they fail the Address filter 1 0: MAC forwards all control packets to the application even if they fail the Address filter 1 1: MAC forwards the control packets that pass the Address filter	R/W
8	SAIF	SA Inverse Filtering When this bit is set, the Address Check block operates in the inverse filtering mode for SA address comparison. If the SA of a packet matches the values programmed in the SA registers, it is marked as failing the SA Address filter. When this bit is reset, if the SA of a packet does not match the values programmed in the SA registers, it is marked as failing the SA Address filter. 0: SA Inverse Filtering is disabled 1: SA Inverse Filtering is enabled	R/W
9	SAF	Source Address Filter Enable*1 0: SA Filtering is disabled. MAC forwards the received packet to the application. Also, MAC updates the result of SA address comparison in the SA Filter Fail, bit [48] of the MAC Rx status word. For more details about MAC Rx status word, see "Receive Status Word at the MAC Interface" table in the Architecture chapter. 1: SA Filtering is enabled. MAC compares the SA field of the received packets with the values you programmed in the MAC_ADDRESSn_High and MAC_ADDRESSn_Low registers. If the comparison fails, the MAC drops the packet.	R/W
10	HPF	Hash or Perfect Filter When this bit is set, the address filter passes a packet if it matches either the perfect filtering or hash filtering as set by the HMC or HUC bit. When this bit is reset and the HUC or HMC bit is set, the packet is passed only if it matches the Hash filter. 0: Hash or Perfect Filter is disabled 1: Hash or Perfect Filter is enabled	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	VTFE	VLAN Tag Filter Enable When this bit is set, the MAC drops the VLAN tagged packets that do not match the VLAN Tag. When this bit is reset, the MAC forwards all packets irrespective of the match status of the VLAN Tag. 0: LAN Tag Filter is disabled 1: VLAN Tag Filter is enabled	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	IPFE	Layer 3 and Layer 4 Filter Enable When this bit is set, the MAC drops packets that do not match the enabled Layer 3 and Layer 4 filters. If Layer 3 or Layer 4 filters are not enabled for matching, this bit does not have any effect. When this bit is reset, the MAC forwards all packets irrespective of the match status of the Layer 3 and Layer 4 fields. 0: Layer 3 and Layer 4 Filters are disabled 1: Layer 3 and Layer 4 Filters are enabled	R/W
21	DNTU	Drop Non-TCP/UDP over IP Packets When this bit is set, the MAC drops the non-TCP or UDP over IP packets. The MAC forward only those packets that are processed by the Layer 4 filter. When this bit is reset, the MAC forwards all non-TCP or UDP over IP packets. 0: Forward Non-TCP/UDP over IP Packets 1: Drop Non-TCP/UDP over IP Packets	R/W
30:22	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31	RA	Receive All 0: Receive All is disabled. The MAC receiver transfers only the packets that pass the SA or DA address filter, to the application. 1: Receive All is enabled. The MAC receiver transfers all the received packets to the application regardless of the address filter status. MAC updates the SA or DA filtering status in the corresponding bit of the MAC filter status word. For more details, see the "MAC Filter Status Word Format" table in the Architecture chapter.	R/W

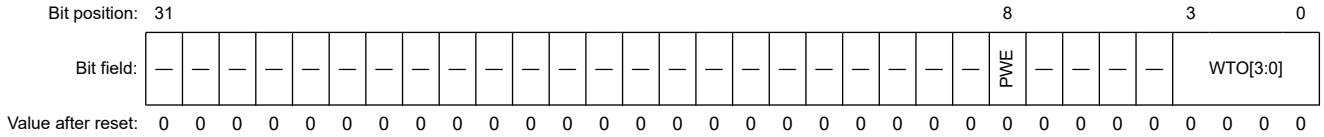
Note 1. According to the IEEE specification, Bit 47 of the SA is reserved. However, in GMAC, the MAC compares all 48 bits. The application must consider this while programming the MAC address registers for SA.

The MAC_Packet_Filter register contains the filter controls for receiving packets. Some of the controls from this register go to the address check block of the MAC which performs the first level of address filtering. The second level of filtering is performed on the incoming packet based on other controls such as Pass Bad Packets and Pass Control Packets.

28.3.5 MAC_Watchdog_Timeout : Watchdog Timeout register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x000C



Bit	Symbol	Function	R/W
3:0	WTO[3:0]	Watchdog Timeout When the PWE bit is set and the WD bit of the MAC_Configuration register is reset, this field is used as watchdog timeout for a received packet. If the length of a received packet exceeds the value of this field, such packet is terminated and declared as an error packet.*1 0x0: 2 KB 0x1: 3 KB 0x2: 4 KB 0x3: 5 KB 0x4: 6 KB 0x5: 7 KB 0x6: 8 KB 0x7: 9 KB 0x8: 10 KB 0x9: 11 KB 0xA: 12 KB 0xB: 13 KB 0xC: 14 KB 0xD: 15 KB 0xE: 16383 Bytes 0xF: Reserved	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PWE	Programmable Watchdog Enable When this bit is set and the WD bit of the MAC_Configuration register is reset, the WTO field is used as watchdog timeout for a received packet. When this bit is cleared, the watchdog timeout for a received packet is controlled by setting of WD and JE bits in MAC_Configuration register. 0: Programmable Watchdog is disabled 1: Programmable Watchdog is enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the PWE bit is set, the value in this field should be more than 1,522 (0x05F2). Otherwise, the IEEE 802.3-specified valid tagged packets are declared as error packets and then dropped.

The MAC_Watchdog_Timeout register controls the watchdog timeout for received packets.

28.3.6 MAC_HASH_TABLE_REGn : Hash Table Register n (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0010 + 0x04 × n

Bit position:	31																												0
Bit field:	HT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	HT[31:0]	MAC Hash Table n-th 32 Bits This field contains the n-the 32 Bits [(n+1)*32-1 : n*32] of the Hash table.	R/W

The Hash Table Register 0 contains the first 32 bits of the hash table, when the width of the hash table is 256 bits.

The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper eight bits of the CRC register are used to index the content of the Hash table. The most significant bits determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 10111111b (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5.

The hash value of the destination address is calculated in the following way:

- Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).
- Perform bitwise reversal for the value obtained in Step 1.
- Take the upper 8 bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC_Packet_Filter, all multicast packets are accepted regardless of the multicast hash values.

If the Hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits [31:24] of the Hash Table Register n registers are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

28.3.7 MAC_VLAN_Tag_Ctrl : VLAN Tag Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EIVLRXS	—	EIVLS[1:0]		ERIVLT	EDVLP	VTHM	EVLRXS	—	EVLS[1:0]		DOVLTC	ERSVLM	ESVL	VTIM	ETV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	OFS[3:0]				CT	OB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OB	Operation Busy ● Software writes 1 in this bit to initiate an indirect read or write access to the MAC_VLAN_Tag register. ● MAC writes 0 in this bit when it completes the indirect read or write access to the MAC_VLAN_Tag register. MAC can initiate the next indirect register access when this bit has a value 0. ● During a write operation, MAC writes 0 in this bit after it completes writing the data into the MAC_VLAN_Tag register. ● During a read operation, the application must read the data from the MAC_VLAN_Tag_Data register only after MAC writes 0 in this bit. 0: Operation Busy is disabled 1: Operation Busy is enabled	R/W
1	CT	Command Type This bit indicates if the current register access is a read or a write. 0: Write operation. Indicates a write operation. 1: Read operation. Indicates a read operation.	R/W
5:2	OFS[3:0]	Offset This field holds the address offset (i) of the MAC_VLAN_Tag_Filter i Register (i = 0 to 15) which the application is trying to access.	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
16	ETV	Enable 12-Bit VLAN Tag Comparison for VLAN Hash Filtering 0: 12-Bit VLAN Tag Comparison is disabled. All 16 bits of the 15th and 16th bytes of the received VLAN packet are used for VLAN hash filtering. 1: 12-Bit VLAN Tag Comparison is enabled. A 12-bit VLAN identifier is used for VLAN Hash filtering instead of the complete 16-bit VLAN tag. Bits [11:0] of VLAN tag in the received VLAN-tagged packet are used for hash-based VLAN filtering.	R/W
17	VTIM	VLAN Tag Inverse Match Enable 0: VLAN Tag Inverse Match is disabled. Enables the VLAN Tag perfect matching. The packets with matched VLAN Tag are marked as matched. 1: VLAN Tag Inverse Match is enabled. Enables the VLAN Tag inverse matching. The packets without matching VLAN Tag are marked as matched.	R/W
18	ESVL	Enable S-VLAN 0: S-VLAN is disabled. 1: S-VLAN is enabled. MAC transmitter and receiver consider the S-VLAN packets (Type = 0x88A8) as valid VLAN tagged packets.	R/W
19	ERSVLM	Enable Receive S-VLAN Match for VLAN Hash Filtering The ERIVLT bit determines the VLAN tag position considered for VLAN Hash filtering or matching. 0: Receive S-VLAN Match is disabled. MAC receiver enables VLAN Hash filtering or matching for C-VLAN (Type = 0x8100) packets. 1: Receive S-VLAN Match is enabled. MAC receiver enables VLAN Hash filtering or matching for S-VLAN (Type = 0x88A8) packets.	R/W
20	DOVLTC	Disable VLAN Type Check for VLAN Hash Filtering 0: VLAN Type Check is enabled. MAC VLAN Hash Filter filters or matches the VLAN Tag specified by the ERIVLT bit only when VLAN Tag type is similar to the one specified by the ERSVLM bit. 1: VLAN Type Check is disabled. MAC VLAN Hash Filter does not check whether the VLAN Tag specified by the ERIVLT bit is of type S-VLAN or C-VLAN.	R/W

Bit	Symbol	Function	R/W
22:21	EVLS[1:0]	Enable VLAN Tag Stripping on Receive This field indicates the stripping operation on the outer VLAN Tag in received packet. 0 0: Do not strip 0 1: Strip if VLAN filter passes 1 0: Strip if VLAN filter fails 1 0: Always strip	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
24	EVLRXS	Enable VLAN Tag in Rx status 0: VLAN Tag in Rx status is disabled. MAC does not provide the outer VLAN Tag in Rx status. 1: VLAN Tag in Rx status is enabled. MAC provides the outer VLAN Tag in the Rx status.	R/W
25	VTHM	VLAN Tag Hash Table Match Enable*1 0: VLAN Tag Hash Table Match is disabled. MAC does not perform the VLAN Hash Match operation. 1: VLAN Tag Hash Table Match is enabled. MAC uses the most significant four bits of CRC of VLAN Tag to index the content of the MAC_VLAN_Hash_Table register. A value of 1 in the MAC_VLAN_Hash_Table register, corresponding to the index, indicates that the packet matched the VLAN hash table. ● When ETV = 1, MAC uses the CRC of the 12-bit VLAN Identifier (VID) for comparison. ● When ETV = 0, MAC uses the CRC of the 16-bit VLAN Identifier (VID) for comparison.	R/W
26	EDVLP	Enable Double VLAN Processing 0: Double VLAN Processing is disabled. MAC enables processing of up to one VLAN Tag on Tx and Rx (if present). 1: Double VLAN Processing is enabled. MAC enables processing of up to two VLAN Tags on Tx and Rx (if present).	R/W
27	ERIVLT	Enable Inner VLAN Tag Comparison When this bit, VTHM bit and the EDVLP field are set, the MAC receiver enables VLAN Hash filtering operation on the inner VLAN Tag (if present). When this bit is reset and VTHM bit is set, the MAC receiver enables VLAN Hash filtering operation on the outer VLAN Tag (if present). The ERSVLM bit and DOVLTC bit determines which VLAN type is enabled for filtering. 0: VLAN tag is disabled 1: VLAN tag is enabled	R/W
29:28	EIVLS[1:0]	Enable Inner VLAN Tag Stripping on Receive This field indicates the stripping operation on inner VLAN Tag in received packet. 0 0: Do not strip 0 1: Strip if VLAN filter passes 1 0: Strip if VLAN filter fails 1 1: Always strip	R/W
30	—	This bit is read as 0. The write value should be 0.	R/W
31	EIVLRXS	Enable Inner VLAN Tag in Rx Status 0: Inner VLAN Tag in Rx status is disabled. MAC does not provide the inner VLAN Tag in Rx status. 1: Inner VLAN Tag in Rx status is enabled. MAC provides the inner VLAN Tag in the Rx status.	R/W

Note 1. For more details, see section 28.6.3.2. VLAN Tag Hash Filtering.

This register is the redefined format of the MAC VLAN Tag Register. It is used for indirect addressing. It contains the address offset, command type and Busy Bit for CSR access of the Per VLAN Tag registers.

28.3.8 MAC_VLAN_Tag_Data : VLAN Tag Data Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0054

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	DMACHN[2:0]			DMAC HEN	—	—	—	RIVLT	ERSV LM	DOVL TC	ETV	VEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	VID[15:0]	VLAN Tag ID This field holds the VLAN Tag value which is used by the MAC for perfect comparison. It is valid when VLAN Tag Enable is set.	R/W
16	VEN	VLAN Tag Enable This bit is used to enable or disable the VLAN Tag. 0: VLAN Tag is disabled. MAC does not compare the VLAN Tag of received packet with the VLAN Tag ID, irrespective of the programming of the other fields. 1: VLAN Tag is enabled. MAC compares the VLAN Tag of the received packet with the VLAN Tag ID.	R/W
17	ETV	12-Bit or 16-Bit VLAN Comparison This bit is valid only when you enable the VLAN Tag by programming the VEN field of the corresponding MAC_VLAN_Tag_Filter i register. 0: 16 bit VLAN comparison. 1: 12 bit VLAN comparison. MAC uses a 12-bit VLAN tag for the comparison and filtering instead of the 16-bit VLAN tag. MAC compares the bits [11:0] of the VLAN tag with the corresponding field in the received VLAN-tagged packet.	R/W
18	DOVLTC	Disable VLAN Type Comparison This bit is valid only when you enable the VLAN Tag by programming the VEN field of the MAC_VLAN_Tag_Filter i register. 0: VLAN type comparison is enabled. MAC considers the VLAN Tag type during the comparison. The VLAN Tag type, S-VLAN or C-VLAN, used by the MAC for comparison is determined by the value you program in the ERSVLM field of the MAC_VLAN_Tag_Ctrl register. 1: VLAN type comparison is disabled. MAC does not check the type of the VLAN Tag during the VLAN tag comparison. MAC performs either inner or outer VLAN Tag comparison based on the value you program in the ERIVLT field of the MAC_VLAN_Tag_Filter i register.	R/W
19	ERSVLM	Enable S-VLAN Match for received Frames This bit is valid only when you enable the VLAN Tag by programming the VEN field of the MAC_VLAN_Tag_Filter i register. When this bit is set, the MAC receiver enables filtering or matching for S-VLAN (Type = 0x88A8) packets. When this bit is reset, the MAC receiver enables filtering or matching for C-VLAN (Type = 0x8100) packets. 0: Receive S-VLAN Match is disabled 1: Receive S-VLAN Match is enabled	R/W

Bit	Symbol	Function	R/W
20	RIVLT	Enable Inner VLAN Tag Comparison This bit is valid only when Double VLAN Tag Enable of the Filter is set. When this bit and the EDVLP field are set, the MAC receiver enables operation on the inner VLAN Tag (if present). When this bit is reset, the MAC receiver enables operation on the outer VLAN Tag (if present). 0: Inner VLAN tag comparison is disabled 1: Inner VLAN tag comparison is enabled	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	DMACHEN	DMA Channel Number Enable 0: DMA Channel Number is disabled. 1: DMA Channel Number is enabled. Enables the DMA channel number that you program in the DMACHN field.	R/W
27:25	DMACHN[2:0]	DMA Channel Number The DMA Channel number to which the VLAN Tagged Frame is to be routed if it passes this VLAN Tag Filter is programmed in this field. If the Routing based on VLAN Tag Filter is not necessary, this field need not be programmed.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

This register holds the read or write data for the indirect access of the MAC_VLAN_Tag_Filter i (i = 0 to 15) registers.

- For the read operation, this register contains valid read data only after the MAC writes 0 to the OB bit of the MAC_VLAN_Tag_Ctrl register.
- For the write operation, Software must ensure valid data in this register before writing 1 to the OB bit of the MAC_VLAN_Tag_Ctrl register.

28.3.9 MAC_VLAN_Hash_Table : VLAN Hash Table Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0058

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLHT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	VLHT[15:0]	VLAN Hash Table This field contains the 16-bit VLAN Hash Table.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

When VTHM bit of the MAC_VLAN_Tag register is set, the 16-bit VLAN Hash Table register is used for group address filtering based on the VLAN tag. For hash filtering, the content of the 16-bit VLAN tag or 12-bit VLAN ID (based on the ETV bit of MAC_VLAN_Tag Register) in the incoming packet is passed through the CRC logic. The upper four bits of the calculated hash value are used to index the contents of the VLAN Hash table. For example, hash value of 1000b selects Bit 8 of the VLAN Hash table.

The hash value of the destination address is calculated in the following way:

- Calculate the 32-bit CRC for the VLAN tag or ID (For steps to calculate CRC32, see Section 3.2.8 of IEEE 802.3).
- Perform bitwise reversal for the value obtained in step 1.
- Take the upper four bits from the value obtained in step 2.

If the VLAN hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits [15:8] of this register are written.

- If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

28.3.10 MAC_Q0_Tx_Flow_Ctrl : Flow Control register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DZPQ	PLT[2:0]			—	—	TFE	FCB_ BPA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FCB_BPA	Flow Control Busy or Back pressure Activate*1 This bit initiates a Pause packet in the full-duplex mode and activates the back pressure function in the half-duplex mode if the TFE bit is set. **Full-Duplex Mode:** In the full-duplex mode, this bit should be read as 0 before writing to this register. To initiate a Pause packet, the application must set this bit to 1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When Pause packet transmission is complete, the MAC resets this bit to 0. You should not write to this register until this bit is cleared. **Half-Duplex Mode:** When this bit is set (and TFE bit is set) in the half-duplex mode, the MAC asserts the back pressure. During back pressure, when the MAC receives a new packet, the transmitter starts sending a JAM pattern resulting in a collision. When the MAC is configured for the full-duplex mode, the BPA is automatically disabled. 0: Flow Control Busy or Back pressure Activate is disabled 1: Flow Control Busy or Back pressure Activate is enabled	R/W
1	TFE	Transmit Flow Control Enable **Full-Duplex Mode:** In the full-duplex mode, when this bit is set, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. **Half-Duplex Mode:** In the half-duplex mode, when this bit is set, the MAC enables the back pressure operation. When this bit is reset, the back pressure feature is disabled. 0: Transmit Flow Control is disabled 1: Transmit Flow Control is enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6:4	PLT[2:0]	Pause Low Threshold This field configures the threshold of the Pause timer for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits [31:16]. For example, if PT = 0x100 (256 slot times), and PLT = 001b, a second Pause packet is automatically transmitted if the trigger is asserted at 228 (256 - 28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 0 0 0: Pause Time minus 4 Slot Times (PT - 4 slot times) 0 0 1: Pause Time minus 28 Slot Times (PT - 28 slot times) 0 1 0: Pause Time minus 36 Slot Times (PT - 36 slot times) 0 1 1: Pause Time minus 144 Slot Times (PT - 144 slot times) 1 0 0: Pause Time minus 256 Slot Times (PT - 256 slot times) 1 0 1: Pause Time minus 512 Slot Times (PT - 512 slot times) Others: Reserved	R/W
7	DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion from the FIFO layer. When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0: Zero-Quanta Pause packet generation is enabled 1: Zero-Quanta Pause packet generation is disabled	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
31:16	PT[15:0]	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.	R/W

Note 1. Access restrictions apply.

- MAC initiates the pause packet or activates the back pressure, when application writes 1.
- MAC writes 0 after the operation is complete.
- MAC ignores when application writes 0.

The MAC_Q0_Tx_Flow_Ctrl register controls the generation and reception of the Control (Pause Command) packets by the Flow control module of the MAC. A Write to a register with the Busy bit set to 1 triggers the Flow Control block to generate a Pause packet. The fields of the control packet are selected as specified in the 802.3x specification, and the Pause Time value from this register is used in the Pause Time field of the control packet. The Busy bit remains set until the control packet is transferred onto the cable. The application must make sure that the Busy bit is cleared before writing to the register.

When the PFCE bit in the MAC_Rx_Flow_Ctrl register is enabled, this register controls the generation of Priority Flow Control (PFC) frames with priorities mapped according to PSRQ0 in the MAC_RxQ_Ctrl2 register.

28.3.11 MAC_Rx_Flow_Ctrl : Receive Flow Control register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UP	RFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFE	Receive Flow Control Enable When this bit is set and the MAC is operating in full-duplex mode, the MAC decodes the received Pause packet and disables its transmitter for a specified (Pause) time. When this bit is reset or the MAC is operating in half-duplex mode, the decode function of the Pause packet is disabled. When PFC is enabled, flow control is enabled for PFC packets. The MAC decodes the received PFC packet and disables the Transmit queue, with matching priorities, for a duration of received Pause time. 0: Receive Flow Control is disabled 1: Receive Flow Control is enabled	R/W
1	UP	Unicast Pause Packet Detect A pause packet is processed when it has the unique multicast address specified in the IEEE 802.3. When this bit is set, the MAC can also detect Pause packets with unicast address of the station. This unicast address should be as specified in MAC_ADDRESS0_High and MAC_ADDRESS0_Low. When this bit is reset, the MAC only detects Pause packets with unique multicast address.*1 0: Unicast Pause Packet Detect disabled 1: Unicast Pause Packet Detect enabled	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The MAC does not process a Pause packet if the multicast address is different from the unique multicast address. This is also applicable to the received PFC packet when the Priority Flow Control (PFC) is enabled. The unique multicast address (0x01_80_C2_00_00_01) is as specified in IEEE 802.1 Qbb-2011.

The MAC_Rx_Flow_Ctrl register controls the pausing of MAC Transmit based on the received Pause packet.

28.3.12 MAC_RxQ_Ctrl4 : Receive Queue Control 4 register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0094

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	VFFQ[2:0]			VFFQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MFFQ[2:0]			MFFQE	—	—	—	—	UFFQ[2:0]			UFFQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UFFQE	Unicast Address Filter Fail Packets Queuing Enable When this bit is set, the Unicast packets which fail the Destination or Source address filter is routed to the Rx Queue Number programmed in the UFFQ. When this bit is reset, the Unicast packets which fail the Destination or Source address filter is routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set. 0: Unicast Address Filter Fail Packets Queuing is disabled 1: Unicast Address Filter Fail Packets Queuing is enabled	R/W
3:1	UFFQ[2:0]	Unicast Address Filter Fail Packets Queue This field holds the Rx queue number to which the Unicast packets failing the Destination or Source Address filter are routed to. This field is valid only when the UFFQE bit is set.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	MFFQE	Multicast Address Filter Fail Packets Queuing Enable When this bit is set, the Multicast packets which fail the Destination or Source address filter is routed to the Rx Queue Number programmed in the MFFQ. When this bit is reset, the Multicast packets which fail the Destination or Source address filter is routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set. 0: Multicast Address Filter Fail Packets Queuing is disabled 1: Multicast Address Filter Fail Packets Queuing is enabled	R/W
11:9	MFFQ[2:0]	Multicast Address Filter Fail Packets Queue This field holds the Rx queue number to which the Multicast packets failing the Destination or Source Address filter are routed to. This field is valid only when the MFFQE bit is set.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	VFFQE	VLAN Tag Filter Fail Packets Queuing Enable When this bit is set, the tagged packets which fail the Destination or Source address filter or fail the VLAN tag filter, are routed to the Rx Queue Number programmed in the VFFQ. When this bit is reset, the tagged packets which fail the Destination or Source address filter or fail the VLAN tag filter are routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set. 0: VLAN tag Filter Fail Packets Queuing is disabled 1: VLAN tag Filter Fail Packets Queuing is enabled	R/W
19:17	VFFQ[2:0]	VLAN Tag Filter Fail Packets Queue This field holds the Rx queue number to which the tagged packets failing the Destination or Source Address filter (and UFFQE/MFFQE not enabled) or failing the VLAN tag filter must be routed to. This field is valid only when the VFFQE bit is set.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The MAC_RxQ_Ctrl4 register controls the routing of unicast and multicast packets that fail the Destination or Source address filter to the Rx queues.

28.3.13 MAC_RxQ_Ctrl0 : Receive Queue Control 0 register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXQ7EN[1:0]		RXQ6EN[1:0]		RXQ5EN[1:0]		RXQ4EN[1:0]		RXQ3EN[1:0]		RXQ2EN[1:0]		RXQ1EN[1:0]		RXQ0EN[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RXQ0EN[1:0]	Receive Queue 0 Enable This field indicates whether Rx Queue 0 is enabled for AV or DCB. 0 0: Queue not enabled 0 1: Queue enabled for AV 1 0: Queue enabled for DCB/Generic 1 1: Reserved	R/W
3:2	RXQ1EN[1:0]	Receive Queue 1 Enable This field is similar to the RXQ0EN[1:0] field. 0 0: Queue not enabled 0 1: Queue enabled for AV 1 0: Queue enabled for DCB/Generic 1 1: Reserved	R/W

Bit	Symbol	Function	R/W
5:4	RXQ2EN[1:0]	Receive Queue 2 Enable This field is similar to the RXQ0EN[1:0] field. 0 0: Queue not enabled 0 1: Queue enabled for AV 1 0: Queue enabled for DCB/Generic 1 1: Reserved	R/W
7:6	RXQ3EN[1:0]	Receive Queue 3 Enable This field is similar to the RXQ0EN[1:0] field. 0 0: Queue not enabled 0 1: Queue enabled for AV 1 0: Queue enabled for DCB/Generic 1 1: Reserved	R/W
9:8	RXQ4EN[1:0]	Receive Queue 4 Enable This field is similar to the RXQ0EN[1:0] field. 0 0: Queue not enabled 0 1: Queue enabled for AV 1 0: Queue enabled for DCB/Generic 1 1: Reserved	R/W
11:10	RXQ5EN[1:0]	Receive Queue 5 Enable This field is similar to the RXQ0EN[1:0] field. 0 0: Queue not enabled 0 1: Queue enabled for AV 1 0: Queue enabled for DCB/Generic 1 1: Reserved	R/W
13:12	RXQ6EN[1:0]	Receive Queue 6 Enable This field is similar to the RXQ0EN[1:0] field. 0 0: Queue not enabled 0 1: Queue enabled for AV 1 0: Queue enabled for DCB/Generic 1 1: Reserved	R/W
15:14	RXQ7EN[1:0]	Receive Queue 7 Enable This field is similar to the RXQ0EN[1:0] field. 0 0: Queue not enabled 0 1: Queue enabled for AV 1 0: Queue enabled for DCB/Generic 1 1: Reserved	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The MAC_RxQ_Ctrl0 register controls the queue management in the MAC Receiver.

Note: All the queues are disabled by default. Enable the Rx queue by programming the corresponding field in this register.

28.3.14 MAC_RxQ_Ctrl1 : Receive Queue Control 1 register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	TBRQE	OMCBCQ	—	FPRQ[2:0]			TPQC[1:0]		TACPQE	MCBCQEN	—	MCBCQ[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	UPQ[2:0]			—	—	—	—	—	PTPQ[2:0]			—	AVCPQ[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	AVCPQ[2:0]	AV Untagged Control Packets Queue This field specifies the Receive queue on which the received AV tagged and untagged control packets are routed. The AV tagged (when TACPQE bit is set) and untagged control packets are routed to Receive queue specified by this field. 0 0 0: Receive Queue 0 0 0 1: Receive Queue 1 0 1 0: Receive Queue 2 0 1 1: Receive Queue 3 1 0 0: Receive Queue 4 1 0 1: Receive Queue 5 1 1 0: Receive Queue 6 1 1 1: Receive Queue 7	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	PTPQ[2:0]	PTP Packets Queue This field specifies the Rx queue on which the PTP packets sent over the Ethernet payload (not over IPv4 or IPv6) are routed. When the AV8021ASMEN bit of MAC_Timestamp_Control register is set, only untagged PTP over Ethernet packets are routed on an Rx Queue. If the bit is not set, then based on programming of TPQC field, both tagged and untagged PTPoE packets can be routed to this Rx Queue. 0 0 0: Receive Queue 0 0 0 1: Receive Queue 1 0 1 0: Receive Queue 2 0 1 1: Receive Queue 3 1 0 0: Receive Queue 4 1 0 1: Receive Queue 5 1 1 0: Receive Queue 6 1 1 1: Receive Queue 7	R/W
11:7	—	These bits are read as 0. The write value should be 0.	R/W
14:12	UPQ[2:0]	Untagged Packet Queue This field indicates the Rx Queue to which Untagged Packets are to be routed. Any Rx Queue enabled for Generic/DCB/AV traffic can be used to route the Untagged Packets. 0 0 0: Receive Queue 0 0 0 1: Receive Queue 1 0 1 0: Receive Queue 2 0 1 1: Receive Queue 3 1 0 0: Receive Queue 4 1 0 1: Receive Queue 5 1 1 0: Receive Queue 6 1 1 1: Receive Queue 7	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	MCBCQ[2:0]	Multicast and Broadcast Queue This field specifies the Rx Queue onto which Multicast or Broadcast Packets are routed. Any Rx Queue enabled for Generic/DCB/AV traffic can be used to route the Multicast or Broadcast Packets. 0 0 0: Receive Queue 0 0 0 1: Receive Queue 1 0 1 0: Receive Queue 2 0 1 1: Receive Queue 3 1 0 0: Receive Queue 4 1 0 1: Receive Queue 5 1 1 0: Receive Queue 6 1 1 1: Receive Queue 7	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	MCBCQEN	Multicast and Broadcast Queue Enable This bit specifies that Multicast or Broadcast packets routing to the Rx Queue is enabled and the Multicast or Broadcast packets must be routed to Rx Queue specified in MCBCQ field. 0: Multicast and Broadcast Queue is disabled 1: Multicast and Broadcast Queue is enabled	R/W

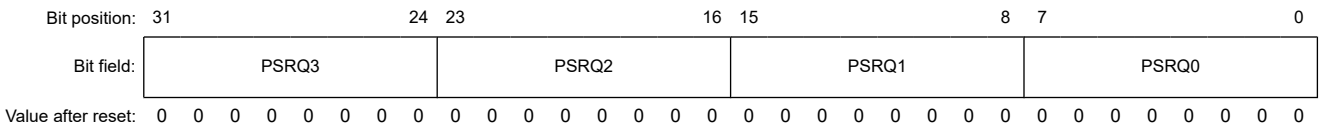
Bit	Symbol	Function	R/W
21	TACPQE	Tagged AV Control Packets Queuing Enable When set, the MAC routes the received Tagged AV Control packets to the Rx queue specified by AVCPQ field. When reset, the MAC routes the received Tagged AV Control packets based on the tag priority matching the PSRQ fields in MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers. 0: Tagged AV Control Packets Queuing is disabled 1: Tagged AV Control Packets Queuing is enabled	R/W
23:22	TPQC[1:0]	Tagged PTP over Ethernet Packets Queuing Control This field controls the routing of the VLAN Tagged PTPoE packets. The following programmable options are allowed. 0 0: VLAN Tagged PTPoE packets are routed as generic VLAN Tagged packet (based on PSRQ for only non-AV enabled Rx Queues) 0 1: VLAN Tagged PTPoE packets are routed to Rx Queue specified by PTPQ field (That Rx Queue can be enabled for AV or non-AV traffic) 1 0: VLAN Tagged PTPoE packets are routed to only AV enabled Rx Queues based on PSRQ 1 1: Reserved	R/W
26:24	FPRQ[2:0]	Frame Preemption Residue Queue This field holds the Rx queue number to which the residual preemption frames must be forwarded. Preemption frames that are tagged and pass the SA/DA/VLAN filtering are routed based on PSRQ and all other frames are treated as residual frames and is routed to the queue number mentioned in this field. The Queue-0 is used as a default queue for express frames, so this field cannot be programmed to a value 0.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	OMCBCQ	Over-riding MC-BC queue priority select 0: Overriding MCBCQ priority disabled. Received Multicast/Broadcast packet is routed to MCBCQ. 1: Overriding MCBCQ priority enabled. Priority of MCBCQ is reduced and the received packet is first routed to PTPQ, AVCPQ, DCBCPQ depending on packet type.	R/W
29	TBRQE	Type Field Based Rx Queuing Enable When this bit is set, it enables Type Field based Rx Queuing where the Type field of received packet is compared with programmed TYP field in MAC_TMRQ_REGSn and if a match occurs the packet is routed to the corresponding TMRQ field.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

The MAC_RxQ_Ctrl1 register controls the routing of multicast, broadcast, AV, DCB, and untagged packets to the Rx queues.

28.3.15 MAC_RxQ_Ctrl2 : Receive Queue Control 2 register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00A8



Bit	Symbol	Function	R/W
7:0	PSRQ0	Priorities Selected in the Receive Queue 0 This field decides the priorities assigned to Rx Queue 0. All packets with priorities that match the values set in this field are routed to Rx Queue 0. For example, if PSRQ0[5] is set, packets with USP field equal to 5 are routed to Rx Queue 0. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.	R/W

Bit	Symbol	Function	R/W
15:8	PSRQ1	Priorities Selected in the Receive Queue 1 This field decides the priorities assigned to Rx Queue 1. All packets with priorities that match the values set in this field are routed to Rx Queue 1. For example, if PSRQ1[4] is set, packets with USP field equal to 4 are routed to Rx Queue 1. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.	R/W
23:16	PSRQ2	Priorities Selected in the Receive Queue 2 This field decides the priorities assigned to Rx Queue 2. All packets with priorities that match the values set in this field are routed to Rx Queue 2. For example, if PSRQ2[1, 0] are set, packets with USP field equal to 1 or 0 are routed to Rx Queue 2. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.	R/W
31:24	PSRQ3	Priorities Selected in the Receive Queue 3 This field decides the priorities assigned to Rx Queue 3. Packets whose priorities match the values set in this field are routed to Rx Queue 3. For example, if PSRQ3[6, 3] have value 1, the packets whose USP field of the VLAN Tag has value 3 or 6 are routed to Rx Queue 3. Application must ensure that values in the PSRQ fields of all the Rx queues within a set that you enable generically or for AV or DCB features, are mutually exclusive. This is because multiple Rx queues cannot have the same priority within a set.	R/W

This register controls the routing of tagged packets based on the USP (user Priority) field of the received packets to the Rx Queues 0 to 3.

28.3.16 MAC_RxQ_Ctrl3 : Receive Queue Control 3 register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00AC

Bit position:	31							24	23							16	15							8	7							0
Bit field:	PSRQ7								PSRQ6								PSRQ5								PSRQ4							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PSRQ4	Priorities Selected in the Receive Queue 4 This field decides the priorities assigned to Rx Queue 4. All packets with priorities that match the values set in this field are routed to Rx Queue 4. For example, if PSRQ4[7:4] is set, packets with USP field equal to 7, 6, 5, or 4 are routed to Rx Queue 4. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.	R/W
15:8	PSRQ5	Priorities Selected in the Receive Queue 5 This field decides the priorities assigned to Rx Queue 5. All packets with priorities that match the values set in this field are routed to Rx Queue 5. For example, if PSRQ5[6] is set, packets with USP field equal to 6 are routed to Rx Queue 5. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.	R/W
23:16	PSRQ6	Priorities Selected in the Receive Queue 6 This field decides the priorities assigned to Rx Queue 6. All packets with priorities that match the values set in this field are routed to Rx Queue 6. For example, if PSRQ6[5] are set, packets with USP field equal to 5 are routed to Rx Queue 6. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.	R/W
31:24	PSRQ7	Priorities Selected in the Receive Queue 7 This field decides the priorities assigned to Rx Queue 7. All packets with priorities that match the values set in this field are routed to Rx Queue 7. For example, if PSRQ7[7, 4] are set, packets with USP field equal to 7 or 4 are routed to Rx Queue 7. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.	R/W

This register controls the routing of tagged packets based on the USP (user Priority) field of the received packets to the Rx Queues 4 to 7.

28.3.17 MAC_Interrupt_Status : MAC Interrupt Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MFRIS	MFTIS	MDIOIS	FPEIS	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	RXSTSIS	TXSTSIS	TSIS	MMCRXIPIS	MMCTXIS	MMCRXIS	MMCIS	—	—	LPIIS	PMTIS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
4	PMTIS	PMT Interrupt Status This bit is set when a Magic packet or Wake-on-LAN packet is received in the power-down mode (RWKPRCVD and MGKPRCVD bits in MAC_PMT_Control_Status register). This bit is cleared when corresponding interrupt source bit are cleared because of a Read operation to the MAC_PMT_Control_Status register (or corresponding interrupt source bit of MAC_PMT_Control_Status register is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set). 0: PMT Interrupt status not active 1: PMT Interrupt status active	R
5	LPIIS	LPI Interrupt Status When the Energy Efficient Ethernet feature is enabled, this bit is set for any LPI state entry or exit in the MAC Transmitter or Receiver. This bit is cleared when the corresponding interrupt source bit of MAC_LPI_Control_Status register is read (or corresponding interrupt source bit of MAC_LPI_Control_Status register is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set). 0: LPI Interrupt status not active 1: LPI Interrupt status active	R
7:6	—	These bits are read as 0.	R
8	MMCIS	MMC Interrupt Status This bit is set high when Bit 11, Bit 10, or Bit 9 is set high. This bit is cleared only when all these bits are low. 0: MMC Interrupt status not active 1: MMC Interrupt status active	R
9	MMCRXIS	MMC Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. 0: MMC Receive Interrupt status not active 1: MMC Receive Interrupt status active	R
10	MMCTXIS	MMC Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. 0: MMC Transmit Interrupt status not active 1: MMC Transmit Interrupt status active	R
11	MMCRXIPIS	MMC Receive Checksum Offload Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. 0: MMC Receive Checksum Offload Interrupt status not active 1: MMC Receive Checksum Offload Interrupt status active	R

Bit	Symbol	Function	R/W
12	TSIS	Timestamp Interrupt Status If you select Enable Timestamp feature, this bit is 1, when any of the following conditions is true: An overflow in the MAC_System_Time_Seconds register. If you select Auxiliary Snapshot feature, this bit is 1. ● When the auxiliary snapshot trigger is asserted. ● When you enable drop transmit status in MTL, this bit is 1 when the MAC updates the captured transmit timestamp in the MAC_Tx_Timestamp_Status_Nanoseconds and MAC_Tx_Timestamp_Status_Seconds registers. ● When you select Enable PTP Offload feature, this bit is 1, when the MAC updates the captured transmit timestamp in the MAC_Tx_Timestamp_Status_Nanoseconds and MAC_Tx_Timestamp_Status_Seconds registers, for the PTO generated Delay Request and Pdelay request packets. MAC writes 0 to this bit when ● Application reads the corresponding interrupt source bit in the MAC_Timestamp_Status register. ● Application writes 1 to the corresponding interrupt source bit in the MAC_Timestamp_Status register, when RCWE = 1 in the MAC_CSR_SW_Ctrl register. 0: Timestamp Interrupt status not active 1: Timestamp Interrupt status active	R
13	TXSTSIS	Transmit Status Interrupt This bit indicates the status of transmitted packets. This bit is set when any of the following bits is set in the MAC_Rx_Tx_Status register: ● Excessive Collision (EXCOL) ● Late Collision (LCOL) ● Excessive Deferral (EXDEF) ● Loss of Carrier (LCARR) ● No Carrier (NCARR) ● Jabber Timeout (TJT) This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set) in the MAC_Rx_Tx_Status register. 0: Transmit Interrupt status not active 1: Transmit Interrupt status active	R
14	RXSTSIS	Receive Status Interrupt This bit indicates the status of received packets. This bit is set when the RWT bit is set in the MAC_Rx_Tx_Status register. This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set) in the MAC_Rx_Tx_Status register. 0: Receive Interrupt status not active 1: Receive Interrupt status active	R
16:15	—	These bits are read as 0.	R
17	FPEIS	Frame Preemption Interrupt Status This bit indicates an interrupt event during the operation of Frame Preemption (Bits [19:16] of MAC_FPE_CTRL_STS register is set). To reset this bit, the application must clear the event in MAC_FPE_CTRL_STS that has caused the Interrupt. 0: Frame Preemption Interrupt status not active 1: Frame Preemption Interrupt status active	R
18	MDIOIS	MDIO Interrupt Status This bit indicates an interrupt event after the completion of MDIO operation. Following access restrictions apply. ● Clears on read ● Clears on write 1, when RCWE = 1 in the MAC_CSR_SW_Ctrl register ● Self-set to 1 on internal event 0: MDIO Interrupt status not active. 1: MDIO Interrupt status active.	R
19	MFTIS	MMC FPE Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC FPE Transmit Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. 0: MMC FPE Transmit Interrupt status not active 1: MMC FPE Transmit Interrupt status active	R

Bit	Symbol	Function	R/W
20	MFRIS	MMC FPE Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC FPE Receive Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. 0: MMC FPE Receive Interrupt status not active. 1: MMC FPE Receive Interrupt status active	R
31:21	—	These bits are read as 0.	R

The MAC_Interrupt_Status register contains the status of interrupts.

28.3.18 MAC_Interrupt_Enable : MAC Interrupt Enable Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00B4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	MDIOIE	FPEIE	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	RXSTSIE	TXSTSIE	TSIE	—	—	—	—	—	—	LPIIE	PMTIE	PHYIE	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	PHYIE	PHY Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of PHYIS bit in MAC_Interrupt_Status register. 0: PHY Interrupt is disabled 1: PHY Interrupt is enabled	R/W
4	PMTIE	PMT Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of PMTIS bit in MAC_Interrupt_Status register. 0: PMT Interrupt is disabled 1: PMT Interrupt is enabled	R/W
5	LPIIE	LPI Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of LPIIS bit in MAC_Interrupt_Status register. 0: LPI Interrupt is disabled 1: LPI Interrupt is enabled	R/W
11:6	—	These bits are read as 0. The write value should be 0.	R/W
12	TSIE	Timestamp Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of TSIS bit in MAC_Interrupt_Status register. 0: Timestamp Interrupt is disabled 1: Timestamp Interrupt is enabled	R/W
13	TXSTSIE	Transmit Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of TXSTSIS bit in the MAC_Interrupt_Status register. 0: Timestamp Status Interrupt is disabled 1: Timestamp Status Interrupt is enabled	R/W
14	RXSTSIE	Receive Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of RXSTSIS bit in the MAC_Interrupt_Status register. 0: Receive Status Interrupt is disabled 1: Receive Status Interrupt is enabled	R/W

Bit	Symbol	Function	R/W
16:15	—	These bits are read as 0. The write value should be 0.	R/W
17	FPEIE	Frame Preemption Interrupt Enable When this bit is set, it enables the assertion of the interrupt when FPEIS field is set in the MAC_Interrupt_Status register. 0: Frame Preemption Interrupt is disabled 1: Frame Preemption Interrupt is enabled	R/W
18	MDIOIE	MDIO Interrupt Enable When this bit is set, it enables the assertion of the interrupt when MDIOIS field is set in the MAC_Interrupt_Status register. 0: MDIO Interrupt is disabled 1: MDIO Interrupt is enabled	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The MAC_Interrupt_Enable register contains the masks for generating the interrupts.

28.3.19 MAC_Rx_Tx_Status : Receive and Transmit Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00B8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RWT	—	—	EXCOL	LCOL	EXDEF	LCARR	NCARR	TJT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TJT	Transmit Jabber Timeout*1 This bit indicates that the Transmit Jabber Timer expired which happens when the packet size exceeds 2,048 bytes (10,240 bytes when the Jumbo packet is enabled) and JD bit is reset in the MAC_Configuration register. This bit is set when the packet size exceeds 16,383 bytes and the JD bit is set in the MAC_Configuration register. 0: No Transmit Jabber Timeout 1: Transmit Jabber Timeout occurred	R
1	NCARR	No Carrier*1 When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the carrier signal from the PHY is not present at the end of preamble transmission. 0: Carrier is present 1: No carrier	R
2	LCARR	Loss of Carrier*1 When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the loss of carrier occurred during packet transmission, that is, the CRS signal was inactive for one or more transmission clock periods during packet transmission. This bit is valid only for packets transmitted without collision. 0: Carrier is present 1: Loss of carrier	R
3	EXDEF	Excessive Deferral*1 When the DTXSTS bit is set in the MTL_Operation_Mode register and the DC bit is set in the MAC_Configuration register, this bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 in 1000 Mbps mode or when Jumbo packet is enabled). 0: No Excessive deferral 1: Excessive deferral	R

Bit	Symbol	Function	R/W
4	LCOL	Late Collision*1 When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the packet transmission aborted because a collision occurred after the collision window (64 bytes including Preamble in MII mode; 512 bytes including Preamble and Carrier Extension in GMII mode). This bit is not valid if the Underflow error occurs. 0: No collision 1: Late collision is sensed	R
5	EXCOL	Excessive Collisions*1 When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the transmission aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the MAC_Configuration register, this bit is set after the first collision and the packet transmission is aborted. 0: No collision 1: Excessive collision is sensed	R
7:6	—	These bits are read as 0.	R
8	RWT	Receive Watchdog Timeout*1 This bit is set when a packet with length greater than 2,048 bytes is received (10,240 bytes when Jumbo Packet mode is enabled) and the WD bit is reset in the MAC_Configuration register. This bit is set when a packet with length greater than 16,383 bytes is received and the WD bit is set in the MAC_Configuration register. 0: No receive watchdog timeout 1: Receive watchdog timed out	R
31:9	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.

The MAC_Rx_Tx_Status register contains the Receive and Transmit Error status.

28.3.20 MAC_PMT_Control_Status : PMT Control and Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RWKFILTRST	—	—	RWKPTR[4:0]					—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	RWKPFE	GLBLUCAST	—	—	RWKPRCVD	MGKPRCVD	—	—	RWKPKTEN	MGKPKTEN	PWRDWN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PWRDWN	Power Down When this bit is set, the MAC receiver drops all received packets until it receives the expected magic packet or remote wake-up packet. This bit is then self-cleared and the power-down mode is disabled. The software can clear this bit before the expected magic packet or remote wake-up packet is received. The packets received by the MAC after this bit is cleared are forwarded to the application. This bit must only be set when the Magic Packet Enable, Global Unicast, or Remote Wake-Up Packet Enable bit is set high.*1 *3 0: Power down is disabled 1: Power down is enabled	R/W

Bit	Symbol	Function	R/W
1	MGKPKTEN	Magic Packet Enable When this bit is set, a power management event is generated when the MAC receives a magic packet. 0: Magic Packet is disabled 1: Magic Packet is enabled	R/W
2	RWKPKTEN	Remote Wake-Up Packet Enable When this bit is set, a power management event is generated when the MAC receives a remote wake-up packet. 0: Remote wake-up packet is disabled 1: Remote wake-up packet is enabled	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	MGKPRCVD	Magic Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a magic packet. This bit is cleared when this register is read.*4 0: No Magic packet is received 1: Magic packet is received	R
6	RWKPRCVD	Remote Wake-Up Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a remote wake-up packet. This bit is cleared when this register is read.*4 0: No Remote wake-up packet is received 1: Remote wake-up packet is received	R
8:7	—	These bits are read as 0. The write value should be 0.	R/W
9	GLBLUCAST	Global Unicast When this bit set, any unicast packet filtered by the MAC (DAF) address recognition is detected as a remote wake-up packet. 0: Global unicast is disabled 1: Global unicast is enabled	R/W
10	RWKPFE	Remote Wake-up Packet Forwarding Enable When this bit is set along with RWKPKTEN, the MAC receiver drops all received frames until it receives the expected Wake-up frame. All frames after that event including the received wake-up frame are forwarded to application. This bit is then self-cleared on receiving the wake-up packet. The application can also clear this bit before the expected wake-up frame is received. In such cases, the MAC reverts to the default behavior where packets received are forwarded to the application. This bit must only be set when RWKPKTEN is set high and PWRDWN is set low. The setting of this bit has no effect when PWRDWN is set high.*2 *3 0: Remote Wake-up Packet Forwarding is disabled 1: Remote Wake-up Packet Forwarding is enabled	R/W
23:11	—	These bits are read as 0. The write value should be 0.	R/W
28:24	RWKPTR[4:0]	Remote Wake-up FIFO Pointer This field gives the current value (0 to 7, 15, or 31 when 4, 8, or 16 Remote Wake-up Packet Filters are selected) of the Remote Wake-up Packet Filter register pointer. When the value of this pointer is equal to maximum for the selected number of Remote Wake-up Packet Filters, the contents of the Remote Wake-up Packet Filter Register are transferred to the RX_CLK domain when a Write occurs to that register.	R
30:29	—	These bits are read as 0. The write value should be 0.	R/W
31	RWKFILTRST	Remote Wake-Up Packet Filter Register Pointer Reset When this bit is set, the remote wake-up packet filter register pointer is reset to 000b. It is automatically cleared after 1 clock cycle.*3 0: Remote Wake-Up Packet Filter Register Pointer is not Reset 1: Remote Wake-Up Packet Filter Register Pointer is Reset	R/W

Note 1. You can gate-off the CSR clock during the power-down mode. However, when the CSR clock is gated-off, you cannot perform any read or write operations on this register. Therefore, the software cannot clear this bit.
Note 2. If Magic Packet Enable and Wake-Up Frame Enable are both set along with setting of this bit and Magic Packet is received prior to wake-up frame, this bit is self-cleared on receiving Magic Packet, the received Magic packet is dropped, and all frames after received Magic Packet are forwarded to application.
Note 3. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
Note 4. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.

The MAC_PMT_Control_Status Register

28.3.21 MAC_RWK_Packet_Filter : Remote Wakeup Filter Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00C4

Bit position:	31																												0
Bit field:	WKUPFRMFTR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	WKUPFRMFTR[31:0]	RWK Packet Filter This field contains the various controls of RWK Packet filter.	R/W

MAC_RWK_Packet_Filter register is used to program CRC, offset, and byte mask of the pattern embedded in remote wakeup packet and the filter operation commands in a filter lookup table. See
section 28.13.1.2. Description of Remote Wakeup Packet Mode
in detail.

28.3.22 MAC_LPI_Control_Status : LPI Control and Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	LPIATE	LPITXA	—	PLS	LPIEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RLPIST	TLPIST	—	—	—	—	RLPIEX	RLPIEN	TLPIEX	TLPIEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TLPIEN	Transmit LPI Entry When this bit is set, it indicates that the MAC Transmitter has entered the LPI state because of the setting of the LPIEN bit. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). 0: Transmit LPI entry not detected 1: Transmit LPI entry detected	R
1	TLPIEX	Transmit LPI Exit When this bit is set, it indicates that the MAC transmitter exited the LPI state after the application cleared the LPIEN bit and the LPI TW Timer has expired. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). 0: Transmit LPI exit not detected 1: Transmit LPI exit detected	R
2	RLPIEN	Receive LPI Entry When this bit is set, it indicates that the MAC Receiver has received an LPI pattern and entered the LPI state. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set).*1 0: Receive LPI entry not detected 1: Receive LPI entry detected	R

Bit	Symbol	Function	R/W
3	RLPIEX	Receive LPI Exit When this bit is set, it indicates that the MAC Receiver has stopped receiving the LPI pattern on the GMII or MII interface, exited the LPI state, and resumed the normal reception. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set).*1 0: Receive LPI exit not detected 1: Receive LPI exit detected	R
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	TLPIST	Transmit LPI State When this bit is set, it indicates that the MAC is transmitting the LPI pattern on the GMII or MII interface. 0: Transmit LPI state not detected 1: Transmit LPI state detected	R
9	RLPIST	Receive LPI State When this bit is set, it indicates that the MAC is receiving the LPI pattern on the GMII or MII interface. 0: Receive LPI state not detected 1: Receive LPI state detected	R
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	LPIEN	LPI Enable When this bit is set, it instructs the MAC Transmitter to enter the LPI state. When this bit is reset, it instructs the MAC to exit the LPI state and resume normal transmission. This bit is cleared when the LPITXA bit is set and the MAC exits the LPI state because of the arrival of a new packet for transmission. 0: LPI state is disabled 1: LPI state is enabled	R/W
17	PLS	PHY Link Status This bit indicates the link status of the PHY. The MAC Transmitter asserts the LPI pattern only when the link status is up (OKAY) at least for the time indicated by the LPI LS TIMER. When this bit is set, the link is considered to be okay (UP) and when this bit is reset, the link is considered to be down. 0: Link is down 1: Link is okay (UP)	R/W
18	—	This bit is read as 0. The write value should be 0.	R/W
19	LPITXA	LPI Tx Automate This bit controls the behavior of the MAC when it is entering or coming out of the LPI mode on the Transmit side. If the LPITXA and LPIEN bits are set to 1, the MAC enters the LPI mode only after all outstanding packets (in the IP) and pending packets (in the application interface) have been transmitted. The MAC comes out of the LPI mode when the application sends any packet for transmission or the application issues a Tx FIFO Flush command. In addition, the MAC automatically clears the LPIEN bit when it exits the LPI state. If Tx FIFO Flush is set in the FTQ bit of MTL_TxQ0_Operation_Mode register, when the MAC is in the LPI mode, it exits the LPI mode. When this bit is 0, the LPIEN bit directly controls behavior of the MAC when it is entering or coming out of the LPI mode. 0: LPI Tx Automate is disabled 1: LPI Tx Automate is enabled	R/W
20	LPIATE	LPI Timer Enable This bit controls the automatic entry of the MAC Transmitter into and exit out of the LPI state. When LPIATE, LPITXA and LPIEN bits are set, the MAC Transmitter enters LPI state only when the complete MAC TX data path is IDLE for a period indicated by the MAC_LPI_Entry_Timer register. After entering LPI state, if the data path becomes non-IDLE (due to a new packet being accepted for transmission), the Transmitter exits LPI state but does not clear LPIEN bit. This enables the re-entry into LPI state when it is IDLE again. When LPIATE is 0, the LPI Auto timer is disabled and MAC Transmitter enters LPI state based on the settings of LPITXA and LPIEN bit descriptions. 0: LPI Timer is disabled 1: LPI Timer is enabled	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit might not be set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than three clock cycles of CSR clock.

The MAC_LPI_Control_Status register controls the LPI functions and provides the LPI interrupt status. The status bits are cleared when this register is read.

28.3.23 MAC_LPI_Timers_Control : LPI Timers Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00D4

Bit position:	31						25									16	15														0
Bit field:	—	—	—	—	—	—	LST[9:0]										TWT[15:0]															
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TWT[15:0]	LPI TW Timer This field specifies the minimum time (in microseconds) for which the MAC waits after it stops transmitting the LPI pattern to the PHY and before it resumes the normal transmission. The TLPIEX status bit is set after the expiry of this timer.	R/W
25:16	LST[9:0]	LPI LS Timer This field specifies the minimum time (in milliseconds) for which the link status from the PHY should be up (OKAY) before the LPI pattern can be transmitted to the PHY. The MAC does not transmit the LPI pattern even when the LPIEN bit is set unless the LPI LS Timer reaches the programmed terminal count. The default value of the LPI LS Timer is 1000 (1 sec) as defined in the IEEE standard.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The MAC_LPI_Timers_Control register controls the timeout values in the LPI states. It specifies the time for which the MAC transmits the LPI pattern and also the time for which the MAC waits before resuming the normal transmission.

28.3.24 MAC_LPI_Entry_Timer : LPI entry timer Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00D8

Bit position:	31											19																3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	LPIET[16:0]																—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

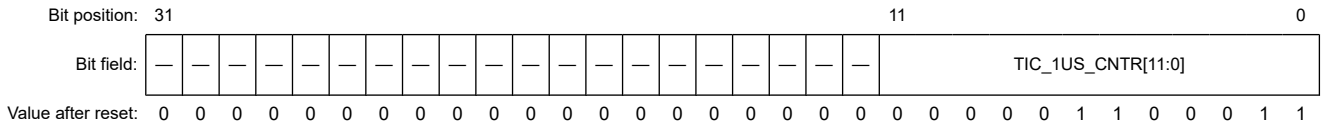
Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
19:3	LPIET[16:0]	LPI Entry Timer This field specifies the time in microseconds the MAC waits to enter LPI mode, after it has transmitted all the frames. This field is valid and used only when LPITE and LPITXA are set to 1. Bits [2:0] are read-only so that the granularity of this timer is in steps of 8 micro-seconds.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

This register controls the Tx LPI entry timer. This counter is enabled only when bit [20] (LPITE) bit of MAC_LPI_Control_Status is set to 1.

28.3.25 MAC_1US_Tic_Counter : LP1 1us Tick Counter Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x00DC



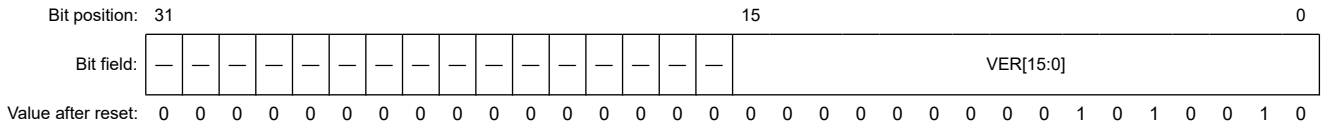
Bit	Symbol	Function	R/W
11:0	TIC_1US_CNTR[11:0]	1US TIC Counter The application must program this counter so that the number of clock cycles of CSR clock is 1 µs. (Subtract 1 from the value before programming). For example if the CSR clock is 100 MHz then this field needs to be programmed to value 100 - 1 = 99 (which is 0x63). This is required to generate the 1 US events that are used to update some of the EEE related counters.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

This register controls the generation of the Reference time (1 microsecond tic) for all the LPI timers. This timer has to be programmed by the software initially.

28.3.26 MAC_Version : Version Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0110



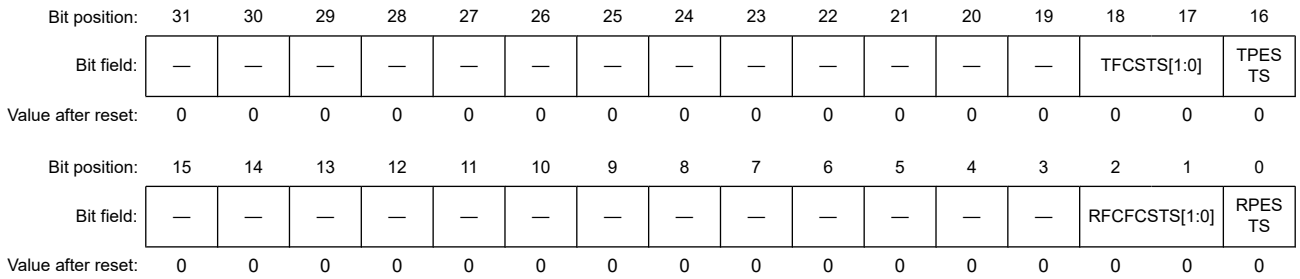
Bit	Symbol	Function	R/W
15:0	VER[15:0]	Version	R
31:16	—	These bits are read as 0.	R

The MAC_Version register identifies the version of the GMAC.

28.3.27 MAC_Debug : MAC Debug Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0114



Bit	Symbol	Function	R/W
0	RPESTS	MAC GMII or MII Receive Protocol Engine Status When this bit is set, it indicates that the MAC GMII or MII receive protocol engine is actively receiving data, and it is not in the Idle state. 0: MAC GMII or MII Receive Protocol Engine Status not detected 1: MAC GMII or MII Receive Protocol Engine Status detected	R
2:1	RFCFCSTS[1:0]	MAC Receive Packet Controller FIFO Status When this bit is set, this field indicates the active state of the small FIFO Read and Write controllers of the MAC Receive Packet Controller module.	R
15:3	—	These bits are read as 0.	R
16	TPESTS	MAC GMII or MII Transmit Protocol Engine Status When this bit is set, it indicates that the MAC GMII or MII transmit protocol engine is actively transmitting data, and it is not in the Idle state. 0: MAC GMII or MII Transmit Protocol Engine Status not detected 1: MAC GMII or MII Transmit Protocol Engine Status detected	R
18:17	TFCSTS[1:0]	MAC Transmit Packet Controller Status This field indicates the state of the MAC Transmit Packet Controller module. 0 0: Idle state 0 1: Waiting for one of the following: Status of the previous packet OR IPG or back off period to be over 1 0: Generating and transmitting a Pause control packet (in full-duplex mode) 1 1: Transferring input packet for transmission	R
31:19	—	These bits are read as 0.	R

The MAC_Debug register provides the debug status of various MAC blocks.

28.3.28 MAC_HW_Feature0 : MAC Hardware Feature 0 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x011C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	TSTSSEL[1:0]		—	—	ADDMACADRSEL[4:0]					—	RXCOESEL
Value after reset:	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	TXCOESEL	EEESEL	TSSEL	—	—	ARPOFFSEL	MMCSEL	MGKSEL	RWKSEL	SMASEL	VLHASH	—	HDSEL	GMIISEL	MIISEL
Value after reset:	0	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1

Bit	Symbol	Function	R/W
0	MIISEL	10 or 100 Mbps Support This bit is set to 1 when 10/100 Mbps is selected as the Mode of Operation. 0: No 10 or 100 Mbps support 1: 10 or 100 Mbps support	R
1	GMIISEL	1000 Mbps Support This bit is set to 1 when 1000 Mbps is selected as the Mode of Operation. 0: No 1000 Mbps support 1: 1000 Mbps support	R
2	HDSEL	Half-duplex Support This bit is set to 1 when the half-duplex mode is selected. 0: No Half-duplex support 1: Half-duplex support	R
3	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
4	VLHASH	VLAN Hash Filter Selected This bit is set to 1 when the Enable VLAN Hash Table Based Filtering option is selected. 0: VLAN Hash Filter not selected 1: VLAN Hash Filter selected	R
5	SMASEL	SMA (MDIO) Interface This bit is set to 1 when the Enable Station Management (MDIO Interface) option is selected. 0: SMA (MDIO) Interface not selected 1: SMA (MDIO) Interface selected	R
6	RWKSEL	PMT Remote Wake-up Packet Enable This bit is set to 1 when the Enable Remote Wake-Up Packet Detection option is selected. 0: PMT Remote Wake-up Packet Enable option is not selected 1: PMT Remote Wake-up Packet Enable option is selected	R
7	MGKSEL	PMT Magic Packet Enable This bit is set to 1 when the Enable Magic Packet Detection option is selected. 0: PMT Magic Packet Enable option is not selected 1: PMT Magic Packet Enable option is selected	R
8	MMCSEL	RMON Module Enable This bit is set to 1 when the Enable MAC Management Counters (MMC) option is selected. 0: RMON Module Enable option is not selected 1: RMON Module Enable option is selected	R
9	ARPOFFSEL	ARP Offload Enabled This bit is set to 1 when the Enable IPv4 ARP Offload option is selected. 0: ARP Offload Enable option is not selected 1: ARP Offload Enable option is selected	R
11:10	—	These bits are read as 0.	R
12	TSSEL	IEEE 1588-2008 Timestamp Enabled This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected. 0: IEEE 1588-2008 Timestamp Enable option is not selected 1: IEEE 1588-2008 Timestamp Enable option is selected	R
13	EEESEL	Energy Efficient Ethernet Enabled This bit is set to 1 when the Enable Energy Efficient Ethernet (EEE) option is selected. 0: Energy Efficient Ethernet Enable option is not selected 1: Energy Efficient Ethernet Enable option is selected	R
14	TXCOESEL	Transmit Checksum Offload Enabled This bit is set to 1 when the Enable Transmit TCP/IP Checksum Insertion option is selected. 0: Transmit Checksum Offload Enable option is not selected 1: Transmit Checksum Offload Enable option is selected	R
15	—	This bit is read as 0.	R
16	RXCOESEL	Receive Checksum Offload Enabled This bit is set to 1 when the Enable Receive TCP/IP Checksum Check option is selected. 0: Receive Checksum Offload Enable option is not selected 1: Receive Checksum Offload Enable option is selected	R
17	—	This bit is read as 0.	R
22:18	ADDMACADRSEL[4:0]	MAC Addresses 1-31 Selected This bit is set to 1 when the non-zero value is selected for Enable Additional 1-31 MAC Address Registers option.	R
24:23	—	These bits are read as 0.	R
26:25	TSSTSSEL[1:0]	Timestamp System Time Source This bit indicates the source of the Timestamp system time: This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected. 0 0: Internal 0 1: External 1 0: Both 1 1: Reserved	R
31:27	—	These bits are read as 0.	R

This register indicates the presence of first set of the optional features or functions of the GMAC. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

28.3.29 MAC_HW_Feature1 : MAC Hardware Feature 1 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0120

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	L3L4FNUM[3:0]				—	HASHTBLSZ[1:0]		POUOST	—	—	AVSEL	—	—	SPHEN	—
Value after reset:	0	1	0	0	0	0	1	1	1	0	0	1	0	0	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADDR64[1:0]		—	PTOEN	OSTEN	TXFIFOSIZE[4:0]					—	RXFIFOSIZE[4:0]				
Value after reset:	0	0	0	1	1	0	0	1	1	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
4:0	RXFIFOSIZE[4:0]	MTL Receive FIFO Size This field contains the configured value of MTL Rx FIFO in bytes. 0x6: 8192 bytes	R
5	—	This bit is read as 0.	R
10:6	TXFIFOSIZE[4:0]	MTL Transmit FIFO Size This field contains the configured value of MTL Tx FIFO in bytes. 0x6: 8192 bytes	R
11	OSTEN	One-Step Timestamping Enable This bit is set to 1 when the Enable One-Step Timestamp Feature is selected. 0: One-Step Timestamping feature is not selected 1: One-Step Timestamping feature is selected	R
12	PTOEN	PTP Offload Enable This bit is set to 1 when the Enable PTP Timestamp Offload Feature is selected. 0: PTP Offload feature is not selected 1: PTP Offload feature is selected	R
13	—	This bit is read as 0.	R
15:14	ADDR64[1:0]	Address Width This field indicates the configured address width: 0 0: 32 0 1: 40 1 0: 48 1 1: Reserved	R
16	—	This bit is read as 0.	R
17	SPHEN	Split Header Feature Enable This bit is set to 1 when the Enable Split Header Structure option is selected. 0: Split Header Feature is not selected 1: Split Header Feature is selected	R
19:18	—	These bits are read as 0.	R
20	AVSEL	AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option is selected. 0: AV Feature is not selected 1: AV Feature is selected	R
22:21	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
23	POUOST	One Step for PTP over UDP/IP Feature Enable This bit is set to 1 when the Enable One step timestamp for PTP over UDP/IP feature is selected. 0: One Step for PTP over UDP/IP Feature is not selected 1: One Step for PTP over UDP/IP Feature is selected	R
25:24	HASHTBLSZ[1:0]	Hash Table Size This field indicates the size of the hash table: 0 0: No hash table 0 1: 64 1 0: 128 1 1: 256	R
26	—	This bit is read as 0.	R
30:27	L3L4FNUM[3:0]	Total number of L3 or L4 Filters This field indicates the total number of L3 or L4 filters: 0x0: No L3 or L4 Filters 0x1: 1 L3 or L4 Filter 0x2: 2 L3 or L4 Filters 0x3: 3 L3 or L4 Filters 0x4: 4 L3 or L4 Filters 0x5: 5 L3 or L4 Filters 0x6: 6 L3 or L4 Filters 0x7: 7 L3 or L4 Filters 0x8: 8 L3 or L4 Filters Others: Reserved	R
31	—	This bit is read as 0.	R

This register indicates the presence of second set of the optional features or functions of the GMAC. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

28.3.30 MAC_HW_Feature2 : MAC Hardware Feature 2 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0124

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	AUXSNAPNUM[2:0]			—	—	—	—	—	—	TXCHCNT[3:0]				—	—
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXCHCNT[3:0]				—	—	TXQCNT[3:0]				—	—	RXQCNT[3:0]			
Value after reset:	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1

Bit	Symbol	Function	R/W
3:0	RXQCNT[3:0]	Number of MTL Receive Queues This field indicates the number of MTL Receive queues: 0x0: 1 MTL Rx Queue 0x1: 2 MTL Rx Queues 0x2: 3 MTL Rx Queues 0x3: 4 MTL Rx Queues 0x4: 5 MTL Rx Queues 0x5: 6 MTL Rx Queues 0x6: 7 MTL Rx Queues 0x7: 8 MTL Rx Queues Others: Reserved	R
5:4	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
9:6	TXQCNT[3:0]	Number of MTL Transmit Queues This field indicates the number of MTL Transmit queues: 0x0: 1 MTL Tx Queue 0x1: 2 MTL Tx Queues 0x2: 3 MTL Tx Queues 0x3: 4 MTL Tx Queues 0x4: 5 MTL Tx Queues 0x5: 6 MTL Tx Queues 0x6: 7 MTL Tx Queues 0x7: 8 MTL Tx Queues Others: Reserved	R
11:10	—	These bits are read as 0.	R
15:12	RXCHCNT[3:0]	Number of DMA Receive Channels This field indicates the number of DMA Receive channels: 0x0: 1 MTL Rx Channel 0x1: 2 MTL Rx Channels 0x2: 3 MTL Rx Channels 0x3: 4 MTL Rx Channels 0x4: 5 MTL Rx Channels 0x5: 6 MTL Rx Channels 0x6: 7 MTL Rx Channels 0x7: 8 MTL Rx Channels Others: Reserved	R
17:16	—	These bits are read as 0.	R
21:18	TXCHCNT[3:0]	Number of DMA Transmit Channels This field indicates the number of DMA Transmit channels: 0x0: 1 MTL Tx Channel 0x1: 2 MTL Tx Channels 0x2: 3 MTL Tx Channels 0x3: 4 MTL Tx Channels 0x4: 5 MTL Tx Channels 0x5: 6 MTL Tx Channels 0x6: 7 MTL Tx Channels 0x7: 8 MTL Tx Channels Others: Reserved	R
27:22	—	These bits are read as 0.	R
30:28	AUXSNAPNUM[2:0]	Number of Auxiliary Snapshot Inputs This field indicates the number of auxiliary snapshot inputs: 0 0 0: No auxiliary input 0 0 1: 1 auxiliary input 0 1 0: 2 auxiliary input 0 1 1: 3 auxiliary input 1 0 0: 4 auxiliary input Others: Reserved	R
31	—	This bit is read as 0.	R

This register indicates the presence of third set of the optional features or functions of the GMAC. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

28.3.31 MAC_HW_Feature3 : MAC Hardware Feature 3 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0128

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	TBSSEL	FPESEL	—	—	—	—	ESTWID[1:0]		ESTDEP[2:0]			ESTSEL
Value after reset:	0	0	0	0	1	1	0	0	0	0	1	1	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PDUPSEL	—	—	—	DVLAN	—	—	NRVF[2:0]		
Value after reset:	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1

Bit	Symbol	Function	R/W
2:0	NRVF[2:0]	Number of Extended VLAN Tag Filters Enabled This field indicates the Number of Extended VLAN Tag Filters selected: 0 0 0: No Extended Rx VLAN Filters 0 0 1: 4 Extended Rx VLAN Filters 0 1 0: 8 Extended Rx VLAN Filters 0 1 1: 16 Extended Rx VLAN Filters 1 0 0: 24 Extended Rx VLAN Filters 1 0 1: 32 Extended Rx VLAN Filters Others: Reserved	R
4:3	—	These bits are read as 0.	R
5	DVLAN	Double VLAN Tag Processing Selected This bit is set to 1 when the Enable Double VLAN Processing Feature is selected. 0: Double VLAN option is not selected 1: Double VLAN option is selected	R
8:6	—	These bits are read as 0.	R
9	PDUPSEL	Broadcast/Multicast Packet Duplication This bit is set to 1 when the Broadcast/Multicast Packet Duplication feature is selected. 0: Broadcast/Multicast Packet Duplication feature is not selected 1: Broadcast/Multicast Packet Duplication feature is selected	R
15:10	—	These bits are read as 0.	R
16	ESTSEL	Enhancements to Scheduled Traffic Enable This bit is set to 1 when the Enable Enhancements to Scheduling Traffic feature is selected. 0: Enable Enhancements to Scheduling Traffic feature is not selected 1: Enable Enhancements to Scheduling Traffic feature is selected	R
19:17	ESTDEP[2:0]	Depth of the Gate Control List Indicates the depth of Gate Control list in bytes. 0 0 0: No Depth configured 0 0 1: 64 0 1 0: 128 0 1 1: 256 1 0 0: 512 1 0 1: 1024 Others: Reserved	R
21:20	ESTWID[1:0]	Width of the Time Interval field in the Gate Control List This field indicates the width of the Configured Time Interval Field. 0 0: Width not configured 0 1: 16 1 0: 20 1 1: 24	R
25:22	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
26	FPESEL	Frame Preemption Enable This bit is set to 1 when the Enable Frame preemption feature is selected. 0: Frame Preemption Enable feature is not selected 1: Frame Preemption Enable feature is selected	R
27	TBSSEL	Time Based Scheduling Enable This bit is set to 1 when the Time Based Scheduling feature is selected. 0: Time Based Scheduling Enable feature is not selected 1: Time Based Scheduling Enable feature is selected	R
31:28	—	These bits are read as 0.	R

This register indicates the presence of fourth set the optional features or functions of the GMAC. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

28.3.32 MAC_MDIO_Address : MDIO Address Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PSE	BTB	PA[4:0]					RDA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	NTC[2:0]			CR[3:0]				—	—	—	SKAP	GOC[1:0]		C45E	GB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GB	GMII Busy*1 The application writes 1 to this bit to instruct the SMA to initiate a read or write access to the MDIO slave. The MAC writes 0 to this bit after the MDIO frame transfer is complete. Therefore, the application must modify the fields of the MAC_MDIO_Address and MAC_MDIO_Data registers only when the value in this bit is 0. For write operation, application must follow these steps: ● Write the 16-bit data in the GDI field of the MAC_MDIO_Data register. ● Update the RA field of the MAC_MDIO_Data register with the register address of the PHY. **Note:** RA field is valid only when C45E field is 1. ● Write 1 to this bit. For read operation, application must follow these steps: ● Update the RA field of the MAC_MDIO_Data register with the register address of the PHY. **Note:** RA field is valid only when C45E field is 1. ● Write 1 to this bit. ● Wait until MAC writes 0 to this bit. When MAC writes 0 to this bit, it implies that the data read from the PHY is available in the GD field of the MAC_MDIO_Data register and the read transfer is complete. 0: GMII Busy is disabled 1: GMII Busy is enabled	R/W
1	C45E	Clause 45 PHY Enable 0: Clause 45 PHY is disabled. Clause 22 capable PHY is connected to MDIO. 1: Clause 45 PHY is enabled. Clause 45 capable PHY is connected to MDIO.	R/W

Bit	Symbol	Function	R/W
3:2	GOC[1:0]	GMII Operation Command This is the lower bit of the operation command to the PHY or RevMII. When in SMA mode (MDIO master) this bit along with GOC_1 determines the operation to be performed to the PHY. When only RevMII is selected in configuration this bit is read-only and tied to 1. 0 0: Reserved 0 1: Write 1 0: Post Read Increment Address for Clause 45 PHY 1 1: Read When Clause 22 PHY is enabled, only Write and Read commands are valid.	R/W
4	SKAP	Skip Address Packet 0: Skip Address Packet is disabled. 1: Skip Address Packet is enabled. SMA does not send the address packets before read, write, or post-read increment address packets.*2	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
11:8	CR[3:0]	CSR Clock Range*3 The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency (100 MHz for unit 0, 200 MHz for unit 1, 2). The suggested range of CSR clock frequency applicable for each value (when Bit 11 = 0) ensures that the MDC clock is approximately between 1.0 MHz to 2.5 MHz frequency range. When Bit 11 is set, you can achieve a higher frequency of the MDC clock than the frequency limit of 2.5 MHz (specified in the IEEE 802.3) and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 0xA, the resultant MDC clock is of 12.5 MHz which is beyond the range specified in IEEE 802.3. Note that supported max frequency is 12.5 MHz in this LSI. 0x0: CSR clock/42 0x1: CSR clock/62 0x2: CSR clock/16 0x3: CSR clock/26 0x4: CSR clock/102 0x5: CSR clock/124 0x6: CSR clock/204 0x7: CSR clock/324 0x8: CSR clock/4 0x9: CSR clock/6 0xA: CSR clock/8 0xB: CSR clock/10 0xC: CSR clock/12 0xD: CSR clock/14 0xE: CSR clock/16 0xF: CSR clock/18	R/W
14:12	NTC[2:0]	Number of Trailing Clocks This field controls the number of trailing clock cycles generated on the MDC signal, after the end of MDIO frame transfer. The valid values are 000b to 111b. When you program a value of 011b in this field, MAC generates three clock cycles on the MDC line after the end of MDIO frame transfer.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20:16	RDA[4:0]	Register or Device Address*4 ● For clause 45 PHY: Specifies the selected MDIO device. ● For clause 45 PHY: Specifies the selected MDIO device. 0x00: B10T1S PLCA control 0x01: B10T1S PLCA node control 0x02: B10T1S PLCA status 0x03: B10T1S PLCA timer 0x0B: B10T1S PCS control 0x0C: B10T1S PCS status 0x0D: B10T1S PCS diagnostic 1 0x0E: B10T1S PCS diagnostic 2 0x14: B10T1S PMA/PMD extended ability 0x15: B10T1S PMA/PMD control 0x16: B10T1S PMA control 0x17: B10T1S PMA status 0x18: B10T1S test mode control Others: Reserved	R/W
25:21	PA[4:0]	Physical Layer Address Specifies the address of the PHY device, out of the 32 possible PHY devices.*4	R/W
26	BTB	Back-to-Back Transactions*5 0: Back to Back transactions disabled. MAC considers the read or write command to be complete, that is, MAC clears the GB field, only after generating the trailing clocks. This mode ensures that the NTC is always generated after each frame. 1: Back to Back transactions enabled. If NTC > 0, at the end of the frame transfer, MAC acknowledges the completion of a read or write command before transferring the trailing clocks. So, application can initiate the next command and MAC can execute it immediately without waiting for the transmission of the trailing clocks of the previous frame.	R/W
27	PSE	Preamble Suppression Enable 0: Preamble Suppression disabled. The SMA does not suppress the preamble. So, the MDIO frame has 32-bit preamble as defined in the IEEE specifications. 1: Preamble Suppression enabled. The SMA suppresses the 32-bit preamble and transmits MDIO frames with only 1 preamble bit.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

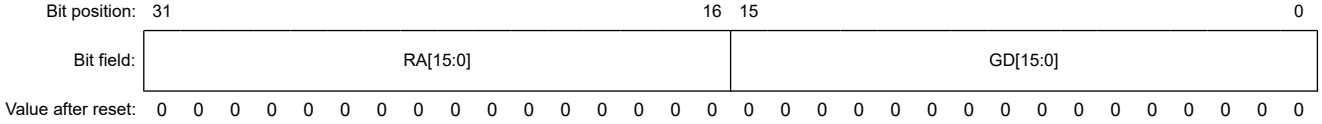
Note 1. Even if the addressed PHY is not present, there is no change in the functionality of this bit.
Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
Note 2. This bit is valid only when C45E is set.
Note 3. These bits are not used when you select Reverse MII interface. These bits are read-only when you select Reverse MII as the single PHY interface.
Note 4. The value you program in the C45E field determines if the PHY device is clause 45 capable or clause 22 capable.
Note 5. Program this field with a value 1, only when NTC > 0.

The MAC_MDIO_Address register controls the management cycles to external PHY through a management interface.

28.3.33 MAC_MDIO_Data : MDIO Data Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0204



Bit	Symbol	Function	R/W
15:0	GD[15:0]	GMII Data This field contains the 16-bit data value read from the PHY or RevMII after a Management Read operation or the 16-bit data value to be written to the PHY or RevMII before a Management Write operation.	R/W
31:16	RA[15:0]	Register Address This field is valid only when C45E is set. It contains the Register Address in the PHY to which the MDIO frame is intended for.	R/W

The MAC_MDIO_Data register stores the Write data to be written to the PHY register located at the address specified in MAC_MDIO_Address. This register also stores the Read data from the PHY register located at the address specified by MDIO Address register.

28.3.34 MAC_ARP_Address : ARP Address Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0210

Bit position:	31																													0
Bit field:	ARPPA[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ARPPA[31:0]	ARP Protocol Address This field contains the IPv4 Destination Address of the MAC. This address is used for perfect match with the Protocol Address of Target field in the received ARP packet. This field is available only when the Enable IPv4 ARP Offload option is selected.	R/W

The MAC_ARP_Address register contains the IPv4 Destination Address of the MAC.

28.3.35 MAC_CSR_SW_Ctrl : CSR Software Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0230

Bit position:	31																						8							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEEN	—	—	—	—	—	—	RCWE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RCWE	Register Clear on Write 1 Enable When this bit is set, the access mode of some register fields changes to Clear on Write 1, the application needs to set that respective bit to 1 to clear it. When this bit is reset, the access mode of these register fields remain as Clear on Read. 0: Register Clear on Write 1 is disabled 1: Register Clear on Write 1 is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	SEEN	Slave Error Response Enable When this bit is set, the MAC responds with Slave Error for accesses to reserved registers in CSR space. When this bit is reset, the MAC responds with Okay response to any register accessed from CSR space. 0: Slave Error Response is disabled 1: Slave Error Response is enabled	R/W

Bit	Symbol	Function	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

This register contains software programmable controls for changing the CSR access response and status bits clearing.

28.3.36 MAC_FPE_CTRL_STS : Frame Preemption Control and Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0234

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	TRSP	TVER	RRSP	RVER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	S1_SET_0	SRSP	SVER	EFPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EFPE	Enable Tx Frame Preemption When set Frame Preemption Tx functionality is enabled. 0: Tx Frame Preemption is disabled 1: Tx Frame Preemption is enabled	R/W
1	SVER	Send Verify mPacket*1 When set indicates hardware to send a verify mPacket. Reset by hardware after sending the Verify mPacket. 0: Send Verify mPacket is disabled 1: Send Verify mPacket is enabled	R/W
2	SRSP	Send Respond mPacket*1 When set indicates hardware to send a Respond mPacket. Reset by hardware after sending the Respond mPacket. 0: Send Respond mPacket is disabled 1: Send Respond mPacket is enabled	R/W
3	S1_SET_0	Synopsys Reserved, Must be set to "0". This field is reserved for Synopsys Internal use, and must always be set to "0" unless instructed by Synopsys. Setting to "1" might cause unexpected behavior in the IP.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
16	RVER	Received Verify Frame*2 Set when a Verify mPacket is received. An interrupt can be generated for this event if FPEIE bit of MAC_Interrupt_Enable is set. 0: Not received Verify Frame 1: Received Verify Frame	R/W
17	RRSP	Received Respond Frame*2 Set when a Respond mPacket is received. An interrupt can be generated for this event if FPEIE bit of MAC_Interrupt_Enable is set. 0: Not received Respond Frame 1: Received Respond Frame	R/W
18	TVER	Transmitted Verify Frame*2 Set when a Verify mPacket is transmitted (triggered by setting SVER field). An interrupt can be generated for this event if FPEIE bit of MAC_Interrupt_Enable is set. 0: Not transmitted Verify Frame 1: Transmitted Verify Frame	R/W

Bit	Symbol	Function	R/W
19	TRSP	Transmitted Respond Frame*2 Set when a Respond mPacket is transmitted (triggered by setting SRSP field). An interrupt can be generated for this event if FPEIE bit of MAC_Interrupt_Enable is set. 0: Not transmitted Respond Frame 1: Transmitted Respond Frame	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
Note 2. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.

This register controls the operation of Frame Preemption.

28.3.37 MAC_Ext_Cfg1 : Split Mode Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0238

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	SAVE	—	SAVO[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPLM[1:0]		—	SPLOFST[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

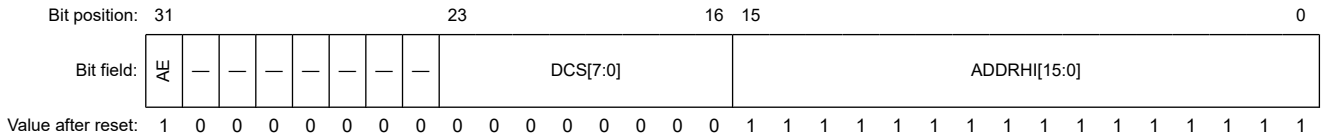
Bit	Symbol	Function	R/W
6:0	SPLOFST[6:0]	Split Offset These bits indicate the value of offset from the beginning of Length/Type field at which header split should take place when the appropriate SPLM[1:0] is selected. The reset value of this field is 2 bytes indicating a split at L2 header. Value is in terms of bytes.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
9:8	SPLM[1:0]	Split Mode These bits indicate the mode of splitting the incoming Rx packets. They are: 0 0: Split at L3/L4 header 0 1: Split at L2 header with an offset. Always Split at SPLOFST[6:0] bytes from the beginning of Length/Type field of the Frame 1 0: Combination mode: Split similar to SPLM[1:0] = 00b for IP packets that are untagged or tagged and VLAN stripped 1 1: Reserved	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
22:16	SAVO[6:0]	Split AV Offset When SAVE bit is set to 1, and the received packet is an AV Type packet, these bits indicate the value of the offset from the beginning of Length/Type field at which header should be split, when appropriate SPLM[1:0] is selected. The reset value of this field is 2 bytes, indicating a split at L2 header. Value is in terms of bytes.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
24	SAVE	Split AV Enable • When this bit is set to 1, and the received packet is an AV Type packet, the header is split at SAVO[6:0] bytes from the beginning of Length/Type field of the packet, for L2 Split. • When this bit is set to 0, header is split at SPLOFST[6:0] bytes from the beginning of Length/Type field of the frame, for L2 Split.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

This register contains Split mode control field and offset field for Split Header feature.

28.3.38 MAC_ADDRESS0_HIGH : MAC Address 0 High register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0300



Bit	Symbol	Function	R/W
15:0	ADDRHI[15:0]	MAC Address0 [47:32] This field contains the upper 16 bits [47:32] of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.	R/W
23:16	DCS[7:0]	DMA Channel Select This field indicates the Rx DMA channel number or DMA channel numbers to which the MAC routes the received packet. When the MAC receives a packet with multicast or broadcast destination address, MAC routes the packet to multiple Rx DMA channels. The PDC field of the MAC_Ext_Configuration register determines if the MAC interprets the value of this field as a binary number or a one-hot encoded number.	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	AE	Address Enable This bit must be always set to 1.	R

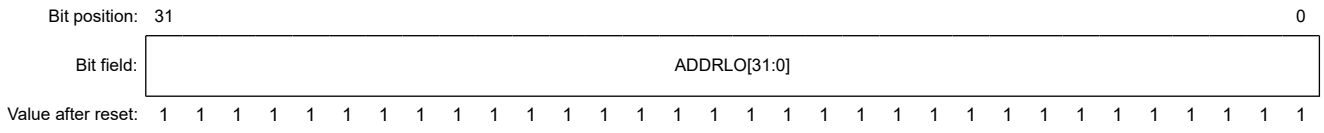
The MAC_ADDRESS0_HIGH register holds the upper 16 bits of the first 6-byte MAC address of the station. The first DA byte that is received on the (G)MII interface corresponds to the LS byte (Bits [7:0]) of the MAC Address Low register. For example, if 0x112233445566 is received (0x11 in lane 0 of the first column) on the (G)MII as the destination address, then the MacAddress0 Register [47:0] is compared with 0x665544332211.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, then the synchronization is triggered only when Bits [31:24] of the MAC Address0 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

28.3.39 MAC_ADDRESS0_LOW : MAC Address 0 Low register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0304



Bit	Symbol	Function	R/W
31:0	ADDRLO[31:0]	MAC Address0 [31:0] This field contains the lower 32 bits of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.	R/W

The MAC_ADDRESS0_LOW register holds the lower 32 bits of the 6-byte first MAC address of the station.

28.3.40 MAC_ADDRESSn_HIGH : MAC Address n High register (n = 1 to 31)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0308 + 0x08 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AE	SA	MBC[5:0]						DCS[7:0]								ADDRHI[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	ADDRHI[15:0]	MAC Address1 [47:32] This field contains the upper 16 bits [47:32] of the second 6-byte MAC address.	R/W
23:16	DCS[7:0]	DMA Channel Select If the PDC bit of MAC_Ext_Configuration register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC_ADDRESSn content is routed. If the PDC bit of MAC_Ext_Configuration register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC_ADDRESSn content is routed.	R/W
29:24	MBC[5:0]	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address1 registers. Each bit controls the masking of the bytes as follows: Bit 29: MAC_ADDRESSn_High[15:8] Bit 28: MAC_ADDRESSn_High[7:0] Bit 27: MAC_ADDRESSn_Low[31:24] ⋮ Bit 24: MAC_ADDRESSn_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.	R/W
30	SA	Source Address When this bit is set, the MAC Address1 [47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC Address1 [47:0] is used to compare with the DA fields of the received packet. 0: Compare with Destination Address 1: Compare with Source Address	R/W
31	AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0: Address is ignored 1: Address is enabled	R/W

The MAC_ADDRESSn_HIGH register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits [31:24] of the MAC Address1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

28.3.41 MAC_ADDRESSn_LOW : MAC Address n Low register (n = 1 to 31)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x030C + 0x08 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADDRLO[31:0]																															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	ADDRLO[31:0]	MAC Address1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.	R/W

The MAC_ADDRESSn_LOW register holds the lower 32 bits of the second 6-byte MAC address of the station.

28.3.42 MMC_Control : MMC Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0700

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	UCDBC	—	—	CNTPRSTLVL	CNTPRST	CNTFREEZ	RSTONRD	CNTSTOPRO	CNTRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CNTRST	Counters Reset*1 When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. 0: Counters are not reset 1: All counters are reset	R/W
1	CNTSTOPRO	Counter Stop Rollover When this bit is set, the counter does not roll over to zero after reaching the maximum value. 0: Counter Stop Rollover is disabled 1: Counter Stop Rollover is enabled	R/W
2	RSTONRD	Reset on Read When this bit is set, the MMC counters are reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (Bits [7:0]) is read. 0: Reset on Read is disabled 1: Reset on Read is enabled	R/W
3	CNTFREEZ	MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. Until this bit is reset to 0, no MMC counter is updated because of any transmitted or received packet. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode. 0: MMC Counter Freeze is disabled 1: MMC Counter Freeze is enabled	R/W
4	CNTPRST	Counters Preset*1 When this bit is set, all counters are initialized or preset to almost full or almost half according to the CNTPRSTLVL bit. This bit is cleared automatically after 1 clock cycle. This bit, along with the CNTPRSTLVL bit, is useful for debugging and testing the assertion of interrupts because of MMC counter becoming half-full or full. 0: Counters Preset is disabled 1: Counters Preset is enabled	R/W

Bit	Symbol	Function	R/W
5	CNTPRSTLVL	Full-Half Preset When this bit is low and the CNTPRST bit is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (Half 2 KBytes) and all packet-counters gets preset to 0x7FFF_FFF0 (Half 16). When this bit is high and the CNTPRST bit is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (Full 2 KBytes) and all packet-counters gets preset to 0xFFFF_FFF0 (Full 16). For 16-bit counters, the almost-half preset values are 0x7800 and 0x7FF0 for the respective octet and packet counters. Similarly, the almost-full preset values for the 16-bit counters are 0xF800 and 0xFFF0. 0: Full-Half Preset is disabled 1: Full-Half Preset is enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	UCDBC	Update MMC Counters for Dropped Broadcast Packets*2 When set, the MAC updates all related MMC Counters for Broadcast packets that are dropped because of the setting of the DBF bit of MAC_Packet_Filter register. When reset, the MMC Counters are not updated for dropped Broadcast packets. 0: Update MMC Counters for Dropped Broadcast Packets is disabled 1: Update MMC Counters for Dropped Broadcast Packets is enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restrictions apply.

- Initiates the counter reset when application writes 1.
- MAC writes 0 after the counter reset is complete.
- MAC ignores when application writes 0.

Note 2. The CNTRST bit has a higher priority than the CNTPRST bit. Therefore, when the software tries to set both bits in the same write cycle, all counters are cleared and the CNTPRST bit is not set.

This register establishes the operating mode of MMC.

28.3.43 MMC_Rx_Interrupt : MMC Receive Interrupt Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0704

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	RXLPITRCIS	RXLPIUSCIS	RXCTRLPIS	RXRCVERRPIS	RXWDOGPIS	RXVLANGBPIS	RXFOVPIS	RXPAUSPIS	RXORANGEPIS	RXLENERPIS	RXUCGPIS	RX1024TMAXOCTGBPIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RX512T1023OCTGBPIS	RX256T511OCTGBPIS	RX128T255OCTGBPIS	RX65T127OCTGBPIS	RX64OCTGBPIS	RXOSIZEGPIS	RXUSIZEGPIS	RXJABERPIS	RXRUNTPIS	RXALGNERPIS	RXCRCERPIS	RXMCGPIS	RXBCGPIS	RXGOCTIS	RXGBOCTIS	RXGBPKTIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RXGBPKTIS	MMC Receive Good Bad Packet Counter Interrupt Status*1 This bit is set when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive Good Bad Packet Counter Interrupt Status not detected 1: MMC Receive Good Bad Packet Counter Interrupt Status detected	R
1	RXGBOCTIS	MMC Receive Good Bad Octet Counter Interrupt Status*1 This bit is set when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive Good Bad Octet Counter Interrupt Status not detected 1: MMC Receive Good Bad Octet Counter Interrupt Status detected	R

Bit	Symbol	Function	R/W
2	RXGOCTIS	MMC Receive Good Octet Counter Interrupt Status*1 This bit is set when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Good Octet Counter Interrupt Status not detected 1: MMC Receive Good Octet Counter Interrupt Status detected	R
3	RXBCGPIS	MMC Receive Broadcast Good Packet Counter Interrupt Status*1 This bit is set when the rxbroadcastpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Broadcast Good Packet Counter Interrupt Status not detected 1: MMC Receive Broadcast Good Packet Counter Interrupt Status detected	R
4	RXMCGPIS	MMC Receive Multicast Good Packet Counter Interrupt Status*1 This bit is set when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Multicast Good Packet Counter Interrupt Status not detected 1: MMC Receive Multicast Good Packet Counter Interrupt Status detected	R
5	RXCRCERPIS	MMC Receive CRC Error Packet Counter Interrupt Status*1 This bit is set when the rxcrcerror counter reaches half of the maximum value or the maximum value. 0: MMC Receive CRC Error Packet Counter Interrupt Status not detected 1: MMC Receive CRC Error Packet Counter Interrupt Status detected	R
6	RXALGNERPIS	MMC Receive Alignment Error Packet Counter Interrupt Status*1 This bit is set when the rxalignmenterror counter reaches half of the maximum value or the maximum value. 0: MMC Receive Alignment Error Packet Counter Interrupt Status not detected 1: MMC Receive Alignment Error Packet Counter Interrupt Status detected	R
7	RXRUNTPIS	MMC Receive Runt Packet Counter Interrupt Status*1 This bit is set when the rxrunterror counter reaches half of the maximum value or the maximum value. 0: MMC Receive Runt Packet Counter Interrupt Status not detected 1: MMC Receive Runt Packet Counter Interrupt Status detected	R
8	RXJABERPIS	MMC Receive Jabber Error Packet Counter Interrupt Status*1 This bit is set when the rxjabbererror counter reaches half of the maximum value or the maximum value. 0: MMC Receive Jabber Error Packet Counter Interrupt Status not detected 1: MMC Receive Jabber Error Packet Counter Interrupt Status detected	R
9	RXUSIZEGPIS	MMC Receive Undersize Good Packet Counter Interrupt Status*1 This bit is set when the rxundersize_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Undersize Good Packet Counter Interrupt Status not detected 1: MMC Receive Undersize Good Packet Counter Interrupt Status detected	R
10	RXOSIZEGPIS	MMC Receive Oversize Good Packet Counter Interrupt Status*1 This bit is set when the rxoversize_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Oversize Good Packet Counter Interrupt Status not detected 1: MMC Receive Oversize Good Packet Counter Interrupt Status detected	R
11	RX64OCTGBPIS	MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the rx64octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status detected	R
12	RX65T127OCTGBPIS	MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status detected	R

Bit	Symbol	Function	R/W
13	RX128T255OCTGBPIS	MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status detected	R
14	RX256T511OCTGBPIS	MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status detected	R
15	RX512T1023OCTGBPIS	MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status detected	R
16	RX1024TMAXOCTGBPIS	MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status detected	R
17	RXUCGPIS	MMC Receive Unicast Good Packet Counter Interrupt Status*1 This bit is set when the rxunicastpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Unicast Good Packet Counter Interrupt Status not detected 1: MMC Receive Unicast Good Packet Counter Interrupt Status detected	R
18	RXLENERPIS	MMC Receive Length Error Packet Counter Interrupt Status*1 This bit is set when the rxlengtherror counter reaches half of the maximum value or the maximum value. 0: MMC Receive Length Error Packet Counter Interrupt Status not detected 1: MMC Receive Length Error Packet Counter Interrupt Status detected	R
19	RXORANGEPIS	MMC Receive Out Of Range Error Packet Counter Interrupt Status.*1 This bit is set when the rxoutofrangetype counter reaches half of the maximum value or the maximum value. 0: MMC Receive Out Of Range Error Packet Counter Interrupt Status not detected 1: MMC Receive Out Of Range Error Packet Counter Interrupt Status detected	R
20	RXPAUSPIS	MMC Receive Pause Packet Counter Interrupt Status*1 This bit is set when the rxpausepackets counter reaches half of the maximum value or the maximum value. 0: MMC Receive Pause Packet Counter Interrupt Status not detected 1: MMC Receive Pause Packet Counter Interrupt Status detected	R
21	RXFOVPIS	MMC Receive FIFO Overflow Packet Counter Interrupt Status*1 This bit is set when the rxfifooverflow counter reaches half of the maximum value or the maximum value. 0: MMC Receive FIFO Overflow Packet Counter Interrupt Status not detected 1: MMC Receive FIFO Overflow Packet Counter Interrupt Status detected	R
22	RXVLANGBPIS	MMC Receive VLAN Good Bad Packet Counter Interrupt Status*1 This bit is set when the rxvlanpackets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive VLAN Good Bad Packet Counter Interrupt Status not detected 1: MMC Receive VLAN Good Bad Packet Counter Interrupt Status detected	R

Bit	Symbol	Function	R/W
23	RXWDOGPIS	MMC Receive Watchdog Error Packet Counter Interrupt Status*1 This bit is set when the rxwatchdog error counter reaches half of the maximum value or the maximum value. 0: MMC Receive Watchdog Error Packet Counter Interrupt Status not detected 1: MMC Receive Watchdog Error Packet Counter Interrupt Status detected	R
24	RXRCVERRPIS	MMC Receive Error Packet Counter Interrupt Status*1 This bit is set when the rxrcverror counter reaches half of the maximum value or the maximum value. 0: MMC Receive Error Packet Counter Interrupt Status not detected 1: MMC Receive Error Packet Counter Interrupt Status detected	R
25	RXCTRLPIS	MMC Receive Control Packet Counter Interrupt Status*1 This bit is set when the rxctrlpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Control Packet Counter Interrupt Status not detected 1: MMC Receive Control Packet Counter Interrupt Status detected	R
26	RXLPIUSCIS	MMC Receive LPI microsecond counter interrupt status*1 This bit is set when the Rx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Receive LPI microsecond Counter Interrupt Status not detected 1: MMC Receive LPI microsecond Counter Interrupt Status detected	R
27	RXLPITRCIS	MMC Receive LPI transition counter interrupt status*1 This bit is set when the Rx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Receive LPI transition Counter Interrupt Status not detected 1: MMC Receive LPI transition Counter Interrupt Status detected	R
31:28	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

This register maintains the interrupts generated from all Receive statistics counters.

The MMC_Rx_Interrupt register maintains the interrupts that are generated when the following occur:

- Receive statistic counters reach half of their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter)
- Receive statistic counters cross their maximum values (0xFFFF_FFFF for 32 bit counter and 0xFFFF for 16 bit counter)

When the Counter Stop Rollover is set, interrupts are set but the counter remains at all-ones. The MMC Receive Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits [7:0]) of the respective counter must be read to clear the interrupt bit.

Note: R_SS_RC means that this register bit is set internally, and it is cleared when the Counter register is read.

28.3.44 MMC_Tx_Interrupt : MMC Transmit Interrupt Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0708

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	TXLPITRCIS	TXLPIUSCIS	TXOSIZEGPIS	TXVLANGPIS	TXPAUSPIS	TXEXDEFPIS	TXGPKTIS	TXGOCTIS	TXCARERPIS	TXEXCOLPIS	TXLATCOLPIS	TXDEFPIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TXMCOLGPIS	TXSCOLGPIS	TXUFLOWERPIS	TXBCGBPIS	TXMCGBPIS	TXUCGBPIS	TX1024TMAXOCTGBPIS	TX512T1023OCTGBPIS	TX256T511OCTGBPIS	TX128T255OCTGBPIS	TX65T127OCTGBPIS	TX64OCTGBPIS	TXMCGPIS	TXBCGPIS	TXGBPKTIS	TXGBOCTIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXGBOCTIS	MMC Transmit Good Bad Octet Counter Interrupt Status*1 This bit is set when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Good Bad Octet Counter Interrupt Status not detected 1: MMC Transmit Good Bad Octet Counter Interrupt Status detected	R
1	TXGBPKTIS	MMC Transmit Good Bad Packet Counter Interrupt Status*1 This bit is set when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Good Bad Packet Counter Interrupt Status not detected 1: MMC Transmit Good Bad Packet Counter Interrupt Status detected	R
2	TXBCGPIS	MMC Transmit Broadcast Good Packet Counter Interrupt Status*1 This bit is set when the txbroadcastpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Broadcast Good Packet Counter Interrupt Status not detected 1: MMC Transmit Broadcast Good Packet Counter Interrupt Status detected	R
3	TXMCGPIS	MMC Transmit Multicast Good Packet Counter Interrupt Status*1 This bit is set when the txmulticastpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Multicast Good Packet Counter Interrupt Status not detected 1: MMC Transmit Multicast Good Packet Counter Interrupt Status detected	R
4	TX64OCTGBPIS	MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the tx64octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status detected	R
5	TX65T127OCTGBPIS	MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the tx65to127octets_gb counter reaches half the maximum value, and also when it reaches the maximum value. 0: MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status detected	R
6	TX128T255OCTGBPIS	MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status detected	R
7	TX256T511OCTGBPIS	MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status detected	R
8	TX512T1023OCTGBPIS	MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status detected	R
9	TX1024TMAXOCTGBPIS	MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status*1 This bit is set when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status not detected 1: MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status detected	R

Bit	Symbol	Function	R/W
10	TXUCGBPIS	MMC Transmit Unicast Good Bad Packet Counter Interrupt Status*1 This bit is set when the txunicastpackets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Unicast Good Bad Packet Counter Interrupt Status not detected 1: MMC Transmit Unicast Good Bad Packet Counter Interrupt Status detected	R
11	TXMCGBPIS	MMC Transmit Multicast Good Bad Packet Counter Interrupt Status*1 The bit is set when the txmulticastpackets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Multicast Good Bad Packet Counter Interrupt Status not detected 1: MMC Transmit Multicast Good Bad Packet Counter Interrupt Status detected	R
12	TXBCGBPIS	MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status*1 This bit is set when the txbroadcastpackets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status not detected 1: MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status detected	R
13	TXUFLOWERPIS	MMC Transmit Underflow Error Packet Counter Interrupt Status*1 This bit is set when the txunderflowerror counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Underflow Error Packet Counter Interrupt Status not detected 1: MMC Transmit Underflow Error Packet Counter Interrupt Status detected	R
14	TXSCOLGPIS	MMC Transmit Single Collision Good Packet Counter Interrupt Status*1 This bit is set when the txsinglecol_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Single Collision Good Packet Counter Interrupt Status not detected 1: MMC Transmit Single Collision Good Packet Counter Interrupt Status detected	R
15	TXMCOLGPIS	MMC Transmit Multiple Collision Good Packet Counter Interrupt Status*1 This bit is set when the txmulticol_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Multiple Collision Good Packet Counter Interrupt Status not detected 1: MMC Transmit Multiple Collision Good Packet Counter Interrupt Status detected	R
16	TXDEFPIS	MMC Transmit Deferred Packet Counter Interrupt Status*1 This bit is set when the txdeferred counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Deferred Packet Counter Interrupt Status not detected 1: MMC Transmit Deferred Packet Counter Interrupt Status detected	R
17	TXLATCOLPIS	MMC Transmit Late Collision Packet Counter Interrupt Status*1 This bit is set when the txlatecol counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Late Collision Packet Counter Interrupt Status not detected 1: MMC Transmit Late Collision Packet Counter Interrupt Status detected	R
18	TXEXCOLPIS	MMC Transmit Excessive Collision Packet Counter Interrupt Status*1 This bit is set when the txexesscol counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Excessive Collision Packet Counter Interrupt Status not detected 1: MMC Transmit Excessive Collision Packet Counter Interrupt Status detected	R
19	TXCARERPIS	MMC Transmit Carrier Error Packet Counter Interrupt Status*1 This bit is set when the txcarriererror counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Carrier Error Packet Counter Interrupt Status not detected 1: MMC Transmit Carrier Error Packet Counter Interrupt Status detected	R
20	TXGOCTIS	MMC Transmit Good Octet Counter Interrupt Status*1 This bit is set when the txoctetcount_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Good Octet Counter Interrupt Status not detected 1: MMC Transmit Good Octet Counter Interrupt Status detected	R

Bit	Symbol	Function	R/W
21	TXGPKTIS	MMC Transmit Good Packet Counter Interrupt Status*1 This bit is set when the txpacketcount_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Good Packet Counter Interrupt Status not detected 1: MMC Transmit Good Packet Counter Interrupt Status detected	R
22	TXEXDEFPIS	MMC Transmit Excessive Deferral Packet Counter Interrupt Status*1 This bit is set when the txexcessdef counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Excessive Deferral Packet Counter Interrupt Status not detected 1: MMC Transmit Excessive Deferral Packet Counter Interrupt Status detected	R
23	TXPAUSPIS	MMC Transmit Pause Packet Counter Interrupt Status*1 This bit is set when the txpausepacketserror counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Pause Packet Counter Interrupt Status not detected 1: MMC Transmit Pause Packet Counter Interrupt Status detected	R
24	TXVLANGPIS	MMC Transmit VLAN Good Packet Counter Interrupt Status*1 This bit is set when the txvlanpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit VLAN Good Packet Counter Interrupt Status not detected 1: MMC Transmit VLAN Good Packet Counter Interrupt Status detected	R
25	TXOSIZEGPIS	MMC Transmit Oversize Good Packet Counter Interrupt Status*1 This bit is set when the txoversize_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Oversize Good Packet Counter Interrupt Status not detected 1: MMC Transmit Oversize Good Packet Counter Interrupt Status detected	R
26	TXLPIUSCIS	MMC Transmit LPI microsecond counter interrupt status*1 This bit is set when the Tx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Transmit LPI microsecond Counter Interrupt Status not detected 1: MMC Transmit LPI microsecond Counter Interrupt Status detected	R
27	TXLPITRCIS	MMC Transmit LPI transition counter interrupt status*1 This bit is set when the Tx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Transmit LPI transition Counter Interrupt Status not detected 1: MMC Transmit LPI transition Counter Interrupt Status detected	R
31:28	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

This register maintains the interrupts generated from all Transmit statistics counters.

The MMC_Tx_Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values(0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter).

When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones.

The MMC_Tx_Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read.

The least significant byte lane (Bits [7:0]) of the respective counter must be read to clear the interrupt bit.

28.3.45 MMC_Rx_Interrupt_Mask : MMC Receive Interrupt Mask Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x070C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	RXLPITRCIM	RXLPIUSCIM	RXCTRLPIM	RXRCVERRPIM	RXWDOGPIM	RXVLANGBPIM	RXFOVPIM	RXPAUSPIM	RXORANGEPIM	RXLENERPIM	RXUCGPIM	RX1024TMAXOCTGBPIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RX512T1023OCTGBPIM	RX256T511OCTGBPIM	RX128T255OCTGBPIM	RX65T127OCTGBPIM	RX64OCTGBPIM	RXOSIZEGPIM	RXUSIZEGPIM	RXJABERPIM	RXRUNTPIM	RXALGNERPIM	RXCRCERPIM	RXMCGPIM	RXBCGPIM	RXGOCTIM	RXGBOCTIM	RXGBPKTIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RXGBPKTIM	MMC Receive Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Receive Good Bad Packet Counter Interrupt Mask is enabled	R/W
1	RXGBOCTIM	MMC Receive Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive Good Bad Octet Counter Interrupt Mask is disabled 1: MMC Receive Good Bad Octet Counter Interrupt Mask is enabled	R/W
2	RXGOCTIM	MMC Receive Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Good Octet Counter Interrupt Mask is disabled 1: MMC Receive Good Octet Counter Interrupt Mask is enabled	R/W
3	RXBCGPIM	MMC Receive Broadcast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxbroadcastpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Broadcast Good Packet Counter Interrupt Mask is disabled 1: MMC Receive Broadcast Good Packet Counter Interrupt Mask is enabled	R/W
4	RXMCGPIM	MMC Receive Multicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Multicast Good Packet Counter Interrupt Mask is disabled 1: MMC Receive Multicast Good Packet Counter Interrupt Mask is enabled	R/W
5	RXCRCERPIM	MMC Receive CRC Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxcrcerror counter reaches half of the maximum value or the maximum value. 0: MMC Receive CRC Error Packet Counter Interrupt Mask is disabled 1: MMC Receive CRC Error Packet Counter Interrupt Mask is enabled	R/W
6	RXALGNERPIM	MMC Receive Alignment Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxalignmenterror counter reaches half of the maximum value or the maximum value. 0: MMC Receive Alignment Error Packet Counter Interrupt Mask is disabled 1: MMC Receive Alignment Error Packet Counter Interrupt Mask is enabled	R/W
7	RXRUNTPIM	MMC Receive Runt Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxrunterror counter reaches half of the maximum value or the maximum value. 0: MMC Receive Runt Packet Counter Interrupt Mask is disabled 1: MMC Receive Runt Packet Counter Interrupt Mask is enabled	R/W

Bit	Symbol	Function	R/W
8	RXJABERPIM	MMC Receive Jabber Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxjabbererror counter reaches half of the maximum value or the maximum value. 0: MMC Receive Jabber Error Packet Counter Interrupt Mask is disabled 1: MMC Receive Jabber Error Packet Counter Interrupt Mask is enabled	R/W
9	RXUSIZEGPIM	MMC Receive Undersize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxundersize_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Undersize Good Packet Counter Interrupt Mask is disabled 1: MMC Receive Undersize Good Packet Counter Interrupt Mask is enabled	R/W
10	RXOSIZEGPIM	MMC Receive Oversize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoversize_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Oversize Good Packet Counter Interrupt Mask is disabled 1: MMC Receive Oversize Good Packet Counter Interrupt Mask is enabled	R/W
11	RX64OCTGBPIM	MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx64octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W
12	RX65T127OCTGBPIM	MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W
13	RX128T255OCTGBPIM	MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W
14	RX256T511OCTGBPIM	MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W
15	RX512T1023OCTGBPIM	MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W
16	RX1024TMAXOCTGBPIM	MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask. Setting this bit masks the interrupt when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W
17	RXUCGPIM	MMC Receive Unicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxunicastpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Unicast Good Packet Counter Interrupt Mask is disabled 1: MMC Receive Unicast Good Packet Counter Interrupt Mask is enabled	R/W

Bit	Symbol	Function	R/W
18	RXLENERPIM	MMC Receive Length Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxlengtherror counter reaches half of the maximum value or the maximum value. 0: MMC Receive Length Error Packet Counter Interrupt Mask is disabled 1: MMC Receive Length Error Packet Counter Interrupt Mask is enabled	R/W
19	RXORANGEPIM	MMC Receive Out Of Range Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoutofrangetype counter reaches half of the maximum value or the maximum value. 0: MMC Receive Out Of Range Error Packet Counter Interrupt Mask is disabled 1: MMC Receive Out Of Range Error Packet Counter Interrupt Mask is enabled	R/W
20	RXPAUSPIM	MMC Receive Pause Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxpausepackets counter reaches half of the maximum value or the maximum value. 0: MMC Receive Pause Packet Counter Interrupt Mask is disabled 1: MMC Receive Pause Packet Counter Interrupt Mask is enabled	R/W
21	RXFOVPIM	MMC Receive FIFO Overflow Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxfifooverflow counter reaches half of the maximum value or the maximum value. 0: MMC Receive FIFO Overflow Packet Counter Interrupt Mask is disabled 1: MMC Receive FIFO Overflow Packet Counter Interrupt Mask is enabled	R/W
22	RXVLANGBPIM	MMC Receive VLAN Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxvlanpackets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Receive VLAN Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Receive VLAN Good Bad Packet Counter Interrupt Mask is enabled	R/W
23	RXWDOGPIM	MMC Receive Watchdog Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxwatchdog counter reaches half of the maximum value or the maximum value. 0: MMC Receive Watchdog Error Packet Counter Interrupt Mask is disabled 1: MMC Receive Watchdog Error Packet Counter Interrupt Mask is enabled	R/W
24	RXRCVERRPIM	MMC Receive Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxrcverror counter reaches half of the maximum value or the maximum value. 0: MMC Receive Error Packet Counter Interrupt Mask is disabled 1: MMC Receive Error Packet Counter Interrupt Mask is enabled	R/W
25	RXCTRLPIM	MMC Receive Control Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxctrlpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Receive Control Packet Counter Interrupt Mask is disabled 1: MMC Receive Control Packet Counter Interrupt Mask is enabled	R/W
26	RXLPIUSCIM	MMC Receive LPI microsecond counter interrupt Mask Setting this bit masks the interrupt when the Rx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Receive LPI microsecond counter interrupt Mask is disabled 1: MMC Receive LPI microsecond counter interrupt Mask is enabled	R/W
27	RXLPITRCIM	MMC Receive LPI transition counter interrupt Mask Setting this bit masks the interrupt when the Rx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Receive LPI transition counter interrupt Mask is disabled 1: MMC Receive LPI transition counter interrupt Mask is enabled	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

This register maintains the masks for interrupts generated from all Receive statistics counters.

The MMC_Rx_Interrupt_Mask register maintains the masks for the interrupts generated when receive statistic counters reach half of their maximum value or the maximum values.

28.3.46 MMC_Tx_Interrupt_Mask : MMC Transmit Interrupt Mask Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0710

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	TXLPITRCIM	TXLPIUSCIM	TXOSIZEGPIM	TXVLANGPIM	TXPAUSPIM	TXEXDEFPIM	TXGPKTIM	TXGOCTIM	TXCARERPIM	TXEXCOLPIM	TXLATCOLPIM	TXDEFPIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TXMCOLGPIM	TXSCOLGPIM	TXUFLOWERPIM	TXBCGBPIM	TXMCGBPIM	TXUCGBPIM	TX1024TMAXOCTGBPIM	TX512T1023OCTGBPIM	TX256T511OCTGBPIM	TX128T255OCTGBPIM	TX65T127OCTGBPIM	TX64OCTGBPIM	TXMCGPIM	TXBCGPIM	TXGBPKTIM	TXGBOCTIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXGBOCTIM	MMC Transmit Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Good Bad Octet Counter Interrupt Mask is disabled 1: MMC Transmit Good Bad Octet Counter Interrupt Mask is enabled	R/W
1	TXGBPKTIM	MMC Transmit Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Transmit Good Bad Packet Counter Interrupt Mask is enabled	R/W
2	TXBCGPIM	MMC Transmit Broadcast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txbroadcastpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Broadcast Good Packet Counter Interrupt Mask is disabled 1: MMC Transmit Broadcast Good Packet Counter Interrupt Mask is enabled	R/W
3	TXMCGPIM	MMC Transmit Multicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticastpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Multicast Good Packet Counter Interrupt Mask is disabled 1: MMC Transmit Multicast Good Packet Counter Interrupt Mask is enabled	R/W
4	TX64OCTGBPIM	MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx64octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W
5	TX65T127OCTGBPIM	MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx65to127octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W
6	TX128T255OCTGBPIM	MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W

Bit	Symbol	Function	R/W
7	TX256T511OCTGBPIM	MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W
8	TX512T1023OCTGBPIM	MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W
9	TX1024TMAXOCTGBPIM	MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is enabled	R/W
10	TXUCGBPIM	MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txunicastpackets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask is enabled	R/W
11	TXMCGBPIM	MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticastpackets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask is enabled	R/W
12	TXBCGBPIM	MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txbroadcastpackets_gb counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask is disabled 1: MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask is enabled	R/W
13	TXUFLOWERPIM	MMC Transmit Underflow Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txunderflowerror counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Underflow Error Packet Counter Interrupt Mask is disabled 1: MMC Transmit Underflow Error Packet Counter Interrupt Mask is enabled	R/W
14	TXSCOLGPIM	MMC Transmit Single Collision Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txsinglecol_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Single Collision Good Packet Counter Interrupt Mask is disabled 1: MMC Transmit Single Collision Good Packet Counter Interrupt Mask is enabled	R/W
15	TXMCOLGPIM	MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticol_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask is disabled 1: MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask is enabled	R/W
16	TXDEFPIM	MMC Transmit Deferred Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txdeferred counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Deferred Packet Counter Interrupt Mask is disabled 1: MMC Transmit Deferred Packet Counter Interrupt Mask is enabled	R/W
17	TXLATCOLPIM	MMC Transmit Late Collision Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txlatecol counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Late Collision Packet Counter Interrupt Mask is disabled 1: MMC Transmit Late Collision Packet Counter Interrupt Mask is enabled	R/W

Bit	Symbol	Function	R/W
18	TXEXCOLPIM	MMC Transmit Excessive Collision Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txexcesscol counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Excessive Collision Packet Counter Interrupt Mask is disabled 1: MMC Transmit Excessive Collision Packet Counter Interrupt Mask is enabled	R/W
19	TXCARERPIM	MMC Transmit Carrier Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txcarriererror counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Carrier Error Packet Counter Interrupt Mask is disabled 1: MMC Transmit Carrier Error Packet Counter Interrupt Mask is enabled	R/W
20	TXGOCTIM	MMC Transmit Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Good Octet Counter Interrupt Mask is disabled 1: MMC Transmit Good Octet Counter Interrupt Mask is enabled	R/W
21	TXGPKTIM	MMC Transmit Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Good Packet Counter Interrupt Mask is disabled 1: MMC Transmit Good Packet Counter Interrupt Mask is enabled	R/W
22	TXEXDEFPIM	MMC Transmit Excessive Deferral Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txexcessdef counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Excessive Deferral Packet Counter Interrupt Mask is disabled 1: MMC Transmit Excessive Deferral Packet Counter Interrupt Mask is enabled	R/W
23	TXPAUSPIM	MMC Transmit Pause Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpausepackets counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Pause Packet Counter Interrupt Mask is disabled 1: MMC Transmit Pause Packet Counter Interrupt Mask is enabled	R/W
24	TXVLANGPIM	MMC Transmit VLAN Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txvlanpackets_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit VLAN Good Packet Counter Interrupt Mask is disabled 1: MMC Transmit VLAN Good Packet Counter Interrupt Mask is enabled	R/W
25	TXOSIZEGPIM	MMC Transmit Oversize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txoversize_g counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Oversize Good Packet Counter Interrupt Mask is disabled 1: MMC Transmit Oversize Good Packet Counter Interrupt Mask is enabled	R/W
26	TXLPIUSCIM	MMC Transmit LPI microsecond counter interrupt Mask Setting this bit masks the interrupt when the Tx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Transmit LPI microsecond counter interrupt Mask is disabled 1: MMC Transmit LPI microsecond counter interrupt Mask is enabled	R/W
27	TXLPITRCIM	MMC Transmit LPI transition counter interrupt Mask Setting this bit masks the interrupt when the Tx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Transmit LPI transition counter interrupt Mask is disabled 1: MMC Transmit LPI transition counter interrupt Mask is enabled	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

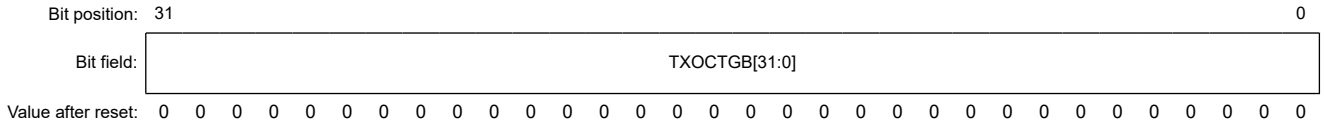
This register maintains the masks for interrupts generated from all Transmit statistics counters.

The MMC_Tx_Interrupt_Mask register maintains the masks for the interrupts generated when the transmit statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

28.3.47 Tx_Octet_Count_Good_Bad : Transmit Octet Count for Good and Bad 64 Byte Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0714



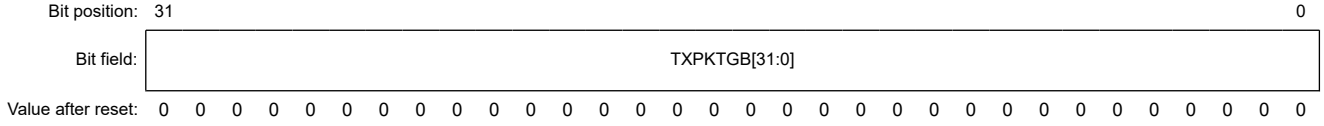
Bit	Symbol	Function	R/W
31:0	TXOCTGB[31:0]	Tx Octet Count Good Bad This field indicates the number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad packets.	R

This register provides the number of bytes transmitted by the GMAC, exclusive of preamble and retried bytes, in good and bad packets.

28.3.48 Tx_Packet_Count_Good_Bad : Transmit Packet Count for Good and Bad Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0718



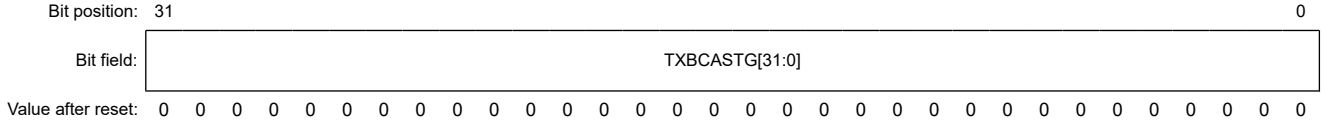
Bit	Symbol	Function	R/W
31:0	TXPKTGB[31:0]	Tx Packet Count Good Bad This field indicates the number of good and bad packets transmitted, exclusive of retried packets.	R

This register provides the number of good and bad packets transmitted by GMAC, exclusive of retried packets.

28.3.49 Tx_Broadcast_Packets_Good : Transmit Packet Count for Good Broadcast Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x071C



Bit	Symbol	Function	R/W
31:0	TXBCASTG[31:0]	Tx Broadcast Packets Good This field indicates the number of good broadcast packets transmitted.	R

This register provides the number of good broadcast packets transmitted by GMAC.

28.3.50 Tx_Multicast_Packets_Good : Transmit Packet Count for Good Multicast Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0720

Bit position:	31																												0
Bit field:	TXMCASTG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXMCASTG[31:0]	Tx Multicast Packets Good This field indicates the number of good multicast packets transmitted.	R

This register provides the number of good multicast packets transmitted by GMAC.

28.3.51 Tx_64Octets_Packets_Good_Bad : Transmit Octet Count for Good and Bad 64 Byte Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0724

Bit position:	31																												0
Bit field:	TX64OCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TX64OCTGB[31:0]	Tx 64Octets Packets Good_Bad This field indicates the number of good and bad packets transmitted with length 64 bytes, exclusive of preamble and retried packets.	R

This register provides the number of good and bad packets transmitted by GMAC with length 64 bytes, exclusive of preamble and retried packets.

28.3.52 Tx_65To127Octets_Packets_Good_Bad : Transmit Octet Count for Good and Bad 65 to 127 Bytes Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0728

Bit position:	31																												0
Bit field:	TX65_127OCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TX65_127OCTGB[31:0]	Tx 65To127Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried packets.	R

This register provides the number of good and bad packets transmitted by GMAC with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried packets.

28.3.53 Tx_128To255Octets_Packets_Good_Bad : Transmit Octet Count for Good and Bad 128 to 255 Bytes Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x072C

Bit position:	31																												0
Bit field:	TX128_255OCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TX128_255OCTGB[31:0]	Tx 128To255Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 128 and 255 (inclusive) bytes, exclusive of preamble and retried packets.	R

This register provides the number of good and bad packets transmitted by GMAC with length between 128 to 255 (inclusive) bytes, exclusive of preamble and retried packets.

28.3.54 Tx_256To511Octets_Packets_Good_Bad : Transmit Octet Count for Good and Bad 256 to 511 Bytes Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0730

Bit position:	31																												0
Bit field:	TX256_511OCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TX256_511OCTGB[31:0]	Tx 256To511Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 256 and 511 (inclusive) bytes, exclusive of preamble and retried packets.	R

This register provides the number of good and bad packets transmitted by GMAC with length between 256 to 511 (inclusive) bytes, exclusive of preamble and retried packets.

28.3.55 Tx_512To1023Octets_Packets_Good_Bad : Transmit Octet Count for Good and Bad 512 to 1023 Bytes Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0734

Bit position:	31																												0
Bit field:	TX512_1023OCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

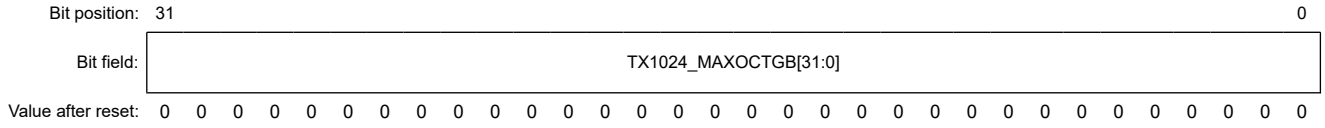
Bit	Symbol	Function	R/W
31:0	TX512_1023OCTGB[31:0]	Tx 512To1023Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 512 and 1023 (inclusive) bytes, exclusive of preamble and retried packets.	R

This register provides the number of good and bad packets transmitted by GMAC with length 512 to 1023 (inclusive) bytes, exclusive of preamble and retried packets.

28.3.56 Tx_1024ToMaxOctets_Packets_Good_Bad : Transmit Octet Count for Good and Bad 1024 to Maxsize Bytes Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0738



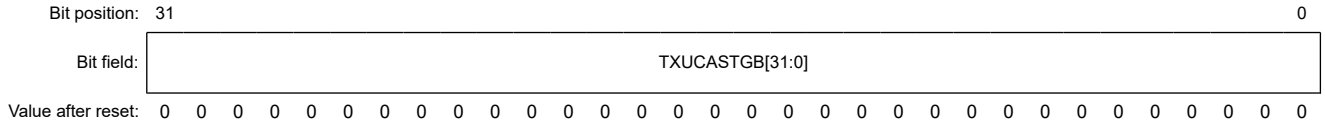
Bit	Symbol	Function	R/W
31:0	TX1024_MAXOCTGB[31:0]	Tx 1024ToMaxOctets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 1024 and maxsize (inclusive) bytes, exclusive of preamble and retried packets.	R

This register provides the number of good and bad packets transmitted by GMAC with length 1024 to maxsize (inclusive) bytes, exclusive of preamble and retried packets.

28.3.57 Tx_Unicast_Packets_Good_Bad : Transmit Packet Count for Good and Bad Unicast Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x073C



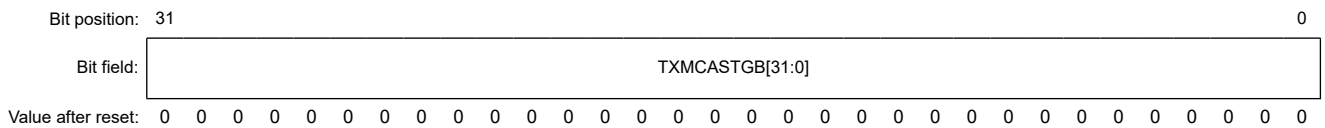
Bit	Symbol	Function	R/W
31:0	TXUCASTGB[31:0]	Tx Unicast Packets Good Bad This field indicates the number of good and bad unicast packets transmitted.	R

This register provides the number of good and bad unicast packets transmitted by GMAC.

28.3.58 Tx_Multicast_Packets_Good_Bad : Transmit Packet Count for Good and Bad Multicast Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0740



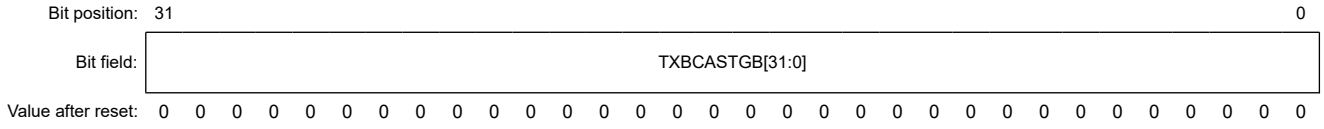
Bit	Symbol	Function	R/W
31:0	TXMCASTGB[31:0]	Tx Multicast Packets Good Bad This field indicates the number of good and bad multicast packets transmitted.	R

This register provides the number of good and bad multicast packets transmitted by GMAC.

28.3.59 Tx_Broadcast_Packets_Good_Bad : Transmit Packet Count for Good and Bad Broadcast Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0744



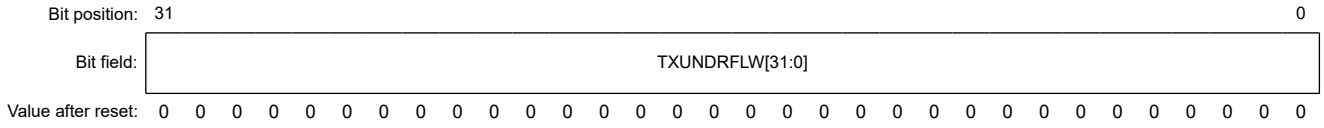
Bit	Symbol	Function	R/W
31:0	TXBCASTGB[31:0]	Tx Broadcast Packets Good Bad This field indicates the number of good and bad broadcast packets transmitted.	R

This register provides the number of good and bad broadcast packets transmitted by GMAC.

28.3.60 Tx_Underflow_Error_Packets : Transmit Packet Count for Underflow Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0748



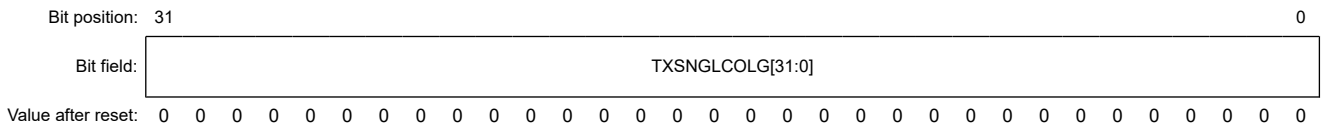
Bit	Symbol	Function	R/W
31:0	TXUNDRFLW[31:0]	Tx Underflow Error Packets This field indicates the number of packets aborted because of packets underflow error.	R

This register provides the number of packets aborted by GMAC because of packets underflow error.

28.3.61 Tx_Single_Collision_Good_Packets : Transmit Packet Count for Packets Transmitted after Single Collision

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x074C



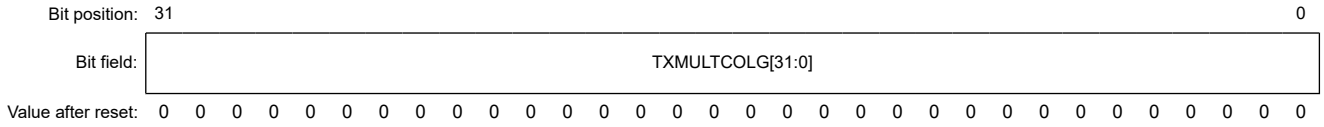
Bit	Symbol	Function	R/W
31:0	TXSNGLCOLG[31:0]	Tx Single Collision Good Packets This field indicates the number of successfully transmitted packets after a single collision in the half-duplex mode.	R

This register provides the number of successfully transmitted packets by GMAC after a single collision in the half-duplex mode.

28.3.62 Tx_Multiple_Collision_Good_Packets : Transmit Packet Count for Packets Transmitted after Multiple Collision

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0750



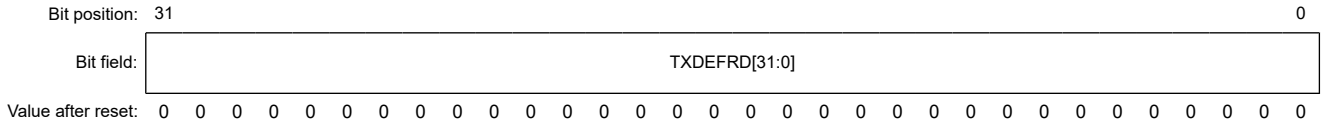
Bit	Symbol	Function	R/W
31:0	TXMULTCOLG[31:0]	Tx Multiple Collision Good Packets This field indicates the number of successfully transmitted packets after multiple collisions in the half-duplex mode.	R

This register provides the number of successfully transmitted packets by GMAC after multiple collisions in the half-duplex mode.

28.3.63 Tx_Deferred_Packets : Transmit Packet Count for Deferred Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0754



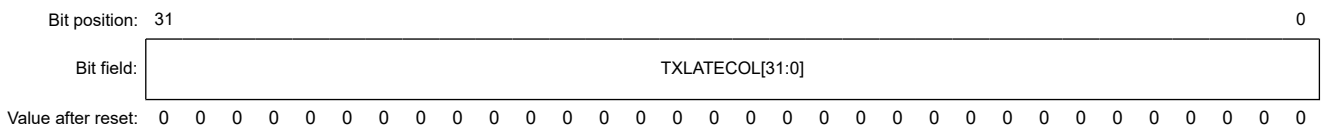
Bit	Symbol	Function	R/W
31:0	TXDEFRD[31:0]	Tx Deferred Packets This field indicates the number of successfully transmitted after a deferral in the half-duplex mode.	R

This register provides the number of successfully transmitted by GMAC after a deferral in the half-duplex mode.

28.3.64 Tx_Late_Collision_Packets : Transmit Packet Count for Late Collision Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0758



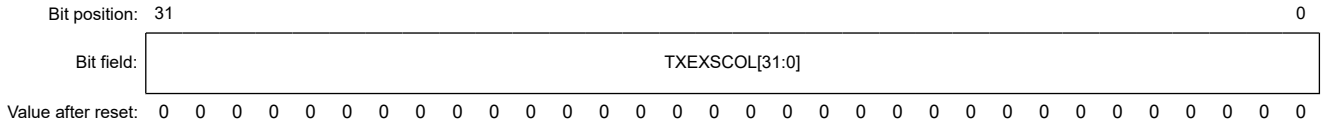
Bit	Symbol	Function	R/W
31:0	TXLATECOL[31:0]	Tx Late Collision Packets This field indicates the number of packets aborted because of late collision error.	R

This register provides the number of packets aborted by GMAC because of late collision error.

28.3.65 Tx_Excessive_Collision_Packets : Transmit Packet Count for Excessive Collision Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x075C



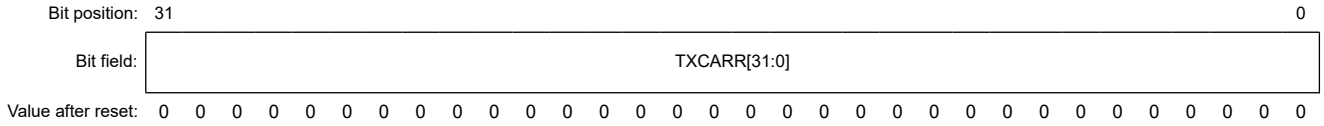
Bit	Symbol	Function	R/W
31:0	TXEXSCOL[31:0]	Tx Excessive Collision Packets This field indicates the number of packets aborted because of excessive (16) collision errors.	R

This register provides the number of packets aborted by GMAC because of excessive (16) collision errors.

28.3.66 Tx_Carrier_Error_Packets : Transmit Packet Count for Carrier Sense Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0760



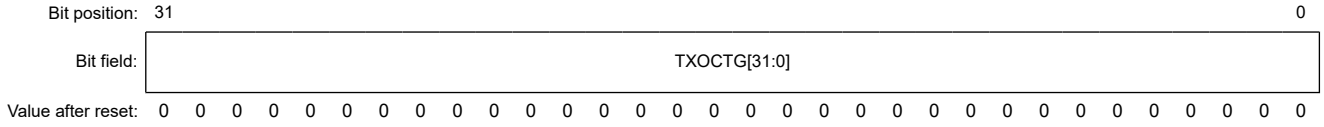
Bit	Symbol	Function	R/W
31:0	TXCARR[31:0]	Tx Carrier Error Packets This field indicates the number of packets aborted because of carrier sense error (no carrier or loss of carrier).	R

This register provides the number of packets aborted by GMAC because of carrier sense error (no carrier or loss of carrier).

28.3.67 Tx_Octet_Count_Good : Transmit Octet Count for Good Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0764



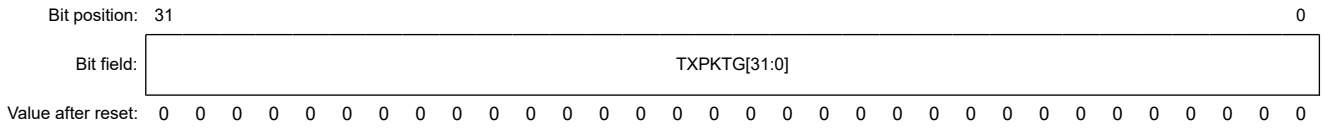
Bit	Symbol	Function	R/W
31:0	TXOCTG[31:0]	Tx Octet Count Good This field indicates the number of bytes transmitted, exclusive of preamble, only in good packets.	R

This register provides the number of bytes transmitted by GMAC, exclusive of preamble, only in good packets.

28.3.68 Tx_Packet_Count_Good : Transmit Packet Count for Good Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0768



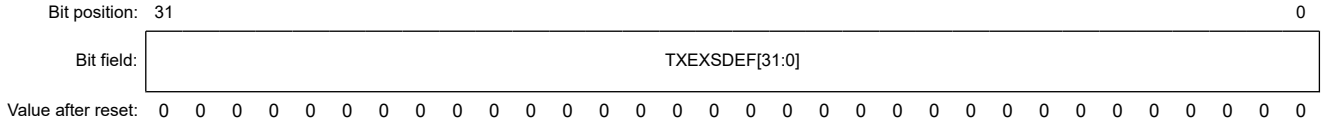
Bit	Symbol	Function	R/W
31:0	TXPKTG[31:0]	Tx Packet Count Good This field indicates the number of good packets transmitted.	R

This register provides the number of good packets transmitted by GMAC.

28.3.69 Tx_Excessive_Deferral_Error : Transmit Packet Count for Excessive Deferral Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x076C



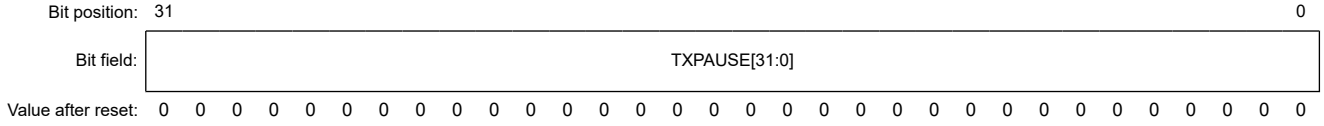
Bit	Symbol	Function	R/W
31:0	TXEXSDEF[31:0]	Tx Excessive Deferral Error This field indicates the number of packets aborted because of excessive deferral error (deferred for more than two max-sized packet times).	R

This register provides the number of packets aborted by GMAC because of excessive deferral error (deferred for more than two max-sized packet times).

28.3.70 Tx_Pause_Packets : Transmit Packet Count for Good PAUSE Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0770



Bit	Symbol	Function	R/W
31:0	TXPAUSE[31:0]	Tx Pause Packets This field indicates the number of good Pause packets transmitted.	R

This register provides the number of good Pause packets transmitted by GMAC.

28.3.71 Tx_VLAN_Packets_Good : Transmit Packet Count for Good VLAN Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0774

Bit position:	31																												0
Bit field:	TXVLANG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXVLANG[31:0]	Tx VLAN Packets Good This field provides the number of good VLAN packets transmitted.	R

This register provides the number of good VLAN packets transmitted by GMAC.

28.3.72 Tx_OSize_Packets_Good : Transmit Packet Count for Good Oversize Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0778

Bit position:	31																												0
Bit field:	TXOSIZG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXOSIZG[31:0]	Tx OSize Packets Good This field indicates the number of packets transmitted without errors and with length greater than the maxsize (1,518 or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in S2KP bit of the MAC_Configuration register).	R

This register provides the number of packets transmitted by GMAC without errors and with length greater than the maxsize (1,518 or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in S2KP bit of the MAC_Configuration register).

28.3.73 Rx_Packets_Count_Good_Bad : Receive Packet Count for Good and Bad Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0780

Bit position:	31																												0
Bit field:	RXPKTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXPKTGB[31:0]	Rx Packets Count Good Bad This field indicates the number of good and bad packets received.	R

This register provides the number of good and bad packets received by GMAC.

28.3.74 Rx_Octet_Count_Good_Bad : Receive Octet Count for Good and Bad Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0784

Bit position:	31																												0
Bit field:	RXOCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXOCTGB[31:0]	Rx Octet Count Good Bad This field indicates the number of bytes received, exclusive of preamble, in good and bad packets.	R

This register provides the number of bytes received by GMAC, exclusive of preamble, in good and bad packets.

28.3.75 Rx_Octet_Count_Good : Receive Octet Count for Good Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0788

Bit position:	31																												0
Bit field:	RXOCTG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXOCTG[31:0]	Rx Octet Count Good This field indicates the number of bytes received, exclusive of preamble, only in good packets.	R

This register provides the number of bytes received by GMAC, exclusive of preamble, only in good packets.

28.3.76 Rx_Broadcast_Packets_Good : Receive Packet Count for Good Broadcast Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x078C

Bit position:	31																												0
Bit field:	RXBCASTG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXBCASTG[31:0]	Rx Broadcast Packets Good This field indicates the number of good broadcast packets received.	R

This register provides the number of good broadcast packets received by GMAC.

28.3.77 Rx_Multicast_Packets_Good : Receive Packet Count for Good Multicast Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0790

Bit position:	31																												0
Bit field:	RXMCASTG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXMCASTG[31:0]	Rx Multicast Packets Good This field indicates the number of good multicast packets received.	R

This register provides the number of good multicast packets received by GMAC.

28.3.78 Rx_CRC_Error_Packets : Receive Packet Count for CRC Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0794

Bit position:	31																												0
Bit field:	RXCRCERR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXCRCERR[31:0]	Rx CRC Error Packets This field indicates the number of packets received with CRC error.	R

This register provides the number of packets received by GMAC with CRC error.

28.3.79 Rx_Alignment_Error_Packets : Receive Packet Count for Alignment Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0798

Bit position:	31																												0
Bit field:	RXALGNERR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXALGNERR[31:0]	Rx Alignment Error Packets This field indicates the number of packets received with alignment (dribble) error. It is valid only in 10/100 mode.	R

This register provides the number of packets received by GMAC with alignment (dribble) error. It is valid only in 10/100 mode.

28.3.80 Rx_Runt_Error_Packets : Receive Packet Count for Runt Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x079C

Bit position:	31																												0
Bit field:	RXRUNTERR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXRUNTERR[31:0]	Rx Runt Error Packets This field indicates the number of packets received with runt (length less than 64 bytes and CRC error) error.	R

This register provides the number of packets received by GMAC with runt (length less than 64 bytes and CRC error) error.

28.3.81 Rx_Jabber_Error_Packets : Receive Packet Count for Jabber Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07A0

Bit position:	31																												0
Bit field:	RXJABERR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXJABERR[31:0]	Rx Jabber Error Packets This field indicates the number of giant packets received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Packet mode is enabled, packets of length greater than 9,018 bytes (9,022 bytes for VLAN tagged) are considered as giant packets.	R

This register provides the number of giant packets received by GMAC with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Packet mode is enabled, packets of length greater than 9,018 bytes (9,022 bytes for VLAN tagged) are considered as giant packets.

28.3.82 Rx_Undersize_Packets_Good : Receive Packet Count for Undersize Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07A4

Bit position:	31																												0
Bit field:	RXUNDERSZG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXUNDERSZG[31:0]	Rx Undersize Packets Good This field indicates the number of packets received with length less than 64 bytes, without any errors.	R

This register provides the number of packets received by GMAC with length less than 64 bytes, without any errors.

28.3.83 Rx_Oversize_Packets_Good : Receive Packet Count for Oversize Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07A8

Bit position:	31																												0
Bit field:	RXOVERSZG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXOVERSZG[31:0]	Rx Oversize Packets Good This field indicates the number of packets received without errors, with length greater than the maxsize (1,518 bytes or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in the S2KP bit of the MAC_Configuration register).	R

This register provides the number of packets received by GMAC without errors, with length greater than the maxsize (1,518 bytes or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in the S2KP bit of the MAC_Configuration register).

28.3.84 Rx_64Octets_Packets_Good_Bad : Receive Packet Count for Good and Bad 64 Byte Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07AC

Bit position:	31																												0
Bit field:	RX64OCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RX64OCTGB[31:0]	Rx 64 Octets Packets Good Bad This field indicates the number of good and bad packets received with length 64 bytes, exclusive of the preamble.	R

This register provides the number of good and bad packets received by GMAC with length 64 bytes, exclusive of the preamble.

28.3.85 Rx_65To127Octets_Packets_Good_Bad : Receive Packet Count for Good and Bad 65 to 127 Bytes Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07B0

Bit position:	31																												0
Bit field:	RX65_127OCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

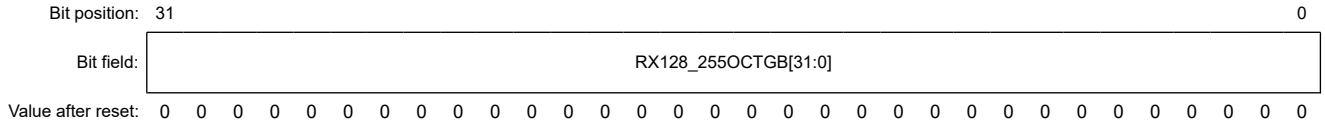
Bit	Symbol	Function	R/W
31:0	RX65_127OCTGB[31:0]	Rx 65-127 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 65 and 127 (inclusive) bytes, exclusive of the preamble.	R

This register provides the number of good and bad packets received by GMAC with length between 65 and 127 (inclusive) bytes, exclusive of the preamble.

28.3.86 Rx_128To255Octets_Packets_Good_Bad : Receive Packet Count for Good and Bad 128 to 255 Bytes Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07B4



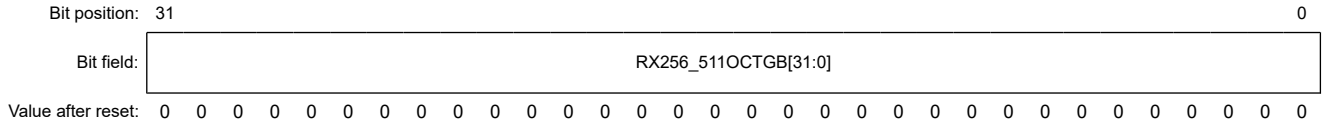
Bit	Symbol	Function	R/W
31:0	RX128_255OCTGB[31:0]	Rx 128-255 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 128 and 255 (inclusive) bytes, exclusive of the preamble.	R

This register provides the number of good and bad packets received by GMAC with length between 128 and 255 (inclusive) bytes, exclusive of the preamble.

28.3.87 Rx_256To511Octets_Packets_Good_Bad : Receive Packet Count for Good and Bad 256 to 511 Bytes Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07B8



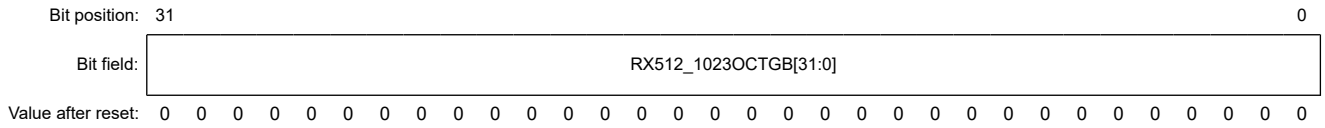
Bit	Symbol	Function	R/W
31:0	RX256_511OCTGB[31:0]	Rx 256-511 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 256 and 511 (inclusive) bytes, exclusive of the preamble.	R

This register provides the number of good and bad packets received by GMAC with length between 256 and 511 (inclusive) bytes, exclusive of the preamble.

28.3.88 Rx_512To1023Octets_Packets_Good_Bad : Receive Packet Count for Good and Bad 512 to 1,023 Bytes Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07BC



Bit	Symbol	Function	R/W
31:0	RX512_1023OCTGB[31:0]	RX 512-1023 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 512 and 1023 (inclusive) bytes, exclusive of the preamble.	R

This register provides the number of good and bad packets received by GMAC with length between 512 and 1023 (inclusive) bytes, exclusive of the preamble.

28.3.89 Rx_1024ToMaxOctets_Packets_Good_Bad : Receive Packet Count for Good and Bad 1,024 to Maxsize Bytes Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07C0

Bit position:	31																												0
Bit field:	RX1024_MAXOCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RX1024_MAXOCTGB[31:0]	Rx 1024-Max Octets Good Bad This field indicates the number of good and bad packets received with length between 1024 and maxsize (inclusive) bytes, exclusive of the preamble.	R

This register provides the number of good and bad packets received by GMAC with length between 1024 and maxsize (inclusive) bytes, exclusive of the preamble.

28.3.90 Rx_Unicast_Packets_Good : Receive Packet Count for Good Unicast Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07C4

Bit position:	31																												0
Bit field:	RXUCASTG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXUCASTG[31:0]	Rx Unicast Packets Good This field indicates the number of good unicast packets received.	R

This register provides the number of good unicast packets received by GMAC.

28.3.91 Rx_Length_Error_Packets : Receive Packet Count for Length Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07C8

Bit position:	31																												0
Bit field:	RXLENERR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

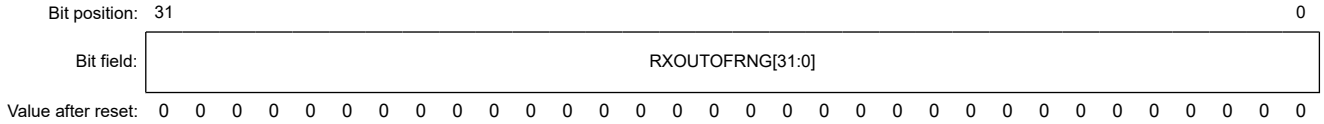
Bit	Symbol	Function	R/W
31:0	RXLENERR[31:0]	Rx Length Error Packets This field indicates the number of packets received with length error (Length Type field not equal to packet size), for all packets with valid length field.	R

This register provides the number of packets received by GMAC with length error (Length Type field not equal to packet size), for all packets with valid length field.

28.3.92 Rx_Out_Of_Range_Type_Packets : Receive Packet Count for Out of Range Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07CC



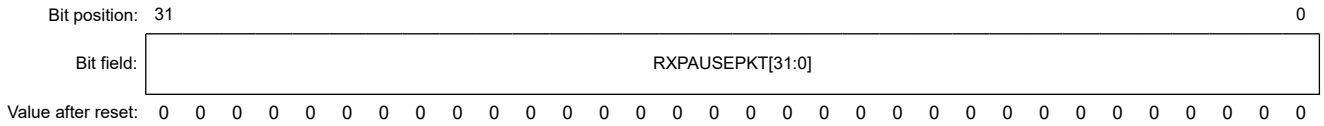
Bit	Symbol	Function	R/W
31:0	RXOUTOFRNG[31:0]	Rx Out of Range Type Packet This field indicates the number of packets received with length field not equal to the valid packet size (greater than 1,500 but less than 1,536).	R

This register provides the number of packets received by GMAC with length field not equal to the valid packet size (greater than 1,500 but less than 1,536).

28.3.93 Rx_Pause_Packets : Receive Packet Count for PAUSE Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07D0



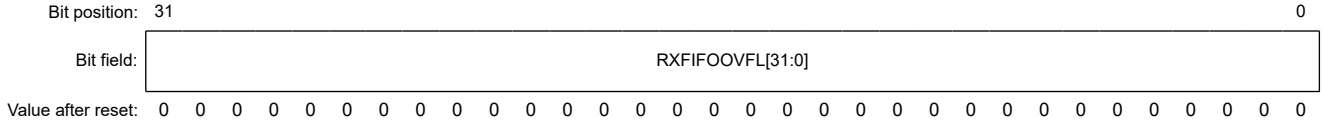
Bit	Symbol	Function	R/W
31:0	RXPAUSEPKT[31:0]	Rx Pause Packets This field indicates the number of good and valid Pause packets received.	R

This register provides the number of good and valid Pause packets received by GMAC.

28.3.94 Rx_FIFO_Overflow_Packets : Receive Packet Count for FIFO Overflow Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07D4



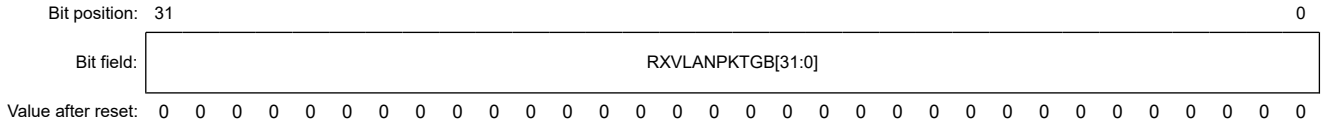
Bit	Symbol	Function	R/W
31:0	RXFIFOOVFL[31:0]	Rx FIFO Overflow Packets This field indicates the number of missed received packets because of FIFO overflow.	R

This register provides the number of missed received packets because of FIFO overflow in GMAC.

28.3.95 Rx_VLAN_Packets_Good_Bad : Receive Packet Count for Good and Bad VLAN Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07D8



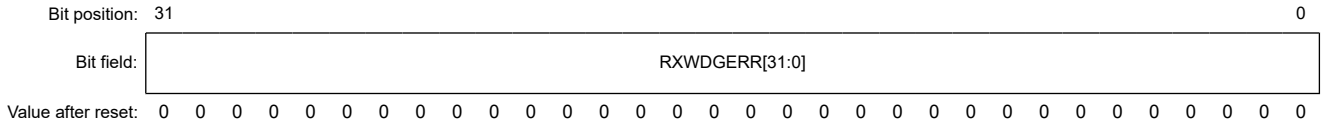
Bit	Symbol	Function	R/W
31:0	RXVLANPKTGB[31:0]	Rx VLAN Packets Good Bad This field indicates the number of good and bad VLAN packets received.	R

This register provides the number of good and bad VLAN packets received by GMAC.

28.3.96 Rx_Watchdog_Error_Packets : Receive Packet Count for Watchdog Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07DC



Bit	Symbol	Function	R/W
31:0	RXWDGERR[31:0]	Rx Watchdog Error Packets This field indicates the number of packets received with error because of watchdog timeout error (packets with a data load larger than 2,048 bytes (when JE and WD bits are reset in MAC_Configuration register), 10,240 bytes (when JE bit is set and WD bit is reset in MAC_Configuration register), 16,384 bytes (when WD bit is set in MAC_Configuration register) or the value programmed in the MAC_Watchdog_Timeout register).	R

This register provides the number of packets received by GMAC with error because of watchdog timeout error (packets with a data load larger than 2,048 bytes (when JE and WD bits are reset in MAC_Configuration register), 10,240 bytes (when JE bit is set and WD bit is reset in MAC_Configuration register), 16,384 bytes (when WD bit is set in MAC_Configuration register) or the value programmed in the MAC_Watchdog_Timeout register).

28.3.97 Rx_Receive_Error_Packets : Receive Packet Count for Receive Error Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07E0

Bit position:	31																												0
Bit field:	RXRCVERR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXRCVERR[31:0]	Rx Receive Error Packets This field indicates the number of packets received with Receive error or Packet Extension error on the GMII or MII interface.	R

This register provides the number of packets received by GMAC with Receive error or Packet Extension error on the GMII or MII interface.

28.3.98 Rx_Control_Packets_Good : Receive Packet Count for Good Control Packets

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07E4

Bit position:	31																												0
Bit field:	RXCTRLG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXCTRLG[31:0]	Rx Control Packets Good This field indicates the number of good control packets received.	R

This register provides the number of good control packets received by GMAC.

28.3.99 Tx_LPI_USEC_Cntr : Tx LPI Microseconds Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07EC

Bit position:	31																												0
Bit field:	TXLPIUSC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXLPIUSC[31:0]	Tx LPI Microseconds Counter This field indicates the number of microseconds Tx LPI is asserted. For every Tx LPI Entry and Exit, the Timer value can have an error of ±1 microsecond.	R

This register provides the number of microseconds Tx LPI is asserted by GMAC.

28.3.100 Tx_LPI_Tran_Cntr : Tx LPI Transition Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07F0

Bit position:	31																												0
Bit field:	TXLPITRC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXLPITRC[31:0]	Tx LPI Transition counter This field indicates the number of times Tx LPI Entry has occurred. Even if Tx LPI Entry occurs in Automate Mode (because of LPITXA bit set in the LPI Control and Status register), the counter increments.	R

This register provides the number of times GMAC has entered Tx LPI.

28.3.101 Rx_LPI_USEC_Cntr : Rx LPI Microseconds Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07F4

Bit position:	31																												0
Bit field:	RXLPIUSC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXLPIUSC[31:0]	Rx LPI Microseconds Counter This field indicates the number of microseconds Rx LPI is asserted. For every Rx LPI Entry and Exit, the Timer value can have an error of ±1 microsecond.	R

This register provides the number of microseconds Rx LPI is sampled by GMAC.

28.3.102 Rx_LPI_Tran_Cntr : Rx LPI Transition Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x07F8

Bit position:	31																												0
Bit field:	RXLPITRC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXLPITRC[31:0]	Rx LPI Transition counter This field indicates the number of times Rx LPI Entry has occurred.	R

This register provides the number of times GMAC has entered Rx LPI.

28.3.103 MMC_IPC_Rx_Interrupt_Mask : MMC Receive Checksum Off load Interrupt Mask Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0800

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	RXICMPEROIM	RXICMPGOIM	RXTCPEROIM	RXTCPGOIM	RXUDPEROIM	RXUDPGOIM	RXIPV6NOPAYOIM	RXIPV6HEROIM	RXIPV6GOIM	RXIPV4UDSBLOIM	RXIPV4FRAGOIM	RXIPV4NOPAYOIM	RXIPV4HEROIM	RXIPV4GOIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RXICMPERPIM	RXICMPGPIM	RXTCPERPIM	RXTCPGPIM	RXUDPERPIM	RXUDPGPIM	RXIPV6NOPAYPIM	RXIPV6HERPIM	RXIPV6GPIM	RXIPV4UDSBLPIM	RXIPV4FRAGPIM	RXIPV4NOPAYPIM	RXIPV4HERPIM	RXIPV4GPIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RXIPV4GPIM	MMC Receive IPV4 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Good Packet Counter Interrupt Mask is disabled 1: MMC Receive IPV4 Good Packet Counter Interrupt Mask is enabled	R/W
1	RXIPV4HERPIM	MMC Receive IPV4 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is disabled 1: MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is enabled	R/W
2	RXIPV4NOPAYPIM	MMC Receive IPV4 No Payload Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_nopay_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 No Payload Packet Counter Interrupt Mask is disabled 1: MMC Receive IPV4 No Payload Packet Counter Interrupt Mask is enabled	R/W
3	RXIPV4FRAGPIM	MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_frag_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask is disabled 1: MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask is enabled	R/W
4	RXIPV4UDSBLPIM	MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_udsbl_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask is disabled 1: MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask is enabled	R/W
5	RXIPV6GPIM	MMC Receive IPV6 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 Good Packet Counter Interrupt Mask is disabled 1: MMC Receive IPV6 Good Packet Counter Interrupt Mask is enabled	R/W
6	RXIPV6HERPIM	MMC Receive IPV6 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is disabled 1: MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is enabled	R/W

Bit	Symbol	Function	R/W
7	RXIPV6NOPAYPIM	MMC Receive IPV6 No Payload Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_nopay_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 No Payload Packet Counter Interrupt Mask is disabled 1: MMC Receive IPV6 No Payload Packet Counter Interrupt Mask is enabled	R/W
8	RXUDPGPIM	MMC Receive UDP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_gd_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive UDP Good Packet Counter Interrupt Mask is disabled 1: MMC Receive UDP Good Packet Counter Interrupt Mask is enabled	R/W
9	RXUDPERPIM	MMC Receive UDP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive UDP Error Packet Counter Interrupt Mask is disabled 1: MMC Receive UDP Error Packet Counter Interrupt Mask is enabled	R/W
10	RXTCPGPIM	MMC Receive TCP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_gd_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive TCP Good Packet Counter Interrupt Mask is disabled 1: MMC Receive TCP Good Packet Counter Interrupt Mask is enabled	R/W
11	RXTCPERPIM	MMC Receive TCP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive TCP Error Packet Counter Interrupt Mask is disabled 1: MMC Receive TCP Error Packet Counter Interrupt Mask is enabled	R/W
12	RXICMPGPIM	MMC Receive ICMP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_gd_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive ICMP Good Packet Counter Interrupt Mask is disabled 1: MMC Receive ICMP Good Packet Counter Interrupt Mask is enabled	R/W
13	RXICMPERPIM	MMC Receive ICMP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive ICMP Error Packet Counter Interrupt Mask is disabled 1: MMC Receive ICMP Error Packet Counter Interrupt Mask is enabled	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	RXIPV4GOIM	MMC Receive IPV4 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_gd_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Good Octet Counter Interrupt Mask is disabled 1: MMC Receive IPV4 Good Octet Counter Interrupt Mask is enabled	R/W
17	RXIPV4HEROIM	MMC Receive IPV4 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is disabled 1: MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is enabled	R/W
18	RXIPV4NOPAYOIM	MMC Receive IPV4 No Payload Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_nopay_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 No Payload Octet Counter Interrupt Mask is disabled 1: MMC Receive IPV4 No Payload Octet Counter Interrupt Mask is enabled	R/W
19	RXIPV4FRAGOIM	MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_frag_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask is disabled 1: MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask is enabled	R/W

Bit	Symbol	Function	R/W
20	RXIPV4UDSBLOIM	MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_udsbl_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask is disabled 1: MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask is enabled	R/W
21	RXIPV6GOIM	MMC Receive IPV6 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_gd_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 Good Octet Counter Interrupt Mask is disabled 1: MMC Receive IPV6 Good Octet Counter Interrupt Mask is enabled	R/W
22	RXIPV6HEROIM	MMC Receive IPV6 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 Good Octet Counter Interrupt Mask is disabled 1: MMC Receive IPV6 Good Octet Counter Interrupt Mask is enabled	R/W
23	RXIPV6NOPAYOIM	MMC Receive IPV6 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is disabled 1: MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is enabled	R/W
24	RXUDPGOIM	MMC Receive IPV6 No Payload Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_gd_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 No Payload Octet Counter Interrupt Mask is disabled 1: MMC Receive IPV6 No Payload Octet Counter Interrupt Mask is enabled	R/W
25	RXUDPEROIM	MMC Receive UDP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive UDP Good Octet Counter Interrupt Mask is disabled 1: MMC Receive UDP Good Octet Counter Interrupt Mask is enabled	R/W
26	RXTCPGOIM	MMC Receive TCP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_gd_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive TCP Good Octet Counter Interrupt Mask is disabled 1: MMC Receive TCP Good Octet Counter Interrupt Mask is enabled	R/W
27	RXTCPEROIM	MMC Receive TCP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive TCP Error Octet Counter Interrupt Mask is disabled 1: MMC Receive TCP Error Octet Counter Interrupt Mask is enabled	R/W
28	RXICMPGOIM	MMC Receive ICMP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_gd_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive ICMP Good Octet Counter Interrupt Mask is disabled 1: MMC Receive ICMP Good Octet Counter Interrupt Mask is enabled	R/W
29	RXICMPEROIM	MMC Receive ICMP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive ICMP Error Octet Counter Interrupt Mask is disabled 1: MMC Receive ICMP Error Octet Counter Interrupt Mask is enabled	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

This register maintains the mask for the interrupt generated from the receive IPC statistic counters.

The MMC_IPC_Rx_Interrupt_Mask register maintains the masks for the interrupts generated when the receive IPC (Checksum Off load) statistic counters reach half their maximum value, and when they reach their maximum values. This register is 32 bits wide.

28.3.104 MMC_IPC_Rx_Interrupt : MMC Receive Checksum Off load Interrupt Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0808

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	RXICMPERIS	RXICMPGOIS	RXTCPEROIS	RXTCPGOIS	RXUDPEROIS	RXUDPGOIS	RXIPV6NOPAYOIS	RXIPV6HERIS	RXIPV6GOIS	RXIPV4UDSBLOIS	RXIPV4FRAGOIS	RXIPV4NOPAYOIS	RXIPV4HERIS	RXIPV4GOIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RXICMPERPIS	RXICMPGPIS	RXTCPERPIS	RXTCPGPIS	RXUDPERPIS	RXUDPGPIS	RXIPV6NOPAYPIS	RXIPV6HERPIS	RXIPV6GPIS	RXIPV4UDSBLPIS	RXIPV4FRAGPIS	RXIPV4NOPAYPIS	RXIPV4HERPIS	RXIPV4GPIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RXIPV4GPIS	MMC Receive IPV4 Good Packet Counter Interrupt Status*1 This bit is set when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Good Packet Counter Interrupt Status not detected 1: MMC Receive IPV4 Good Packet Counter Interrupt Status detected	R
1	RXIPV4HERPIS	MMC Receive IPV4 Header Error Packet Counter Interrupt Status*1 This bit is set when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Header Error Packet Counter Interrupt Status not detected 1: MMC Receive IPV4 Header Error Packet Counter Interrupt Status detected	R
2	RXIPV4NOPAYPIS	MMC Receive IPV4 No Payload Packet Counter Interrupt Status*1 This bit is set when the rxipv4_nopay_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 No Payload Packet Counter Interrupt Status not detected 1: MMC Receive IPV4 No Payload Packet Counter Interrupt Status detected	R
3	RXIPV4FRAGPIS	MMC Receive IPV4 Fragmented Packet Counter Interrupt Status*1 This bit is set when the rxipv4_frag_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Fragmented Packet Counter Interrupt Status not detected 1: MMC Receive IPV4 Fragmented Packet Counter Interrupt Status detected	R
4	RXIPV4UDSBLPIS	MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status*1 This bit is set when the rxipv4_udsbl_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status not detected 1: MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status detected	R
5	RXIPV6GPIS	MMC Receive IPV6 Good Packet Counter Interrupt Status*1 This bit is set when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 Good Packet Counter Interrupt Status not detected 1: MMC Receive IPV6 Good Packet Counter Interrupt Status detected	R
6	RXIPV6HERPIS	MMC Receive IPV6 Header Error Packet Counter Interrupt Status*1 This bit is set when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 Header Error Packet Counter Interrupt Status not detected 1: MMC Receive IPV6 Header Error Packet Counter Interrupt Status detected	R
7	RXIPV6NOPAYPIS	MMC Receive IPV6 No Payload Packet Counter Interrupt Status*1 This bit is set when the rxipv6_nopay_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 No Payload Packet Counter Interrupt Status not detected 1: MMC Receive IPV6 No Payload Packet Counter Interrupt Status detected	R

Bit	Symbol	Function	R/W
8	RXUDPGPIS	MMC Receive UDP Good Packet Counter Interrupt Status*1 This bit is set when the rxudp_gd_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive UDP Good Packet Counter Interrupt Status not detected 1: MMC Receive UDP Good Packet Counter Interrupt Status detected	R
9	RXUDPERPIS	MMC Receive UDP Error Packet Counter Interrupt Status*1 This bit is set when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive UDP Error Packet Counter Interrupt Status not detected 1: MMC Receive UDP Error Packet Counter Interrupt Status detected	R
10	RXTCPGPIS	MMC Receive TCP Good Packet Counter Interrupt Status*1 This bit is set when the rxtcp_gd_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive TCP Good Packet Counter Interrupt Status not detected 1: MMC Receive TCP Good Packet Counter Interrupt Status detected	R
11	RXTCPERPIS	MMC Receive TCP Error Packet Counter Interrupt Status*1 This bit is set when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive TCP Error Packet Counter Interrupt Status not detected 1: MMC Receive TCP Error Packet Counter Interrupt Status detected	R
12	RXICMPGPIS	MMC Receive ICMP Good Packet Counter Interrupt Status*1 This bit is set when the rxicmp_gd_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive ICMP Good Packet Counter Interrupt Status not detected 1: MMC Receive ICMP Good Packet Counter Interrupt Status detected	R
13	RXICMPERPIS	MMC Receive ICMP Error Packet Counter Interrupt Status*1 This bit is set when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. 0: MMC Receive ICMP Error Packet Counter Interrupt Status not detected 1: MMC Receive ICMP Error Packet Counter Interrupt Status detected	R
15:14	—	These bits are read as 0.	R
16	RXIPV4GOIS	MMC Receive IPV4 Good Octet Counter Interrupt Status*1 This bit is set when the rxipv4_gd_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Good Octet Counter Interrupt Status not detected 1: MMC Receive IPV4 Good Octet Counter Interrupt Status detected	R
17	RXIPV4HEROIS	MMC Receive IPV4 Header Error Octet Counter Interrupt Status*1 This bit is set when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Header Error Octet Counter Interrupt Status not detected 1: MMC Receive IPV4 Header Error Octet Counter Interrupt Status detected	R
18	RXIPV4NOPAYOIS	MMC Receive IPV4 No Payload Octet Counter Interrupt Status*1 This bit is set when the rxipv4_nopay_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 No Payload Octet Counter Interrupt Status not detected 1: MMC Receive IPV4 No Payload Octet Counter Interrupt Status detected	R
19	RXIPV4FRAGOIS	MMC Receive IPV4 Fragmented Octet Counter Interrupt Status*1 This bit is set when the rxipv4_frag_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 Fragmented Octet Counter Interrupt Status not detected 1: MMC Receive IPV4 Fragmented Octet Counter Interrupt Status detected	R
20	RXIPV4UDSBLOIS	MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status*1 This bit is set when the rxipv4_udsbl_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status not detected 1: MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status detected	R

Bit	Symbol	Function	R/W
21	RXIPV6GOIS	MMC Receive IPV6 Good Octet Counter Interrupt Status*1 This bit is set when the rxipv6_gd_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 Good Octet Counter Interrupt Status not detected 1: MMC Receive IPV6 Good Octet Counter Interrupt Status detected	R
22	RXIPV6HEROIS	MMC Receive IPV6 Header Error Octet Counter Interrupt Status*1 This bit is set when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 Header Error Octet Counter Interrupt Status not detected 1: MMC Receive IPV6 Header Error Octet Counter Interrupt Status detected	R
23	RXIPV6NOPAYOIS	MMC Receive IPV6 No Payload Octet Counter Interrupt Status*1 This bit is set when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive IPV6 No Payload Octet Counter Interrupt Status not detected 1: MMC Receive IPV6 No Payload Octet Counter Interrupt Status detected	R
24	RXUDPGOIS	MMC Receive UDP Good Octet Counter Interrupt Status*1 This bit is set when the rxudp_gd_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive UDP Good Octet Counter Interrupt Status not detected 1: MMC Receive UDP Good Octet Counter Interrupt Status detected	R
25	RXUDPEROIS	MMC Receive UDP Error Octet Counter Interrupt Status*1 This bit is set when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive UDP Error Octet Counter Interrupt Status not detected 1: MMC Receive UDP Error Octet Counter Interrupt Status detected	R
26	RXTCPGOIS	MMC Receive TCP Good Octet Counter Interrupt Status*1 This bit is set when the rxtcp_gd_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive TCP Good Octet Counter Interrupt Status not detected 1: MMC Receive TCP Good Octet Counter Interrupt Status detected	R
27	RXTCPEROIS	MMC Receive TCP Error Octet Counter Interrupt Status*1 This bit is set when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive TCP Error Octet Counter Interrupt Status not detected 1: MMC Receive TCP Error Octet Counter Interrupt Status detected	R
28	RXICMPGOIS	MMC Receive ICMP Good Octet Counter Interrupt Status*1 This bit is set when the rxicmp_gd_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive ICMP Good Octet Counter Interrupt Status not detected 1: MMC Receive ICMP Good Octet Counter Interrupt Status detected	R
29	RXICMPEROIS	MMC Receive ICMP Error Octet Counter Interrupt Status*1 This bit is set when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. 0: MMC Receive ICMP Error Octet Counter Interrupt Status not detected 1: MMC Receive ICMP Error Octet Counter Interrupt Status detected	R
31:30	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

This register maintains the interrupt that the receive IPC statistic counters generate.

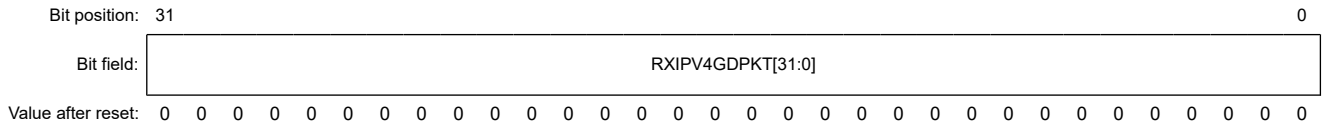
The MMC_IPC_Rx_Interrupt register maintains the interrupts generated when receive IPC statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32 bit counter and 0xFFFF for 16 bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones.

The MMC_IPC_Rx_Interrupt register is 32 bit wide. When the MMC IPC counter that caused the interrupt is read, its corresponding interrupt bit is cleared. The counter's least-significant byte lane (Bits [7:0]) must be read to clear the interrupt bit.

28.3.105 RxIPv4_Good_Packets : RxIPv4 Good Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0810



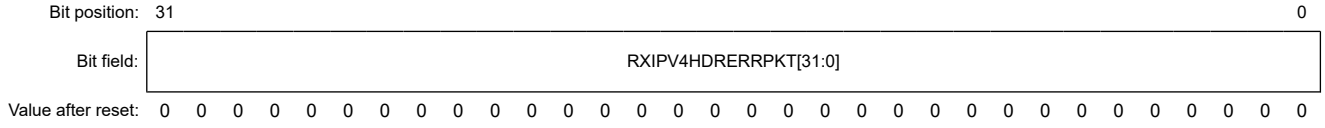
Bit	Symbol	Function	R/W
31:0	RXIPV4GDPKT[31:0]	RxIPv4 Good Packets This field indicates the number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload.	R

This register provides the number of good IPv4 datagrams received by GMAC with the TCP, UDP, or ICMP payload.

28.3.106 RxIPv4_Header_Error_Packets : RxIPv4 Header Error Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0814



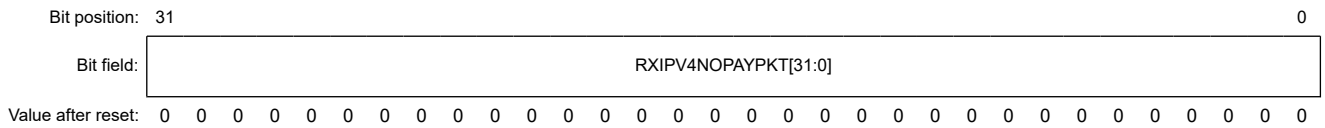
Bit	Symbol	Function	R/W
31:0	RXIPV4HDRERRPKT[31:0]	RxIPv4 Header Error Packets This field indicates the number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors.	R

This register provides the number of IPv4 datagrams received by GMAC with header (checksum, length, or version mismatch) errors.

28.3.107 RxIPv4_No_Payload_Packets : RxIPv4 Payload Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0818



Bit	Symbol	Function	R/W
31:0	RXIPV4NOPAYPKT[31:0]	RxIPv4 Payload Packets This field indicates the number of IPv4 datagram packets received that did not have a TCP, UDP, or ICMP payload.	R

This register provides the number of IPv4 datagram packets received by GMAC that did not have a TCP, UDP, or ICMP payload.

28.3.108 RxIPv4_Fragmented_Packets : RxIPv4 Fragmented Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x081C

Bit position:	31																												0
Bit field:	RXIPV4FRAGPKT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXIPV4FRAGPKT[31:0]	RxIPv4 Fragmented Packets This field indicates the number of good IPv4 datagrams received with fragmentation.	R

This register provides the number of good IPv4 datagrams received by GMAC with fragmentation.

28.3.109 RxIPv4_UDP_Checksum_Disabled_Packets : RxIPv4 UDP Checksum Disabled Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0820

Bit position:	31																												0
Bit field:	RXIPV4UDSBLPKT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXIPV4UDSBLPKT[31:0]	RxIPv4 UDP Checksum Disabled Packets This field indicates the number of good IPv4 datagrams received that had a UDP payload with checksum disabled.	R

This register provides the number of good IPv4 datagrams received by GMAC that had a UDP payload with checksum disabled.

28.3.110 RxIPv6_Good_Packets : RxIPv6 Good Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0824

Bit position:	31																												0
Bit field:	RXIPV6GDPKT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

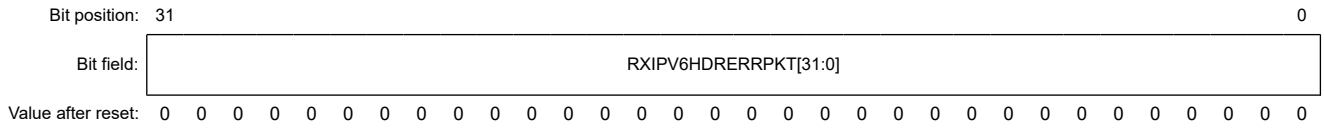
Bit	Symbol	Function	R/W
31:0	RXIPV6GDPKT[31:0]	RxIPv6 Good Packets This field indicates the number of good IPv6 datagrams received with the TCP, UDP, or ICMP payload.	R

This register provides the number of good IPv6 datagrams received by GMAC with the TCP, UDP, or ICMP payload.

28.3.111 RxIPv6_Header_Error_Packets : RxIPv6 Header Error Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0828



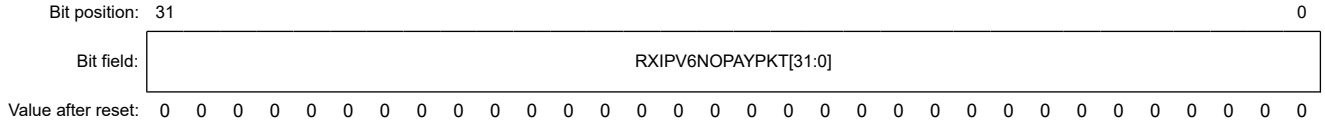
Bit	Symbol	Function	R/W
31:0	RXIPV6HDRERRPKT[31:0]	RxIPv6 Header Error Packets This field indicates the number of IPv6 datagrams received with header (length or version mismatch) errors.	R

This register provides the number of IPv6 datagrams received by GMAC with header (length or version mismatch) errors.

28.3.112 RxIPv6_No_Payload_Packets : RxIPv6 No Payload Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x082C



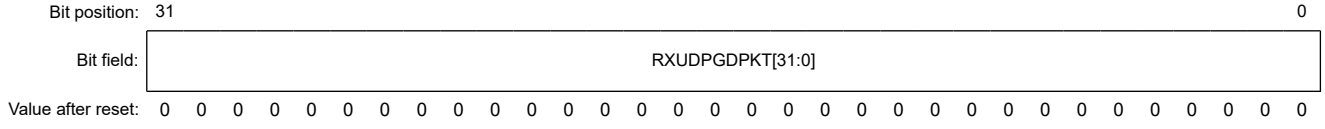
Bit	Symbol	Function	R/W
31:0	RXIPV6NOPAYPKT[31:0]	RxIPv6 Payload Packets This field indicates the number of IPv6 datagram packets received that did not have a TCP, UDP, or ICMP payload. This includes all IPv6 datagrams with fragmentation or security extension headers.	R

This register provides the number of IPv6 datagram packets received by GMAC that did not have a TCP, UDP, or ICMP payload. This includes all IPv6 datagrams with fragmentation or security extension headers.

28.3.113 RxUDP_Good_Packets : RxUDP Good Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0830



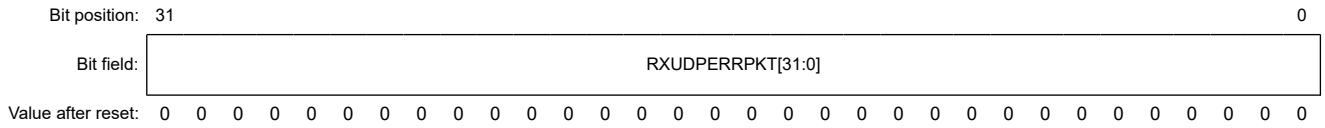
Bit	Symbol	Function	R/W
31:0	RXUDPGDPKT[31:0]	RxUDP Good Packets This field indicates the number of good IP datagrams received with a good UDP payload. This counter is not updated when the RxIPv4_UDP_Checksum_Disabled_Packets counter is incremented.	R

This register provides the number of good IP datagrams received by GMAC with a good UDP payload. This counter is not updated when the RxIPv4_UDP_Checksum_Disabled_Packets counter is incremented.

28.3.114 RxUDP_Error_Packets : RxUDP Error Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0834



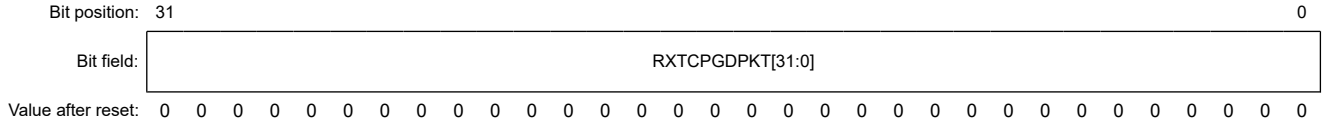
Bit	Symbol	Function	R/W
31:0	RXUDPERRPKT[31:0]	RxUDP Error Packets This field indicates the number of good IP datagrams received whose UDP payload has a checksum error.	R

This register provides the number of good IP datagrams received by GMAC whose UDP payload has a checksum error.

28.3.115 RxTCP_Good_Packets : RxTCP Good Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0838



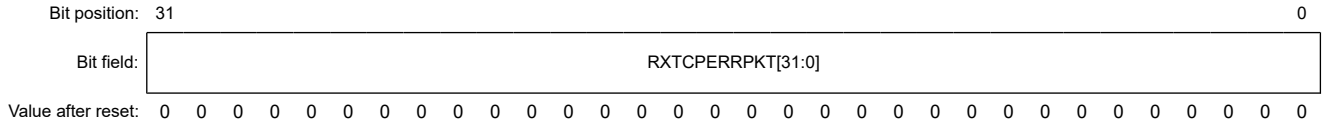
Bit	Symbol	Function	R/W
31:0	RXTCPGDPKT[31:0]	RxTCP Good Packets This field indicates the number of good IP datagrams received with a good TCP payload.	R

This register provides the number of good IP datagrams received by GMAC with a good TCP payload.

28.3.116 RxTCP_Error_Packets : RxTCP Error Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x083C



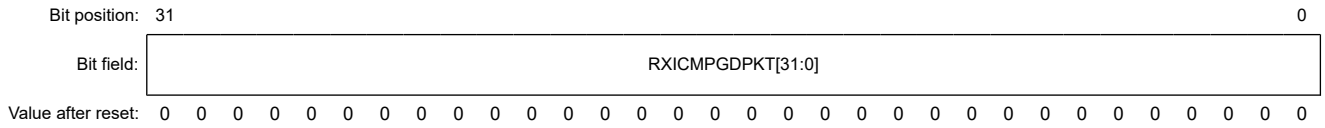
Bit	Symbol	Function	R/W
31:0	RXTCPERRPKT[31:0]	RxTCP Error Packets This field indicates the number of good IP datagrams received whose TCP payload has a checksum error.	R

This register provides the number of good IP datagrams received by GMAC whose TCP payload has a checksum error.

28.3.117 RxICMP_Good_Packets : RxICMP Good Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0840



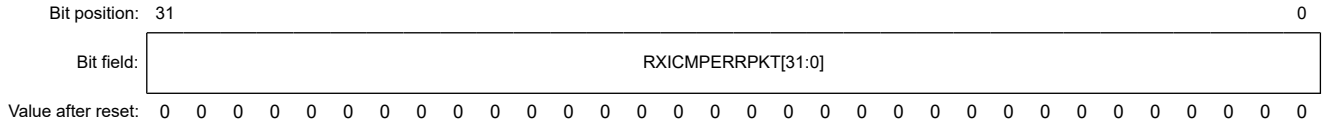
Bit	Symbol	Function	R/W
31:0	RXICMPGDPKT[31:0]	RxICMP Good Packets This field indicates the number of good IP datagrams received with a good ICMP payload.	R

This register provides the number of good IP datagrams received by GMAC with a good ICMP payload.

28.3.118 RxICMP_Error_Packets : RxICMP Error Packets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0844



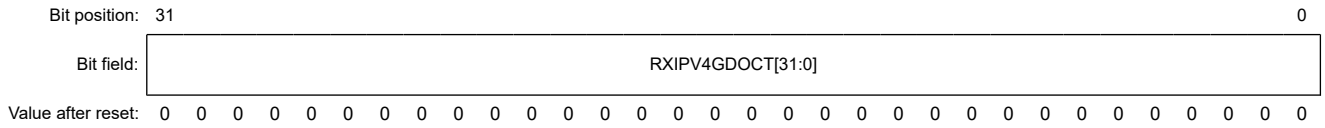
Bit	Symbol	Function	R/W
31:0	RXICMPERRPKT[31:0]	RxICMP Error Packets This field indicates the number of good IP datagrams received whose ICMP payload has a checksum error.	R

This register provides the number of good IP datagrams received by GMAC whose ICMP payload has a checksum error.

28.3.119 RxIPv4_Good_Octets : RxIPv4 Good Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0850



Bit	Symbol	Function	R/W
31:0	RXIPV4GDOCT[31:0]	RxIPv4 Good Octets This field indicates the number of bytes received in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)	R

This register provides the number of bytes received by GMAC in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)

28.3.120 RxIPv4_Header_Error_Octets : RxIPv4 Header Error Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0854

Bit position:	31																												0
Bit field:	RXIPV4HDRERROCT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXIPV4HDRERROCT[31:0]	RxIPv4 Header Error Octets This field indicates the number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)	R

This register provides the number of bytes received by GMAC in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)

28.3.121 RxIPv4_No_Payload_Octets : RxIPv4 No Payload Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0858

Bit position:	31																												0
Bit field:	RXIPV4NOPAYOCT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXIPV4NOPAYOCT[31:0]	RxIPv4 Payload Octet This field indicates the number of bytes received in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)	R

This register provides the number of bytes received by GMAC in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)

28.3.122 RxIPv4_Fragmented_Octets : RxIPv4 Fragmented Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x085C

Bit position:	31																												0
Bit field:	RXIPV4FRAGOCT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXIPV4FRAGOCT[31:0]	RxIPv4 Fragmented Octets This field indicates the number of bytes received in fragmented IPv4 datagrams. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)	R

This register provides the number of bytes received by GMAC in fragmented IPv4 datagrams. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)

28.3.123 RxIPv4_UDP_Checksum_Disable_Octets : RxIPv4 UDP Checksum Disable Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0860

Bit position:	31																												0
Bit field:	RXIPV4UDSBLOCT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXIPV4UDSBLOCT[31:0]	RxIPv4 UDP Checksum Disable Octets This field indicates the number of bytes received in a UDP segment that had the UDP checksum disabled. This counter does not count IP Header bytes. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)	R

This register provides the number of bytes received by GMAC in a UDP segment that had the UDP checksum disabled. This counter does not count IP Header bytes. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)

28.3.124 RxIPv6_Good_Octets : RxIPv6 Good Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0864

Bit position:	31																												0
Bit field:	RXIPV6GDOCT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXIPV6GDOCT[31:0]	RxIPv6 Good Octets This field indicates the number of bytes received in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)	R

This register provides the number of bytes received by GMAC in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)

28.3.125 RxIPv6_Header_Error_Octets : RxIPv6 Header Error Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0868

Bit position:	31																												0
Bit field:	RXIPV6HDRERROCT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXIPV6HDRERROCT[31:0]	RxIPv6 Header Error Octets This field indicates the number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)	R

This register provides the number of bytes received by GMAC in IPv6 datagrams with header errors (length, version mismatch). The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)

28.3.126 RxIPv6_No_Payload_Octets : RxIPv6 No Payload Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x086C

Bit position:	31																												0
Bit field:	RXIPV6NOPAYOCT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXIPV6NOPAYOCT[31:0]	RxIPv6 Payload Octets This field indicates the number of bytes received in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)	R

This register provides the number of bytes received by GMAC in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.)

28.3.127 RxUDP_Good_Octets : RxUDP Good Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0870

Bit position:	31																												0
Bit field:	RXUDPGDOCT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

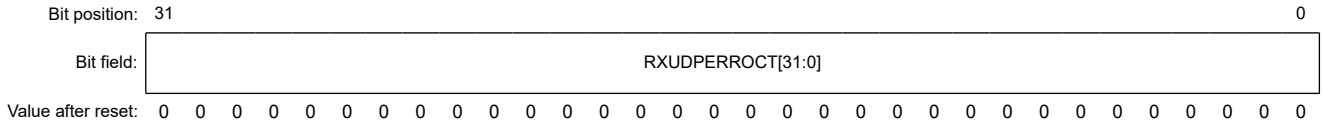
Bit	Symbol	Function	R/W
31:0	RXUDPGDOCT[31:0]	RxUDP Good Octets This field indicates the number of bytes received in a good UDP segment. This counter does not count IP header bytes.	R

This register provides the number of bytes received by GMAC in a good UDP segment. This counter does not count IP header bytes.

28.3.128 RxUDP_Error_Octets : RxUDP Error Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0874



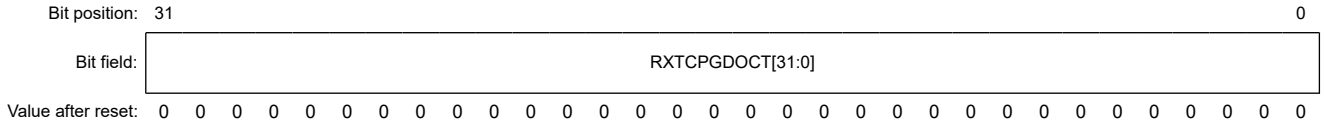
Bit	Symbol	Function	R/W
31:0	RXUDPERROCT[31:0]	RxUDP Error Octets This field indicates the number of bytes received in a UDP segment that had checksum errors. This counter does not count IP header bytes.	R

This register provides the number of bytes received by GMAC in a UDP segment that had checksum errors. This counter does not count IP header bytes.

28.3.129 RxTCP_Good_Octets : RxTCP Good Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0878



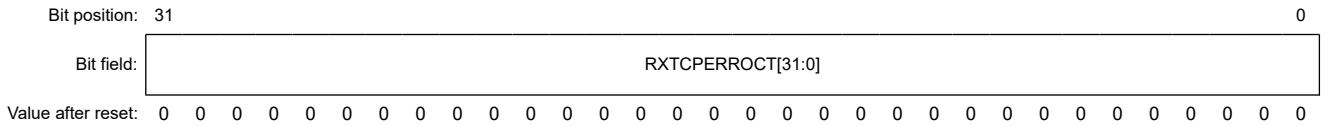
Bit	Symbol	Function	R/W
31:0	RXTCPGDOCT[31:0]	RxTCP Good Octets This field indicates the number of bytes received in a good TCP segment. This counter does not count IP header bytes.	R

This register provides the number of bytes received by GMAC in a good TCP segment. This counter does not count IP header bytes.

28.3.130 RxTCP_Error_Octets : RxTCP Error Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x087C



Bit	Symbol	Function	R/W
31:0	RXTCPERROCT[31:0]	RxTCP Error Octets This field indicates the number of bytes received in a TCP segment that had checksum errors. This counter does not count IP header bytes.	R

This register provides the number of bytes received by GMAC in a TCP segment that had checksum errors. This counter does not count IP header bytes.

28.3.131 RxICMP_Good_Octets : RxICMP Good Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0880

Bit position:	31																												0
Bit field:	RXICMPGDOCT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXICMPGDOCT[31:0]	RxICMP Good Octets This field indicates the number of bytes received in a good ICMP segment. This counter does not count IP header bytes.	R

This register provides the number of bytes received by GMAC in a good ICMP segment. This counter does not count IP header bytes.

28.3.132 RxICMP_Error_Octets : RxICMP Error Octets Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0884

Bit position:	31																												0
Bit field:	RXICMPERROCT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXICMPERROCT[31:0]	RxICMP Error Octets This field indicates the number of bytes received in a ICMP segment that had checksum errors. This counter does not count IP header bytes.	R

This register provides the number of bytes received by GMAC in a ICMP segment that had checksum errors. This counter does not count IP header bytes.

28.3.133 MMC_FPE_Tx_Interrupt : MMC FPE Transmit Interrupt Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x08A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HRCIS	FCIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FCIS	MMC Tx FPE Fragment Counter Interrupt status*1 This bit is set when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Tx FPE Fragment Counter Interrupt status not detected 1: MMC Tx FPE Fragment Counter Interrupt status detected	R
1	HRCIS	MMC Tx Hold Request Counter Interrupt Status*1 This bit is set when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Tx Hold Request Counter Interrupt Status not detected 1: MMC Tx Hold Request Counter Interrupt Status detected	R
31:2	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

This register maintains the interrupts generated from all FPE related Transmit statistics counters. The MMC_FPE_Tx_Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones. The MMC_FPE_Tx_Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits [7:0]) of the respective counter must be read to clear the interrupt bit.

28.3.134 MMC_FPE_Tx_Interrupt_Mask : MMC FPE Transmit Interrupt Mask Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x08A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HRCIM	FCIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FCIM	MMC Transmit Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Fragment Counter Interrupt Mask is disabled 1: MMC Transmit Fragment Counter Interrupt Mask is enabled	R/W
1	HRCIM	MMC Transmit Hold Request Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Transmit Hold Request Counter Interrupt Mask is disabled 1: MMC Transmit Hold Request Counter Interrupt Mask is enabled	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

This register maintains the masks for interrupts generated from all FPE related Transmit statistics counters. The MMC_FPE_Tx_Interrupt_Mask register maintains the masks for the interrupts generated when FPE related receive statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

28.3.135 MMC_Tx_FPE_Fragment_Cntr : Tx FPE Fragment Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x08A8

Bit position:	31																												0
Bit field:	TXFFC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXFFC[31:0]	Tx FPE Fragment counter This field indicates the number of additional mPackets that has been transmitted due to preemption.	R

This register provides the number of additional mPackets transmitted due to preemption.

28.3.136 MMC_Tx_Hold_Req_Cntr : Tx Hold Request Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x08AC

Bit position:	31																												0
Bit field:	TXHRC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXHRC[31:0]	Tx Hold Request Counter This field indicates count of number of a hold request is given to MAC.	R

This register provides the count of number of times a hold request is given to MAC.

28.3.137 MMC_FPE_Rx_Interrupt : MMC FPE Receive Interrupt Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x08C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FCIS	PAOCIS	PSECIS	PAECIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PAECIS	MMC Rx Packet Assembly Error Counter Interrupt Status[*1] This bit is set when the Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Rx Packet Assembly Error Counter Interrupt Status not detected 1: MMC Rx Packet Assembly Error Counter Interrupt Status detected	R

Bit	Symbol	Function	R/W
1	PSECIS	MMC Rx Packet SMD Error Counter Interrupt Status*1 This bit is set when the Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Rx Packet SMD Error Counter Interrupt Status not detected 1: MMC Rx Packet SMD Error Counter Interrupt Status detected	R
2	PAOCIS	MMC Rx Packet Assembly OK Counter Interrupt Status*1 This bit is set when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Rx Packet Assembly OK Counter Interrupt Status not detected 1: MMC Rx Packet Assembly OK Counter Interrupt Status detected	R
3	FCIS	MMC Rx FPE Fragment Counter Interrupt Status*1 This bit is set when the Rx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Rx FPE Fragment Counter Interrupt Status not detected 1: MMC Rx FPE Fragment Counter Interrupt Status detected	R
31:4	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

This register maintains the interrupts generated from all FPE related Receive statistics counters. The MMC_FPE_Rx_Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones. The MMC_FPE_Rx_Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits [7:0]) of the respective counter must be read to clear the interrupt bit.

28.3.138 MMC_FPE_Rx_Interrupt_Mask : MMC FPE Receive Interrupt Mask Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x08C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FCIM	PAOCIM	PSECIM	PAECIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PAECIM	MMC Rx Packet Assembly Error Counter Interrupt Mask Setting this bit masks the interrupt when the Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Rx Packet Assembly Error Counter Interrupt Mask is disabled 1: MMC Rx Packet Assembly Error Counter Interrupt Mask is enabled	R/W
1	PSECIM	MMC Rx Packet SMD Error Counter Interrupt Mask Setting this bit masks the interrupt when the Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Rx Packet SMD Error Counter Interrupt Mask is disabled 1: MMC Rx Packet SMD Error Counter Interrupt Mask is enabled	R/W
2	PAOCIM	MMC Rx Packet Assembly OK Counter Interrupt Mask Setting this bit masks the interrupt when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Rx Packet Assembly OK Counter Interrupt Mask is disabled 1: MMC Rx Packet Assembly OK Counter Interrupt Mask is enabled	R/W

Bit	Symbol	Function	R/W
3	FCIM	MMC Rx FPE Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. 0: MMC Rx FPE Fragment Counter Interrupt Mask is disabled 1: MMC Rx FPE Fragment Counter Interrupt Mask is enabled	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

This register maintains the masks for interrupts generated from all FPE related Receive statistics counters. The MMC_FPE_Rx_Interrupt_Mask register maintains the masks for the interrupts generated when FPE related receive statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

28.3.139 MMC_Rx_Packet_Assembly_Err_Cntr : Rx Packet Assembly Error Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x08C8

Bit position:	31																												0
Bit field:	PAEC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PAEC[31:0]	Rx Packet Assembly Error Counter This field indicates the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value.	R

This register provides the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value.

28.3.140 MMC_Rx_Packet_SMD_Err_Cntr : Rx Packet SMD Error Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x08CC

Bit position:	31																												0
Bit field:	PSEC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PSEC[31:0]	Rx Packet SMD Error Counter This field indicates the number of MAC frames rejected due to unknown SMD value and MAC frame fragments rejected due to arriving with an SMD-C when there was no preceding preempted frame.	R

This register provides the number of received MAC frames rejected due to unknown SMD value and MAC frame fragments rejected due to arriving with an SMD-C when there was no preceding preempted frame.

28.3.141 MMC_Rx_Packet_Assembly_OK_Cntr : Rx Packet Assembly OK Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x08D0

Bit position:	31																												0
Bit field:	PAOC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PAOC[31:0]	Rx Packet Assembly OK Counter This field indicates the number of MAC frames that were successfully reassembled and delivered to MAC.	R

This register provides the number of MAC frames that were successfully reassembled and delivered to MAC.

28.3.142 MMC_Rx_FPE_Fragment_Cntr : Rx FPE Fragment Counter

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x08D4

Bit position:	31																												0
Bit field:	FFC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FFC[31:0]	Rx FPE Fragment Counter This field indicates the number of additional mPackets received due to preemption.	R

This register provides the number of additional mPackets received due to preemption.

28.3.143 MAC_L3_L4_CONTROLn : MAC Layer 3 and Layer 4 Control Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0900 + 0x30 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	DMCH EN0	—	DMCHN0[2:0]			—	—	L4DPI M0	L4DP M0	L4SPI M0	L4SP M0	—	L4PEN 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	L3HDBM0[4:0]					L3HSBM0[4:0]					L3DAI M0	L3DA M0	L3SAI M0	L3SA M0	—	L3PEN 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	L3PEN0	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. 0: Layer 3 Protocol is disabled 1: Layer 3 Protocol is enabled	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	L3SAM0	Layer 3 IP SA Match Enable*1 When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. 0: Layer 3 IP SA Match is disabled 1: Layer 3 IP SA Match is enabled	R/W
3	L3SAIM0	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0: Layer 3 IP SA Inverse Match is disabled 1: Layer 3 IP SA Inverse Match is enabled	R/W
4	L3DAM0	Layer 3 IP DA Match Enable*2 When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. 0: Layer 3 IP DA Match is disabled 1: Layer 3 IP DA Match is enabled	R/W
5	L3DAIM0	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0: Layer 3 IP DA Inverse Match is disabled 1: Layer 3 IP DA Inverse Match is enabled	R/W
10:6	L3HSBM0[4:0]	Layer 3 IP SA Higher Bits Match **IPv4 Packets:** 0x00: No bits are masked 0x01: LSb [0] is masked 0x02: Two LSbs [1:0] are masked ⋮ 0x1F: All bits except MSb are masked **IPv6 Packets:** This field contains Bits [4:0] of L3HSBM0. These bits indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high. See description of the L3HDBM0 field.	R/W
15:11	L3HDBM0[4:0]	Layer 3 IP DA Higher Bits Match*3 **IPv4 Packets:** 0x00: No bits are masked 0x01: LSb [0] is masked 0x02: Two LSbs [1:0] are masked ⋮ 0x1F: All bits except MSb are masked **IPv6 Packets:** Bits [12:11] of this field correspond to Bits [6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: 0x00: No bits are masked 0x01: LSb [0] is masked 0x02: Two LSbs [1:0] are masked ⋮ 0x7F: All bits except MSb are masked	R/W

Bit	Symbol	Function	R/W
16	L4PEN0	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set. 0: Layer 4 Protocol is disabled 1: Layer 4 Protocol is enabled	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
18	L4SPM0	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0: Layer 4 Source Port Match is disabled 1: Layer 4 Source Port Match is enabled	R/W
19	L4SPIM0	Layer 4 Source Port Inverse Match Enable*4 When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. 0: Layer 4 Source Port Inverse Match is disabled 1: Layer 4 Source Port Inverse Match is enabled	R/W
20	L4DPM0	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0: Layer 4 Destination Port Match is disabled 1: Layer 4 Destination Port Match is enabled	R/W
21	L4DPIM0	Layer 4 Destination Port Inverse Match Enable*5 When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. 0: Layer 4 Destination Port Inverse Match is disabled 1: Layer 4 Destination Port Inverse Match is enabled	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
26:24	DMCHN0[2:0]	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	DMCHEN0	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0: DMA Channel Select is disabled 1: DMA Channel Select is enabled	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering.
Note 2. When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering.
Note 3. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
Note 4. This bit is valid and applicable only when the L4SPM0 bit is set high.
Note 5. This bit is valid and applicable only when the L4DPM0 bit is set high.

The MAC_L3_L4_CONTROLn register controls the operations of filter 0 of Layer 3 and Layer 4.

28.3.144 MAC_LAYER4_ADDRESSn : MAC Layer 4 Address n Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0904 + 0x30 × n

Bit position:	31															16	15													0
Bit field:	L4DP0[15:0]																L4SP0[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	L4SP0[15:0]	Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_Control0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.	R/W
31:16	L4DP0[15:0]	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_Control0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.	R/W

The MAC_LAYER4_ADDRESSn, MAC_L3_L4_CONTROLn, MAC_LAYER3_ADDR0_REGn, MAC_LAYER3_ADDR1_REGn, MAC_LAYER3_ADDR2_REGn, and MAC_LAYER3_ADDR3_REGn registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the IP.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the IP. When you select this option, the synchronization is triggered only when Bits [31:24] of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

28.3.145 MAC_LAYER3_ADDR0_REGn : MAC Layer 3 Address 0 Register n (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0910 + 0x30 × n

Bit position:	31																													0
Bit field:	L3A00[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

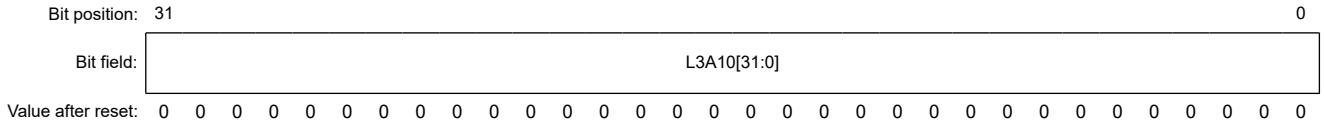
Bit	Symbol	Function	R/W
31:0	L3A00[31:0]	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits [31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits [31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.	R/W

For IPv4 packets, the Layer 3 Address 0 Register 0 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits [31:0] of the 128-bit IP Source Address or Destination Address field.

28.3.146 MAC_LAYER3_ADDR1_REGn : MAC Layer 3 Address 1 Register n (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0914 + 0x30 × n



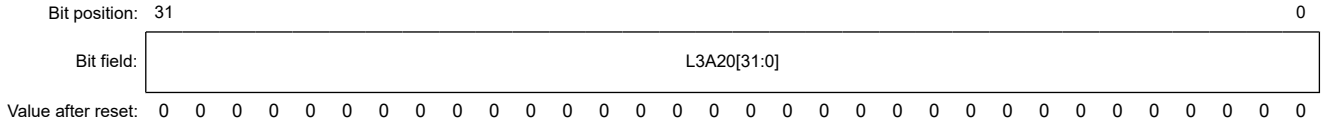
Bit	Symbol	Function	R/W
31:0	L3A10[31:0]	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits [63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits [63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.	R/W

For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits [63:32] of the 128-bit IP Source Address or Destination Address field.

28.3.147 MAC_LAYER3_ADDR2_REGn : MAC Layer 3 Address 2 Register n (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0918 + 0x30 × n



Bit	Symbol	Function	R/W
31:0	L3A20[31:0]	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits [95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits [95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_Control0 register, this field is not used.	R/W

The Layer 3 Address 2 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits [95:64] of 128-bit IP Source Address or Destination Address field.

28.3.148 MAC_LAYER3_ADDR3_REGn : MAC Layer 3 Address 3 Register n (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x091C + 0x30 × n

Bit position:	31																												0
Bit field:	L3A30[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	L3A30[31:0]	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits [127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits [127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_Control0 register, this field is not used.	R/W

The Layer 3 Address 3 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits [127:96] of 128-bit IP Source Address or Destination Address field.

28.3.149 MAC_Indir_Access_Ctrl : MAC Indirect Access Control and Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0A70

Bit position:	31											19		16	15				8		5			1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MSEL[3:0]			AOFF[7:0]						AUTO			COM	OB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OB	Operation Busy This bit is set along with a read or write command for initiating the indirect access to MAC_<MSEL>_<AOFF> register. This bit is reset when the read or write command to MAC_<MSEL>_<AOFF> register is complete. The next indirect register access can be initiated only after this bit is reset. During a write operation, the bit is reset only after the data has been written into MAC_<MSEL>_<AOFF> register. During a read operation, the data should be read from MAC_Indir_Access_Data register only after this bit is reset.	R/W
1	COM	Command type This bit indicates the register access type. 0: Write operation. Indicates a write operation. 1: Read operation. Indicates a read operation.	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	AUTO	Auto increment 0: AOFF is not incremented automatically. Software should program the correct Address Offset for each access. 1: AOFF is incremented by 1. Software should ensure not to cause a wrap condition. Byte wise read/write is not supported when auto increment is enabled.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	AOFF[7:0]	Address Offset This field is used in indirect access of MAC_<MSEL>_<AOFF>. This field must be set along with initiation of read/write to MAC_<MSEL>_<AOFF> and should not be changed until the OB is reset. 0x00: Reg0 (Indirect register 0) 0x01: Reg1 (Indirect register 1) ⋮ 0x07: Reg7 (Indirect register 7) Others: Reserved	R/W
19:16	MSEL[3:0]	Mode Select This field is used in indirect access of MAC_<MSEL>_<AOFF>. This field must be set along with initiation of read/write to MAC_<MSEL>_<AOFF> and should not be changed until the OB is reset. 0x0: TMRQ (Type based RXQ mapping) Others: Reserved	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

This register provides the Indirect Access control and status for MAC_<MSEl>_<AOFF> registers.

28.3.150 MAC_Indir_Access_Data : MAC Indirect Access Data Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0A74

Bit position:	31																													0
Bit field:	DATA[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DATA[31:0]	This field contains data to read/write for Indirect address access associated with MAC_Indir_Access_Ctrl register.	R/W

This register holds the read/write data for Indirect Access of MAC_<MSEL>_<AOFF> registers.

28.3.151 MAC_Timestamp_Control : Timestamp Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	AV8021ASMEN	—	—	—	TXTSTSM	—	—	—	—	CSC	TSENMACADDR	SNAPTYPSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSMSTRENA	TSEVNTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	—	—	—	TSTRIG	—	—	—	TSENA
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSENA	Enable Timestamp When this bit is set, the timestamp is added for Transmit and Receive packets. When disabled, timestamp is not added for transmit and receive packets and the Timestamp Generator is also suspended. You need to initialize the Timestamp (system time) after enabling this mode. On the Receive side, the MAC processes the 1588 packets only if this bit is set. 0: Timestamp is disabled 1: Timestamp is enabled	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TSTRIG	Enable Timestamp Interrupt Trigger*1 When this bit is set, the timestamp interrupt is generated when the System Time becomes greater than the value written in the Target Time register. This bit is reset after the Timestamp Trigger Interrupt is generated. 0: Timestamp Interrupt Trigger is not enabled 1: Timestamp Interrupt Trigger is enabled	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	TSENALL	Enable Timestamp for All Packets When this bit is set, the timestamp snapshot is enabled for all packets received by the MAC. 0: Timestamp for All Packets disabled 1: Timestamp for All Packets enabled	R/W
9	TSCTRLSSR	Timestamp Digital or Binary Rollover Control When this bit is set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. When this bit is reset, the rollover value of sub-second register is 0x7FFF_FFFF. The sub-second increment must be programmed correctly depending on the PTP reference clock frequency and the value of this bit. 0: Timestamp Digital or Binary Rollover Control is disabled 1: Timestamp Digital or Binary Rollover Control is enabled	R/W
10	TSVER2ENA	Enable PTP Packet Processing for Version 2 Format When this bit is set, the IEEE 1588 version 2 format is used to process the PTP packets. When this bit is reset, the IEEE 1588 version 1 format is used to process the PTP packets. The IEEE 1588 formats are described in 'PTP Processing and Control'. 0: PTP Packet Processing for Version 2 Format is disabled 1: PTP Packet Processing for Version 2 Format is enabled	R/W
11	TSIPENA	Enable Processing of PTP over Ethernet Packets When this bit is set, the MAC receiver processes the PTP packets encapsulated directly in the Ethernet packets. When this bit is reset, the MAC ignores the PTP over Ethernet packets. 0: Processing of PTP over Ethernet Packets is disabled 1: Processing of PTP over Ethernet Packets is enabled	R/W
12	TSIPV6ENA	Enable Processing of PTP Packets Sent over IPv6-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv6-UDP packets. When this bit is clear, the MAC ignores the PTP transported over IPv6-UDP packets. 0: Processing of PTP Packets Sent over IPv6-UDP is disabled 1: Processing of PTP Packets Sent over IPv6-UDP is enabled	R/W
13	TSIPV4ENA	Enable Processing of PTP Packets Sent over IPv4-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv4-UDP packets. When this bit is reset, the MAC ignores the PTP transported over IPv4-UDP packets. This bit is set by default. 0: Processing of PTP Packets Sent over IPv4-UDP is disabled 1: Processing of PTP Packets Sent over IPv4-UDP is enabled	R/W
14	TSEVNTENA	Enable Timestamp Snapshot for Event Messages When this bit is set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). When this bit is reset, the snapshot is taken for all messages except Announce, Management, and Signaling. For more information about the timestamp snapshots, see Timestamp Snapshot Dependency on Register Bits Table. 0: Timestamp Snapshot for Event Messages is disabled 1: Timestamp Snapshot for Event Messages is enabled	R/W

Bit	Symbol	Function	R/W
15	TSMSTRENA	Enable Snapshot for Messages Relevant to Master When this bit is set, the snapshot is taken only for the messages that are relevant to the master node. Otherwise, the snapshot is taken for the messages relevant to the slave node. 0: Snapshot for Messages Relevant to Master is disabled 1: Snapshot for Messages Relevant to Master is enabled	R/W
17:16	SNAPTYPSEL[1:0]	Select PTP packets for Taking Snapshots These bits, along with Bits 15 and 14, decide the set of PTP packet types for which snapshot needs to be taken. The encoding is given in Timestamp Snapshot Dependency on Register Bits Table.	R/W
18	TSENMACADDR	Enable MAC Address for PTP Packet Filtering When this bit is set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP packets when PTP is directly sent over Ethernet. When this bit is set, received PTP packets with DA containing a special multicast or unicast address that matches the one programmed in MAC address registers are considered for processing as indicated, when PTP is directly sent over Ethernet. For normal time stamping operation, MAC address registers 0 to 31 is considered for unicast destination address matching. For PTP offload, only MAC address register 0 is considered for unicast destination address matching. 0: MAC Address for PTP Packet Filtering is disabled 1: MAC Address for PTP Packet Filtering is enabled	R/W
19	CSC	Enable checksum correction during OST for PTP over UDP/IPv4 packets When this bit is set, the last two bytes of PTP message sent over UDP/IPv4 is updated to keep the UDP checksum correct, for changes made to origin timestamp and/or correction field as part of one step timestamp operation. The application forms the packet with these two dummy bytes. When reset, no updates are done to keep the UDP checksum correct. The application forms the packet with UDP checksum set to 0. 0: Checksum correction during OST for PTP over UDP/IPv4 packets is disabled 1: Checksum correction during OST for PTP over UDP/IPv4 packets is enabled	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	TXTSSTSM	Transmit Timestamp Status Mode When this bit is set, the MAC overwrites the earlier transmit timestamp status even if it is not read by the software. The MAC indicates this by setting the TXTSSMIS bit of the MAC_Tx_Timestamp_Status_Nanoseconds register. When this bit is reset, the MAC ignores the timestamp status of current packet if the timestamp status of previous packet is not read by the software. The MAC indicates this by setting the TXTSSMIS bit of the MAC_Tx_Timestamp_Status_Nanoseconds register. 0: Transmit Timestamp Status Mode is disabled 1: Transmit Timestamp Status Mode is enabled	R/W
27:25	—	These bits are read as 0. The write value should be 0.	R/W
28	AV8021ASMEN	AV 802.1AS Mode Enable When this bit is set, the MAC processes only untagged PTP over Ethernet packets for providing PTP status and capturing timestamp snapshots, that is, IEEE 802.1AS mode of operation. When PTP offload feature is enabled, for the purpose of PTP offload, the transport specific field in the PTP header is generated and checked based on the value of this bit. 0: AV 802.1AS Mode is disabled 1: AV 802.1AS Mode is enabled	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

This register controls the operation of the System Time generator and processing of PTP packets for timestamping in the Receiver.

28.3.152 MAC_System_Time_Seconds : Timestamp Seconds Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B08

Bit position:	31																												0
Bit field:	TSS[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TSS[31:0]	Timestamp Second The value in this field indicates the current value in seconds of the System Time maintained by the MAC.	R

The MAC_System_Time_Seconds register, along with System Time Nanoseconds register, indicates the current value of the system time maintained by the MAC. Though it is updated on a continuous basis, there is some delay from the actual time because of clock domain transfer latencies (from PTP clock to CSR clock).

28.3.153 MAC_System_Time_Nanoseconds : Timestamp Nanoseconds Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B0C

Bit position:	31	30																											0
Bit field:	—	TSSS[30:0]																											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
30:0	TSSS[30:0]	Timestamp Sub Seconds The value in this field has the sub-second representation of time, with an accuracy of 0.46 ns. When Bit 9 is set in MAC_Timestamp_Control, each bit represents 1 ns. The maximum value is 0x3B9A_C9FF after which it rolls-over to zero.	R
31	—	This bit is read as 0.	R

The MAC_System_Time_Nanoseconds register, along with System Time Seconds register, indicates the current value of the system time maintained by the MAC.

28.3.154 MAC_Timestamp_Status : Timestamp Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	ATSNS[4:0]					ATSSTM	—	—	—	—	—	—	ATSSTN[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TXTSSIS	—	—	—	—	—	—	—	—	—	—	—	—	AUXTSTRIG	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0.	R
2	AUXTSTRIG	Auxiliary Timestamp Trigger Snapshot*1 This bit is set high when the auxiliary snapshot is written to the FIFO. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected. 0: Auxiliary Timestamp Trigger Snapshot status not detected 1: Auxiliary Timestamp Trigger Snapshot status detected	R
14:3	—	These bits are read as 0.	R
15	TXTSSIS	Tx Timestamp Status Interrupt Status This bit is set when the captured transmit timestamp is updated in the MAC_Tx_Timestamp_Status_Nanoseconds and MAC_Tx_Timestamp_Status_Seconds registers When PTP offload feature is enabled, this bit is set when the captured transmit timestamp is updated in the MAC_Tx_Timestamp_Status_Nanoseconds and MAC_Tx_Timestamp_Status_Seconds registers, for PTO generated Delay Request and Pdelay request packets. This bit is cleared when the MAC_Tx_Timestamp_Status_Seconds register is read (or write to MAC_Tx_Timestamp_Status_Seconds register when RCWE bit of MAC_CSR_SW_Ctrl register is set). 0: Tx Timestamp Status Interrupt status not detected 1: Tx Timestamp Status Interrupt status detected	R
17:16	ATSSTN[1:0]	Auxiliary Timestamp Snapshot Trigger Identifier*2 These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. When more than one bit is set at the same time, it means that corresponding auxiliary triggers were sampled at the same clock. These bits are applicable only if the number of Auxiliary snapshots is more than one. One bit is assigned for each trigger as shown in the following list: Bit 16: Auxiliary trigger 0 Bit 17: Auxiliary trigger 1	R
23:18	—	These bits are read as 0.	R
24	ATSSTM	Auxiliary Timestamp Snapshot Trigger Missed This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. This indicates that the latest snapshot is not stored in the FIFO. 0: Auxiliary Timestamp Snapshot Trigger Missed status not detected 1: Auxiliary Timestamp Snapshot Trigger Missed status detected	R
29:25	ATSNS[4:0]	Number of Auxiliary Timestamp Snapshots This field indicates the number of Snapshots available in the FIFO. A value equal to the selected depth of FIFO 4 indicates that the Auxiliary Snapshot FIFO is full. These bits are cleared (to 0x00) when the Auxiliary snapshot FIFO clear bit is set.	R
31:30	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.
Note 2. The software can read this register to find the triggers that are set when the timestamp is taken.

This register indicates the status of the Timestamp. All bits except Bits [27:25] gets cleared when the application reads this register.

28.3.155 MAC_Tx_Timestamp_Status_Nanoseconds : Transmit Timestamp Status Nanoseconds Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B30

Bit position:	31	30																												0
Bit field:	TXTSSMIS	TXTSSLO[30:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
30:0	TXTSSLO[30:0]	Transmit Timestamp Status Low This field contains the 31 bits of the Nanoseconds field of the Transmit packet's captured timestamp.	R
31	TXTSSMIS	Transmit Timestamp Status Missed*1 *2 0: Transmit Timestamp Status Missed status not detected 1: Transmit Timestamp Status Missed status detected	R

Note 1. Access restriction applies. Clears on read. Self-set to 1 on internal event.
Note 2. When this bit is set, it indicates one of the following:

- The timestamp of the current packet is ignored if TXTSSTSM bit of the MAC_Timestamp_Control register is reset.
- The timestamp of the previous packet is overwritten with timestamp of the current packet if TXTSSTSM bit of the MAC_Timestamp_Control register is set.

This register contains the nanosecond part of timestamp captured for Transmit packets when Tx status is disabled.

The MAC_Tx_Timestamp_Status_Nanoseconds register, along with MAC_Tx_Timestamp_Status_Seconds, gives the 64-bit timestamp captured for the PTP packet successfully transmitted by the MAC. This value is considered to be read by the application when the last byte of MAC_Tx_Timestamp_Status_Nanoseconds is read. This means when bits [31:24] are read; If the application does not read these registers and timestamp of another packet is captured, then either the current timestamp is lost (overwritten) or the new timestamp is lost (dropped), depending on the setting of the TXTSSTSM bit of the MAC_Timestamp_Control register. The status bit TXTSSIS bit [15] in MAC_Timestamp_Status register is set when the MAC transmitter captures the timestamp.

28.3.156 MAC_Tx_Timestamp_Status_Seconds : Transmit Timestamp Status Seconds Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B34

Bit position:	31																														0
Bit field:	TXTSSHI[31:0]																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXTSSHI[31:0]	Transmit Timestamp Status High This field contains the 32 bits of the Seconds field of Transmit packet's captured timestamp.	R

The register contains the 32 bits of the timestamp (in seconds) captured when a PTP packet is transmitted.

28.3.157 MAC_Auxiliary_Control : Auxiliary Timestamp Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	ATSEN1	ATSEN0	—	—	—	ATSFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ATSFC	Auxiliary Snapshot FIFO Clear*1 When set, this bit resets the pointers of the Auxiliary Snapshot FIFO. This bit is cleared when the pointers are reset and the FIFO is empty. When this bit is high, the auxiliary snapshots are stored in the FIFO. 0: Auxiliary Snapshot FIFO Clear is disabled 1: Auxiliary Snapshot FIFO Clear is enabled	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	ATSEN0	Auxiliary Snapshot 0 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 0. When this bit is set, the auxiliary snapshot of the event on GMAC_PTPTRG0 input is enabled. When this bit is reset, the events on this input are ignored. 0: Auxiliary Snapshot 0 is disabled 1: Auxiliary Snapshot 0 is enabled	R/W
5	ATSEN1	Auxiliary Snapshot 1 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 1. When this bit is set, the auxiliary snapshot of the event on GMAC_PTPTRG1 input is enabled. When this bit is reset, the events on this input are ignored. 0: Auxiliary Snapshot 1 is disabled 1: Auxiliary Snapshot 1 is enabled	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

The MAC_Auxiliary_Control register controls the Auxiliary Timestamp snapshot.

28.3.158 MAC_Auxiliary_Timestamp_Nanoseconds : Auxiliary Timestamp Nanoseconds Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B48

Bit position:	31	30																												0
Bit field:	—	AUXTSLO[30:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
30:0	AUXTSLO[30:0]	Auxiliary Timestamp Contains the lower 31 bits (nanoseconds field) of the auxiliary timestamp.	R
31	—	This bit is read as 0.	R

The MAC_Auxiliary_Timestamp_Nanoseconds register, along with MAC_Auxiliary_Timestamp_Seconds, gives the 64-bit timestamp stored as auxiliary snapshot. These two registers form the read port of a 64-bit wide FIFO with a depth of 4.

You can store multiple snapshots in this FIFO. Bits [29:25] in MAC_Timestamp_Status indicate the fill-level of the FIFO. The top of the FIFO is removed only when the last byte of MAC_Auxiliary_Timestamp_Seconds register is read. This means when Bits [31:24] are read.

28.3.159 MAC_Auxiliary_Timestamp_Seconds : Auxiliary Timestamp Seconds Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B4C

Bit position:	31																													0
Bit field:	AUXTSHI[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	AUXTSHI[31:0]	Auxiliary Timestamp Contains the 32 bits of the Seconds field of the auxiliary timestamp.	R

The MAC_Auxiliary_Timestamp_Seconds register contains the 32 bits of the Seconds field of the auxiliary timestamp register.

28.3.160 MAC_Timestamp_Ingress_Asym_Corr : Timestamp Ingress Asymmetry Correction Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B50

Bit position:	31																												0
Bit field:	OSTIAC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	OSTIAC[31:0]	One-Step Timestamp Ingress Asymmetry Correction This field contains the ingress path asymmetry value to be added to correctionField of Pdelay_Resp PTP packet. The programmed value should be in units of nanoseconds and multiplied by 2^{16}. For example, 2.5 ns is represented as 0x00028000. The value can also be negative, which is represented in 2's complement form with bit 31 representing the sign bit.	R/W

The MAC_Timestamp_Ingress_Asym_Corr register contains the Ingress Asymmetry Correction value to be used while updating correction field in PDelay_Resp PTP messages.

28.3.161 MAC_Timestamp_Egress_Asym_Corr : Timestamp Egress Asymmetry Correction Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B54

Bit position:	31																												0
Bit field:	OSTEAC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

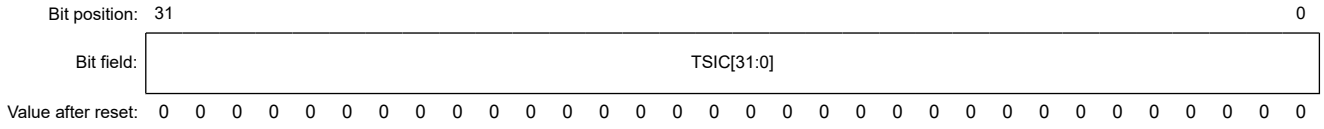
Bit	Symbol	Function	R/W
31:0	OSTEAC[31:0]	One-Step Timestamp Egress Asymmetry Correction This field contains the egress path asymmetry value to be subtracted from correctionField of Pdelay_Resp PTP packet. The programmed value must be the negated value in units of nanoseconds multiplied by 2^{16}. For example, if the required correction is +2.5 ns, the programmed value must be 0xFFFD_8000, which is the 2's complement of 0x0002_8000 (2.5 × 216). Similarly, if the required correction is -3.3 ns, the programmed value is 0x0003_4CCC (3.3 × 216).	R/W

The MAC_Timestamp_Egress_Asym_Corr register contains the Egress Asymmetry Correction value to be used while updating the correction field in PDelay_Req PTP messages.

28.3.162 MAC_Timestamp_Ingress_Corr_Nanosecond : Timestamp Ingress Correction Nanosecond Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B58



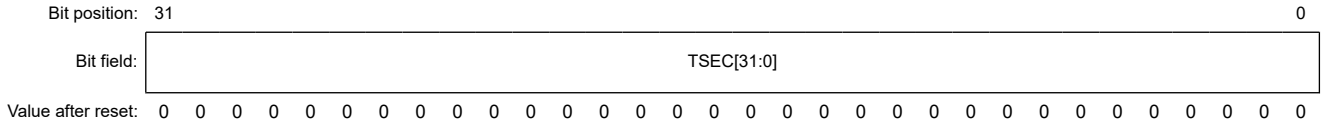
Bit	Symbol	Function	R/W
31:0	TSIC[31:0]	Timestamp Ingress Correction This field contains the ingress path correction value as defined by the Ingress Correction expression.	R/W

This register contains the correction value in nanoseconds to be used with the captured timestamp value in the ingress path.

28.3.163 MAC_Timestamp_Egress_Corr_Nanosecond : Timestamp Egress Correction Nanosecond Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B5C



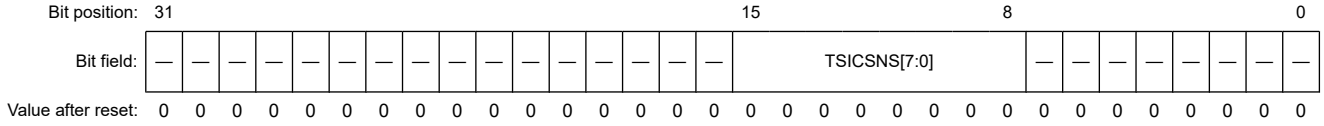
Bit	Symbol	Function	R/W
31:0	TSEC[31:0]	Timestamp Egress Correction This field contains the nanoseconds part of the egress path correction value as defined by the Egress Correction expression.	R/W

This register contains the correction value in nanoseconds to be used with the captured timestamp value in the egress path.

28.3.164 MAC_Timestamp_Ingress_Corr_Subnanosec : Timestamp Ingress Correction Subnanosecond Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B60



Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
15:8	TSICSNS[7:0]	Timestamp Ingress Correction, sub-nanoseconds This field contains the sub-nanoseconds part of the ingress path correction value as defined by the "Ingress Correction" expression.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

This register contains the sub-nanosecond part of the correction value to be used with the captured timestamp value, for ingress direction.

28.3.165 MAC_Timestamp_Egress_Corr_Subnanosec : Timestamp Egress Correction Subnanosecond Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B64

Bit position:	31																15								8							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSECSNS[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
15:8	TSECSNS[7:0]	Timestamp Egress Correction, sub-nanoseconds This field contains the sub-nanoseconds part of the egress path correction value as defined by the "Egress Correction" expression.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

This register contains the sub-nanosecond part of the correction value to be used with the captured timestamp value, for egress direction.

28.3.166 MAC_Timestamp_Ingress_Latency : Timestamp Ingress Latency Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B68

Bit position:	31				27											16	15							8							0
Bit field:	—	—	—	—	ITLNS[11:0]												ITLSNS[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0.	R
15:8	ITLSNS[7:0]	Ingress Timestamp Latency, in nanoseconds This register holds the average latency in nanoseconds between the input ports RXD of MAC and the actual point (GMII/MII) where the ingress timestamp is taken. Ingress correction value is computed as described in the section, "Ingress Correction".	R
27:16	ITLNS[11:0]	Ingress Timestamp Latency, in sub-nanoseconds This register holds the average latency in sub-nanoseconds between the input ports RXD of MAC and the actual point (GMII/MII) where the ingress timestamp is taken. Ingress correction value is computed as described in the section, "Ingress Correction".	R
31:28	—	These bits are read as 0.	R

This register holds the Ingress MAC latency.

28.3.167 MAC_Timestamp_Egress_Latency : Timestamp Egress Latency Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0B6C

Bit position:	31				27											16	15								8							0
Bit field:	—	—	—	—	ETLNS[11:0]												ETLSNS[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0.	R
15:8	ETLSNS[7:0]	Egress Timestamp Latency, in sub-nanoseconds This register holds the average latency in sub-nanoseconds between the actual point (GMII/MII) where the egress timestamp is taken and the output ports TXD of the MAC. Ingress correction value is computed as described in the section "Ingress Correction".	R
27:16	ETLNS[11:0]	Egress Timestamp Latency, in nanoseconds This register holds the average latency in nanoseconds between the actual point (GMII/MII) where the egress timestamp is taken and the output ports TXD of the MAC. Ingress correction value is computed as described in the section, "Ingress Correction".	R
31:28	—	These bits are read as 0.	R

This register holds the Egress MAC latency.

28.3.168 MAC_PTO_Control : PTP Offload Engine Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0BC0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DN[7:0]								PDRDIS	DRRDIS	APDREQTRIG	ASYNCTRIG	—	APDREQEN	ASYNCEN	PTOEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PTOEN	PTP Offload Enable When this bit is set, the PTP Offload feature is enabled. 0: PTP Offload feature is disabled 1: PTP Offload feature is enabled	R/W
1	ASYNCEN	Automatic PTP SYNC message Enable When this bit is set, PTP SYNC message is generated periodically based on interval programmed or trigger from application, when the MAC is programmed to be in Clock Master mode. 0: Automatic PTP SYNC message is disabled 1: Automatic PTP SYNC message is enabled	R/W
2	APDREQEN	Automatic PTP Pdelay_Req message Enable When this bit is set, PTP Pdelay_Req message is generated periodically based on interval programmed or trigger from application, when the MAC is programmed to be in Peer-to-Peer Transparent mode. 0: Automatic PTP Pdelay_Req message is disabled 1: Automatic PTP Pdelay_Req message is enabled	R/W

Bit	Symbol	Function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	ASYNCTRIG	Automatic PTP SYNC message Trigger*1 When this bit is set, one PTP SYNC message is transmitted. This bit is automatically cleared after the PTP SYNC message is transmitted. The application should set the ASYNCEN bit for this operation. 0: Automatic PTP SYNC message Trigger is disabled 1: Automatic PTP SYNC message Trigger is enabled	R/W
5	APDREQTRIG	Automatic PTP Pdelay_Req message Trigger*1 When this bit is set, one PTP Pdelay_Req message is transmitted. This bit is automatically cleared after the PTP Pdelay_Req message is transmitted. The application should set the APDREQEN bit for this operation. 0: Automatic PTP Pdelay_Req message Trigger is disabled 1: Automatic PTP Pdelay_Req message Trigger is enabled	R/W
6	DRRDIS	Disable PTO Delay Request/Response response generation When this bit is set, the Delay Request and Delay response is not generated for received SYNC and Delay request packet respectively, as required by the programmed mode. 0: PTO Delay Request/Response response generation is enabled 1: PTO Delay Request/Response response generation is disabled	R/W
7	PDRDIS	Disable Peer Delay Response response generation*2 When this bit is set, the Peer Delay Response (Pdelay_Resp) response is not be generated for received Peer Delay Request (Pdelay_Req) request packet, as required by the programmed mode. 0: Peer Delay Response response generation is enabled 1: Peer Delay Response response generation is disabled	R/W
15:8	DN[7:0]	Domain Number This field indicates the domain Number in which the PTP node is operating.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

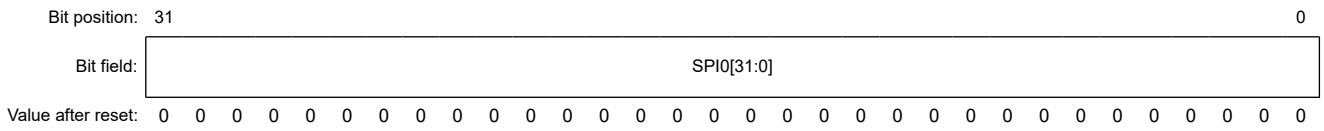
Note 1. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
Note 2. Setting this bit to 1 affects the normal PTP Offload operation and the time synchronization. So, this bit must be set only if there is problem with Pdelay_Resp generation in hardware and/or Pdelay_Resp generation is handled by software.

This register controls the PTP Offload Engine operation. This register is available only when the Enable PTP Timestamp Offload feature is selected.

28.3.169 MAC_Source_Port_Identity0 : PTP Source Port Identity 0 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0BC4



Bit	Symbol	Function	R/W
31:0	SPI0[31:0]	Source Port Identity 0 This field indicates bits [31:0] of sourcePortIdentity of PTP node.	R/W

This register contains Bits [31:0] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

28.3.170 MAC_Source_Port_Identity1 : PTP Source Port Identity 1 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0BC8

Bit position:	31																												0
Bit field:	SPI1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SPI1[31:0]	Source Port Identity 1 This field indicates bits [63:32] of sourcePortIdentity of PTP node.	R/W

This register contains Bits [63:32] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

28.3.171 MAC_Source_Port_Identity2 : PTP Source Port Identity 2 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0BCC

Bit position:	31															15													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPI2[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SPI2[15:0]	Source Port Identity 2 This field indicates bits [79:64] of sourcePortIdentity of PTP node.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

This register contains Bits [79:64] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

28.3.172 MAC_Log_Message_Interval : PTP Log Message Interval Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0BD0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	LMPDRI[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	DRSYNCR[2:0]			LSI[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	LSI[7:0]	Log Sync Interval This field indicates the periodicity of the automatically generated SYNC message when the PTP node is Master. Allowed values are -15 to 15. Negative value must be represented in 2's-complement form. For example, if the required value is -1, the value programmed must be 0xFF.	R/W
10:8	DRSYNCR[2:0]	Delay_Req to SYNC Ratio*1 *2 The master sends this information (logMinDelayReqInterval) in the DelayResp PTP messages to the slave. The GMAC Receiver processes this value from the received DelayResp messages and updates this field accordingly. In the Slave mode, the host must not write/update this register unless it has to override the received value. In Master mode, the sum of this field and logSyncInterval (LSI) field is provided in the logMinDelayReqInterval field of the generated multicast Delay_Resp PTP message. 0 0 0: DelayReq generated for every received SYNC 0 0 1: DelayReq generated every alternate reception of SYNC 0 1 0: For every 4 SYNC messages 0 1 1: For every 8 SYNC messages 1 0 0: For every 16 SYNC messages 1 0 1: For every 32 SYNC messages Others: Reserved	R/W
23:11	—	These bits are read as 0. The write value should be 0.	R/W
31:24	LMPDRI[7:0]	Log Min Pdelay_Req Interval This field indicates logMinPdelayReqInterval of PTP node. This is used to schedule the periodic Pdelay request packet transmission. Allowed values are -15 to 15. Negative value must be represented in 2's-complement form. For example, if the required value is -1, the value programmed must be 0xFF.	R/W

Note 1. In Slave mode, it is used for controlling frequency of Delay_Req messages transmitted.
Note 2. Access restriction applies. Updated based on the event. Setting 1 sets. Setting 0 clears.

This register contains the periodic intervals for automatic PTP packet generation. This register is available only when the Enable PTP Timestamp Offload feature is selected.

28.3.173 MTL_Operation_Mode : MTL Operation Mode Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CNTCLR	CNTPRST	—	SCHALG[1:0]		—	—	RAA	DTXSTS	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	DTXSTS	Drop Transmit Status 0: Drop Transmit Status is disabled. Tx packet status received from the MAC is forwarded to the application 1: Drop Transmit Status is enabled. Tx packet status received from the MAC is dropped in the MTL	R/W
2	RAA	Receive Arbitration Algorithm This field is used to select the arbitration algorithm for the Rx side. 0: Strict priority (SP). Queue 0 has the lowest priority and the last queue has the highest priority. 1: Weighted Strict Priority (WSP).	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6:5	SCHALG[1:0]	Tx Scheduling Algorithm This field indicates the algorithm for Tx scheduling: 0 0: WRR algorithm 0 1: WFQ algorithm when DCB feature is selected. Otherwise, Reserved. 1 0: DWRR algorithm when DCB feature is selected. Otherwise, Reserved. 1 1: Strict priority algorithm	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CNTPRST	Counters Preset*1 When this bit is set, ● MTL_TxQ[0-7]_UNDERFLOW register is initialized/preset to 0x7F0. ● Missed Packet and Overflow Packet counters in MTL_RxQ[0-7]_MISSED_PACKET_OVERFLOW_CNT register is initialized/preset to 0x7F0. 0: Counters Preset is disabled 1: Counters Preset is enabled	R/W
9	CNTCLR	Counters Reset*1 When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. If this bit is set along with CNT_PRESET bit, CNT_PRESET has precedence. 0: Counters are not reset 1: All counters are reset	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restrictions apply.

- MAC initiates the counters preset when application writes 1.
- MAC writes 0 after the counters preset is complete.
- MAC ignores when application writes 0.

The MTL_Operation_Mode register establishes the Transmit and Receive operating modes and commands.

28.3.174 MTL_Interrupt_Status : MTL Interrupt Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	ESTIS	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	Q7IS	Q6IS	Q5IS	Q4IS	Q3IS	Q2IS	Q1IS	Q0IS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	Q0IS	Queue 0 Interrupt status This bit indicates that there is an interrupt from Queue 0. To reset this bit, the application must read Queue 0 Interrupt Control and Status register to get the exact cause of the interrupt and clear its source. 0: Queue 0 Interrupt status not detected 1: Queue 0 Interrupt status detected	R
1	Q1IS	Queue 1 Interrupt status This bit indicates that there is an interrupt from Queue 1. To reset this bit, the application must read the MTL_Q1_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0: Queue 1 Interrupt status not detected 1: Queue 1 Interrupt status detected	R

Bit	Symbol	Function	R/W
2	Q2IS	Queue 2 Interrupt status This bit indicates that there is an interrupt from Queue 2. To reset this bit, the application must read the MTL_Q2_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0: Queue 2 Interrupt status not detected 1: Queue 2 Interrupt status detected	R
3	Q3IS	Queue 3 Interrupt status This bit indicates that there is an interrupt from Queue 3. To reset this bit, the application must read the MTL_Q3_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0: Queue 3 Interrupt status not detected 1: Queue 3 Interrupt status detected	R
4	Q4IS	Queue 4 Interrupt status This bit indicates that there is an interrupt from Queue 4. To reset this bit, the application must read the MTL_Q4_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0: Queue 4 Interrupt status not detected 1: Queue 4 Interrupt status detected	R
5	Q5IS	Queue 5 Interrupt status This bit indicates that there is an interrupt from Queue 5. To reset this bit, the application must read the MTL_Q5_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0: Queue 5 Interrupt status not detected 1: Queue 5 Interrupt status detected	R
6	Q6IS	Queue 6 Interrupt status This bit indicates that there is an interrupt from Queue 6. To reset this bit, the application must read the MTL_Q6_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0: Queue 6 Interrupt status not detected 1: Queue 6 Interrupt status detected	R
7	Q7IS	Queue 7 Interrupt status This bit indicates that there is an interrupt from Queue 7. To reset this bit, the application must read the MTL_Q7_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0: Queue 7 Interrupt status not detected 1: Queue 7 Interrupt status detected	R
17:8	—	These bits are read as 0.	R
18	ESTIS	EST (TAS- 802.1Qbv) Interrupt Status This bit indicates an interrupt event during the operation of 802.1Qbv. To reset this bit, the application must clear the error/event that has caused the Interrupt. 0: EST (TAS- 802.1Qbv) Interrupt status not detected 1: EST (TAS- 802.1Qbv) Interrupt status detected	R
31:19	—	These bits are read as 0.	R

The software driver (application) reads this register during interrupt service routine or polling to determine the interrupt status of MTL queues and the MAC.

28.3.175 MTL_RxQ_DMA_Map0 : Receive Queue and DMA Channel Mapping 0 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	Q3DD MACH	—	Q3MDMACH[2:0]			—	—	—	Q2DD MACH	—	Q2MDMACH[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	Q1DD MACH	—	Q1MDMACH[2:0]			—	—	—	Q0DD MACH	—	Q0MDMACH[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	Q0MDMACH[2:0]	Queue 0 Mapped to DMA Channel This field is valid when the Q0DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values might be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000b and 001b are valid, the other bits are reserved. This field controls the routing of the packet received in Queue 0 to the DMA channel: 0 0 0: DMA Channel 0 0 0 1: DMA Channel 1 0 1 0: DMA Channel 2 0 1 1: DMA Channel 3 1 0 0: DMA Channel 4 1 0 1: DMA Channel 5 1 1 0: DMA Channel 6 1 1 1: DMA Channel 7	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	Q0DDMACH	Queue 0 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 0 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 0 are routed to the DMA Channel programmed in the Q0MDMACH field. 0: Queue 0 disabled for DA-based DMA Channel Selection 1: Queue 0 enabled for DA-based DMA Channel Selection	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
10:8	Q1MDMACH[2:0]	Queue 1 Mapped to DMA Channel This field is valid when the Q1DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values might be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000b and 001b are valid, the other bits are reserved. This field controls the routing of the received packet in Queue 1 to the DMA channel: 0 0 0: DMA Channel 0 0 0 1: DMA Channel 1 0 1 0: DMA Channel 2 0 1 1: DMA Channel 3 1 0 0: DMA Channel 4 1 0 1: DMA Channel 5 1 1 0: DMA Channel 6 1 1 1: DMA Channel 7	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	Q1DDMACH	Queue 1 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 1 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 1 are routed to the DMA Channel programmed in the Q1MDMACH field (Bits [10:8]). 0: Queue 1 disabled for DA-based DMA Channel Selection 1: Queue 1 enabled for DA-based DMA Channel Selection	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
18:16	Q2MDMACH[2:0]	Queue 2 Mapped to DMA Channel This field is valid when the Q2DDMACH field is reset. This field controls the routing of the received packet in Queue 2 to the DMA channel: 0 0 0: DMA Channel 0 0 0 1: DMA Channel 1 0 1 0: DMA Channel 2 0 1 1: DMA Channel 3 1 0 0: DMA Channel 4 1 0 1: DMA Channel 5 1 1 0: DMA Channel 6 1 1 1: DMA Channel 7	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	Q2DDMACH	Queue 2 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 2 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 2 are routed to the DMA Channel programmed in the Q2MDMACH field (Bits [18:16]). 0: Queue 2 disabled for DA-based DMA Channel Selection 1: Queue 2 enabled for DA-based DMA Channel Selection	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
26:24	Q3MDMACH[2:0]	Queue 3 Mapped to DMA Channel This field is valid when the Q3DDMACH field is reset.*1 This field controls the routing of the received packet in Queue 3 to the DMA channel: 0 0 0: DMA Channel 0 0 0 1: DMA Channel 1 0 1 0: DMA Channel 2 0 1 1: DMA Channel 3 1 0 0: DMA Channel 4 1 0 1: DMA Channel 5 1 1 0: DMA Channel 6 1 1 1: DMA Channel 7	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	Q3DDMACH	Queue 3 Enabled for Dynamic (per packet) DMA Channel Selection When set, this bit indicates that the packets received in Queue 3 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 3 are routed to the DMA Channel programmed in the Q3MDMACH field (Bits [26:24]). 0: Queue 3 disabled for DA-based DMA Channel Selection 1: Queue 3 enabled for DA-based DMA Channel Selection	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The width of this field depends on the number of RX DMA channels and not all the values might be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000b and 001b are valid, the others are reserved.

28.3.176 MTL_RxQ_DMA_Map1 : Receive Queue and DMA Channel Mapping 1 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	Q7DD MACH	—	Q7MDMACH[2:0]			—	—	—	Q6DD MACH	—	Q6MDMACH[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	Q5DD MACH	—	Q5MDMACH[2:0]			—	—	—	Q4DD MACH	—	Q4MDMACH[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

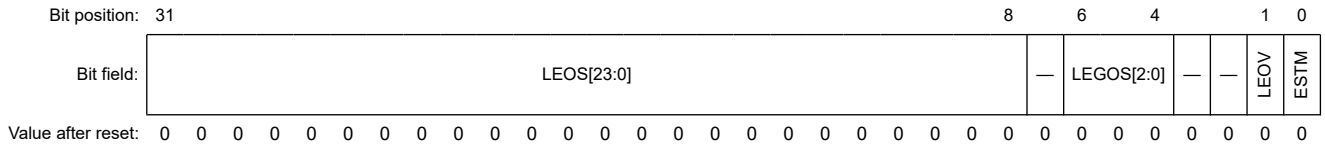
Bit	Symbol	Function	R/W
2:0	Q4MDMACH[2:0]	Queue 4 Mapped to DMA Channel This field is valid when the Q4DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values might be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000b and 001b are valid, the other bits are reserved. This field controls the routing of the packet received in Queue 4 to the DMA channel: 0 0 0: DMA Channel 0 0 0 1: DMA Channel 1 0 1 0: DMA Channel 2 0 1 1: DMA Channel 3 1 0 0: DMA Channel 4 1 0 1: DMA Channel 5 1 1 0: DMA Channel 6 1 1 1: DMA Channel 7	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	Q4DDMACH	Queue 4 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 4 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 4 are routed to the DMA Channel programmed in the Q4MDMACH field. 0: Queue 4 disabled for DA-based DMA Channel Selection 1: Queue 4 enabled for DA-based DMA Channel Selection	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
10:8	Q5MDMACH[2:0]	Queue 5 Mapped to DMA Channel This field is valid when the Q5DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values might be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000b and 001b are valid, the other bits are reserved. This field controls the routing of the packets received in Queue 5 to the DMA channel: 0 0 0: DMA Channel 0 0 0 1: DMA Channel 1 0 1 0: DMA Channel 2 0 1 1: DMA Channel 3 1 0 0: DMA Channel 4 1 0 1: DMA Channel 5 1 1 0: DMA Channel 6 1 1 1: DMA Channel 7	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	Q5DDMACH	Queue 5 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 5 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 5 are routed to the DMA Channel programmed in the Q5MDMACH field. 0: Queue 5 disabled for DA-based DMA Channel Selection 1: Queue 5 enabled for DA-based DMA Channel Selection	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
18:16	Q6MDMACH[2:0]	Queue 6 Mapped to DMA Channel This field is valid when the Q6DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values might be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000b and 001b are valid, the other bits are reserved. This field controls the routing of the packet received in Queue 6 to the DMA channel: 0 0 0: DMA Channel 0 0 0 1: DMA Channel 1 0 1 0: DMA Channel 2 0 1 1: DMA Channel 3 1 0 0: DMA Channel 4 1 0 1: DMA Channel 5 1 1 0: DMA Channel 6 1 1 1: DMA Channel 7	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	Q6DDMACH	Queue 6 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 6 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 6 are routed to the DMA Channel programmed in the Q6MDMACH field. 0: Queue 6 disabled for DA-based DMA Channel Selection 1: Queue 6 enabled for DA-based DMA Channel Selection	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
26:24	Q7MDMACH[2:0]	Queue 7 Mapped to DMA Channel This field is valid when the Q7DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values might be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000b and 001b are valid, the other bits are reserved. This field controls the routing of the packet received in Queue 7 to the DMA channel: 0 0 0: DMA Channel 0 0 0 1: DMA Channel 1 0 1 0: DMA Channel 2 0 1 1: DMA Channel 3 1 0 0: DMA Channel 4 1 0 1: DMA Channel 5 1 1 0: DMA Channel 6 1 1 1: DMA Channel 7	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	Q7DDMACH	Queue 7 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 7 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 7 are routed to the DMA Channel programmed in the Q7MDMACH field. 0: Queue 7 disabled for DA-based DMA Channel Selection 1: Queue 7 enabled for DA-based DMA Channel Selection	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

28.3.177 MTL_TBS_CTRL : Time Based Scheduling Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C40



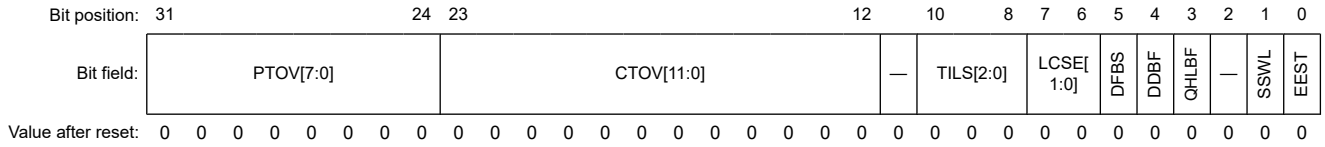
Bit	Symbol	Function	R/W
0	ESTM	EST offset Mode When this bit is set, the Launch Time value used in Time Based Scheduling is interpreted as an EST offset value and is added to the Base Time Register (BTR) of the current list. When reset, the Launch Time value is used as an absolute value that should be compared with the System time [39:8]. 0: EST offset Mode is disabled 1: EST offset Mode is enabled	R/W
1	LEOV	Launch Expiry Offset Valid When set indicates the LEOS field is valid. When not set, indicates the Launch Expiry Offset is not valid and the MTL must not check for Launch expiry time. 0: LEOS field is invalid 1: LEOS field is valid	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	LEGOS[2:0]	Launch Expiry GSN Offset The number GSN slots that has to be added to the Launch GSN to compute the Launch Expiry time. Value valid only when LEOV is set.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
31:8	LEOS[23:0]	Launch Expiry Offset The value in units of 256 nanoseconds that has to be added to the Launch time to compute the Launch Expiry time. Value valid only when LEOV is set. Max value: 999,999,999 ns, additionally should be smaller than CTR - 1 value when ESTM mode is set since this value is a modulo CTR value.	R/W

This register controls the operation of Time Based Scheduling.

28.3.178 MTL_EST_Control : EST Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C50



Bit	Symbol	Function	R/W
0	EEST	Enable EST When reset, the gate control list processing is halted and all gates are assumed to be in Open state. Should be set for the hardware to start processing the gate control lists. During the toggle from 0 to 1, the gate control list processing starts only after the SSWL bit it set. 0: EST is disabled 1: EST is enabled	R/W

Bit	Symbol	Function	R/W
1	SSWL	Switch to S/W owned list*1 When set indicates that the software has programmed that list that it currently owns (SWOL) and the IP must switch to the new list based on the new BTR. IP clears this bit when the switch to the SWOL happens to indicate the completion of the switch or when a BTR error (BTRE in Status register) is set. When BTRE is set, this bit is cleared but SWOL is not updated as the switch was not successful. 0: Switch to S/W owned list is disabled 1: Switch to S/W owned list is enabled	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	QHLBF	Quick Assertion of HLBF Error When set, Time Window for Head-of-Line blocking due to Frame Size Error is 1 to 2 loop count of GCL list. On reset, Time Window for Head-of-Line blocking due to Frame Size Error is 2 to 3 loop counts of GCL list. 0: Disable Quick assertion of HLBF Error 1: Quick Assertion of HLBF Error	R/W
4	DDBF	Do not Drop frames during Frame Size Error When set, frames are not be dropped during Head-of-Line blocking due to Frame Size Error (HLBF field of EST_Status register). 0: Drop frames during Frame Size Error 1: Do not Drop frames during Frame Size Error	R/W
5	DFBS	Drop Frames causing Scheduling Error When set frames reported to cause HOL Blocking due to not getting scheduled (HLBS field of EST_Status register) after 4, 8, 16, 32 (based on LCSE field of this register) GCL iterations are dropped. 0: Do not Drop Frames causing Scheduling Error 1: Drop Frames causing Scheduling Error	R/W
7:6	LCSE[1:0]	Loop Count to report Scheduling Error Programmable number of GCL list iterations before reporting an HLBS error defined in EST_Status register. 0 0: 4 iterations 0 1: 8 iterations 1 0: 16 iterations 1 1: 32 iterations	R/W
10:8	TILS[2:0]	Time Interval Left Shift Amount This field provides the left shift amount for the programmed Time Interval values used in the Gate Control Lists. 0 0 0: No left shift needed (equal to x1 ns) 0 0 1: Left shift TI by 1 bit (equal to x2 ns) 0 1 0: Left shift TI by 2 bits (equal to x4 ns) ⋮ 1 0 0: Left shift TI by 7 bits (equal to x128 ns) Others: Reserved	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
23:12	CTOV[11:0]	Current Time Offset Value Provides a 12 bit time offset value in nanosecond that is added to the current time to compensate for all the implementation pipeline delays such as the CDC sync delay, buffering delays, data path delays and so on. This offset helps to ensure that the impact of gate controls is visible on the line exactly at the pre-determined schedule (or as close to the schedule as possible).	R/W
31:24	PTOV[7:0]	PTP Time Offset Value The value of PTP Clock period multiplied by 6 in nanoseconds. This value is needed to avoid transmission overruns at the beginning of the installation of a new GCL.	R/W

Note 1. Access restrictions apply.

- MAC initiates switching to software owned list, when application writes 1.
- MAC writes 0 after switching to software owned list is complete.
- MAC ignores when application writes 0.

This register controls the operation of Enhancements to Scheduled Transmission (IEEE802.1Qbv).

28.3.179 MTL_EST_Ext_Control : EST Extended Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C54

Bit position:	31																								5	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVHD[5:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	OVHD[5:0]	Overhead Bytes Value This field indicates the fixed overhead for every packet to account for EST Scheduler Delay, IPG or EIPG, and Preamble bytes.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

This register indicates the number of Overhead bytes for EST related scheduling.

28.3.180 MTL_EST_Status : EST Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C58

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CGSN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BTRL[7:0]								SWOL	—	—	CGCE	HLBS	HLBF	BTRE	SWLC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SWLC	Switch to S/W owned list Complete[*1] When "1" indicates the hardware has successfully switched to the SWOL, and the SWOL bit has been updated to that effect. Cleared when the SSWL of EST_Control register transitions from 0 to 1, or on a software write. 0: Switch to S/W owned list Complete not detected 1: Switch to S/W owned list Complete detected	R/W
1	BTRE	BTR Error[*1] When "1" indicates a programming error in the BTR of SWOL where the programmed value is less than current time. If the BTRL = "0xFF", SWOL is not updated and software should reprogram the BTR to a value greater than current time and then set SSWL to reinitiate the switch to SWOL. Else if the value of BTRL < "0xFF", SWOL is updated and this field indicates the number of iterations (of + CycleTime) taken by hardware to update the BTR to a value greater than Current Time. 0: BTR Error not detected 1: BTR Error detected	R/W

Bit	Symbol	Function	R/W
2	HLBF	Head-Of-Line Blocking due to Frame Size Set when HOL Blocking is noticed on one or more Queues as a result of none of the Time Intervals of gate open in the GCL being greater than or equal to the duration needed for frame size (or frame fragment size when preemption is enabled) transmission. The one hot encoded Queue numbers that are experiencing HLBF are indicated in the MTL_EST_Frm_Size_Error register. Additionally, the first Queue number that experienced HLBF along with the frame size is captured in MTL_EST_Frm_Size_Capture register. Bit cleared when MTL_EST_Frame_Size_ Error register is all zeros. 0: Head-Of-Line Blocking due to Frame Size not detected 1: Head-Of-Line Blocking due to Frame Size detected	R
3	HLBS	Head-Of-Line Blocking due to Scheduling Set when the frame is not able to win arbitration and get scheduled even after 4 iterations of the GCL. Indicates to software a potential programming error. The one hot encoded values of the Queue Numbers that are not able to make progress are indicated in the MTL_EST_Sch_Error register. Bit cleared when MTL_EST_Sch_Error register is all zeros. 0: Head-Of-Line Blocking due to Scheduling not detected 1: Head-Of-Line Blocking due to Scheduling detected	R
4	CGCE	Constant Gate Control Error*1 This error occurs when the list length (LLR) is 1 and the Cycle Time (CTR) is less than or equal to the programmed Time Interval (TI) value after the optional Left Shifting. This implies, Gates are either always Closed or always Open based on the Gate Control values; the same effect can be achieved by other simpler (non TSN) programming mechanisms. Since the implementation does not support such a programming an error is reported. 0: Constant Gate Control Error not detected 1: Constant Gate Control Error detected	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SWOL	S/W owned list When "0" indicates Gate control list number "0" is owned by software and when "1" indicates the Gate Control list "1" is owned by the software. Any reads/writes by the software (using indirect access through GCL_Control) is directed to the list indicated by this value by default. The inverse of this value is treated as HWOL. R/W operations performed by hardware are directed to the list pointed by HWOL by default. 0: Gate control list number "0" is owned by software 1: Gate control list number "1" is owned by software	R
15:8	BTRL[7:0]	BTR Error Loop Count Provides the minimum count (N) for which the equation Current Time ≤ New BTR + (N × New Cycle Time) becomes true. N = "0xFF" indicates the iterations exceeded the value of 128 and the hardware was not able to update New BTR to be equal to or greater than Current Time. Software intervention is needed to update the New BTR. Value cleared when BTRE field of this register is cleared.	R
19:16	CGSN[3:0]	Current GCL Slot Number Indicates the slot number of the GCL list. Slot number is a modulo 16 count of the GCL List loops executed so far. Even if a new GCL list is installed, the count is incremental.	R
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

This register provides Status related to Enhancements to Scheduled Transmission (IEEE802.1Qbv).

28.3.181 MTL_EST_Sch_Error : EST Scheduing Error Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C60

Bit position:	31																								7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEQN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SEQN[7:0]	Schedule Error Queue Number*1 The One Hot Encoded Queue Numbers that have experienced error/timeout described in HLBS field of status register.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

This register provides the One Hot encoded Queue Numbers that are having the Scheduling related error (timeout).

28.3.182 MTL_EST_Frm_Size_Error : EST Frame Size Error Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C64

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FEQN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	FEQN[7:0]	Frame Size Error Queue Number*1 The One Hot Encoded Queue Numbers that have experienced error described in HLBF field of status register.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

This register provides the One Hot encoded Queue Numbers that are having the Frame Size related error.

28.3.183 MTL_EST_Frm_Size_Capture : EST Frame Size Error Capture Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	HBFQ[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	HBFS[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	HBFS[14:0]	Frame Size of HLBF Captures the Frame Size of the dropped frame related to queue number indicated in HBFQ field of this register. Contents of this register should be considered invalid, if this field is zero. Cleared when MTL_EST_Frm_Size_Error register is all zeros.	R
15	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
18:16	HBFQ[2:0]	Queue Number of HLBF Captures the binary value of the of the first Queue (number) experiencing HLBF error (see HLBF field of status register). Value once written is not altered by any subsequent queue errors of similar nature. Once cleared the queue number of the next occurring HLBF error is captured. Width is based on the number of Tx Queues configured; remaining bits are Read-Only. Cleared when MTL_EST_Frm_Size_Error register is all zeros.	R
31:19	—	These bits are read as 0.	R

This register captures the Frame Size and Queue Number of the first occurrence of the Frame Size related error. Up on clearing it captures the data of immediate next occurrence of a similar error.

28.3.184 MTL_EST_Intr_Enable : EST Interrupt Enable Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C70

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CGCE	IEHS	IEHF	IEBE	IECC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IECC	Interrupt Enable for Switch List When set, generates interrupt when the configuration change is successful and the hardware has switched to the new list. When reset this event does not generate an interrupt. 0: Interrupt for Switch List is disabled 1: Interrupt for Switch List is enabled	R/W
1	IEBE	Interrupt Enable for BTR Error When set, generates interrupt when the BTR Error occurs and is indicated in the status. When reset this event does not generate an interrupt. 0: Interrupt for BTR Error is disabled 1: Interrupt for BTR Error is enabled	R/W
2	IEHF	Interrupt Enable for HLBF When set, generates interrupt when the Head-of-Line Blocking due to Frame Size error occurs and is indicated in the status. When reset this event does not generate an interrupt. 0: Interrupt for HLBF is disabled 1: Interrupt for HLBF is enabled	R/W
3	IEHS	Interrupt Enable for HLBS When set, generates interrupt when the Head-of-Line Blocking due to Scheduling issue and is indicated in the status. When reset this event does not generate an interrupt. 0: Interrupt for HLBS is disabled 1: Interrupt for HLBS is enabled	R/W
4	CGCE	Interrupt Enable for CGCE When set, generates interrupt when the Constant Gate Control Error occurs and is indicated in the status. When reset this event does not generate an interrupt. 0: Interrupt for CGCE is disabled 1: Interrupt for CGCE is enabled	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

This register implements the Interrupt Enable bits for the various events that generate an interrupt. Bit positions have a 1 to 1 correlation with the status bit positions in MTL_ETS_Status register.

28.3.185 MTL_EST_GCL_Control : EST GCL Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADDR[7:0]								—	—	DBGB	DBGM	—	GCRR	R1W0	SRWO
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRWO	Start Read or Write Operation*1 ● 1: Indicates the start or progress of a read or write operation. ● MAC writes 0: Indicates that the read or write operation is complete or an error has occurred. ● Reads: Data can be read from MTL_EST_GCL_Data register when this field is 0. ● Writes: You must program the MTL_EST_GCL_Data register with the data before setting this bit. 0: Start Read/Write Op disabled 1: Start Read/Write Op enabled	R/W
1	R1W0	Read '1', Write '0': 0: Write Operation 1: Read Operation	R/W
2	GCRR	Gate Control Related Registers When set to "1" indicates the R/W access is for the GCL related registers (BTR, CTR, TER, LLR) whose address is provided by GCRA. When "0" indicates R/W should be directed to GCL from the address provided by GCLA. 0: Gate Control Related Registers are disabled 1: Gate Control Related Registers are enabled	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	DBGM	Debug Mode When set to "1" indicates R/W in debug mode where the memory bank (for GCL and Time related registers) is explicitly provided by DBGB value, when set to "0" SWOL bit is used to determine which bank to use. 0: Debug Mode is disabled 1: Debug Mode is enabled	R/W
5	DBGB	Debug Mode Bank Select When set to "0" indicates R/W in debug mode should be directed to Bank 0 (GCL0 and corresponding Time related registers). When set to "1" indicates R/W in debug mode should be directed to Bank 1 (GCL1 and corresponding Time related registers). This value is used when DBGM is set and overrides by value of SWOL which is normally used. 0: R/W in debug mode should be directed to Bank 0 1: R/W in debug mode should be directed to Bank 1	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	ADDR[7:0]	Gate Control List Address: (GCLA when GCRR is "0"). Provides the address (row number) of the Gate Control List at which the R/W operation has to be performed. By default the Gate Control List pointed by SWOL of MTL_EST_Status is selected for R/W, however if the DBGM bit of this register is set, a debug mode access is given to R/W from DBGB. Gate Control List Related Registers Address: (GCRA when GCRR is "1"). By default the GCL related register set pointed by SWOL of MTL_EST_Status is selected for R/W, however if the DBGM bit of this register is set, a debug mode access is given to R/W from DBGB. Lower 3 bits are only used in this mode, higher order bits are treated as do not cares. 0x00: BTR Low (31:0) 0x01: BTR High (63:31) 0x02: CTR Low (31:0) 0x03: CTR High (39:32) 0x04: TER (31:0) 0x05: LLR (8:0) Others: Reserved	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restrictions apply.

- MAC initiates the read or write operation when application writes 1.
- MAC writes 0 after the read or write operation is complete.
- MAC ignores when application writes 0.

This register provides the control information for reading/writing to the Gate Control lists.

28.3.186 MTL_EST_GCL_Data : EST GCL Data Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C84

Bit position:	31																													0
Bit field:	GCD[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	GCD[31:0]	Gate Control Data The data corresponding to the address selected in the GCL_Control register. Used for both Read and Write operations.	R/W

This register holds the read data or write data in case of reads and writes, respectively.

28.3.187 MTL_FPE_CTRL_STS : FPE Control and Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C90

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	HRS	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PEC[7:0]								—	—	—	—	—	—	AFSZ[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

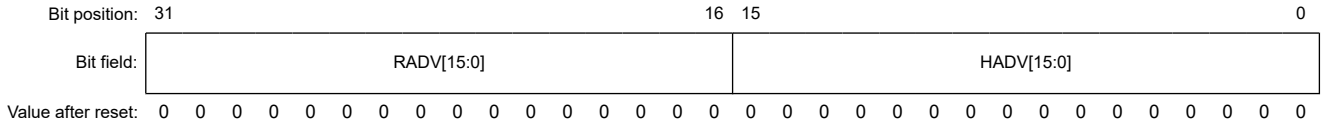
Bit	Symbol	Function	R/W
1:0	AFSZ[1:0]	Additional Fragment Size Used to indicate, in units of 64 bytes, the minimum number of bytes over 64 bytes required in non-final fragments of preempted frames. The minimum non-final fragment size is (AFSZ + 1) × 64 bytes.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PEC[7:0]	Preemption Classification When set indicates the corresponding Queue must be classified as preemptable, when '0' Queue is classified as express. When both EST (Qbv) and Preemption are enabled, Queue-0 is always assumed to be preemptable. When EST (Qbv) is enabled Queues categorized as preemptable here are always assumed to be in "Open" state in the Gate Control List.	R/W
27:16	—	These bits are read as 0. The write value should be 0.	R/W
28	HRS	Hold/Release Status 0: Indicates a Set-and-Release-MAC operation was last executed and the pMAC is in Release State 1: Indicates a Set-and-Hold-MAC operation was last executed and the pMAC is in Hold State	R
31:29	—	These bits are read as 0. The write value should be 0.	R/W

This register controls the operation of, and provides status for Frame Preemption (IEEE802.1Qbu/802.3br).

28.3.188 MTL_FPE_Advance : FPE Advance Time Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0C94



Bit	Symbol	Function	R/W
15:0	HADV[15:0]	Hold Advance The maximum time in nanoseconds that can elapse between issuing a HOLD to the MAC and the MAC ceasing to transmit any preemptable frame that is in the process of transmission or any preemptable frames that are queued for transmission.	R/W
31:16	RADV[15:0]	Release Advance The maximum time in nanoseconds that can elapse between issuing a RELEASE to the MAC and the MAC being ready to resume transmission of preemptable frames, in the absence of there being any express frames available for transmission.	R/W

This register holds the Hold and Release Advance time.

28.3.189 MTL_TXQ0_OPERATION_MODE : Queue 0 Transmit Operation Mode Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TQS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	TTC[2:0]			TXQEN[1:0]		TSF	FTQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FTQ	Flush Transmit Queue*1 When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TxQ1_Operation_Mode register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission.*2 0: Flush Transmit Queue is disabled 1: Flush Transmit Queue is enabled	R/W
1	TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits [6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0: Transmit Store and Forward is disabled 1: Transmit Store and Forward is enabled	R/W
3:2	TXQEN[1:0]	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. 0 0: Not enabled 0 1: Enable in AV mode 1 0: Enabled 1 1: Reserved	R/W
6:4	TTC[2:0]	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 0 0 0: 32 0 0 1: 64 0 1 0: 96 0 1 1: 128 1 0 0: 192 1 0 1: 256 1 1 0: 384 1 1 1: 512	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
20:16	TQS[4:0]	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. This means that value of 0x0 = 256 bytes, 0x1 = 512 bytes and so on. So, program TQS [4:0] = 0x0F to allocate queue size of 4096 (4K) bytes. In general, the size of the Queue = (TQS + 1) × 256 bytes.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restrictions apply.

- MAC initiates the Tx queue flush when application writes 1.

- MAC writes 0 after the Tx queue flush is complete.
- MAC ignores when application writes 0.

Note 2. The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock TX_CLK should be active.

Note 3. All the queues are disabled by default. Enable the Tx queue by programming this field.

The MTL_TXQ0_OPERATION_MODE register establishes the Transmit queue operating modes and commands.

28.3.190 MTL_TXQ0_UNDERFLOW : Queue 0 Underflow Counter Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	UFCNTOVF	UFFRMCNT[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	UFFRMCNT[10:0]	Underflow Packet Counter*1 This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read.	R
11	UFCNTOVF	Overflow Bit for Underflow Packet Counter*1 This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. 0: Overflow not detected for Underflow Packet Counter 1: Overflow detected for Underflow Packet Counter	R
31:12	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

The MTL_TXQ0_UNDERFLOW register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush.

28.3.191 MTL_TXQ0_DEBUG : Queue 0 Transmit Debug Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	STXSTSF[2:0]			—	PTXQ[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TXSTSFSTS	TXQSTS	TWCSTS	TRCSTS[1:0]		TXQPAUSED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: ● Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled. ● Reception of 802.3x Pause packet when PFC is disabled. 0: Transmit Queue in Pause status is not detected 1: Transmit Queue in Pause status is detected	R
2:1	TRCSTS[1:0]	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 0 0: Idle state 0 1: Read state (transferring data to the MAC transmitter) 1 0: Waiting for pending Tx Status from the MAC transmitter 1 1: Flushing the Tx queue because of the Packet Abort request from the MAC	R
3	TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0: MTL Tx Queue Write Controller status is not detected 1: MTL Tx Queue Write Controller status is detected	R
4	TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0: MTL Tx Queue Not Empty status is not detected 1: MTL Tx Queue Not Empty status is detected	R
5	TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0: MTL Tx Status FIFO Full status is not detected 1: MTL Tx Status FIFO Full status is detected	R
15:6	—	These bits are read as 0.	R
18:16	PTXQ[2:0]	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of packets in the Transmit queue.	R
19	—	This bit is read as 0.	R
22:20	STXSTSF[2:0]	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.	R
31:23	—	These bits are read as 0.	R

The MTL_TXQ0_DEBUG register gives the debug status of various blocks related to the Transmit queue.

28.3.192 MTL_TXQ0_ETS_STATUS : Queue 0 ETS Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D14

Bit position:	31								23																						0
Bit field:	—	—	—	—	—	—	—	—	ABS[23:0]																						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	ABS[23:0]	Average Bits per Slot This field contains the average transmitted bits per slot. When the DCB operation is enabled for Queue 0, this field is computed over every 10 million bit times slot (10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.	R
31:24	—	These bits are read as 0.	R

The MTL_TXQ0_ETS_STATUS register provides the average traffic transmitted in Queue 0.

28.3.193 MTL_TXQ0_QUANTUM_WEIGHT : Queue 0 Quantum or Weights Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D18

Bit position:	31											20																			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ISCQW[20:0]																			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
20:0	ISCQW[20:0]	Quantum or Weights When the DCB operation is enabled with DWRR algorithm for Queue 0 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. When DCB operation is enabled with WFQ algorithm for Queue 0 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits [20:14] must be written to zero. The higher the programmed weights lesser the bandwidth allocated for the particular Transmit Queue. This is because the weights are used to compute the packet finish time (weights × packet_size). Lesser the finish time, higher the probability of the packet getting scheduled first and using more bandwidth. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 0 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The MTL_TXQ0_QUANTUM_WEIGHT register contains the quantum value for Deficit Weighted Round Robin (DWRR), weights for the Weighted Round Robin (WRR), and Weighted Fair Queuing (WFQ) for Queue 0.

28.3.194 MTL_Q0_INTERRUPT_CONTROL_STATUS : Queue 0 Interrupt Control and Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	RXOIE	—	—	—	—	—	—	—	RXOVFIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ABPSIE	TXUIE	—	—	—	—	—	—	ABPSIS	TXUNFIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXUNFIS	Transmit Queue Underflow Interrupt Status*1 This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. 0: Transmit Queue Underflow Interrupt Status not detected 1: Transmit Queue Underflow Interrupt Status detected	R/W
1	ABPSIS	Average Bits Per Slot Interrupt Status*1 When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. 0: Average Bits Per Slot Interrupt Status not detected 1: Average Bits Per Slot Interrupt Status detected	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0: Transmit Queue Underflow Interrupt Status is disabled 1: Transmit Queue Underflow Interrupt Status is enabled	R/W
9	ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the GMAC_SBD interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0: Average Bits Per Slot Interrupt is disabled 1: Average Bits Per Slot Interrupt is enabled	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	RXOVFIS	Receive Queue Overflow Interrupt Status*1 This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. 0: Receive Queue Overflow Interrupt Status not detected 1: Receive Queue Overflow Interrupt Status detected	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0: Receive Queue Overflow Interrupt is disabled 1: Receive Queue Overflow Interrupt is enabled	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

This register contains the interrupt enable and status bits for the queue 0 interrupts.

28.3.195 MTL_RXQ0_OPERATION_MODE : Queue 0 Receive Operation Mode Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	RQS[4:0]					—	—	RFD[3:2]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFD[1:0]		—	—	RFA[3:0]				EHFC	DIS_TCP_EF	RSF	FEP	FUP	—	RTC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RTC[1:0]	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 0 0: 64 0 1: 32 1 0: 96 1 1: 128	R/W
2	—	These bits are read as 0. The write value should be 0.	R/W
3	FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01b. 0: Forward Undersized Good Packets is disabled 1: Forward Undersized Good Packets is enabled	R/W
4	FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet might be forwarded to the application or DMA. 0: Forward Error Packets is disabled 1: Forward Error Packets is enabled	R/W
5	RSF	Receive Queue Store and Forward When this bit is set, the GMAC reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0: Receive Queue Store and Forward is disabled 1: Receive Queue Store and Forward is enabled	R/W
6	DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0: Dropping of TCP/IP Checksum Error Packets is enabled 1: Dropping of TCP/IP Checksum Error Packets is disabled	R/W
7	EHFC	Enable Hardware Flow Control When this bit is set, the flow control operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0: Hardware Flow Control is disabled 1: Hardware Flow Control is enabled	R/W
11:8	RFA[3:0]	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.	R/W
13:12	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
17:14	RFD[3:0]	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation:*1 *2 0x0: Full minus 1 KB, that is, FULL 1 KB 0x1: Full minus 1.5 KB, that is, FULL 1.5 KB 0x2: Full minus 2 KB, that is, FULL 2 KB 0x3: Full minus 2.5 KB, that is, FULL 2.5 KB ⋮ 0xE: Full minus 8 KB, that is, FULL 8 KB 0xF: Full minus 8.5 KB, that is, FULL 8.5 KB	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
24:20	RQS[4:0]	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. This means that value of 0x0 = 256 bytes, 0x1 = 512 bytes and so on. So, program RQS [4:0] = 0x0F to allocate queue size of 4096 (4K) bytes. In general, the size of the Queue = (RQS + 1) × 256 bytes.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The de-assertion is effective only after flow control is asserted.
Note 2. The value must be programmed in such a way to make sure that the threshold is a positive number.
When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB.
For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register.

The MTL_RXQ0_OPERATION_MODE register establishes the Receive queue operating modes and command.

28.3.196 MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT : Queue 0 Missed Packet and Overflow Counter Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	MISCNTOVF	MISPKTCNT[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	OVFCNTOVF	OVFPKTCNT[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	OVFPKTCNT[10:0]	Overflow Packet Counter*1 This field indicates the number of packets discarded by the GMAC because of Receive queue overflow. This counter is incremented each time the GMAC discards an incoming packet because of overflow. This counter is reset when this register is read.	R
11	OVFCNTOVF	Overflow Counter Overflow Bit*1 When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. 0: Overflow Counter overflow not detected 1: Overflow Counter overflow detected	R
15:12	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
26:16	MISPKTCNT[10:0]	Missed Packet Counter*1 This field indicates the number of packets missed by the GMAC because the application asserted flush for this queue. This counter is reset when this register is read. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability.	R
27	MISCNTOVF	Missed Packet Counter Overflow Bit*1 When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. 0: Missed Packet Counter overflow not detected 1: Missed Packet Counter overflow detected	R
31:28	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

The MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

28.3.197 MTL_RXQ0_DEBUG : Queue 0 Receive Debug Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	PRXQ[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	RXQSTS[1:0]		—	RRCSTS[1:0]		RWCSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0: MTL Rx Queue Write Controller Active Status not detected 1: MTL Rx Queue Write Controller Active Status detected	R
2:1	RRCSTS[1:0]	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 0 0: Idle state 0 1: Reading packet data 1 0: Reading packet status (or timestamp) 1 1: Flushing the packet data and status	R
3	—	This bit is read as 0.	R
5:4	RXQSTS[1:0]	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 0 0: Rx Queue empty 0 1: Rx Queue fill-level below flow-control deactivate threshold 1 0: Rx Queue fill-level above flow-control activate threshold 1 1: Rx Queue full	R
15:6	—	These bits are read as 0.	R
29:16	PRXQ[13:0]	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256 KB / 16 B = 16 K Packets, that is, Max_Queue_Size/ Min_Packet_Size.	R
31:30	—	These bits are read as 0.	R

The MTL_RXQ0_DEBUG register gives the debug status of various blocks related to the Receive queue.

28.3.198 MTL_RXQ0_CONTROL : Queue 0 Receive Control Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	RXQ_FRM_ARBIT	RXQ_WEGT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RXQ_WEGT[2:0]	Receive Queue Weight*1 This field indicates the weight assigned to the Rx Queue 0. This field needs to be programmed with one value less than the required weight, that is, reset value of 0 indicates weight of 1, value of 1 indicates weight of 2, and so on. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.	R/W
3	RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the GMAC drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the GMAC drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: ● PBL amount of data or ● Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the GMAC drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0: Receive Queue Packet Arbitration is disabled 1: Receive Queue Packet Arbitration is enabled	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The change in value of RXQ_WEGT takes effect only after the completion of current service round or when there is change from RAA = SP to RAA = WSP algorithm. This approach is taken so that there is smooth transition. For the RXQ_WEGT value to take effect at the start, the MTL_RXQ0_CONTROL registers must be programmed before the MTL_Operation_Mode register.

The MTL_RXQ0_CONTROL register controls the receive arbitration and passing of received packets to the application.

28.3.199 MTL_TXQn_OPERATION_MODE : Queue n Transmit Operation Mode Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D40 + 0x40 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TQS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	TTC[2:0]			TXQEN[1:0]		TSF	FTQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FTQ	Flush Transmit Queue*1 When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ1_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. **Note**: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock TX_CLK should be active. 0: Flush Transmit Queue is disabled 1: Flush Transmit Queue is enabled	R/W
1	TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits [6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0: Transmit Store and Forward is disabled 1: Transmit Store and Forward is enabled	R/W
3:2	TXQEN[1:0]	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. All the queues are disabled by default. Enable the Tx queue by programming this field. 0 0: Not enabled 0 1: Enable in AV mode 1 0: Enabled 1 1: Reserved	R/W
6:4	TTC[2:0]	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 0 0 0: 32 0 0 1: 64 0 1 0: 96 0 1 1: 128 1 0 0: 192 1 0 1: 256 1 1 0: 384 1 1 1: 512	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20:16	TQS[4:0]	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. This means that value of 0x0 = 256 bytes, 0x1 = 512 bytes and so on. So, program TQS [4:0] = 0x0F to allocate queue size of 4096 (4K) bytes. In general, the size of the Queue = (TQS + 1) × 256 bytes.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restrictions apply.

- MAC initiates the Tx queue flush when application writes 1.
- MAC writes 0 after the Tx queue flush is complete.
- MAC ignores when application writes 0.

The MTL_TXQn_OPERATION_MODE register establishes the Transmit queue operating modes and commands.

28.3.200 MTL_TXQn_UNDERFLOW : Queue n Underflow Counter Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D44 + 0x40 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	UFCNTOVF	UFFRMCNT[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	UFFRMCNT[10:0]	Underflow Packet Counter*1 This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read.	R
11	UFCNTOVF	Overflow Bit for Underflow Packet Counter*1 This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. 0: Overflow not detected for Underflow Packet Counter 1: Overflow detected for Underflow Packet Counter	R
31:12	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

The MTL_TXQn_UNDERFLOW register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush.

28.3.201 MTL_TXQn_DEBUG : Queue n Transmit Debug Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D48 + 0x40 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	STXSTSF[2:0]			—	PTXQ[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TXSTSFSTS	TXQSTS	TWCSTS	TRCSTS[1:0]		TXQPAUSED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: ● Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled. ● Reception of 802.3x Pause packet when PFC is disabled. 0: Transmit Queue in Pause status is not detected 1: Transmit Queue in Pause status is detected	R
2:1	TRCSTS[1:0]	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 0 0: Idle state 0 1: Read state (transferring data to the MAC transmitter) 1 0: Waiting for pending Tx Status from the MAC transmitter 1 1: Flushing the Tx queue because of the Packet Abort request from the MAC	R
3	TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0: MTL Tx Queue Write Controller status is not detected 1: MTL Tx Queue Write Controller status is detected	R
4	TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0: MTL Tx Queue Not Empty status is not detected 1: MTL Tx Queue Not Empty status is detected	R
5	TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0: MTL Tx Status FIFO Full status is not detected 1: MTL Tx Status FIFO Full status is detected	R
15:6	—	These bits are read as 0.	R
18:16	PTXQ[2:0]	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of packets in the Transmit queue.	R
19	—	This bit is read as 0.	R
22:20	STXSTSF[2:0]	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.	R
31:23	—	These bits are read as 0.	R

The MTL_TXQn_DEBUG register gives the debug status of various blocks related to the Transmit queue.

28.3.202 MTL_TXQn_ETS_CONTROL : Queue n ETS Control Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D50 + 0x40 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	SLC[2:0]			CC	AVALG	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	AVALG	AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. When reset, strict priority is selected. 0: CBS Algorithm is disabled 1: CBS Algorithm is enabled	R/W
3	CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. The credit accumulates even when there is no packet waiting in Channel 1 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 1. When there is no packet waiting in Channel 1 and other channel is transmitting, no credit is accumulated. 0: Credit Control is disabled 1: Credit Control is enabled	R/W
6:4	SLC[2:0]	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots over which the average transmitted bits per slot, provided in the MTL_TXQn_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 0 0 0: 1 slot 0 0 1: 2 slots 0 1 0: 4 slots 0 1 1: 8 slots 1 0 0: 16 slots Others: Reserved	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The MTL_TXQn_ETS_CONTROL register controls the enhanced transmission selection operation.

28.3.203 MTL_TXQn_ETS_STATUS : Queue n ETS Status Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D54 + 0x40 × (n - 1)

Bit position:	31								23																						0
Bit field:	—	—	—	—	—	—	—	—	ABS[23:0]																							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	ABS[23:0]	Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQn_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.	R
31:24	—	These bits are read as 0.	R

The MTL_TXQn_ETS_STATUS register provides the average traffic transmitted in Queue 1.

28.3.204 MTL_TXQn_QUANTUM_WEIGHT : Queue n Quantum or Weights Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D58 + 0x40 × (n - 1)

Bit position:	31											20																			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ISCQW[20:0]																			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
20:0	ISCQW[20:0]	idleSlopeCredit, Quantum, or Weights*1 *2 *3 **idleSlopeCredit** When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000 Mbps mode and 0x1000 in 100 Mbps mode. Bits [20:14] must be written to zero. **Quantum** When the DCB operation is enabled with DWRR algorithm for Queue 1 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. **Weights** When DCB operation is enabled with WFQ algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits [20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is MTL_Operation_Mode register.

Note 2. For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula:
(previous_finish_time of particular Transmit Queue + (weights × packet_size))

Note 3. The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

The MTL_TXQn_QUANTUM_WEIGHT register provides the average traffic transmitted in Queue 1.

28.3.205 MTL_TXQn_SENDSLOPECREDIT : Queue n CBS SendSlope Credit Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D5C + 0x40 × (n - 1)

Bit position:	31														13														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSC[13:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	SSC[13:0]	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns, 8 ns for 100 Mbps, 1000 Mbps, respectively) when the credit is decreasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 1 is selected for transmission.	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The MTL_TXQn_SENDSLOPECREDIT register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

28.3.206 MTL_TXQn_HICREDIT : Queue n CBS Hi Credit Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D60 + 0x40 × (n - 1)

Bit position:	31			28																									0
Bit field:	—	—	—	HC[28:0]																									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	HC[28:0]	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 × 1,024 = 134,217,728 or 0x0800_0000.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

The MTL_TXQn_HICREDIT register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

28.3.207 MTL_TXQn_LOCREDIT : Queue n CBS Lo Credit Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D64 + 0x40 × (n - 1)

Bit position:	31			28																										0
Bit field:	—	—	—	LC[28:0]																										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	LC[28:0]	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192 × 2) × 8 × 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

The MTL_TXQn_LOCREDIT register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

28.3.208 MTL_Qn_INTERRUPT_CONTROL_STATUS : Queue n Interrupt Control and Status Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D6C + 0x40 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	RXOIE	—	—	—	—	—	—	—	RXOVFIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ABPSIE	TXUIE	—	—	—	—	—	—	ABPSIS	TXUNFIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXUNFIS	Transmit Queue Underflow Interrupt Status*1 This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. 0: Transmit Queue Underflow Interrupt Status not detected 1: Transmit Queue Underflow Interrupt Status detected	R/W
1	ABPSIS	Average Bits Per Slot Interrupt Status*1 When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. 0: Average Bits Per Slot Interrupt Status not detected 1: Average Bits Per Slot Interrupt Status detected	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0: Transmit Queue Underflow Interrupt Status is disabled 1: Transmit Queue Underflow Interrupt Status is enabled	R/W
9	ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the GMAC_SBD interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0: Average Bits Per Slot Interrupt is disabled 1: Average Bits Per Slot Interrupt is enabled	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	RXOVFIS	Receive Queue Overflow Interrupt Status*1 This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. 0: Receive Queue Overflow Interrupt Status not detected 1: Receive Queue Overflow Interrupt Status detected	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0: Receive Queue Overflow Interrupt is disabled 1: Receive Queue Overflow Interrupt is enabled	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

This register contains the interrupt enable and status bits for the queue 1 interrupts.

28.3.209 MTL_RXQn_OPERATION_MODE : Queue n Receive Operation Mode Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D70 + 0x40 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	RQS[4:0]					—	—	RFD[3:2]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFD[1:0]		—	—	RFA[3:0]				EHFC	DIS_TCP_EF	RSF	FEP	FUP	—	RTC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RTC[1:0]	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 0 0: 64 0 1: 32 1 0: 96 1 1: 128	R/W

Bit	Symbol	Function	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01b. 0: Forward Undersized Good Packets is disabled 1: Forward Undersized Good Packets is enabled	R/W
4	FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet might be forwarded to the application or DMA. 0: Forward Error Packets is disabled 1: Forward Error Packets is enabled	R/W
5	RSF	Receive Queue Store and Forward When this bit is set, the GMAC reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0: Receive Queue Store and Forward is disabled 1: Receive Queue Store and Forward is enabled	R/W
6	DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0: Dropping of TCP/IP Checksum Error Packets is enabled 1: Dropping of TCP/IP Checksum Error Packets is disabled	R/W
7	EHFC	Enable Hardware Flow Control When this bit is set, the flow control operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0: Hardware Flow Control is disabled 1: Hardware Flow Control is enabled	R/W
11:8	RFA[3:0]	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.	R/W
13:12	—	These bits are read as 0. The write value should be 0.	R/W
17:14	RFD[3:0]	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes)*1 *2 These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: 0x0: Full minus 1 KB, that is, FULL 1 KB 0x1: Full minus 1.5 KB, that is, FULL 1.5 KB 0x2: Full minus 2 KB, that is, FULL 2 KB 0x3: Full minus 2.5 KB, that is, FULL 2.5 KB ⋮ 0xE: Full minus 8 KB, that is, FULL 8 KB 0xF: Full minus 8.5 KB, that is, FULL 8.5 KB	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
24:20	RQS[4:0]	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. This means that value of 0x0 = 256 bytes, 0x1 = 512 bytes and so on. So, program RQS [4:0] = 0x0F to allocate queue size of 4096 (4K) bytes. In general, the size of the Queue = (RQS + 1) × 256 bytes.	R/W

Bit	Symbol	Function	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The de-assertion is effective only after flow control is asserted.
Note 2. The value must be programmed in such a way to make sure that the threshold is a positive number.
When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB.
For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register.

The MTL_RXQn_OPERATION_MODE register establishes the Receive queue operating modes and command.

28.3.210 MTL_RXQn_MISSED_PACKET_OVERFLOW_CNT : Queue n Missed Packet and Overflow Counter Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D74 + 0x40 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	MISCNTOVF	MISPKTCNT[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	OVFCNTOVF	OVFPKTCNT[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	OVFPKTCNT[10:0]	Overflow Packet Counter*1 This field indicates the number of packets discarded by the GMAC because of Receive queue overflow. This counter is incremented each time the GMAC discards an incoming packet because of overflow. This counter is reset when this register is read with.	R
11	OVFCNTOVF	Overflow Counter Overflow Bit*1 When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. 0: Overflow Counter overflow not detected 1: Overflow Counter overflow detected	R
15:12	—	These bits are read as 0.	R
26:16	MISPKTCNT[10:0]	Missed Packet Counter*1 This field indicates the number of packets missed by the GMAC because the application asserted flush for this queue. This counter is reset when this register is read with. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability.	R
27	MISCNTOVF	Missed Packet Counter Overflow Bit*1 When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. 0: Missed Packet Counter overflow not detected 1: Missed Packet Counter overflow detected	R
31:28	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

The MTL_RXQn_MISSED_PACKET_OVERFLOW_CNT register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

28.3.211 MTL_RXQn_DEBUG : Queue n Receive Debug Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D78 + 0x40 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	PRXQ[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	RXQSTS[1:0]		—	RRCSTS[1:0]		RWCSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0: MTL Rx Queue Write Controller Active Status not detected 1: MTL Rx Queue Write Controller Active Status detected	R
2:1	RRCSTS[1:0]	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 0 0: Idle state 0 1: Reading packet data 1 0: Reading packet status (or timestamp) 1 1: Flushing the packet data and status	R
3	—	This bit is read as 0.	R
5:4	RXQSTS[1:0]	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 0 0: Rx Queue empty 0 1: Rx Queue fill-level below flow-control deactivate threshold 1 0: Rx Queue fill-level above flow-control activate threshold 1 1: Rx Queue full	R
15:6	—	These bits are read as 0.	R
29:16	PRXQ[13:0]	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256 KB / 16 B = 16 K Packets, that is, Max_Queue_Size/ Min_Packet_Size.	R
31:30	—	These bits are read as 0.	R

The MTL_RXQn_DEBUG register gives the debug status of various blocks related to the Receive queue.

28.3.212 MTL_RXQn_CONTROL : Queue n Receive Control Register (n = 1 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x0D7C + 0x40 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	RXQ_FRM_ARBIT	RXQ_WEGT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RXQ_WEGT[2:0]	Receive Queue Weight*1 This field indicates the weight assigned to the Rx Queue 0. This field needs to be programmed with one value less than the required weight, that is, reset value of 0 indicates weight of 1, value of 1 indicates weight of 2, and so on. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.	R/W
3	RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the GMAC drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the GMAC drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: ● PBL amount of data or ● Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the GMAC drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0: Receive Queue Packet Arbitration is disabled 1: Receive Queue Packet Arbitration is enabled	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The change in value of RXQ_WEGT takes effect only after the completion of current service round or when there is change from RAA = SP to RAA = WSP algorithm. This approach is taken so that there is smooth transition. For the RXQ_WEGT value to take effect at the start, the MTL_RXQn_CONTROL registers must be programmed before the MTL_Operation_Mode register.

The MTL_RXQn_CONTROL register controls the receive arbitration and passing of received packets to the application.

28.3.213 DMA_Mode : Bus Mode Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INTM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	TXPR	—	—	DSPW	—	—	—	TAA[2:0]			—	SWR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SWR	Software Reset*1 After writing 1 to this bit, wait for at the least 4 CSR clock cycles, before reading this bit. **Note**: The reset operation is complete only when GMAC has de-asserted all resets in all active clock domains. Therefore, it is essential that all PHY input clocks of the selected PHY interface are present for the completion of the software reset. The time required to complete the software reset operation depends on the frequency of the slowest active clock. 0: Software Reset is disabled. 1: Software Reset is enabled. Indicates that MAC and the DMA controller reset the logic and all internal registers of the DMA, MTL, and MAC. This bit is automatically cleared after the reset operation is complete in all GMAC clock domains. Before reprogramming any GMAC register, read a value of zero in this bit.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
4:2	TAA[2:0]	Transmit Arbitration Algorithm This field is used to select the arbitration policy for the Transmit side when you select multiple Tx DMAs. 0 0 0: Fixed priority (Channel 0 has the lowest priority and the last channel has the highest priority) 0 0 1: Weighted Strict Priority (WSP) 0 1 0: Weighted Round-Robin (WRR) Others: Reserved	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	DSPW	Descriptor Posted Write 0: Descriptor Posted Write is disabled. Indicates that the descriptor writes are non-posted. 1: Descriptor Posted Write is enabled. Indicates that descriptor writes are non-posted only when IOC = 1 (Interrupt on completion) in the last descriptor. Otherwise, the descriptor writes are always posted.	R/W
10:9	—	These bits are read as 0. The write value should be 0.	R/W
11	TXPR	Transmit Priority 0: Transmit Priority is disabled. 1: Transmit Priority is enabled. Indicates that the Tx DMA has priority over the Rx DMA during arbitration for the system-side bus. When you enable descriptor cache feature, descriptor reads from DCACHE memory.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
17:16	INTM[1:0]	Interrupt Mode This field defines the interrupt mode of GMAC. The behavior of the following outputs change depending on the following settings: ● GMAC_TXINTn (Transmit Per Channel Interrupt) ● GMAC_RXINTn (Receive Per Channel Interrupt) ● GMAC_SBD (Common Interrupt) It also changes the behavior of the RI/TI bits in the DMA_CH0_STATUS. 0 0: GMAC_TXINTn/GMAC_RXINTn are pulse signals for each TX/RX packet transfer completion events (irrespective of whether corresponding interrupts are enabled) for which IOC bits are enabled in descriptor. GMAC_SBD is also asserted when corresponding interrupts are enabled and cleared only when software clears the corresponding RI/TI status bits. 0 1: GMAC_TXINTn/GMAC_RXINTn are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. The GMAC_SBD is not asserted for these TX/RX packet transfer completion events. 1 0: GMAC_TXINTn/GMAC_RXINTn are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. However, the signal is asserted again if the same event occurred again before it was cleared. The GMAC_SBD is not asserted for these TX/RX packet transfer completion events. 1 1: Reserved	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restrictions apply.

- Initiates the software reset when application writes 1.
- MAC writes 0 after the software reset is complete.
- MAC ignores when application writes 0.

The DMA_Mode register establishes the bus operating modes for the DMA.

28.3.214 DMA_SysBus_Mode : System Bus Mode Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EN_LPI	LPI_XIT_PKT	—	—	WR_OSR_LMT[3:0]				—	—	—	—	RD_OSR_LMT[3:0]			
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	ONEKBBE	AAL	—	AALE	—	—	—	—	—	—	BLEN16	BLEN8	BLEN4	FB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FB	Fixed Burst Length 0: Fixed Burst Length is disabled. The AXI master initiates burst transfers that are equal to or less than the maximum allowed burst length programmed in TxPBL/RxPBL. 1: Fixed Burst Length is enabled. The AXI master initiates burst transfers of specified lengths: ● Burst transfers of fixed burst lengths as indicated by the BLEN16, BLEN8, or BLEN4 field ● Burst transfers of length 1	R/W
1	BLEN4	AXI Burst Length 4*1 When this bit is set to 1 or the FB bit is set to 1, the AXI master can select a burst length of 4 on the AXI interface. 0: No effect 1: AXI Burst Length 4	R/W
2	BLEN8	AXI Burst Length 8*1 When this bit is set to 1 or the FB bit is set to 1, the AXI master can select a burst length of 8 on the AXI interface. 0: No effect 1: AXI Burst Length 8	R/W
3	BLEN16	AXI Burst Length 16*1 When this bit is set to 1 or the FB bit is set to 1, the AXI master can select a burst length of 16 on the AXI interface. 0: No effect 1: AXI Burst Length 16	R/W
9:4	—	These bits are read as 0. The write value should be 0.	R/W
10	AALE	Automatic AXI LPI enable 0: Automatic AXI LPI is disabled. 1: Automatic AXI LPI is enabled. Enables the AXI master to enter into LPI state when there is no activity in the GMAC for certain number of system clock cycles. You can program the number of clock cycles in the LPIEI field of AXI_LPI_Entry_Interval register.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	AAL	Address-Aligned Beats 0: Address-Aligned Beats is disabled. The AXI master performs burst transfers on Read and Write channels without aligning to address boundaries. 1: Address-Aligned Beats is enabled. The AXI master performs address-aligned burst transfers on Read and Write channels.	R/W
13	ONEKBBE	1 KB Boundary Crossing Enable for the AXI Master 0: 1 KB Boundary Crossing for the AXI Master Beats is disabled. The burst transfers performed by the AXI master do not cross 4 KB boundary. 1: 1 KB Boundary Crossing for the AXI Master Beats is enabled. The burst transfers performed by the AXI master do not cross 1 KB boundary.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RD_OSR_LMT[3:0]	AXI Maximum Read Outstanding Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT + 1	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	WR_OSR_LMT[3:0]	AXI Maximum Write Outstanding Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT + 1	R/W
29:28	—	These bits are read as 0. The write value should be 0.	R/W
30	LPI_XIT_PKT	Unlock on Magic Packet or Remote Wake-Up Packet 0: Unlock on Magic Packet or Remote Wake-Up Packet is disabled. Enables the AXI master to come out of the LPI mode when any packet is received. 1: Unlock on Magic Packet or Remote Wake-Up Packet is enabled. Enables the AXI master to come out of the LPI mode only when the magic packet or remote wake-up packet is received.	R/W
31	EN_LPI	Enable Low Power Interface (LPI) 0: Low Power Interface (LPI) is disabled. Disables the LPI mode. Denies the LPI request from the AXI system clock controller. 1: Low Power Interface (LPI) is enabled. Enables the LPI mode and accepts the LPI request from the AXI system clock controller.	R/W

Note 1. When the FB bit is set to 0, setting this bit has no effect.

This register controls the behavior of the AHB or AXI master. It mainly controls burst splitting and the number of outstanding requests.

28.3.215 DMA_Interrupt_Status : DMA Interrupt Status Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MACIS	MTLIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DC7IS	DC6IS	DC5IS	DC4IS	DC3IS	DC2IS	DC1IS	DC0IS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DC0IS	DMA Channel 0 Interrupt Status This bit indicates an interrupt event in DMA Channel 0. To reset this bit to 0, the software must read the corresponding register in DMA Channel 0 to get the exact cause of the interrupt and clear its source. 0: DMA Channel 0 Interrupt Status not detected 1: DMA Channel 0 Interrupt Status detected	R
1	DC1IS	DMA Channel 1 Interrupt Status This bit indicates an interrupt event in DMA Channel 1. To reset this bit to 0, the software must read the corresponding register in DMA Channel 1 to get the exact cause of the interrupt and clear its source. 0: DMA Channel 1 Interrupt Status not detected 1: DMA Channel 1 Interrupt Status detected	R
2	DC2IS	DMA Channel 2 Interrupt Status This bit indicates an interrupt event in DMA Channel 2. To reset this bit to 0, the software must read the corresponding register in DMA Channel 2 to get the exact cause of the interrupt and clear its source. 0: DMA Channel 2 Interrupt Status not detected 1: DMA Channel 2 Interrupt Status detected	R
3	DC3IS	DMA Channel 3 Interrupt Status This bit indicates an interrupt event in DMA Channel 3. To reset this bit to 0, the software must read the corresponding register in DMA Channel 3 to get the exact cause of the interrupt and clear its source. 0: DMA Channel 3 Interrupt Status not detected 1: DMA Channel 3 Interrupt Status detected	R
4	DC4IS	DMA Channel 4 Interrupt Status This bit indicates an interrupt event in DMA Channel 4. To reset this bit to 0, the software must read the corresponding register in DMA Channel 4 to get the exact cause of the interrupt and clear its source. 0: DMA Channel 4 Interrupt Status not detected 1: DMA Channel 4 Interrupt Status detected	R
5	DC5IS	DMA Channel 5 Interrupt Status This bit indicates an interrupt event in DMA Channel 5. To reset this bit to 0, the application must read the corresponding register in DMA Channel 5, get the exact cause of the interrupt and clear the source of the interrupt. 0: DMA Channel 5 Interrupt Status not detected 1: DMA Channel 5 Interrupt Status detected	R
6	DC6IS	DMA Channel 6 Interrupt Status This bit indicates an interrupt event in DMA Channel 6. To reset this bit to 0, the application must read the corresponding register in DMA Channel 6, get the exact cause of the interrupt and clear the source of the interrupt. 0: DMA Channel 6 Interrupt Status not detected 1: DMA Channel 6 Interrupt Status detected	R
7	DC7IS	DMA Channel 7 Interrupt Status This bit indicates an interrupt event in DMA Channel 7. To reset this bit to 0, the application must read the corresponding register in DMA Channel 7, get the exact cause of the interrupt and clear the source of the interrupt. 0: DMA Channel 7 Interrupt Status not detected 1: DMA Channel 7 Interrupt Status detected	R
15:8	—	These bits are read as 0.	R
16	MTLIS	MTL Interrupt Status This bit indicates an interrupt event in the MTL. To reset this bit to 0, the application must read the corresponding register in the MTL, get the exact cause of the interrupt and clear the source of the interrupt. 0: MTL Interrupt Status not detected 1: MTL Interrupt Status detected	R
17	MACIS	MAC Interrupt Status This bit indicates an interrupt event in the MAC. To reset this bit to 0, the application must read the corresponding register in the MAC, get the exact cause of the interrupt and clear the source of the interrupt. 0: MAC Interrupt Status not detected 1: MAC Interrupt Status detected	R

Bit	Symbol	Function	R/W
31:18	—	These bits are read as 0.	R

The application reads this DMA_Interrupt_Status register during interrupt service routine or polling to determine the interrupt status of DMA channels, MTL queues, and the MAC.

28.3.216 DMA_Debug_Status0 : DMA Debug Status 0 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x100C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TPS2[3:0]				RPS2[3:0]				TPS1[3:0]				RPS1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TPS0[3:0]				RPS0[3:0]				—	—	—	—	—	—	AXRH STS	AXWH STS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AXWHSTS	AXI Master Write Channel When high, this bit indicates that the write channel of the AXI master is active, and it is transferring data. 0: AXI Master Write Channel or AHB Master Status not detected 1: AXI Master Write Channel or AHB Master Status detected	R
1	AXRHSTS	AXI Master Read Channel Status 0: AXI Master Read Channel Status not detected. 1: AXI Master Read Channel Status detected. Indicates that the read channel of the AXI master is active, and it is transferring the data.	R
7:2	—	These bits are read as 0.	R
11:8	RPS0[3:0]	DMA Channel 0 Receive Process State This field indicates the Rx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Receive Command issued) 0x1: Running (Fetching Rx Transfer Descriptor) 0x2: Reserved for future use 0x3: Running (Waiting for Rx packet) 0x4: Suspended (Rx Descriptor Unavailable) 0x5: Running (Closing the Rx Descriptor) 0x6: Timestamp write state 0x7: Running (Transferring the received packet data from the Rx buffer to the system memory)	R
15:12	TPS0[3:0]	DMA Channel 0 Transmit Process State This field indicates the Tx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Transmit Command issued) 0x1: Running (Fetching Tx Transfer Descriptor) 0x2: Running (Waiting for status) 0x3: Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4: Timestamp write state 0x5: Reserved for future use 0x6: Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7: Running (Closing Tx Descriptor)	R

Bit	Symbol	Function	R/W
19:16	RPS1[3:0]	DMA Channel 1 Receive Process State This field indicates the Rx DMA FSM state for Channel 1. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Receive Command issued) 0x1: Running (Fetching Rx Transfer Descriptor) 0x2: Reserved for future use 0x3: Running (Waiting for Rx packet) 0x4: Suspended (Rx Descriptor Unavailable) 0x5: Running (Closing the Rx Descriptor) 0x6: Timestamp write state 0x7: Running (Transferring the received packet data from the Rx buffer to the system memory)	R
23:20	TPS1[3:0]	DMA Channel 1 Transmit Process State This field indicates the Tx DMA FSM state for Channel 1. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Transmit Command issued) 0x1: Running (Fetching Tx Transfer Descriptor) 0x2: Running (Waiting for status) 0x3: Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4: Timestamp write state 0x5: Reserved for future use 0x6: Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7: Running (Closing Tx Descriptor)	R
27:24	RPS2[3:0]	DMA Channel 2 Receive Process State This field indicates the Rx DMA FSM state for Channel 2. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Receive Command issued) 0x1: Running (Fetching Rx Transfer Descriptor) 0x2: Reserved for future use 0x3: Running (Waiting for Rx packet) 0x4: Suspended (Rx Descriptor Unavailable) 0x5: Running (Closing the Rx Descriptor) 0x6: Timestamp write state 0x7: Running (Transferring the received packet data from the Rx buffer to the system memory)	R
31:28	TPS2[3:0]	DMA Channel 2 Transmit Process State This field indicates the Tx DMA FSM state for Channel 2. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Transmit Command issued) 0x1: Running (Fetching Tx Transfer Descriptor) 0x2: Running (Waiting for status) 0x3: Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4: Timestamp write state 0x5: Reserved for future use 0x6: Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7: Running (Closing Tx Descriptor)	R

The DMA_Debug_Status0 register gives the Receive and Transmit process status for DMA Channel 0 to Channel 2 for debugging purpose.

28.3.217 DMA_Debug_Status1 : DMA Debug Status 1 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1010

Bit position:	31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
Bit field:	TPS6[3:0]	RPS6[3:0]	TPS5[3:0]	RPS5[3:0]	TPS4[3:0]	RPS4[3:0]	TPS3[3:0]	RPS3[3:0]
Value after reset:	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bit	Symbol	Function	R/W
3:0	RPS3[3:0]	DMA Channel 3 Receive Process State This field indicates the Rx DMA FSM state for Channel 3. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Receive Command issued) 0x1: Running (Fetching Rx Transfer Descriptor) 0x2: Reserved for future use 0x3: Running (Waiting for Rx packet) 0x4: Suspended (Rx Descriptor Unavailable) 0x5: Running (Closing the Rx Descriptor) 0x6: Timestamp write state 0x7: Running (Transferring the received packet data from the Rx buffer to the system memory)	R
7:4	TPS3[3:0]	DMA Channel 3 Transmit Process State This field indicates the Tx DMA FSM state for Channel 3. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Transmit Command issued) 0x1: Running (Fetching Tx Transfer Descriptor) 0x2: Running (Waiting for status) 0x3: Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4: Timestamp write state 0x5: Reserved for future use 0x6: Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7: Running (Closing Tx Descriptor)	R
11:8	RPS4[3:0]	DMA Channel 4 Receive Process State This field indicates the Rx DMA FSM state for Channel 4. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Receive Command issued) 0x1: Running (Fetching Rx Transfer Descriptor) 0x2: Reserved for future use 0x3: Running (Waiting for Rx packet) 0x4: Suspended (Rx Descriptor Unavailable) 0x5: Running (Closing the Rx Descriptor) 0x6: Timestamp write state 0x7: Running (Transferring the received packet data from the Rx buffer to the system memory)	R
15:12	TPS4[3:0]	DMA Channel 4 Transmit Process State This field indicates the Tx DMA FSM state for Channel 4. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Transmit Command issued) 0x1: Running (Fetching Tx Transfer Descriptor) 0x2: Running (Waiting for status) 0x3: Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4: Timestamp write state 0x5: Reserved for future use 0x6: Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7: Running (Closing Tx Descriptor)	R
19:16	RPS5[3:0]	DMA Channel 5 Receive Process State This field indicates the Rx DMA FSM state for Channel 5. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Receive Command issued) 0x1: Running (Fetching Rx Transfer Descriptor) 0x2: Reserved for future use 0x3: Running (Waiting for Rx packet) 0x4: Suspended (Rx Descriptor Unavailable) 0x5: Running (Closing the Rx Descriptor) 0x6: Timestamp write state 0x7: Running (Transferring the received packet data from the Rx buffer to the system memory)	R

Bit	Symbol	Function	R/W
23:20	TPS5[3:0]	DMA Channel 5 Transmit Process State This field indicates the Tx DMA FSM state for Channel 5. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Transmit Command issued) 0x1: Running (Fetching Tx Transfer Descriptor) 0x2: Running (Waiting for status) 0x3: Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4: Timestamp write state 0x5: Reserved for future use 0x6: Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7: Running (Closing Tx Descriptor)	R
27:24	RPS6[3:0]	DMA Channel 6 Receive Process State This field indicates the Rx DMA FSM state for Channel 6. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Receive Command issued) 0x1: Running (Fetching Rx Transfer Descriptor) 0x2: Reserved for future use 0x3: Running (Waiting for Rx packet) 0x4: Suspended (Rx Descriptor Unavailable) 0x5: Running (Closing the Rx Descriptor) 0x6: Timestamp write state 0x7: Running (Transferring the received packet data from the Rx buffer to the system memory)	R
31:28	TPS6[3:0]	DMA Channel 6 Transmit Process State This field indicates the Tx DMA FSM state for Channel 6. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Transmit Command issued) 0x1: Running (Fetching Tx Transfer Descriptor) 0x2: Running (Waiting for status) 0x3: Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4: Timestamp write state 0x5: Reserved for future use 0x6: Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7: Running (Closing Tx Descriptor)	R

The DMA_Debug_Status1 register gives the Receive and Transmit process status for DMA Channel 3 to Channel 6.

28.3.218 DMA_Debug_Status2 : DMA Debug Status 2 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TPS7[3:0]				RPS7[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	RPS7[3:0]	DMA Channel 7 Receive Process State This field indicates the Rx DMA FSM state for Channel 7. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Receive Command issued) 0x1: Running (Fetching Rx Transfer Descriptor) 0x2: Reserved for future use 0x3: Running (Waiting for Rx packet) 0x4: Suspended (Rx Descriptor Unavailable) 0x5: Running (Closing the Rx Descriptor) 0x6: Timestamp write state 0x7: Running (Transferring the received packet data from the Rx buffer to the system memory)	R
7:4	TPS7[3:0]	DMA Channel 7 Transmit Process State This field indicates the Tx DMA FSM state for Channel 7. The MSB of this field always returns 0. This field does not generate an interrupt. 0x0: Stopped (Reset or Stop Transmit Command issued) 0x1: Running (Fetching Tx Transfer Descriptor) 0x2: Running (Waiting for status) 0x3: Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4: Timestamp write state 0x5: Reserved for future use 0x6: Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7: Running (Closing Tx Descriptor)	R
31:8	—	These bits are read as 0.	R

The DMA_Debug_Status2 register gives the Receive and Transmit process status for DMA Channel 7.

28.3.219 AXI_LPI_Entry_Interval : AXI LPI Entry Interval Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1040

Bit position:	31																											3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LPIEI[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	LPIEI[3:0]	LPI Entry Interval Contains the number of system clock cycles, multiplied by 64, to wait for an activity in the GMAC to enter into the AXI low power state. 0 indicates 64 clock cycles.	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

This register is used to control the AXI LPI entry interval.

28.3.220 DMA_TBS_CTRL0 : DMA TBS Attributes Control 0 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	FTOS[23:8]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FTOS[7:0]								—	FGOS[2:0]			—	—	—	FTOV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0: Fetch Time Offset is invalid 1: Fetch Time Offset is valid	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
6:4	FGOS[2:0]	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
31:8	FTOS[23:0]	Fetch Time Offset The value in units of 256 ns, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR - 1 value when ESTM mode is set since this value is a modulo CTR value.	R/W

This register is used to control the TBS attributes.

28.3.221 DMA_TBS_CTRL1 : DMA TBS Attributes Control 1 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1054

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	FTOS[23:8]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FTOS[7:0]								—	FGOS[2:0]			—	—	—	FTOV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0: Fetch Time Offset is invalid 1: Fetch Time Offset is valid	R/W

Bit	Symbol	Function	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
6:4	FGOS[2:0]	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
31:8	FTOS[23:0]	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR - 1 value when ESTM mode is set since this value is a modulo CTR value.	R/W

This register is used to control the TBS attributes.

28.3.222 DMA_TBS_CTRL2 : DMA TBS Attributes Control 2 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1058

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	FTOS[23:8]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FTOS[7:0]								—	FGOS[2:0]			—	—	—	FTOV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0: Fetch Time Offset is invalid 1: Fetch Time Offset is valid	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
6:4	FGOS[2:0]	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
31:8	FTOS[23:0]	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR - 1 value when ESTM mode is set since this value is a modulo CTR value.	R/W

This register is used to control the TBS attributes.

28.3.223 DMA_TBS_CTRL3 : DMA TBS Attributes Control 3 Register

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x105C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	FTOS[23:8]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FTOS[7:0]								—	FGOS[2:0]			—	—	—	FTOV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0: Fetch Time Offset is invalid 1: Fetch Time Offset is valid	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
6:4	FGOS[2:0]	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
31:8	FTOS[23:0]	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR - 1 value when ESTM mode is set since this value is a modulo CTR value.	R/W

This register is used to control the TBS attributes.

28.3.224 DMA_CHn_CONTROL : DMA Channel n Control Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1100 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	SPH	—	—	—	DSL[2:0]			—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
17:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20:18	DSL[2:0]	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	SPH	Split Headers The software must ensure that the header fits into the Receive buffers. If the header length exceeds the receive buffer size, the DMA does not split the header and payload. This bit is available only when you select Enable Split Header Structure feature. 0: Split Headers feature is disabled. 1: Split Headers feature is enabled. DMA splits the header and payload in the receive path. The DMA writes the header to the Buffer Address1 of RDES0. The DMA writes the payload to the buffer to which the Buffer Address2 is pointing.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The DMA_CHn_CONTROL register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting.

28.3.225 DMA_CHn_TX_CONTROL : DMA Channel n Transmit Control Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1104 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	TFSEL[1:0]		EDSE	—	—	—	—	—	—	TxPBL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IPBL	—	—	—	—	—	—	—	—	—	—	OSF	TCW[2:0]			ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ST	Start or Stop Transmission Command The DMA tries to acquire descriptor from either of the following positions: ● The current position in the list This is the base address of the Transmit list set by the DMA_CH0_TXDESC_LIST_ADDRESS register. ● The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH0_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH0_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program DMA_CH0_TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0: Stop Transmission Command. 1: Start Transmission Command. Transmission is placed in the running state. The DMA checks the Tx list at the current position for a packet to be transmitted.	R/W

Bit	Symbol	Function	R/W
3:1	TCW[2:0]	Transmit Channel Weight This field indicates the weight assigned to the corresponding Tx channel. When reset is complete, this field is set to 0 for all channels by default, resulting in equal weights to all channels.	R/W
4	OSF	Operate on Second Packet 0: Operate on Second Packet disabled. 1: Operate on Second Packet enabled. Instructs the DMA to process the second packet of the Tx data even before the status for the first packet is obtained.	R/W
14:5	—	These bits are read as 0. The write value should be 0.	R/W
15	IPBL	Ignore PBL Requirement*1 0: Ignore PBL Requirement is disabled. 1: Ignore PBL Requirement is enabled. DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL might use handshaking to slow the DMA.	R/W
21:16	TxPBL[5:0]	Transmit Programmable Burst Length*2 These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, or 16. Any other value results in undefined behavior.	R/W
27:22	—	These bits are read as 0. The write value should be 0.	R/W
28	EDSE	Enhanced Descriptor Enable 0: Enhanced Descriptor is disabled. The corresponding channel uses the descriptors that are 16 Bytes. 1: Enhanced Descriptor is enabled. The corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors.	R/W
30:29	TFSEL[1:0]	TBS Fetch Select Select bits for one of the four DMA_TBS_CTRL register fields (FTOS, FGSN, FTOV) for the channel. 0 0: DMA_TBS_CTRL0 0 1: DMA_TBS_CTRL1 1 0: DMA_TBS_CTRL2 1 1: DMA_TBS_CTRL3	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. Do not use this mode when you enable multiple Transmit DMA channels. Because, it might block other Transmit and Receive DMA Channels from accessing the Read Data Channel of the AXI bus until space is available in Transmit Queue for current transfer.

Note 2. The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of MTL_TXQn_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, the total locations in Tx Queue of size 512 bytes is 32, TxPBL needs to be programmed to less than or equal to 16.

The DMA_CHn_TX_CONTROL register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

28.3.226 DMA_CHn_RX_CONTROL : DMA Channel n Receive Control Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1108 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RPF	—	—	—	—	—	—	—	—	—	RxPBL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	RBSZ_13_y[9:0]										RBSZ_x_0[3:0]				SR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SR	Start or Stop Receive The DMA tries to acquire descriptor from either of the following positions: ● The current position in the list: This is the address set by the DMA_CH0_RXDESC_LIST_ADDRESS register. ● The position at which the Rx process was previously stopped: If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH0_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH0_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. 0: Stop Receive. Rx DMA operation stops after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 1: Start Receive. DMA attempts to acquire the descriptor from the Receive list and processes the incoming packets.	R/W
4:1	RBSZ_x_0[3:0]	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width.	R
14:5	RBSZ_13_y[9:0]	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16 Kbytes. The buffer size is applicable to payload buffers when split headers are enabled. **Note**: The buffer size must be a multiple of 16 depending on the data bus widths. This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
21:16	RxPBL[5:0]	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, or 16. Any other value results in undefined behavior. **Note**: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of MTL_RXQn_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, the total locations in Rx Queue of size 512 bytes is 32, so RxPBL needs to be programmed to less than or equal to 16.	R/W
30:22	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31	RPF	Rx Packet Flush **Note**: The stopping of packet flow from a Rx DMA Channel to the application by setting RPF works only when there is one-to-one mapping of Rx Queue to Rx DMA channels. In Dynamic mapping mode, setting RPF bit in any DMA_CHn_RX_CONTROL register might flush packets from unintended Rx Queues which are destined to the stopped Rx DMA Channel. 0: Rx Packet Flush is disabled. GMAC does not flush the packet in the Rx Queue destined to this Rx DMA Channel when it is in STOP state. This might cause head-of-line blocking in the corresponding Rx Queue. 1: Rx Packet Flush is enabled. GMAC automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this Rx DMA was stopped, are flushed out. The packets that are received by the MAC after the Rx DMA is re-started are routed to the Rx DMA. The flushing is done on the Read side of the Rx Queue.	R/W

The DMA_CHn_RX_CONTROL register controls the Rx features such as PBL, buffer size, and extended status.

28.3.227 DMA_CHn_TXDESC_LIST_ADDRESS : DMA Channel n Tx Descriptor List Address Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1114 + 0x80 × n

Bit position:	31																												0
Bit field:	TDESLA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TDESLA[31:0]	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (3:0) and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO).	R/W

The DMA_CHn_TXDESC_LIST_ADDRESS register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Lword-aligned. The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in DMA_CH0_Tx_Control register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

28.3.228 DMA_CHn_RXDESC_LIST_ADDRESS : DMA Channel n Rx Descriptor List Address Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x111C + 0x80 × n

Bit position:	31																												0
Bit field:	RDESLA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RDESLA[31:0]	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (3:0) and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO).	R/W

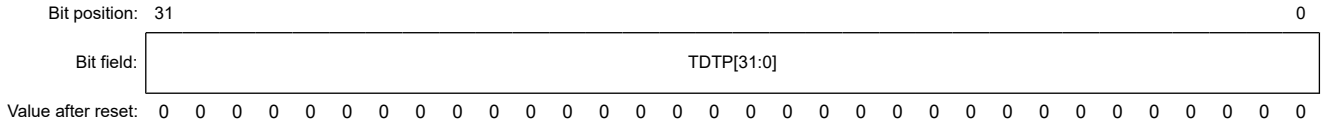
The DMA_CHn_RXDESC_LIST_ADDRESS register points the DMA to the start of Receive descriptor list. This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Lword-aligned. The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given.

You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in DMA_CH0_RX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

28.3.229 DMA_CHn_TXDESC_TAIL_POINTER : DMA Channel n Tx Descriptor Tail Pointer Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1120 + 0x80 × n



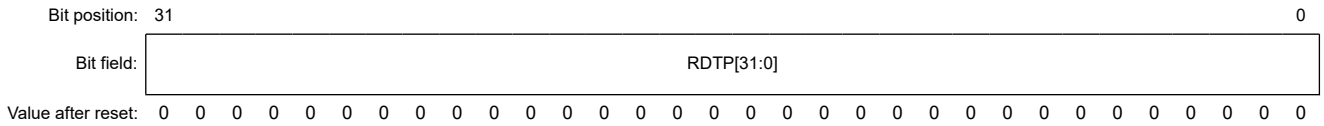
Bit	Symbol	Function	R/W
31:0	TDTP[31:0]	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. LSB (3:0) is ignored.	R/W

The DMA_CHn_TXDESC_TAIL_POINTER register points to an offset from the base and indicates the location of the last valid descriptor.

28.3.230 DMA_CHn_RXDESC_TAIL_POINTER : DMA Channel n Rx Descriptor Tail Pointer Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1128 + 0x80 × n



Bit	Symbol	Function	R/W
31:0	RDTP[31:0]	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. LSB (3:0) is ignored.	R/W

The DMA_CHn_RXDESC_TAIL_POINTER register points to an offset from the base and indicates the location of the last valid descriptor.

28.3.231 DMA_CHn_TXDESC_RING_LENGTH : DMA Channel n Tx Descriptor Ring Length Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x112C + 0x80 × n

Bit position:	31																					9								0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TDRL[9:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	TDRL[9:0]	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. Synopsys recommends a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The DMA_CHn_TXDESC_RING_LENGTH register contains the length of the Transmit descriptor ring.

28.3.232 DMA_CHn_RX_CONTROL2 : DMA Channel n Receive Control 2 Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1130 + 0x80 × n

Bit position:	31								23							18						9								0
Bit field:	—	—	—	—	—	—	—	—	ARBS[5:0]							—	—	—	—	—	—	—	—	RDRL[9:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	RDRL[9:0]	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.	R/W
17:10	—	These bits are read as 0. The write value should be 0.	R/W
23:18	ARBS[5:0]	Alternate Receive Buffer Size Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit, respectively). When ARBS = 0, Rx Buffer1 and Rx Buffer2 sizes are based on RBSZ field of DMA_CHn_RX_CONTROL. Width of ARBS field is 8, 7, or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit, respectively).	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The DMA_CHn_RX_CONTROL2 register controls the Rx features such as Rx Descriptor Ring Length and Alternate Rx Buffer Size.

28.3.233 DMA_CHn_INTERRUPT_ENABLE : DMA Channel n Interrupt Enable Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1134 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	—	—	—	TBUE	TXSE	TIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0: Transmit Interrupt is disabled 1: Transmit Interrupt is enabled	R/W
1	TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0: Transmit Stopped is disabled 1: Transmit Stopped is enabled	R/W
2	TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0: Transmit Buffer Unavailable is disabled 1: Transmit Buffer Unavailable is enabled	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0: Receive Interrupt is disabled 1: Receive Interrupt is enabled	R/W
7	RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0: Receive Buffer Unavailable is disabled 1: Receive Buffer Unavailable is enabled	R/W
8	RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0: Receive Stopped is disabled 1: Receive Stopped is enabled	R/W
9	RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0: Receive Watchdog Timeout is disabled 1: Receive Watchdog Timeout is enabled	R/W
10	ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0: Early Transmit Interrupt is disabled 1: Early Transmit Interrupt is enabled	R/W

Bit	Symbol	Function	R/W
11	ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0: Early Receive Interrupt is disabled 1: Early Receive Interrupt is enabled	R/W
12	FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0: Fatal Bus Error is disabled 1: Fatal Bus Error is enabled	R/W
13	CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0: Context Descriptor Error is disabled 1: Context Descriptor Error is enabled	R/W
14	AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the DMA_CH0_STATUS register: ● Bit 1: Transmit Process Stopped ● Bit 7: Rx Buffer Unavailable ● Bit 8: Receive Process Stopped ● Bit 9: Receive Watchdog Timeout ● Bit 10: Early Transmit Interrupt ● Bit 12: Fatal Bus Error ● Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled. 0: Abnormal Interrupt Summary is disabled 1: Abnormal Interrupt Summary is enabled	R/W
15	NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the DMA_CH0_STATUS register: ● Bit 0: Transmit Interrupt ● Bit 2: Transmit Buffer Unavailable ● Bit 6: Receive Interrupt ● Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0: Normal Interrupt Summary is disabled 1: Normal Interrupt Summary is enabled	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The DMA_CHn_INTERRUPT_ENABLE register enables the interrupts reported by the Status register.

28.3.234 DMA_CHn_RX_INTERRUPT_WATCHDOG_TIMER : DMA Channel n Receive Interrupt Watchdog Timer Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1138 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RWTU[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	RWT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RWT[7:0]	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the DMA_CHn_STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
17:16	RWTU[1:0]	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. For example, when RWT = 2 and RWTU = 1, the watchdog timer is set for 2 × 512 = 1024 system clock cycles. 0 0: 256 0 1: 512 1 0: 1024 1 1: 2048	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The DMA_CHn_RX_INTERRUPT_WATCHDOG_TIMER register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CHn_STATUS register.

28.3.235 DMA_CHn_SLOT_FUNCTION_CONTROL_STATUS : DMA Channel n Slot Function Control and Status Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x113C + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	RSN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SIV[11:0]												—	—	ASC	ESC
Value after reset:	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is ● equal to the reference slot number or ● ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. **Note**: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/ fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0: Slot Comparison is disabled 1: Slot Comparison is enabled	R/W

Bit	Symbol	Function	R/W
1	ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is ● equal to the reference slot number given in the RSN field or ● ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0: Advance Slot Check is disabled 1: Advance Slot Check is enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
15:4	SIV[11:0]	Slot Interval Value This field controls the period of the slot interval in which the Tx DMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 µs while the maximum value 4095 specifies the slot interval of 4096 µs. The default/reset value is 0x07C which corresponds to slot interval of 125 µs.	R/W
19:16	RSN[3:0]	Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.	R
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The DMA_CHn_SLOT_FUNCTION_CONTROL_STATUS register contains the control bits for slot function and the status for Transmit path.

28.3.236 DMA_CHn_CURRENT_APP_TXDESC : DMA Channel n Current Application Transmit Descriptor Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1144 + 0x80 × n

Bit position:	31																												0
Bit field:	CURTDESAPTR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CURTDESAPTR[31:0]	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.	R

The DMA_CHn_CURRENT_APP_TXDESC register points to the current Transmit descriptor read by the DMA.

28.3.237 DMA_CHn_CURRENT_APP_RXDESC : DMA Channel n Current Application Receive Descriptor Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x114C + 0x80 × n

Bit position:	31																												0
Bit field:	CURRDESAPTR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CURRDESAPTR[31:0]	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.	R

The DMA_CHn_CURRENT_APP_RXDESC register points to the current Receive descriptor read by the DMA.

28.3.238 DMA_CHn_CURRENT_APP_TXBUFFER : DMA Channel n Current Application Transmit Buffer Address Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1154 + 0x80 × n

Bit position:	31																												0
Bit field:	CURTBUFAPTR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CURTBUFAPTR[31:0]	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.	R

The DMA_CHn_CURRENT_APP_TXBUFFER register points to the current Tx buffer address read by the DMA.

28.3.239 DMA_CHn_CURRENT_APP_RXBUFFER : DMA Channel n Current Application Receive Buffer Address Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x115C + 0x80 × n

Bit position:	31																												0
Bit field:	CURRBUFAPTR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CURRBUFAPTR[31:0]	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.	R

The DMA_CHn_CURRENT_APP_RXBUFFER register points to the current Rx buffer address read by the DMA.

28.3.240 DMA_CHn_STATUS : DMA Channel n Status Register (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1160 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	REB[2:0]			TEB[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	—	—	—	TBU	TPS	TI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TI	Transmit Interrupt*1 This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. 0: Transmit Interrupt status not detected 1: Transmit Interrupt status detected	R/W
1	TPS	Transmit Process Stopped*1 This bit is set when the transmission is stopped. 0: Transmit Process Stopped status not detected 1: Transmit Process Stopped status detected	R/W
2	TBU	Transmit Buffer Unavailable*1 This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DMA_Debug_Status0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: ● Change the ownership of the descriptor by setting Bit 31 of TDES3. ● Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. 0: Transmit Buffer Unavailable status not detected 1: Transmit Buffer Unavailable status detected	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	RI	Receive Interrupt*1 This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. 0: Receive Interrupt status not detected 1: Receive Interrupt status detected	R/W
7	RBU	Receive Buffer Unavailable*1 This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. 0: Receive Buffer Unavailable status not detected 1: Receive Buffer Unavailable status detected	R/W
8	RPS	Receive Process Stopped*1 This bit is asserted when the Rx process enters the Stopped state. 0: Receive Process Stopped status not detected 1: Receive Process Stopped status detected	R/W
9	RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0: Receive Watchdog Timeout status not detected 1: Receive Watchdog Timeout status detected	R/W
10	ETI	Early Transmit Interrupt*1 This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. 0: Early Transmit Interrupt status not detected 1: Early Transmit Interrupt status detected	R/W
11	ERI	Early Receive Interrupt*1 This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. The setting of RI bit automatically clears this bit. 0: Early Receive Interrupt status not detected 1: Early Receive Interrupt status detected	R/W

Bit	Symbol	Function	R/W
12	FBE	Fatal Bus Error*1 This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. 0: Fatal Bus Error status not detected 1: Fatal Bus Error status detected	R/W
13	CDE	Context Descriptor Error*1 This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. 0: Context Descriptor Error status not detected 1: Context Descriptor Error status detected	R/W
14	AIS	Abnormal Interrupt Summary*1 Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH0_INTERRUPT_ENABLE register: ● Bit 1: Transmit Process Stopped ● Bit 7: Receive Buffer Unavailable ● Bit 8: Receive Process Stopped ● Bit 10: Early Transmit Interrupt ● Bit 12: Fatal Bus Error ● Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. 0: Abnormal Interrupt Summary status not detected 1: Abnormal Interrupt Summary status detected	R/W
15	NIS	Normal Interrupt Summary*1 Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH0_INTERRUPT_ENABLE register: ● Bit 0: Transmit Interrupt ● Bit 2: Transmit Buffer Unavailable ● Bit 6: Receive Interrupt ● Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH0_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. 0: Normal Interrupt Summary status not detected 1: Normal Interrupt Summary status detected	R/W
18:16	TEB[2:0]	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. This field is valid only when the FBE bit is set. This field does not generate an interrupt. ● Bit 18 0: No Error during data transfer by Tx DMA 1: Error during data transfer by Tx DMA ● Bit 17 0: Error during data buffer access 1: Error during descriptor access ● Bit 16 0: Error during write transfer 1: Error during read transfer	R
21:19	REB[2:0]	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. This field is valid only when the FBE bit is set. This field does not generate an interrupt. ● Bit 21 0: Error during data transfer by Rx DMA 1: No Error during data transfer by Rx DMA ● Bit 20 0: Error during data buffer access 1: Error during descriptor access ● Bit 19 0: Error during write transfer 1: Error during read transfer	R

Bit	Symbol	Function	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

28.3.241 DMA_CHn_MISS_FRAME_CNT : DMA Channel n Miss Packet Counter (n = 0 to 7)

Base address: GMAC0 = 0x8010_0000
GMACm = 0x9200_0000 + 0x1_0000 × (m - 1) (m = 1, 2)

Offset address: 0x1164 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MFCO	—	—	—	—	MFC[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	MFC[10:0]	Dropped Packet Counters*1 This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programing RPF field in DMA_CHn_RX_CONTROL register. The counter gets cleared when this register is read.	R
14:11	—	These bits are read as 0.	R
15	MFCO	Overflow status of the MFC Counter*1 When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. 0: Miss Frame Counter overflow not occurred 1: Miss Frame Counter overflow occurred	R
31:16	—	These bits are read as 0.	R

Note 1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CHn_RX_CONTROL register.

28.4 Architecture

28.4.1 AHB Slave Interface

The AHB slave interface provides access to the DMA, MTL, and MAC CSR space.

By default, the slave interface provides OKAY response to all CSR accesses. If SEEN field of the MAC_CSR_SW_Ctrl register is programmed to 1, GMAC provides error response when reserved registers within the CSR space are accessed.

28.4.2 AXI Master Interface

The DMA controller interfaces with the application through the AMBA 3 AXI interface. The AMBA 3 AXI bus interface provides the characteristics to support highly-effective data traffic throughput. The system bus utilization is maximized by allowing simultaneous read and write transfers initiated from different DMA channels. For example, the Tx descriptor read and Rx descriptor write-back (status update) operations can happen simultaneously. However, the Tx descriptor read and Tx descriptor write-back operations cannot happen simultaneously because the Tx DMA (or Rx DMA) does not initiate the next transfer unless the previous one is complete.

The AXI master interface allows multiple numbers of burst requests to be queued for Read or Write operation. The system bus utilization can be further enhanced by supporting Requests reordering and Data interleaving. This gives the AXI master the flexibility to dynamically and independently select multiple slaves for handling Priority Requests and Slow peripheral slaves.

28.4.2.1 Burst Splitting and Burst Selection

The AXI master splits the DMA requests into multiple bursts on the AXI system bus. Splitting is based on the DMA count, the software-controllable burst enable bits (FB, BLEN16, BLEN8, BLEN4), and the software-controllable burst types (INCR and INCR_ALIGNED). For more information, see DMA_SysBus_Mode register.

The AXI master also handles splitting the transfers that cross 4 KB address boundary, with an option to limit it to 1-KB address boundary. The Burst Length Select Priority is in the following sequence: FB, BLEN16, BLEN8, and BLEN4.

28.4.2.2 INCR Burst Type

If FB is disabled, the AXI master always chooses the maximum allowed burst length based on the BLEN16 bit (possible maximum burst lengths are 16). The AXI master chooses a burst length of any value less than the maximum enabled burst length (all lesser burst length enables are redundant) in the following cases:

- The DMA requests are not multiples of the maximum allowed burst length
- The 4K/1K address boundary crossing condition is enabled

For example, when only BLEN16, and BLEN4 are enabled, and the DMA requests a burst transfer of 38 beats size, the AXI master splits it into three bursts of 16, 16 and 6 beats size, respectively.

If FB is enabled, the burst length is based on the priority of the enabled bits in the following order: BLEN16, BLEN8, and BLEN4. When the DMA requests to transfer a burst, the AXI interface splits the requested bursts into multiple transfers by using only the enabled burst lengths. This splitting can occur in following cases:

- The requested burst is not a multiple of the maximum enabled burst
- The 4K/1K address boundary crossing condition is enabled

If the AXI interface cannot choose any of the enabled burst lengths, it selects the burst length as 1. For example, if only BLEN16, and BLEN4 are enabled and the DMA requests a burst transfer of size 38 beats, the AXI interface splits it into multiple bursts of size 16, 16, 4, 1, and 1 beats, respectively. The sequence is in decreasing burst lengths.

28.4.2.3 INCR_ALIGNED Burst Type

When the address-aligned burst type is enabled, then in addition to the burst splitting conditions explained in INCR Burst Type, the AXI interface splits the DMA requested bursts such that each burst length is aligned to the least significant bits of the start address. The AXI interface initially generates smaller bursts so that the remaining transfers can be transferred with the maximum possible (enabled) fixed burst lengths.

28.4.2.4 Outstanding Transactions

The AXI master supports configurable outstanding read/write requests on the AXI bus. You can control the outstanding requests by programming the DMA_SysBus_Mode register.

The AXI master supports up to 16 outstanding Read or Write requests on the AXI bus. You can also control these outstanding requests through software by programming bits [24:21] and bits [19:16] in the DMA_SysBus_Mode register. This is a useful workaround for any system-level issues with handling of multiple requests.

Note: The maximum number of outstanding requests is divided equally between the requests generated by the two internal masters (Rx DMA and Tx DMA). This means that each DMA is limited to a maximum of eight outstanding requests in each Read or Write channel. When you reduce the outstanding request (for example, to four) by programming the respective control bits, all outstanding requests can be from the same DMA. The minimum number of outstanding requests per channel is two.

28.4.2.5 Priority of AXI Requests

The descriptor transfers have higher priority than the data transfers. Therefore, if there are two requests, Rx Descriptor Read and Tx Data Read, the Rx Descriptor Read is given higher priority so that the next Rx Data Write (subsequent to Rx

Descriptor Read) need not wait for the Tx Data Read transfer to complete. If there are descriptor read requests from both DMAs, these requests are serviced based on the first-come first-serve basis. The Rx DMA has higher priority if descriptor read requests, generated from both DMAs, are in the same clock. Similarly, in the write channel, descriptor writes from any DMA have higher priority than the data-write transfers for the Rx DMA.

If the requests are generated from different channels simultaneously in the same clock, the request priority among the channels is: Channel 7, Channel 6, …, Channel 1, and Channel 0.

28.4.2.6 Bursts Reordering and Data Interleaving

The AXI protocol allows reordering of data transfers with different AXI-IDs with respect to the sequence of requests. It also allows data interleaving between transfers of different AXI-IDs for Read channels.

The requests from each internal master (Rx DMA and Tx DMA) are generated with different AXI-ID. Therefore, reordering and data interleaving can be performed as two DMAs operate independently and are allocated different address space on the slave memory.

The AXI master supports reordering and interleaving from the AXI slaves on the Read channel. Each DMA internally ensures that a Read request and Write request is never generated at the same time to the same address (it can occur for only descriptor accesses). The Tx DMA ensures that Tx Descriptor Read, Tx Data Read, and Tx Descriptor Write-Back requests are generated only after the previous requests are complete. The Rx DMA generates only Descriptor reads on the Read channel with a different ID. Therefore, data reordering and interleaving on the Read channel does not cause any problems in the MAC.

28.4.2.7 Posted Writes

In posted writes, the write channel of AXI master transfers an OKAY response to the DMA as soon as the last cycle or beat of data is accepted by the AXI interconnect. The AXI master sends this response without waiting for the response from the target AXI slave on the write response channel. In non-posted writes, the AXI master interface transfers the response received from the AXI slave channel to the DMA.

For transferring the Ethernet data to system memory, the Rx DMA always issues posted write requests to the AXI master interface. This enables pipelining of the data requests without delays. The AXI master does not support out-of-order write transfers. However, the Rx DMA ensures that the sequence is maintained for descriptor writes, descriptor reads, and data transfers.

For writing descriptors (status or timestamp), the DMA always issues the non-posted write requests. This is needed because the Transfer Complete Interrupt is generated based on the descriptor writes for which the DMA needs completion response from the memory slave. This ensures no race condition between the hardware and the software because the interrupt is generated only after the data and descriptor are written to the slave memory.

28.4.2.8 Error Response Handling

Whenever there is an error response from the AXI slave, the respective DMA channel which generated the request gets disabled. The AXI master asserts the interrupt (GMAC_SBD) when the corresponding "Fatal Bus Error" interrupt is enabled. Only the DMA whose specific transaction had the error response gets affected and goes to STOP state. The application can re-initialize that specific DMA and restart the operation. The traffic on the other channels does not get affected if an error is detected for a DMA Channel.

28.4.3 DMA Controller

The DMA has independent Transmit (Tx) and Receive (Rx) engines, and a CSR space. The Tx engine transfers data from the system memory to the device port (MTL), whereas the Rx engine transfers data from the device port to the system memory. The DMA engine uses descriptors to efficiently move data from source to destination with minimal application CPU intervention. The DMA is designed for packet-oriented data transfers such as packets in Ethernet. The DMA controller can be programmed to interrupt the application CPU for situations such as Packet Transmit and Receive Transfer completion, and other normal or error conditions.

The DMA and the application communicate through the following two data structures:

- Control and Status registers (CSR)
- Descriptor lists and data buffers

The DMA supports up to 8 Tx and 8 Rx Descriptor lists (or DMA channels). The base address of each list is written to the respective Tx Descriptor List Address register and Rx Descriptor List Address register. The descriptor list is forward linked and the next descriptor is always considered at a fixed offset to the current one. The offset is controlled by the DSL field of DMA_CHn_CONTROL register. The number of descriptors in the list is programmed in the respective Tx (or Rx) Descriptor Ring Length register. After processing the last descriptor in the list, DMA automatically jumps back to the descriptor in the List Address register to create a descriptor ring.

The descriptor lists reside in the physical memory address space of the application. Each descriptor can point to a maximum of two buffers in the system memory. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the application physical memory space and consists of an entire packet or part of a packet but cannot exceed a single packet. Buffers contain only data. Buffer status is maintained in the descriptor. Data chaining refers to packets that span multiple data buffers. However, a single descriptor cannot span multiple packets. The DMA skips to the data buffer of next packet when EOP is detected.

The GMAC supports the ring structure for the DMA descriptor. For more information on the descriptor, see section 28.17. Descriptors that describes the descriptor structure and how the DMA accesses the descriptors.

28.4.3.1 DMA Application Bus Burst Access

The DMA engines attempts to transfer data in a burst of maximum size as programmed in the PBL fields of Transmit Control and Receive Control registers of the respective DMA. The Rx and Tx descriptors are always accessed in the maximum possible (limited by PBL or 16 × 8/bus width) burst length for 16 bytes to be read. The burst transfers initiated by the DMA can be split into multiple burst transfers as per the Application Interface protocol (AXI) requirements and the settings of DMA_SysBus_Mode register.

The Tx DMA initiates a data transfer only when sufficient space is available in the MTL Tx Queue to accommodate either of the following:

- Bytes corresponding to the configured burst (PBL × bus_width/8)
- Remaining bytes in Tx Buffer without EOP
- Number of bytes till EOP

The Rx DMA initiates a data transfer in the following conditions:

- Sufficient data is available in MTL Rx Queue to accommodate the configured burst
- EOP (when it is less than the configured burst length) is detected in the Rx Queue

The DMA indicates the start address and the number of transfers required to the AXI master interface. When the AXI Interface is configured for fixed-length burst, it transfers the data by using the best combination of INCR4, INCR8, or INCR16 and SINGLE transactions. If EOP is reached before the fixed-burst ends on the AXI interface, dummy transfers are performed in-order to complete the fixed-burst. Otherwise [bit 0 of DMA_SysBus_Mode register is reset], the DMA transfers the data using INCR (undefined length) and SINGLE transactions.

When the AXI interface is configured for address-aligned beats, both DMA engines ensure that the first burst transfer initiated by the AXI is less than or equal to the size of the configured PBL.

Therefore, all subsequent beats start at an address that is aligned to the configured PBL.

(1) DMA Application Data Buffer Alignment

The Tx and Rx data buffers do not have any restrictions on start address alignment. For example, in systems with 32-bit memory, the start address for buffers can be aligned to any of the four bytes. However, the DMA always initiates write transfers with address aligned to the bus width and dummy data (old data) in the invalid byte lanes. This typically happens during the transfer of the beginning or end of an Ethernet packet. The software driver should discard the dummy bytes based on the start address of the buffer and size of the packet.

Table 28.5 Application data buffer alignment examples (1 of 2)

Buffer Read	If the Tx buffer address is 0x00000FF2 (for 32-bit data bus), and 15 bytes is to be transferred, the DMA reads five full words from address 0x00000FF0. But when transferring data to the MTL Tx queue, the extra bytes (the first two bytes) are dropped or ignored. Similarly, the last 3 bytes of the last transfer are also ignored. The DMA always ensures that it transfers a full 32-bit data to the MTL Tx queue, unless it is the end of packet.

Table 28.5 Application data buffer alignment examples (2 of 2)

Buffer Write	If the Rx buffer address is 0x0000FF2 (for 64-bit data bus) and 16 bytes of a received packet is to be transferred, the DMA writes 3 full words from address 0x00000FF0. However, the first 2 bytes of the first transfer and the last 6 bytes of the third transfer have dummy data. The DMA considers the offset address only if it is the first Rx buffer of the packet. The DMA ignores the offset address and performs full word writes for the middle and the last Rx buffer of the packet.

(2) DMA Buffer Size Calculations

The DMA does not update the size fields in the Tx and Rx descriptors. The DMA updates only the status fields (RDES and TDES) of the descriptors. The driver has to perform the size calculations.

The Tx DMA transfers the exact number of bytes (indicated by buffer size field of TDES2) towards the MAC. If a descriptor is marked as first (FD bit of TDES3 is set), the DMA marks the first transfer from the buffer as SOP. If a descriptor is marked as last (LD bit of TDES3), the DMA marks the last transfer from that data buffer as EOP to the MTL.

The Rx DMA transfers data to a buffer until the buffer is full or the end of packet is received from the MTL. When the FD bit of a descriptor is set, the amount of valid data in a buffer is accurately indicated by the buffer size field (programmed in DMA Channel Receive Control register) minus the data buffer pointer offset. The offset is zero when the data buffer pointer is aligned to the data bus width. If a descriptor is marked as last, the buffer may not be full (as indicated by the buffer size in bits [14:1] of Receive Control register). To compute the amount of valid data in this final buffer, the driver must read the packet length (PL bits of RDES3[14:0]) and subtract the sum of the buffer sizes of the preceding buffers in this packet. The Rx DMA always transfers the start of next packet with a new descriptor.

Note: Even when the start address of a Rx buffer is not aligned to the data width of system bus, the system should allocate a Rx buffer of a size aligned to the system bus width. For example, if the system allocates a Rx buffer of 1,024 bytes (1 KB) starting from address 0x1000, the software can program the buffer start address in the Rx descriptor to have a 0x1002 offset. The Rx DMA writes the packet to this buffer with dummy data in the first two locations (0x1000 and 0x1001). The actual packet is written from location 0x1002. Therefore, the actual useful space in this buffer is 1,022 bytes, even though the buffer size is programmed as 1,024 bytes, because of the start address offset.

28.4.3.2 Transmission DMA

(1) Tx DMA Operation: Default (Non-OSP) Mode

The Tx DMA engine in default mode proceeds as follows:

1. The application sets up the Transmit descriptor (TDES0-TDES3) and sets the Own bit (TDES3[31]) after setting up the corresponding data buffer(s) with Ethernet Packet data.
2. The application advances the Descriptor Tail pointer offset value of the Transmit Channel.
3. While in the Run state, the DMA runs an Arbitration cycle to select the next Tx DMA channel from which the packets requiring transmission should be processed.
4. The DMA fetches the descriptor from the application memory.
5. If the DMA detects one of the following conditions, the transmission from that channel is suspended and bit 2 and bit 16 of Status Register of corresponding DMA channel are set and the Tx Engine proceeds to step 11:
   - The descriptor is flagged as owned by the application (TDES3[31] = 0)
   - The Descriptor Tail pointer is equal to the Current Descriptor pointer in Ring Descriptor list Mode
   - An error condition occurs
6. If the acquired descriptor is flagged as owned by the DMA (TDES3[31] = 1), the DMA decodes the Transmit Data Buffer address from the acquired descriptor.
7. The DMA fetches the Transmit data from the system memory and transfers the data to the MTL for transmission.
8. If an Ethernet packet is stored over data buffers in multiple descriptors, the DMA closes the intermediate descriptor and fetches the next descriptor. Steps 3 through 7 are repeated until the end of Ethernet-packet data is transferred to the MTL.
9. When packet transmission is complete, if IEEE 1588 Timestamp feature was enabled for the packet (as indicated in the Tx status), the timestamp value obtained from MTL is written to the Tx descriptor (TDES0 and TDES1) that

contains the EOP buffer. The status information is written to this Tx descriptor (TDES3). The application now owns this descriptor because the Own bit is cleared during this step.
If timestamp feature is not enabled for this packet, the DMA does not alter the contents of TDES0 and TDES1.
By default, the Tx DMA places subsequent request after the current descriptor write request is completed to the descriptor memory. If posted descriptor write is enabled by setting DSPW bit in DMA_Mode register, the Tx DMA places subsequent request after the current descriptor write request is placed without waiting for its completion, thereby improving the throughput.

10. Bit 0 of Status register of corresponding channel is set after completing transmission of a packet that has Interrupt on Completion (TDES2[31]) set in its Last Descriptor. The DMA engine returns to step 3.
11. In the Suspend state, the DMA tries to acquire the descriptor again (and thereby return to step 3). A poll demand command is triggered by writing any value to the DMA_CH0_TxDesc_Tail_Pointer when it receives a Transmit Poll demand and the Underflow Interrupt Status bit is cleared. If the application stopped the DMA by clearing bit 0 of Transmit Control register of corresponding DMA channel, the DMA enters the Stop state.
In non-OSP mode, it is not required for the driver/application to extract the packet status and release the descriptor for the packet, even though FD is set to 1. The driver must track the descriptor that has OWN = 0 and LD = 1, as 1; such a descriptor contains the transmission status for the packet. This indicates that descriptors up to this descriptor can be released for reuse by the driver/application for subsequent packets.

The Tx DMA transmission flow in default mode is shown in Figure 28.3.

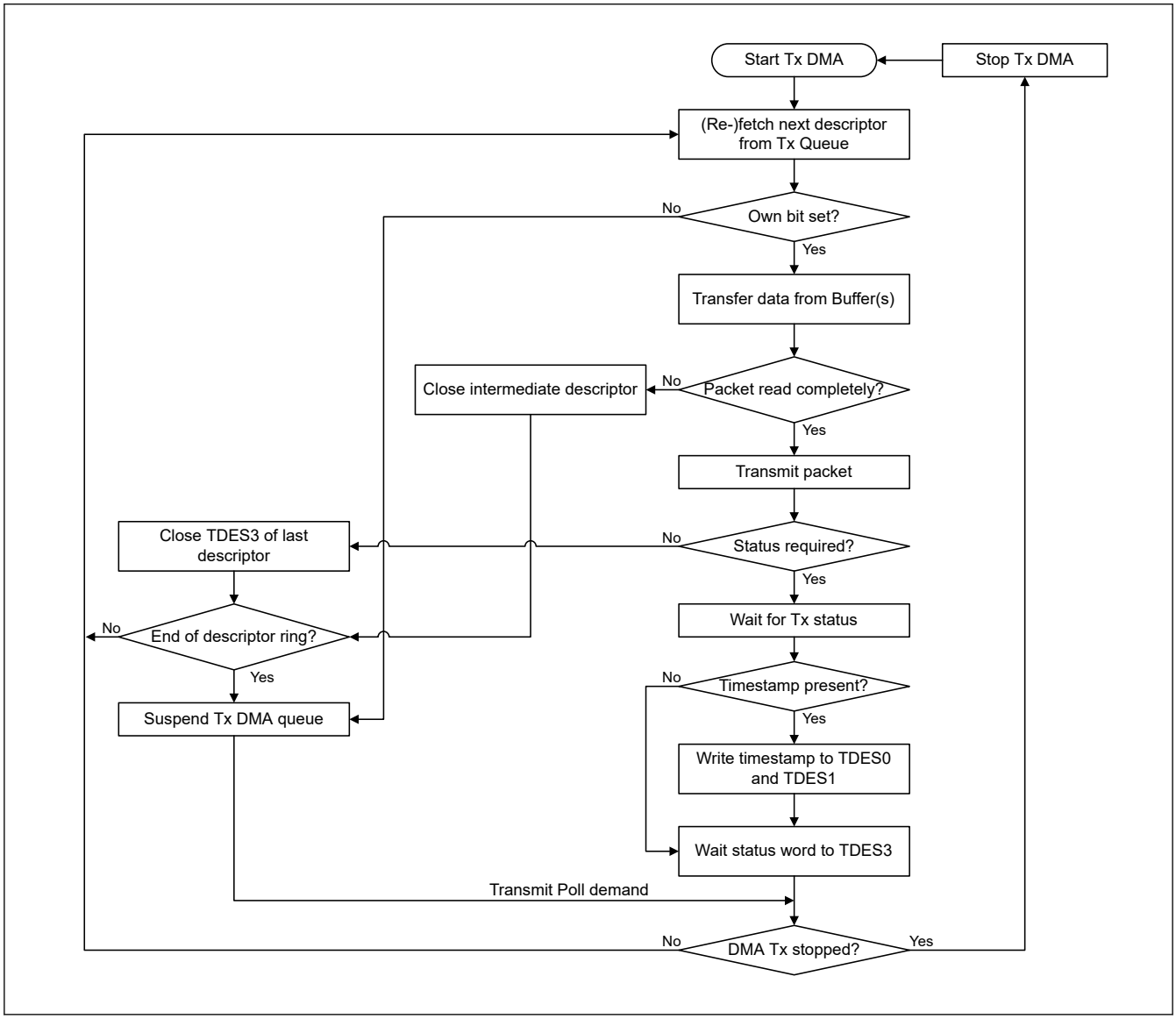


Figure 28.3 Tx DMA operation in default mode

(2) Tx DMA Operation: OSP Mode

In the Run state, if bit 4 is set in the Transmit Control Register of corresponding DMA channel, the Transmit process can simultaneously acquire two packets without closing the Status descriptor of the first packet. As the Transmit process finishes transferring the first packet, it immediately polls the Transmit Descriptor list for the second packet. If the second packet is valid, the Transmit process transfers this packet before writing the status information of the first packet.

In OSP mode, the Run state Tx DMA operates in the following sequence:

1. The DMA operates as described in step 1 to step 7 of the Tx DMA (default mode).
2. The DMA fetches the next descriptor without closing the last descriptor of previous packet.
3. If the DMA owns the acquired descriptor, the DMA decodes the transmit buffer address in this descriptor. If the DMA does not own the descriptor, the DMA goes into Suspend mode and skips to step 7.
4. The DMA fetches the Transmit packet from the system memory and transfers the packet to the MTL until the EOP data is transferred, closing the intermediate descriptors if this packet is split across multiple descriptors.
5. The DMA waits for the packet transmission status and timestamp of previous packet. When the status is available, the DMA writes the timestamp to TDES0 and TDES1 if such timestamp was captured (as indicated by a status bit). The DMA writes the status, with a cleared Own bit, to the corresponding TDES1, thus closing the descriptor.
   If Timestamp feature is not enabled for the previous packet, the DMA does not alter the contents of TDES2 and TDES3.
6. The Transmit interrupt is set (if enabled). The DMA fetches the next descriptor and proceeds to step 3 (when Status is normal). If the previous transmission status shows an underflow error, the DMA goes into Suspend mode (step 7).
7. In Suspend mode, if a pending status and timestamp are received from the MTL, the DMA does the following:
   (a) Writes the timestamp (if enabled for the current packet) to TDES0 and TDES1
   (b) Writes the status to the corresponding TDES1
   (c) Sets relevant interrupts and returns to Suspend mode

   If no status is pending and the application stopped the DMA by clearing bit 0 of Transmit Control Register of corresponding DMA channel, the DMA enters the Stop state.
8. The DMA can exit Suspend mode and enter the Run state (goes to step 1 or step 2 depending on pending status) only after receiving a Transmit Poll demand in Transmit Descriptor Tail Pointer register of corresponding channel.

Note:
- The DMA fetches the next descriptor before closing the current descriptor. Therefore, the descriptor ring length must be more than two. Synopsys recommends having a minimum descriptor length of four.
- In the OSP mode, all except the last descriptor is closed immediately. The last descriptor is closed after the packet is transmitted on the line. Therefore, to minimize the complexity, the transmission status and only the required control bits are updated in the pending last descriptor of the previous packet.

The Tx DMA transmission flow in default mode is shown in Figure 28.4.

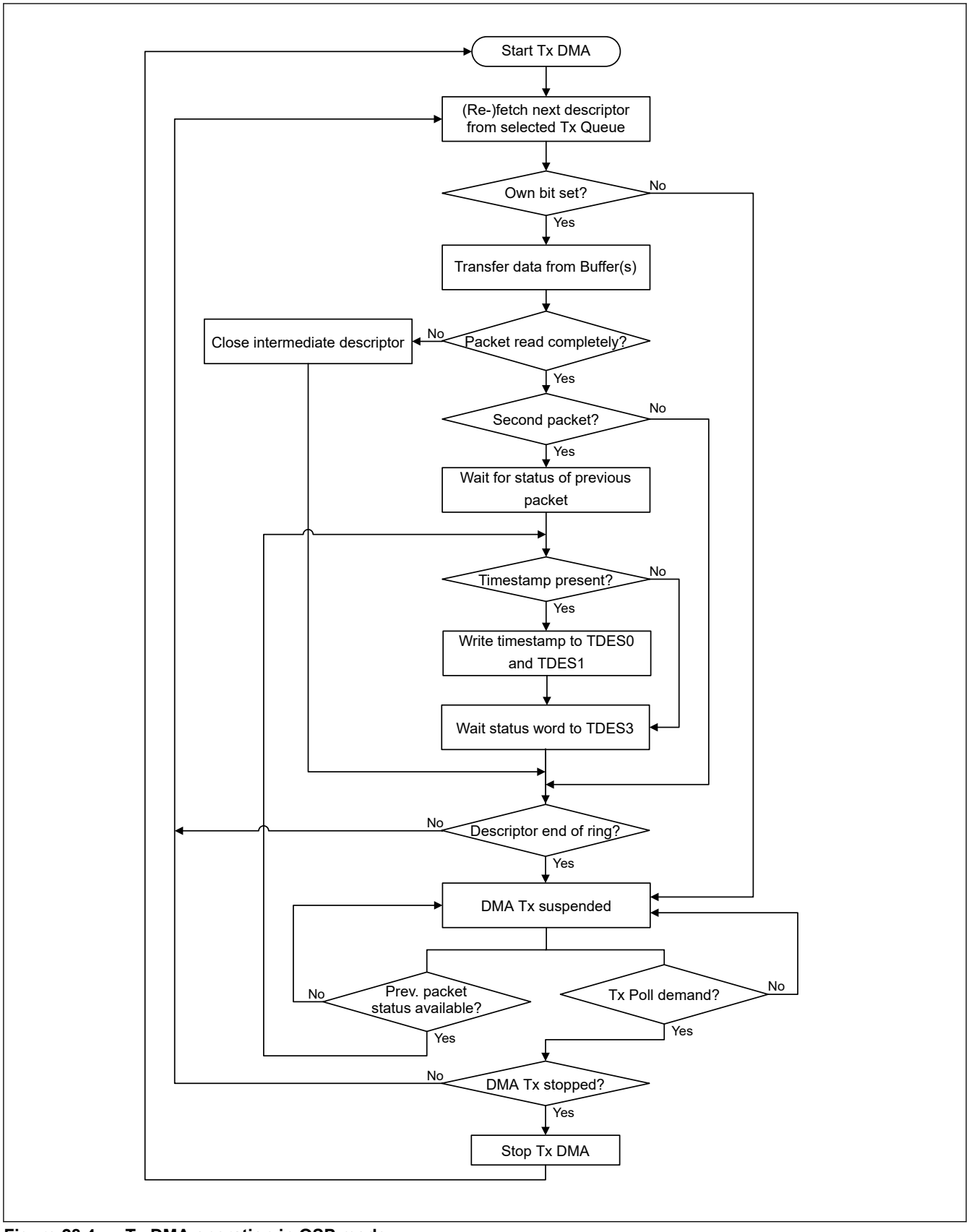


Figure 28.4 Tx DMA operation in OSP mode

(3) Transmit Packet Processing

The Tx DMA expects that the data buffers contain complete Ethernet packets, excluding preamble, pad bytes, and FCS fields. The DA, SA, and Type/Length fields contain valid data. If the Tx Descriptor indicates that the MAC must disable CRC or PAD insertion, the buffer must have complete Ethernet packets (excluding preamble), including the CRC bytes.

Packets can be data-chained and can span several buffers. Packets must be delimited by the First Descriptor (TDES3[29]) and the Last Descriptor (TDES3[28]). As transmission starts, the First Descriptor must have TDES3[29] set. When this occurs, the packet data is transferred from the application buffer to the MTL Tx Queue. Concurrently, if the current packet has the Last Descriptor (TDES3[28]) clear, the Tx Process attempts to acquire the Next Descriptor. The Tx Process expects this descriptor to have TDES3[29] clear. If TDES3[28] is clear, it indicates an intermediary buffer. If TDES3[28] is set, it indicates the last buffer of the packet.

After the last buffer of the packet has been transmitted, the DMA writes back the final status information to the Transmit Descriptor 3 (TDES3) word of the descriptor that has the Last Descriptor bit set in Transmit Descriptor 3 (TDES3[28]). At this time, if Interrupt on Completion (TDES2[31]) is set, bit 0 of Status Register of corresponding DMA channel is set, the Next Descriptor is fetched, and the process repeats. The actual packet transmission begins after either of the following:

- The MTL Tx Queue has reached a programmable Transmit threshold (bits [6:4] of Transmit Operation Mode register of corresponding MTL Transmit Queue)
- A full packet is contained in the FIFO

You can also use the store-and-forward mode (bit 1 of MTL Transmit Operation Mode Register of a queue). In this mode, descriptors are released (Own bit TDES0[31] clears) when the DMA finishes transferring the packet.

Note: To ensure proper transmission of a packet and the next packet, you must specify a non-zero buffer size for the Transmit descriptor that has the Last Descriptor (TDES3[28]) set.

(4) Transmit Polling Suspended

Transmit polling can be suspended by any of the following conditions:

- The DMA detects a descriptor owned by the application (TDES3[31] = 0).
  To resume, the driver must give descriptor ownership to the DMA and then issue a Poll Demand command by writing the Tail Pointer register. If the DMA goes into SUSPEND state because of this condition, bit 15 and bit 2 of Status Register of corresponding DMA channel are set.
- A packet transmission is aborted when a Transmit error is detected because of underflow.
  The appropriate Transmit Descriptor 3 (TDES3) bit is set. When this condition occurs, the following bits are set and the information is written to Transmit Descriptor 0, causing the suspension:
  - Bit 14 of Status Register of corresponding DMA channel
  - Transmit Underflow bit of corresponding queue in MTL_Interrupt_Status
- The DMA detects that the Tail Pointer is equal to the Current descriptor closed by the it.
  To resume, the software driver must modify the Tail Pointer register.

In all conditions, the position in the Transmit List is retained. The retained position is that of the descriptor following the Last Descriptor closed by the DMA. The driver must explicitly issue a Transmit Poll Demand command after rectifying the suspension cause.

(5) Transmit Channel Arbitration

An arbiter provides access to multiple DMAs trying to access the Bus Interface Unit (BIU). Figure 28.5 shows the arbitration process.

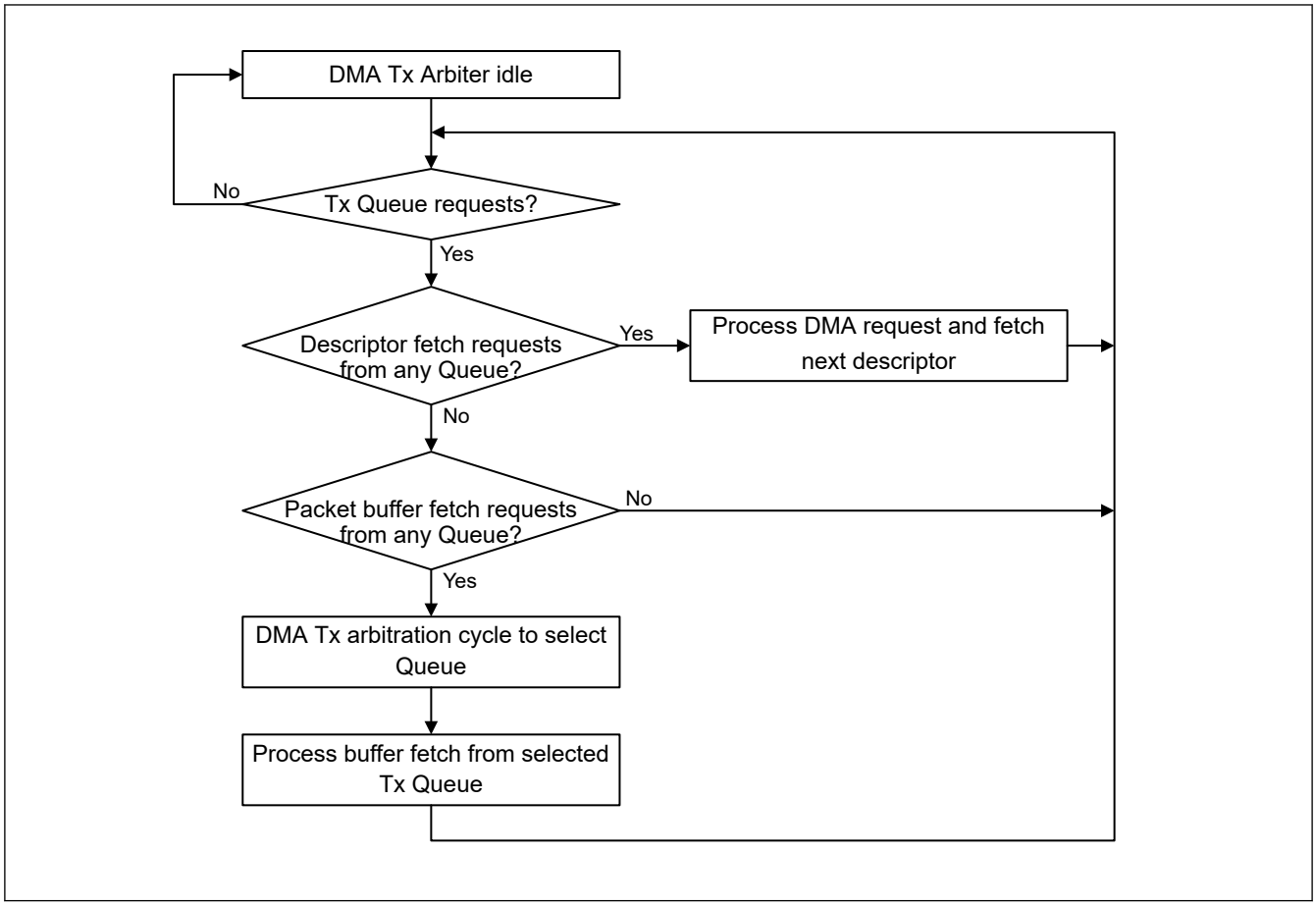


Figure 28.5 DMA transmit channel arbitration process

When there is a request in the Transmit DMA, the DMA arbiter checks the type of the request: packet buffer fetch or descriptor fetch request. The descriptor fetch requests have higher priority than the buffer requests. Therefore, when there is a descriptor fetch request, the DMA arbiter acknowledges the DMA channel that is requesting for a descriptor fetch. If there is no descriptor fetch request, the arbiter looks for packet buffer fetch requests.

The DMA arbiter acknowledges the descriptor fetch request of one DMA channel at a time. Descriptor fetch requests are granted using a fixed priority with the higher channel having higher priority (Channel 1 having priority over Channel 0, Channel 2 having priority over Channel 1 and so on). For packet buffer fetches, the DMA arbiter uses the programmed channel weight and priority to decide which channel to acknowledge. The DMA arbiter performs a burst-by-burst arbitration based on one of the following algorithms, which can be selected by programming the TAA field of the DMA_Mode register. For details, see the DMA_Mode register description.

- Weighted Strict Priority (WSP): In WSP arbitration mode, the arbiter first processes Channel 7 (or the last selected channel) followed by Channel 6, Channel 5, and so on. If a channel does not have a frame to transmit, the weight of the channel is reassigned to Channel7 (or the last selected channel). If Channel 7 has no frames to transmit, the remaining weight is assigned to Channel 6 and so on.
- Weighted Round Robin (WRR): In WRR arbitration mode, the arbiter first selects the channel with the highest weight programmed, and then the channel with next highest weight, and so on. If any channel does not have a frame to transmit, the weight of that channel gets equally distributed to all channels that have frames to transmit. The weight of the channel is programmed in the TCW field of the DMA_CHn_TX_CONTROL register.
- Fixed priority (FP): In Fixed priority mode, Channel 0 has the lowest priority and the last selected channel has the highest priority. The weight programmed in the Transmit Control register of a channel is ignored.

In WSP or WRR arbitration, the channel weight corresponds to the number of DMA burst transfers for which the DMA arbiter grants the bus to a channel. When a channel completes all the DMA burst transfers, the arbiter grants the bus to the next channel.

28.4.3.3 Receive DMA

(1) RX DMA Operation

In the Receive path, the DMA reads a packet from the MTL receive queue and writes it to the packet data buffers of the corresponding DMA channel. The ARI data at the start of the frame indicates the channel number to which the current frame must be written. If only one DMA channel is selected, this information is not provided.

Figure 28.6 shows the reception sequence for Rx DMA engine. The following list describes this sequence:

1. The application sets up the Rx descriptors (RDES0-RDES3) and the Own bit (RDES3[31]).
   The application must set the correct value in the Receive Descriptor Tail Pointer register of corresponding DMA channel.
2. When bit 0 of Receive Control register of corresponding DMA channel is set, the DMA enters the Run state.
   The DMA looks for free descriptors based on the Rx Current Descriptor and Descriptor Tail Pointer register values. If there are no free descriptors, the DMA Channel enters the suspend state and goes to step 11.
3. The DMA fetches the next available descriptor in the ring and decodes the receive data buffer address from acquired descriptors.
4. If IEEE 1588 timestamping is enabled and the timestamp is available for the previous packet, the DMA writes the timestamp (if available) to the RDES0 and RDES1 of current descriptor and sets the CTXT field (RDES3[30]).
5. The DMA processes the incoming packets and places these in the data buffers of acquired descriptor.
6. If the current packet transfer is not complete, the DMA closes the current descriptor as intermediate and goes to step 10.
7. The DMA takes the status of the Receive frame from the MTL and writes the status word to current descriptor with the Own bit cleared and the Last Descriptor bit set.
8. The DMA writes the Frame Length to RDES3 and VLAN Tag to RDES0. The DMA also writes the MAC control frame opcode, OAM control frame code, and extended status information (if available) to RDES1 of the last descriptor.
9. If IEEE 1588 Timestamp feature is enabled, the DMA stores the timestamp (if available). The DMA writes the context descriptor after the last descriptor for the current packet (in the next available descriptor).
10. If more descriptors are available in the Rx DMA Descriptor Ring, go to step 3; otherwise, go to the Suspend state (step 11).
11. The Receive DMA exits the Suspend state when a Receive Poll demand is given and the application advances the Receive Tail Pointer register of a channel.

The engine proceeds to step 2 and re-fetches the next descriptor.

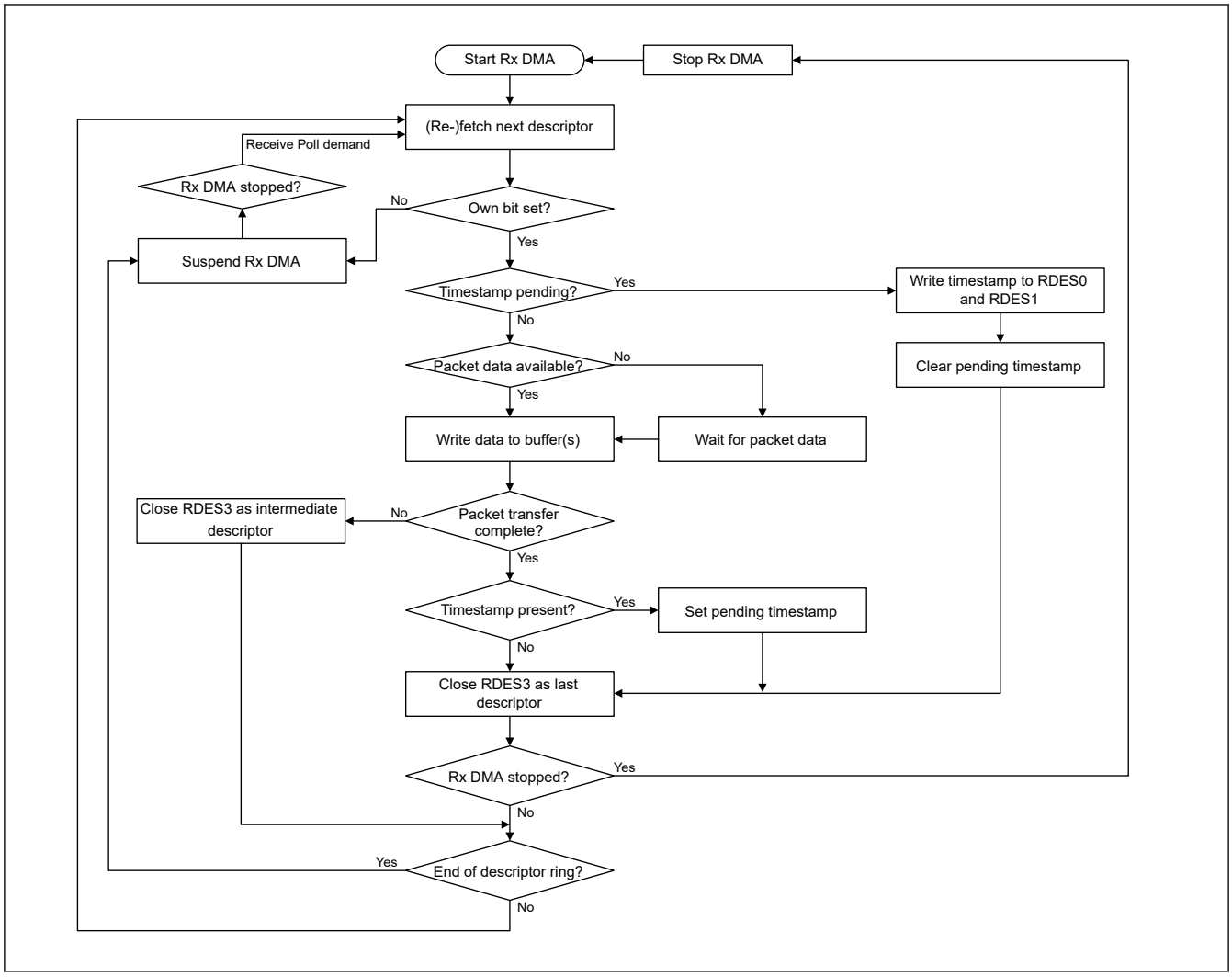


Figure 28.6 Rx DMA operation

(2) Receive Descriptor Acquisition

The Receive Engine always attempts to acquire an extra descriptor in anticipation of an incoming packet. Descriptor acquisition is attempted if any of the following conditions is satisfied:

- Bit 0 of Receive Control Register of corresponding DMA channel is set immediately after being placed in the Run state.
- The Descriptor Tail pointer register value is ahead of the Current Descriptor acquired by the Rx DMA.
- The controller has completed packet reception, but the current Receive Descriptor is not yet closed.
- A Receive poll demand is issued (update of the Tail Pointer register).

(3) Receive Packet Processing

The sequence for processing a Receive packet is as follows:

1. The MAC transfers the received packets to the MTL memory only if the packet passes the address filter. If the packet fails the address filtering, it is dropped in the MAC block (unless bit 31 of MAC_Packet_Filter register is set).
2. If packet size is greater than or equal to configurable threshold bytes set for Rx Queue of MTL, or when the complete packet is written to the queue in the store-and-forward mode, the MTL block requests the DMA block to begin transferring the packet data to the Receive Buffer pointed by the current descriptor.
   Packets smaller than 64 bytes, because of collision or premature termination, are removed from the MTL Receive Queue.
3. When the DMA application Interface (AXI) becomes ready, it transfers the data and sets the following:

- If the packet fits in a single descriptor, the DMA sets both Last Descriptor (RDES3[28]) and First Descriptor (RDES3[29]).
- If the packets fit into more than one descriptor, the DMA sets the First Descriptor (RDES3[29]) to delimit the packet.

4. The DMA releases the descriptors by resetting the Own (RDES3[31]) bit to 0, either because the Receive buffer filled up or the last segment of the packet is transferred to the Receive buffer. The received packets status is updated in the last descriptor.
5. If Interrupt Enabled on Completion (RDES3[30]) bit is set in any of the Descriptors between the First and Last Descriptor of the Packet and bit 6 of Interrupt Enable register of corresponding DMA channel is set, the DMA sets bit 6 of Status register of corresponding DMA channel.

The same process repeats unless the DMA encounters a descriptor flagged as being owned by the application or when there are no more descriptors in the ring. When the DMA finds a descriptor owned by the application and if bit 7 of Interrupt Enable register of corresponding DMA channel is set, the Receive Process sets bit 7 of Status register of corresponding DMA channel and then enters the Suspend state. The position in the receive list is retained.

28.4.4 Transmit and Receive FIFOs

The Transmit FIFO (Tx FIFO) buffers the data transferred from the application to the GMAC. Similarly, the Receive FIFO (Rx FIFO) stores the packets received from the line until they can be transferred to the application. These are asynchronous FIFOs because they also transfer the data between the application clock and the MAC line clocks. TX FIFO and RX FIFO size are 8 KB respectively.

When multiple queues are selected, all Tx queues share the Tx FIFO memory and all Rx queues share the Rx FIFO memory. The application can program the size of the FIFO memory allocated to each Tx or Rx queue.

28.4.5 MAC Transaction Layer

The MAC Transaction Layer (MTL) provides the FIFO memory Interface to buffer and regulate the packets between the application system memory and the MAC. It also enables the data to be transferred between the application clock and MAC clock domains. The MTL layer has two data paths: Transmit path and Receive Path. The data path for both directions is 128-bit wide.

The MTL communicates with the application through Application Transmit Interface (ATI) on the Transmit path and Application Receive Interface (ARI) on the Receive path. The MTL also provides the MAC Control Interface (MCI) as a control path.

28.4.5.1 Transmit Path

The internal DMA handles all transactions for the Transmit path through ATI.

The internal DMA pushes the Ethernet packets read from the system memory into the corresponding queue. The packet is then popped out and transferred to the MAC when the queue threshold is reached (threshold mode) or complete packet is in the queue (store-and-forward mode). When EOP is transferred, the status of the transmission is taken from the MAC and transferred back to the internal DMA.

The Tx queue has a default size of 2 KB. The fill level of the queue is indicated to the internal DMA (using PBL and watermark) so that it can initiate a data fetch in required bursts from the system memory. The internal DMA indicates the SOP and EOP as packet delimiters through ATI interface. In configurations with multiple queues, the internal DMA should also indicate the queue number for which the transaction is being addressed.

(1) Transmit Control Word

The following control information related to packet transmission is provided as a part of the Control Word through ATI interface:

- CRC Pad Control
- TCP/IP Checksum Insertion Control
- One-step Timestamping Control Correction
- Transmit Timestamp Enable

(2) Transmit Operation

The following two modes of operation trigger reading of the data towards the MAC:

- Threshold mode:
  In Threshold (or cut-through) mode, as soon as the number of bytes in the Queue cross the configured threshold level (or when the end of packet is written before the threshold is crossed), the data is ready to be popped out and forwarded to the MAC. The threshold level is configured by using the TTC bits of MTL_TxQ0_Operation_Mode register corresponding to an MTL queue.

- Store-and-forward mode:
  In store-and-forward mode, the MTL pops out the packet towards the MAC only when one or more of the following conditions are true:
  - A complete packet is stored in the Queue
  - The Tx FIFO becomes almost full
  - The ATI watermark becomes low

  The watermark becomes low when the requested Queue does not have space to accommodate the requested burst length on the ATI. Therefore, the MTL when operating in the store-and-forward mode allows packet transmission even if the packet length is bigger than the Tx Queue size.

The application can flush complete content of the Tx Queue by setting bit 0 (FTQ) of Transmit Operation Mode register corresponding to an MTL queue. This bit is self-clearing and initializes the Queue pointers to the default state. If the FTQ bit is set during a packet transfer from the MTL to the MAC, the MTL stops further transfer because the queue is considered to be empty. Therefore, an underflow event occurs at the MAC transmitter.

For information about the initialization and transmit operations for the MTL Layer, see (3) Initialization Flow through (8) Transmit Queue Flush Operation.

(3) Initialization Flow

Upon reset, the MTL is ready to manage the flow of data to and from the DMA and MAC. You need to initialize the Queue size for each of the queues by programming the TQS bits of MTL_TxQ0_Operation_Mode register corresponding to a Transmit queue. You also need to initialize the MAC block. The internal DMA controllers must be individually enabled through their respective CSRs.

(4) Single-Packet Transmit Operation

The following flow is valid when the GMAC is operating in non-OSP mode, that is, when bit 4 (OSP) of the DMA_CHn_TX_CONTROL (n = 0 to 7) register of the DMA channel is set to 0.

During a Transmit operation, the MTL block is a slave for the DMA controller. The general sequence of events for a Transmit operation is as follows:

1. If the system has data to be transferred, the DMA controller fetches the data from the application through the AXI master interface and starts forwarding it to the MTL. The DMA descriptor memory contains the information related to packet control which is used to drive the ATI Control Word. The DMA data buffer memory contains the packet data.
2. The MTL pushes the data received from the DMA into the corresponding queue. This process continues until the EOP is transferred.
3. When the threshold level is crossed or a full packet of data is received into the queue, the MTL reads the packet data and drives it to the MAC.
4. The queue controller continues to transfer data from the queue until a complete packet is transferred to the MAC.
5. When the packet transfer is complete, the MTL receives the status from the MAC and notifies the DMA controller.

Figure 28.7 shows the MTL single-packet transmit operation.

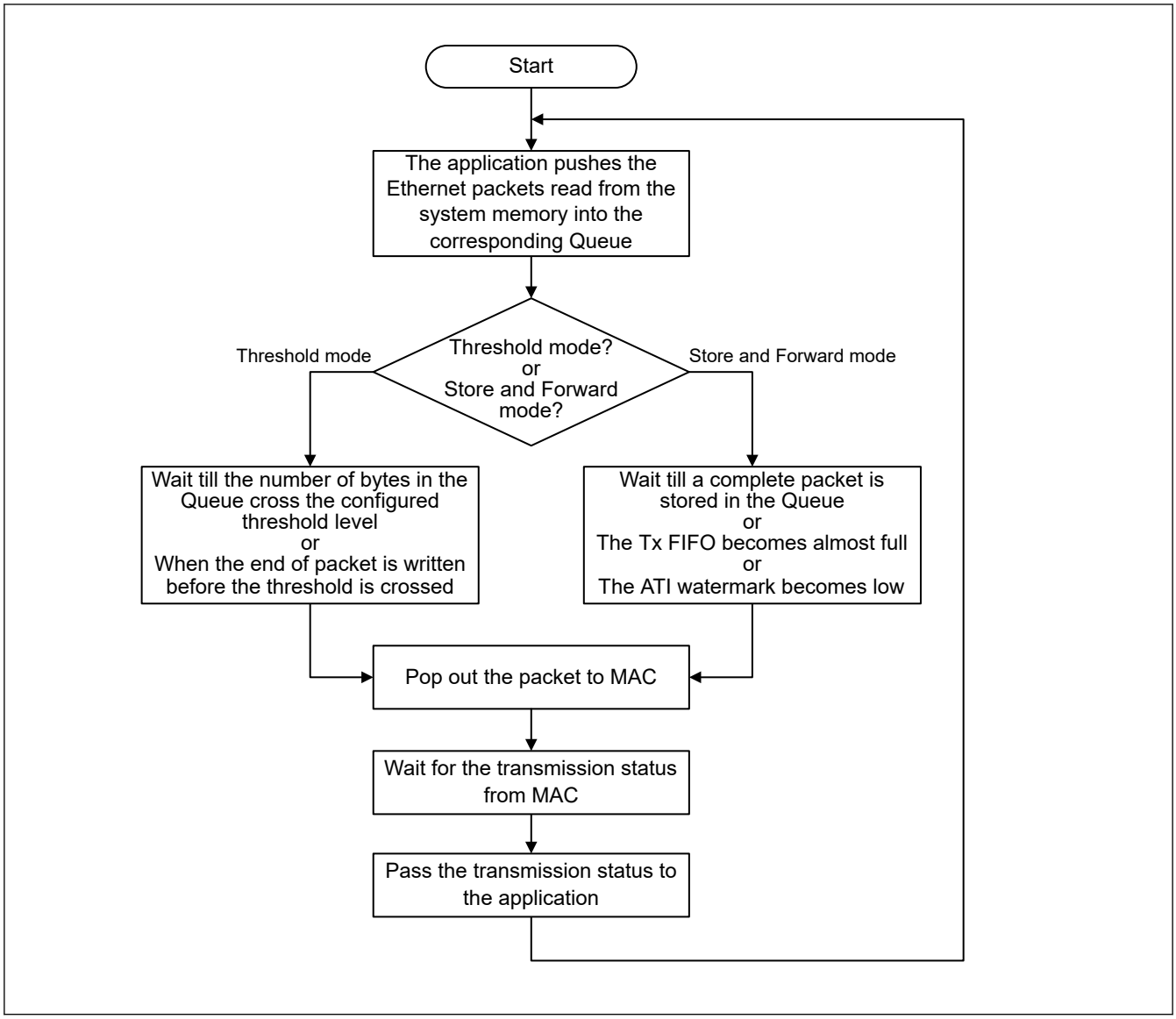


Figure 28.7 MTL single packet transmit flow

Note: When bit 1 (DTXSTS) of MTL_Operation_Mode register is set, the MTL does not provide any status to the DMA.

(5) Transmit Operation - Two Packets in the Buffer

The following flow is valid when the GMAC is operating in OSP mode, that is, when bit 4 (OSP) of the DMA_CHn_TX_CONTROL (n = 0 to 7) register of the DMA channel is set to 1.

The flow is similar to the flow described in (4) Single-Packet Transmit Operation. However, after fetching a packet DMA instead of waiting for the status, it continues to fetch another packet if available in the system memory. So, MTL can receive the second packet while it is processing the first packet. This flow improves the performance because the DMA can process two packets back to back before waiting for the status of the first packet.

(6) Relationship Between the Number of Tx DMA Channels and Tx Queues

The number of Tx DMA channels is always equal to the Tx queues. This is because the DMA is designed to arbitrate among multiple channels (to fetch the descriptor and packet data from the system memory) in terms of PBLs (Programmable Bursts Lengths). However, the Tx queues are packet-based storage memory. Therefore, one to one mapping is required between the DMA channels and Tx queues to maintain the packet-level coherency.

(7) Retransmission During Collision

While a packet is being transferred from the MTL to the MAC, a collision event can occur on the MAC line interface in half-duplex mode. The MAC indicates a Retry attempt to the MTL by giving the status even before the EOP is transferred from the MTL. The MTL then enables the retransmission by popping out the packet again from the queue.

After more than 96 bytes (or 548 bytes in 1000-Mbps mode) are read out towards the MAC, the Queue Controller frees up that space and makes it available to the application or the DMA to push in more data. This means that the retransmission is not possible after this threshold is crossed or when the MAC indicates a late-collision event.

When a packet transmission is aborted because of underflow and a collision event immediately follows (initiating a retry), retry has higher priority than abort.

(8) Transmit Queue Flush Operation

The MTL allows a Tx Queue to be flushed at any moment through bit 0 of MTL_TXQn_OPERATIOn_MODE (n = 0 to 7) register.

1. The application pushes two packets into the Tx queue on the ATI interface.
2. The application performs a write to the Transmit Operation mode register of corresponding MTL queue and sets the Flush Transmit Queue bit.
3. The application reads back the same register to check the status of the Flush operation. The bit [0] is asserted high during CSR read access, indicating the progress of the Flush operation.
4. If the MTL has already transferred a (partial) packet to the MAC, then it terminates that immediately and clears the Tx Queue. The MTL waits for the status of the partial/full packet transfers to the MAC.
5. The MTL indicates the completion of the Flush operation by asserting internal signal, which clears bit 0 in Transmit Operation Mode register of corresponding MTL queue.

28.4.5.2 Receive Path

The MTL Rx module receives the packets from the MAC and pushes them into the Rx Queue. The status (fill level) of the queue is indicated to the DMA when it crosses the configured Receive threshold (RTC bits [1:0] of MTL_RxQ0_Operation_Mode register of corresponding MTL queue), or the complete packet is received. The MTL also indicates the fill level of the queue so that the DMA can initiate preconfigured burst transfers towards the AXI interface.

(1) Receive Operation

During a Receive operation, the MTL is a slave for the MAC. The general sequence of events is as follows:

1. When the MAC receives a packet, it indicates the availability of receive data.
2. The MAC indicates the SOP and EOP delimiters.
3. The MTL accepts the data and pushes it into corresponding Rx queue.
4. After the EOP is transferred, the MAC drives the status word which is also pushed into the corresponding Rx queue by the MTL.
   In Threshold (cut-through) mode, the status words are stored after the packet EOP. In store-and-forward mode, the location for the maximum status words are reserved before writing the SOP and the status is written to reserved locations after writing the EOP.
5. If IEEE 1588 timestamp feature is enabled and the 64-bit timestamp is available along with the packet status, it is pushed into the Rx queue as a part of the status word. Therefore, one additional location is taken per packet to store the timestamp in the Rx queue.
6. The MTL takes the data out of the queue and sends it to the DMA depending on the mode.

(2) Threshold Mode

In the (default) Threshold mode, the MTL reads the data and indicates its availability to the DMA when one of the following occurs:

- Data bytes equal to the threshold amount are written to the Rx queue (RTC bits [1:0] of MTL_RxQ0_Operation_Mode register of corresponding MTL queue)
- A full packet of data is received into the queue

Note: The MTL operates in dynamic threshold mode even though it is programmed for threshold mode. In dynamic threshold mode, MTL starts forwarding to the application only after it receives threshold number of bytes, split-header length, and DMA number from MAC.

(3) Store-and-Forward Mode

In the store-and-forward mode (when bit 5 of MTL_RxQ0_Operation_Mode register of a queue is set to 1), the initial Rx queue locations are reserved for the status words before writing the SOP. A packet is read out only after it is completely written into the Rx queue. In this mode, all error packets are dropped (if configured through bit 4 of MTL_RxQ0_Operation_Mode register of a queue) such that only valid packets are read and forwarded to the application.

(4) Multi-Packet Receive Operation

In Threshold mode, the packet status is available immediately after the packet data. In store-and-forward mode, the packet data is available after the packet status. The MTL is capable of storing any number of packets into the queue as long as it is not full.

If the MAC receives a packet when the corresponding Rx queue is full, the MTL ignores that packet. In addition, the MTL increments the overflow counter in the MTL_RxQ0_Missed_Packet_Overflow_Cnt of corresponding queue.

(5) Error Handling in Receive Operation

If the MTL Rx queue is full before it receives the EOP data from the MAC, the following happens:

1. An overflow is declared

2. The whole packet (including the status word) is dropped

3. The overflow counter in the DMA (Overflow Counter register of corresponding MTL queue) is incremented.

This is true even if bit 4 (FEP) of MTL_RxQ0_Operation_Mode register of corresponding MTL queue is set.

If the start address of such a packet has already been transferred to the Read Controller, the rest of the packet is dropped and a dummy EOP is written to the queue along with the status word with overflow status. The status indicates a partial packet because of overflow. In such packets, the Packet Length field is invalid. If the MTL Receive Queue is configured to operate in the store-and-forward mode and the length of the received packet is more than the queue size, overflow occurs and all such packets are dropped.

The MTL Rx Control logic can filter error and undersized packets, if enabled by using the FEP and FUP bits of MTL_RxQ0_Operation_Mode register of corresponding MTL queue. If the start address of such a packet has already been transferred to the Rx Queue Read Controller, that packet is not filtered. The start address of the packet is transferred to the Read Controller after the packet crosses the receive threshold set by bits [1:0] of MTL_RxQ0_Operation_Mode register of corresponding MTL queue.

If the MTL Receive Queue is configured to operate in the store-and-forward mode, all error packets can be filtered and dropped. The MTL then stops transferring data to the DMA. It internally reads the rest of the packet and drops it. The MTL then starts the transfer of next packet (if available).

28.4.6 MAC

28.4.6.1 MAC Transmission

The MAC communicates with the application side with the MAC Transmit Interface (MTI), MAC Receive Interface (MRI), and MAC Control Interface (MCI).

The MAC transmission process is as follows:

1. Transmission is initiated when the MTL application pushes in data with the SOP signal asserted.
2. When the SOP signal is detected, the MAC accepts the data and begins transmitting to the GMII or MII.
   The time required to transmit the packet data to the GMII or MII after the application initiates the transmission depends on delay factors such as IPG delay, time to transmit preamble or SFD, and any back-off delays for half-duplex mode. While the packets data is being transmitted, the MAC can stop accepting the data received from the MTL.
3. After the EOP is transferred to the MAC, the MAC does one of the following:
   - The MAC completes the normal transmission and gives the transmission status to the MTL.

- If a normal collision (in half-duplex mode) occurs during transmission, the MAC gives the Transmit Status with retry bit set to the MTL. The MAC gives the Retry request till one of the following is true:
  - Packet is successfully transmitted
  - Maximum retry requests expire
    When maximum retry requests expire, the MAC aborts the packet transmission with Excessive Collision Transmit Status. The MAC accepts and drops all further data until the next SOP is received. The MTL block should retransmit the same packet from SOP on observing a Retry request (in the Status) from the MAC.
- If any one of the following occurs, the MAC aborts the packet transmission:
  - No carrier (half-duplex mode)
  - Loss of carrier (half-duplex mode)
  - Excessive deferral (half-duplex mode)
  - Late collisions (half-duplex mode)
  - Jabber

  The MAC accepts and drops all further data until the next SOP is received.

4. The MAC issues an underflow status if the MTL is not able to provide the data continuously during the transmission. The MAC accepts and drops all further data until the next SOP is received.
5. During the normal transfer of a packet from MTL, if the MAC receives a SOP without getting an EOP for the previous packet, it ignores the SOP and considers the new packet as continuation of the previous packet.

Figure 28.8 shows the MAC transmission process flow.

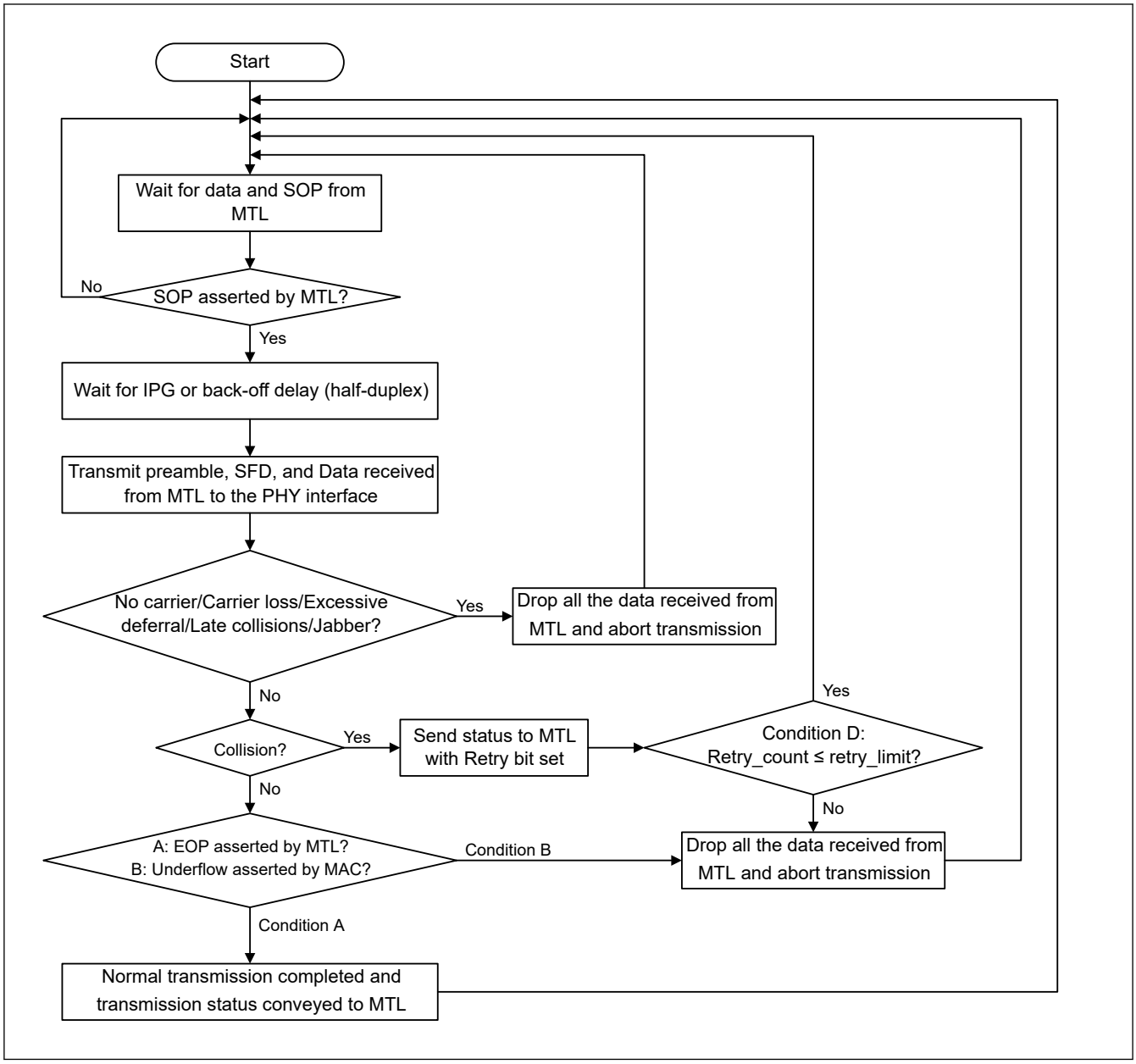


Figure 28.8 Overview of MAC transmission process flow

(1) Transmit Bus Interface Module

The MAC Transmit Bus Interface (TBU) accepts data in 128-bit wide bus and runs on the TX_CLK of GMII interface. The TBU module performs the following functions:

- Outputs the Transmit status to the application at the end of normal transmission or collision
- Outputs the Transmit snapshot register value
- Performs the Endian conversion of data bus by swapping the byte lanes and corresponding byte enables
- Converts the input data into an 8-bit bus towards the Transmit Packet Controller

(2) Transmit Packet Controller Module

The Transmit Packet Controller (TPC) module consists of eight registers to hold the data and the last data control received from the TBU.

The register provides a buffer between the application and the Transmit Protocol Engine (TPE) to regulate data flow.

When the number of bytes received from the application are less than 60 (DA+SA+LT+DATA), the state machine that interfaces with TBU automatically appends zeros to the packet being transmitted. This is done to make the data length

exactly 46 bytes to meet the minimum data field requirement of IEEE 802.3. You can program the MAC to not append any padding.

The cyclic redundancy check (CRC) for the Frame Check Sequence (FCS) field is calculated before transmission to the TPE module. This value is computed by the CTX module. The TPC module receives the computed CRC and appends it to the data being transmitted to the TPE module. When the MAC is programmed to not append the CRC value to the end of Ethernet packets, the TPC module ignores the computed CRC and transmits only the data received from the TBU module to the TPE module. An exception to this rule is that when the MAC is programmed to append pads for packets (DA+SA+LT+DATA) less than 60 bytes sent by the TBU module, the TPC module appends the CRC at the end of padded packet irrespective of the setting value.

(3) Transmit Protocol Engine Module

The Transmit Protocol Engine (TPE) module consists of a Transmit State Machine that controls the operation of Ethernet packet transmission.

The Transmit State Machine of this module contains the following features to meet the IEEE 802.3/802.3z specification:

- Generates preamble and SFD
- Generates jam pattern in the half-duplex mode after normal collision
- Generates carrier extension in the half-duplex (only in the GMII) mode when packet is smaller than 512 bytes
- Supports packet bursting in the half-duplex (only in the GMII) mode
- Supports jabber timeout
- Supports flow control for the half-duplex mode (backpressure)
- Generates Transmit packet status
- Contains timestamp snapshot logic for IEEE 1588 support

When the TPC module requests the TPE module for a new packet transmission, the Transmit State Machine sends out the preamble and SFD, followed by the data received. The preamble is defined as 7 bytes of 10101010b pattern and the SFD is defined as 1 byte of 10101011b pattern.

The collision window is defined as 1 slot time (512-bit times for 10/100 Mbps Ethernet and 4,096-bit times for 1,000 Mbps Ethernet). The jam pattern generation is applicable only to half-duplex mode, not to full duplex mode. In full-duplex mode, the Transmit State Machine ignores the COL signal from the PHY.

In MII mode, if a collision occurs any time from the beginning of the packet to the end of the CRC field, the Transmit State Machine sends a 32-bit jam pattern of 0x55555555 on MII to inform all other stations that a collision has occurred. If the collision is seen during the preamble transmission phase, the Transmit state machine completes the transmission of preamble and SFD, and then sends the jam pattern.

In GMII mode, if a collision occurs any time between the beginning of the packet and the end of the extension field, the Transmit State Machine sends a 32-bit jam pattern of 0x55555555 on GMII to inform all other stations of the collision. If the collision is seen during the preamble transmission phase, the Transmit State Machine completes the transmission of preamble and SFD, and then sends the jam pattern. If a collision occurs during the extension field, the Transmit State Machine sends a 32-bit jam pattern of 0x1F1F1F1F.

If the collision occurs after the collision window and before the end of the FCS field (or the end of Burst if the Packet Burst mode is enabled), the Transmit State Machine sends a 32-bit jam pattern and sets the late collision bit in the Transmit packet status.

Note: At the GMII or MII interface, the collision signal being asynchronous is checked by the transmitter after it is double-synchronized to TX_CLK domain. This additional latency delays the recognition of collision or late-collision event. When the output of COL signal synchronizer is asserted after the complete packet is transmitted, it is not recognized as collision even if the COL signal is high at the GMII interface before the end of transmission. Similarly, an assertion of COL signal at the GMII input at the last byte of normal collision window might be identified as late collision because of the synchronizer delay.

In GMII half-duplex mode (1000 Mbps), the Transmit State Machine ensures that all valid carrier events exceed a slot time of 4,096-bit times. To accomplish this, any Transmit packet shorter than 512 bytes from the TFC module is extended using a carrier extension. On GMII, this is signaled to the PHY by asserting TXER, de-asserting TXEN, and setting TXD[7:0] to 0x0F.

When the Packet Burst mode is enabled, only the first packet of the burst is carrier extended if it is shorter than 512 bytes. The carrier extension is not applicable for MII half-duplex and GMII or MII full-duplex modes. When the Packet Burst mode is enabled, the Transmit State Machine transmits a burst of packets (as long as packets are available from the TFC module) without releasing the carrier of the PHY. To accomplish this, the state machine inserts the carrier extension for a minimum IPG period (96-bit times) between the packets. The Transmit State Machine continues to burst packets as long as additional packets are available from the TPC module and a burst limit of 8,192-byte times has not been exceeded. If an additional packet is not available at the end of the IPG period in the middle of the burst, the Transmit State Machine releases the carrier by de-asserting TXER and TXEN signals on GMII.

Packet bursting is applicable only for the GMII half-duplex mode. It is not applicable in the MII and GMII full-duplex modes. In the GMII half-duplex mode, the size of the first packet in a packet burst should at least be equal to the slot time. If the first packet (including carrier extension) is less than the slot time, the MAC considers all data bytes, received for the first packet and subsequent packets, which end immediately after the slot time is reached as first packet in the burst. This may result in CRC error. However, if the packet burst ends (both RXDV and RXER go low) before the slot time, the subsequent packet is considered as a new packet. This behavior is according to the IEEE 802.3.

The TPE module maintains a jabber timer to stop the transmission of Ethernet packets if the TFC module transfers more than 2,048 (default) bytes. The timeout is changed to 10,240 bytes when the Jumbo packet is enabled.

The Transmit State Machine uses the deferral mechanism for flow control (backpressure) in the half-duplex mode. When the application requests to stop receiving packets, the Transmit State Machine sends a JAM pattern of (0x55) 32 bytes whenever it senses a reception of a packet, provided the Transmit flow control is enabled. This results in a collision and the remote station backs off. The application requests the flow control by setting BPA bit of the Flow Control Register of corresponding MTL queue. If the application requests a packet to be transmitted, it is scheduled and transmitted even when the backpressure is activated. If the backpressure is kept activated for a long time (and more than 16 consecutive collision events occur), the remote stations abort their transmissions because of excessive collisions.

If IEEE 1588 timestamp is enabled for the Transmit packet, this block takes a snapshot of the system time when the SFD is put onto the Transmit GMII or MII bus. The system time source is an external input from ETHSW.

Note: The Collision input is an asynchronous signal and should be asserted for at least 32-bit times (4 clocks for 1Gbps, and 8 clocks for slower speeds).

(4) Transmit Module

The Transmit (STX) module is responsible for scheduling the packet transmission on GMII or MII. It provides an enable signal to the TPE module after satisfying the IPG and back-off delays. The STX module performs the following functions:

- Maintains the inter-packet gap between two transmitted packets
  The STX module maintains an idle period of the configured inter-packet gap (IPG bits of MAC_Configuration register) between any two transmitted packets. If packets from the TPC arrive at the TPE module sooner than the configured IPG time, the TPE module waits for the enable signal from the STX module before starting the transmission on GMII or MII. The STX module starts its IPG counter as soon as the carrier signal of GMII or MII goes inactive. At the end of programmed IPG value, the module issues an enable signal to the TPE module in the full-duplex mode.
  In the half-duplex mode and when IPG is configured for 96-bit times, the STX module follows the rule of deference specified in the IEEE 802.3, Section 4.2.3.2.1. The module resets its IPG counter if a carrier is detected during the first two-thirds (64-bit times for all IPG values) of the IPG interval. If the carrier is detected during the final one-third of the IPG interval, the STX module continues the IPG count and enables the transmitter after the IPG interval.
- Implements the Truncated Binary Exponential Back-off algorithm
  The STX module implements the Truncated Binary Exponential Back-off algorithm when it operates in the half-duplex mode.

(5) Transmit CRC Generator Module

The MAC Transmit CRC Generator (CTX) module interfaces with the TFC module to generate CRC for the FCS field of the Ethernet packet.

The Transmit CRC Generator (CTX) module interfaces with the TFC module to generate CRC for the FCS field of the Ethernet packet.

The TPC module sends the packet data and any necessary padding to the CTX module through an 8-bit interface.

The CTX module calculates the 32-bit CRC for the FCS field of the Ethernet packet. The encoding is defined by the following generating polynomial:

G (x) = x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1

The CTX module gets the byte data of Ethernet packet from the TPC module (DA + SA + LT + DATA + PAD) qualified with a Data Valid signal. The TPC also indicates to the CTX when to reset the previously calculated CRC and to start the new CRC calculation for the coming packet. The TPC module issues the start command before sending the new packet data for calculation. The calculated CRC is valid on the next clock after the data is received.

In GMII mode, the Data Valid signal is valid for every clock, from the first data byte through the last data byte. In MII mode, this signal is valid every alternate clock.

(6) MAC Transmit Flow Control Module

The Transmit Flow Control (FTX) module generates and transmits the Pause packets to the TFC module based on the flow control triggers in full-duplex mode. The TFC module receives the Pause packet from the FTX module, appends the calculated CRC, and sends the packet to the TPE module. For more information about flow control, see section 28.15. Flow Control.

28.4.6.2 MAC Reception

A receive operation is initiated when the MAC detects an SFD on GMII or MII. The MAC strips the preamble and SFD before proceeding to process the packet. The header fields are checked for filtering and the FCS field used to verify the CRC for the packet. The received packet is stored in a shallow buffer until the address filtering is performed. The packet is dropped in the MAC if it fails the address filter.

(1) Receive Protocol Engine Module

The Receive Protocol Engine (RPE) consists of the Receive State Machine which strips the preamble, SFD, and carrier extension of the received Ethernet packet (in half-duplex 1000-Mbps mode).

Figure 28.9 shows the Receive transmission flow in the RPE.

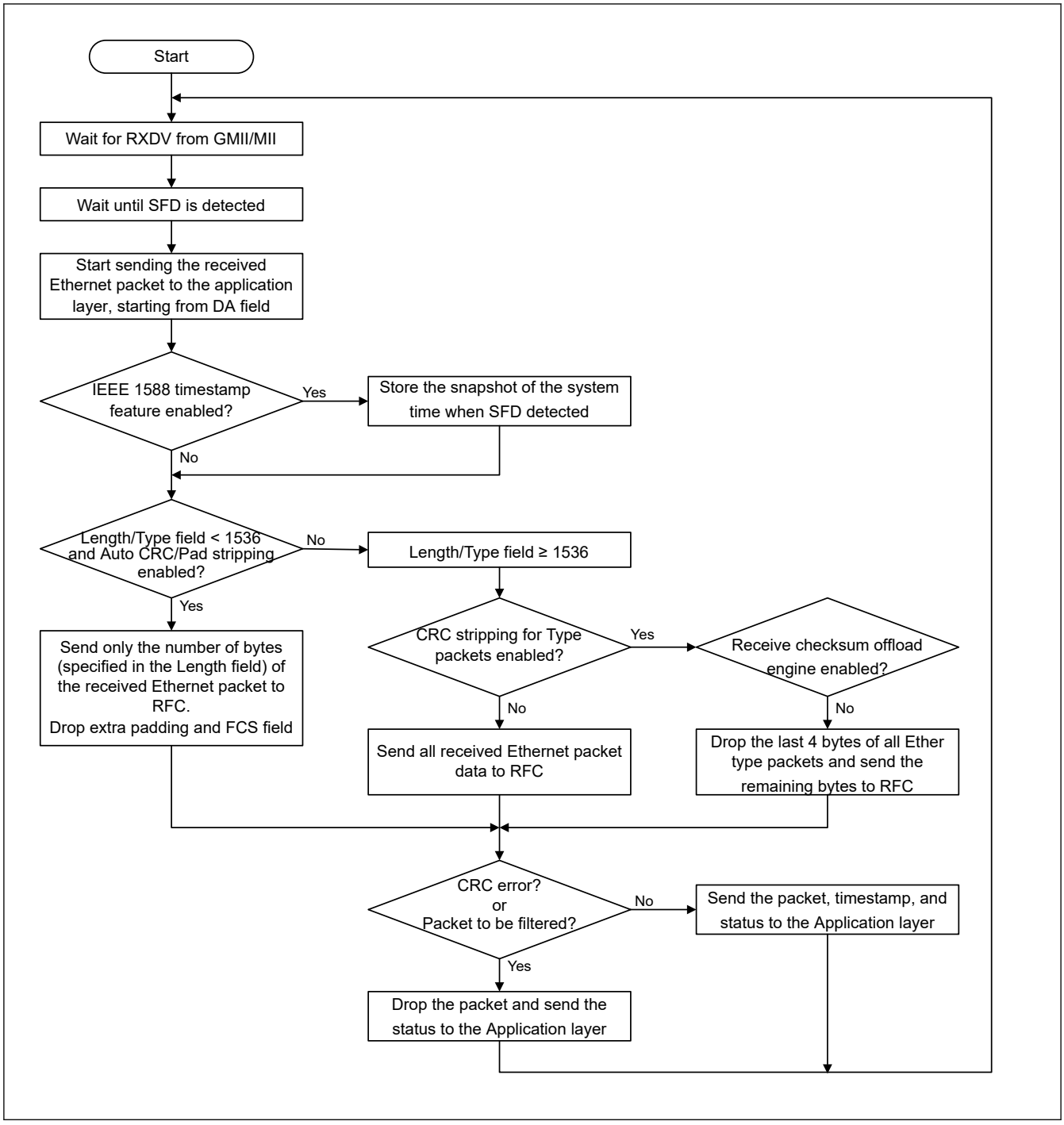


Figure 28.9 MAC receive flow transmission

The sequence is as follows:

1. When the RXDV signal of GMII or MII becomes active, the Receive State Machine of RPE starts looking for the SFD field (byte 0x5D in GMII mode; 0xD nibble in MII mode).
   The state machine drops received packets until it detects SFD.
2. When SFD is detected, the state machine begins sending the data of Ethernet packet to the RPC module, beginning with the first byte following the SFD (destination address).
3. If IEEE 1588 Timestamp feature is enabled, the RPE takes a snapshot of the system time at which SFD of any packet is detected on GMII or MII. If this packet is not dropped during MAC filtering, the timestamp is passed to the application. In MII mode, the RPE converts the received nibble data into bytes and forwards the valid packet data to the RFC module.
4. The Receive State Machine of the RPE module decodes the Length/Type field of the receiving Ethernet packet.

If the Length/Type field is less than 1,536 and if the MAC is programmed for the Auto CRC/Pad Stripping (Bit 20 of the MAC_Configuration register), the state machine sends the data of the packet up to the count specified in the Length/Type field and starts dropping bytes (including the FCS field). The state machine of the RPE module decodes the Length/Type field and checks for the Length interpretation.
In Audio Video (AV) mode, when you select additional Rx queues, the packets that are less than or equal to 16 bytes in length after pad stripping, always get dropped inside the MAC receiver. This happens even if the packets have passed the address filter and have no CRC error.
If the Length/Type field is greater than or equal to 1,536, the RPE module sends all received Ethernet packet data to the RFC module if you have not enabled the CRC stripping for Type packet in bit 21 of the MAC_Configuration register. However, if you have enabled the CRC stripping for Type packets and not enabled the Receive Checksum Offload Engine, the MAC strips and drops the last 4 bytes of all packets of Ether type before forwarding the packets to the application.

5. By default, the MAC is programmed for watchdog timer to be enabled, that is, packets larger than 2,048 (10,240 if Jumbo Packet is enabled) bytes, (DA + SA + LT + DATA + PAD + FCS) are cut off at the RPE module. In addition, you can use a programmable watchdog timer (Bit 16 of MAC_Watchdog_Timeout register) to override the fixed timeout of 2,048 or 10,240 bytes. You can disable the watchdog timer by programming bit 19 of MAC_Configuration register. However, even if the watchdog timer is disabled, a packet greater than 32 KB size is cut off and a watchdog timeout status is given.

At the end of every received packet, the RPE module generates received packet status and sends it to the RPC module. Control, missed packet, and filter fail status are added to the receive status in the RPC module.

Note:
- In half-duplex mode, the first packet in a burst should be at least slot time in length. If the first packet is smaller than the slot time, the MAC considers the bytes received up to the slot time as first packet, which results in CRC error for the packet.
- Packet bursting is applicable only for the GMII half-duplex mode. It is not applicable in the MII and GMII full-duplex modes. In the GMII half-duplex mode, the size of the first packet in a packet burst should at least be equal to the slot time. If the first packet (including carrier extension) is less than the slot time, the MAC considers all data bytes, received for the first packet and subsequent packets, which end immediately after the slot time is reached as first packet in the burst. This may result in CRC error. However, if the packet burst ends (both RXDV and RXER go low) before the slot time, the subsequent packet is considered as a new packet. This behavior is according the IEEE 802.3.

(2) Receive CRC Module

The Receive CRC (CRX) interfaces with the RPE module to check any CRC error in the packet being received. This module calculates the 32-bit CRC for received packet that includes the Destination address field through the FCS field.

The encoding is defined by the following generating polynomial:

G (x) = x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1

The module gets the data from the RPE module (DA+SA+LT+DATA+PAD+FCS). The RPE module also sends a control signal that indicates the validity of the data. Irrespective of the Auto Pad or CRC strip, the CRX module receives the entire packet to compute the CRC check for received packet.

(3) Receive Packet Controller Module

The Receive Packet Controller (RPC) receives the Ethernet packet data and status from the RPE module.

The RPC module consists of a FIFO and two state machines for writing and reading the FIFO. The FIFO holds the received Ethernet packet data and byte enables, along with a control bit to indicate the last data. The state machines manage the FIFO and provide a packet buffering for the Ethernet packet being received from the RPE module. The following are main functions of the RPC module:

- Converting Data path — converts 8-bit data to 32-bit data to the RBU module
- Packet filtering
- Attaching the calculated IP Checksum input from IPC
- Updating the Receive Status and forwarding it to RBU

If RA bit of the MAC_Packet_Filter register is set, the RPC module initiates the data transfer to the RBU module when 4 bytes of Ethernet data are received from the RPE module. At the end of the data transfer, the RPC module sends out the

received packet status that includes the packet filter bits (SA Filterfail and DA Filterfail) and status from the RPC module. These bits are generated based on the filter-fail signals from the AFM module. This status bit indicates to the application whether the received packet has passed the filter controls (both address filter and Packet Filter controls from CSR). The RPC module does not drop any packet on its own in this mode.

If the RA bit is reset, the RPC module performs packet filtering based on the destination or source address. If the application does not want to receive any bad packets such as runt, CRC error packets, the application still needs to perform another level of filtering. The RPC module waits to receive the first 14 bytes of received data (type field) from the RPE module. Until then, the module does not initiate any transfers to the RBU module. After receiving the destination or source address bytes, the RPC checks the filter-fail signal from the AFM module for an address match. On detecting filter-fail from AFB, the packet is dropped at the RPC module and not transferred to the application.

On a delayed filter response from the AFM (this can only occur if you change the AFM logic), the RPC module waits until the FIFO is full, and then proceeds with the packet transfer to the RBU module. However, the RPC module still takes the delayed response from the AFM module and if it is a (DA or SA) filter failure, it drops the rest of the packet and sends the Rx Status Word (with zero packet-length, CRC Error and Runt Error bits set) immediately indicating the filter fail. If there is no response from the AFM until EOP is transmitted, the filter fail status in the Rx Status Word is updated accordingly.

When the optional PMT module is present and configured for power-down mode, this block drops all received packets and does not forward the packets to the application.

(4) Receive Flow Control Module

The Receive Flow Controller (FRX) detects the Pause packet being received and pauses the packet transmission for the delay specified within the received Pause packet. The FRX module is enabled only in the full-duplex mode.

(5) Receive Bus Interface Unit Module

The Receive Bus Interface Unit (RBU) converts the 32-bit data received from the RPC module into a 128-bit FIFO protocol. The RBU module interfaces with the MTL through the MAC receive interface (MRI). The RBU module also outputs the timestamp captured from the received packet, along with the status.

(6) Address Filtering Module

The Address Filtering (AFM) module performs the destination and source address checking function on all received packets and reports the address filtering status to the RPC module.

The address checking is done based on different parameters (Packet Filter register) chosen by the application. These parameters are inputs to the AFM module, and the AFM module reports the status of the address filtering based on the combination of these inputs. The AFM module does not filter the receive packets but reports the status of the address filtering (whether to drop the packet or not) to the RFC module. The AFM module also reports address filter status and whether the received packet is a multicast packet or a broadcast packet.

The AFM module probes the 8-bit receive data path between the RPE module and the RFC module and checks the destination and source address field of each incoming packet. The AFM module gets the physical (MAC) address of the station and the Multicast Hash table from CSR module for address checking. The CSR module provides the Packet Filter register parameters to AFM.

28.4.7 Interrupts

Interrupts can be generated as a result of various events in the GMAC controller. These events are captured in status registers, and interrupt enables are provided for each source of an interrupt such that the interrupt signal (GMAC_SBD) is asserted for an event only when the corresponding interrupt enable is set.

The interrupt status and corresponding enable registers are organized in an hierarchical manner so that it is easier for software to traverse and identify the source of an interrupt event quickly. When GMAC_SBD is asserted, the DMA_Interrupt_Status register is the first level that indicates the major blocks for the interrupt event source. This register is read-only, and it contains bits corresponding to each DMA channel (TX and RX pair), the MTL, and the MAC. The software application must then read one (or more) of the following registers corresponding to the bits that are set:

- MAC_Interrupt_Status register
- MTL_Interrupt_Status register
- DMA_CHn_STATUS (n = 0 to 7) register

28.4.7.1 Interrupts from the MAC

Interrupts can be generated from the MAC as a result of various events in the MAC Receiver, Transmitter, or the optional modules/functions such as RMON counters, EEE and so on.

These interrupt events are combined with the events in the DMA on the GMAC_SBD signal. The MAC interrupts are of level type, that is, the interrupt remains asserted (high) until it is cleared by the application or software.

The MAC_Interrupt_Status register describes the events that can cause an interrupt from the MAC. The MAC interrupts are disabled by default. Each event can assert the interrupt on the GMAC_SBD signal when the corresponding bit is set in the MAC_Interrupt_Enable register.

The interrupt register bits only indicate the block from which the event is reported. You must read the corresponding status registers and other registers to clear the interrupt.

Note: By default, the MAC interrupt status bits are cleared when the register that contains the source of the interrupt is read. If RCWE bit in the MAC_CSR_SW_Ctrl register is programmed to 1, the MAC interrupt status bits are cleared when the bit that contains the source of the interrupt is explicitly written to 1.

28.4.7.2 Interrupts from MTL

GMAC can generate interrupts as a result of events in the MAC layer or in the MTL modules. These interrupt events are combined with the events in the DMA on the GMAC_SBD signal. The MTL interrupts are of level type, that is, the interrupt remains asserted (high) until it is cleared by the application or software. In the MTL, the interrupts are mainly related to exception events in the TxQ or RxQ in the Transmit and Receive paths respectively. The interrupt status are captured and organized in a hierarchical manner in order to identify the root cause quickly.

The MTL_Interrupt_Status register identifies the top level modules that can cause the interrupt to be asserted. Bits [7:0] identify 8 Queues (Tx or Rx). They are read-only bits and the application should read the corresponding MTL_Qn_INTERRUPT_CONTROL_STATUS register to identify the exact cause and set the corresponding bits to 1 to clear the interrupt event. The assertion of interrupt due to these events is enabled by the corresponding enable bits in MTL_Qn_INTERRUPT_CONTROL_STATUS register.

28.4.7.3 Interrupts from DMA

As shown in Figure 28.10, GMAC_SBD interrupt is a level signal and gets de-asserted only when all the enabled interrupt events are cleared in their respective status registers and correspondingly all the bits in the DMA_Interrupt_Status register are cleared.

The DMA_CHn_STATUS register (n = 0 to 7) captures all the interrupt events of that Tx DMA and Rx DMA channel pair. The DMA_CHn_INTERRUPT_ENABLE register (n = 0 to 7) contains the corresponding enable bits for each of the interrupt event. There are two groups of interrupts in the DMA channel namely Normal and Abnormal interrupts. They are indicated by bits [15:14] of DMA_CHn_STATUS register (n = 0 to 7) respectively. The normal group is for events that happen during the normal transfer of packets (TI, RI, TBU) while the abnormal interrupt events are for error events. Interrupt events are cleared by writing 1 to the corresponding bit position. When all the enabled interrupt events are cleared (including the NIS and AIS), the interrupt source from the DMA Channel is cleared and the corresponding bit in DMA_Interrupt_Status register is also cleared.

Interrupts are not queued. If the same interrupt event occurs again before the driver responds to the previous one, no additional interrupts are generated. For example, Receive Interrupt bit [6] of DMA_CHn_STATUS register (n = 0 to 7) indicates that one or more packets were transferred to the application buffer. The driver must scan all descriptors, from the last recorded position to the first one, owned by the DMA to determine how many packets are received.

An interrupt is generated only once for multiple events. The driver must scan the DMA_Interrupt_Status register for the cause of the interrupt and clear the source in the respective Status register. The GMAC_SBD signal is cleared only when all the bits of DMA_Interrupt_Status register are cleared.

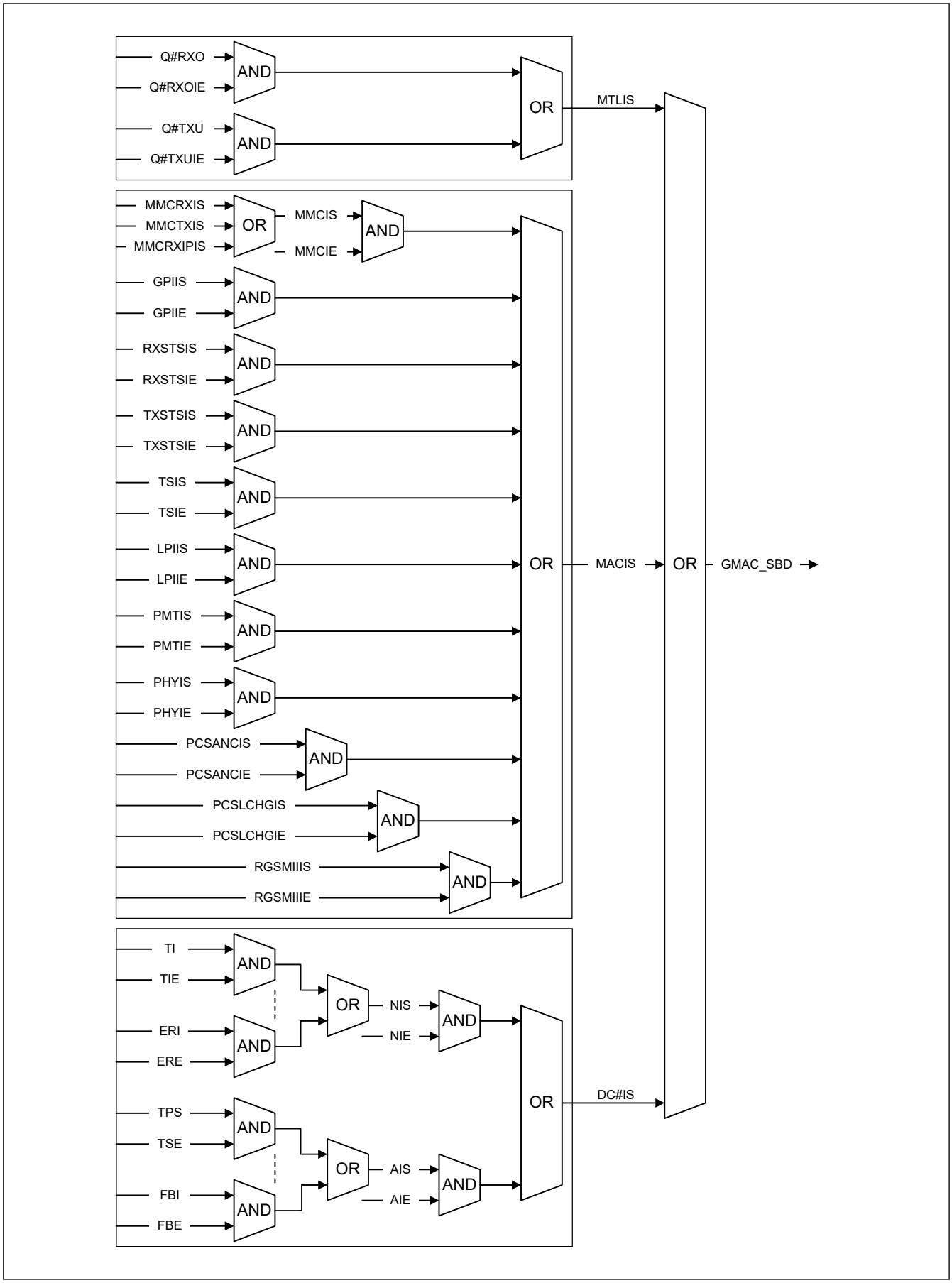


Figure 28.10 GMAC_SBD interrupt generation

28.4.7.4 Periodic Scheduling of Transmit and Receive Interrupt

To improve the throughput and performance, GMAC supports interrupt timer and transmit descriptor to generate interrupts periodically, instead of every DMA transfer.

It is not preferable to generate interrupts for every packet transferred by DMA (RI and TI) for system throughput performance reasons. The GMAC gives the flexibility to schedule the interrupt at regular intervals using two methods:

- Set Interrupt on Completion bit in Transmit descriptor (TDES2[31]) once for every "required" number of packets to be transmitted.
- Similarly, set the IOC (RDES3[30]) bit only at some specific intervals of Receive descriptors. This way, whenever a received packet transfer to system memory is complete and any of the descriptors used for that packet transfer has the IOC bit set, only then the RI event is generated.

In addition, an interrupt timer (DMA_CHn_RX_INTERRUPT_WATCHDOG_TIMER (n = 0 to 7)) is given for flexible control and periodic scheduling of Receive Interrupt. When this interrupt timer is programmed with a nonzero value, it gets activated as soon as the Rx DMA completes a transfer of a received packet to system memory without asserting the Receive Interrupt because the corresponding interrupt of completion IOC bit (RDES3[30]) is not set. When this timer runs out as per the programmed value, RI bit is set and the interrupt is asserted if the corresponding RIE is enabled in DMA_CHn_INTERRUPT_ENABLE register (n = 0 to 7). The timer is stopped and cleared before it expires, if the RI is set for a packet transfer whose descriptor's IOC was set. The timer is reactivated automatically after the next packet transfer is complete without the RI event being generated.

28.4.7.5 Per Channel Transfer Complete Interrupt

The Transmit Transfer complete interrupt (TI) and Receive Transfer complete interrupt (RI) is reflected in DMA_CHn_STATUS register (n = 0 to 7). The TI bit is set whenever the Tx DMA channel closes the transmit descriptor in which the IOC (Interrupt On Completion - TDES2[31]) bit is set. Similarly, the RI bit is set whenever the Rx DMA channel closes the receive descriptor with LD bit set and in any of the descriptors used for transferring that packet, IOC (Interrupt Enable on completion - RDES3[30]) bit is set.

The common GMAC_SBD interrupt signal is asserted for the Transfer complete interrupts only when the corresponding interrupts are enabled in DMA_CHn_INTERRUPT_ENABLE register (n = 0 to 7).

The following per Channel Transfer Complete interrupt signals are also supported.

- GMAC_TXINTn (n = 0 to 7) (Transmit Per Channel Interrupts)
- GMAC_RXINTn (n = 0 to 7) (Receive Per Channel Interrupts)

The behavior of the RI, TI, GMAC_TXINTn (n = 0 to 7) and GMAC_RXINTn (n = 0 to 7) changes depending on the settings of INTM field bits [17:16] in DMA_Mode. Table 28.6 explains the Transfer Complete Interrupt behavior.

Table 28.6 Transfer complete interrupt behavior (1 of 2)

Interrupt Mode	Behavior of GMAC_TXINTn and GMAC_RXINTn	Behavior of TI/RI and GMAC_SBD
INTM = 0	A pulse is asserted on these output signals when corresponding TX/RX transfer complete event (for which IOC descriptor bits are enabled) is detected, irrespective of the corresponding interrupt status.	The TI/RI status signals are set whenever the Transfer Complete event is detected. The bits get cleared whenever the software driver writes'1' to these bits. The GNAC_SBD signal is asserted when ever the corresponding interrupts are also enabled in the DMA_CHn_INTERRUPT_ENABLE register (n = 0 to 7). The NIS status bit is asserted for RI/TI events.
INTM = 1	These signals reflect the value of corresponding TI/RI bits in the DMA_CHn_STATUS register (n = 0 to 7) when the corresponding interrupt enable is set. Therefore, they are level signals and are cleared by the application by writing 1 to the RI/TI status bits. This signal is not asserted when the corresponding interrupt enable bit is not set.	The GMAC_SBD signal and NIS status bit are not asserted for RI/TI events.

Table 28.6 Transfer complete interrupt behavior (2 of 2)

Interrupt Mode	Behavior of GMAC_TXINTn and GMAC_RXINTn	Behavior of TI/RI and GMAC_SBD
INTM = 2	In this mode, RI/TI interrupts are queued. These signals reflect the value of corresponding TI/RI bits in the DMA_CHn_STATUS register (n = 0 to 7) when the corresponding interrupt enable is set. They are level signals and cleared by software by writing 1 to the RI/TI status bits. However, it is set again if another TI/RI event(s) is detected before the TI/RI bits are cleared for the previous event.	The RI/TI status bits are set whenever the Transfer Complete event is detected and gets reset whenever software driver clears those bits by writing 1. However, if another Transfer Complete event is detected before it is cleared (serviced) by the software, then GMAC automatically set these status bits again.However, the GMAC_SBD signal is not generated based on TI/RI. The NIS status bit is not asserted for RI/TI events.

28.5 Station Management Agent

The application can access the PHY registers through the Station Management Agent (SMA). SMA is a two-wire Station Management interface (MIM).

28.5.1 Description of Station Management Agent

For MIM accesses, the maximum operating frequency of the MDC is 2.5 MHz, as specified in the IEEE 802.3. In the GMAC, the MDC clock is derived from the application clock, using a divider-counter. The divide factor depends on the clock range setting (CR field) in the MAC_MDIO_Address register.

Select the clock divide factor as mentioned in the description of CR field of MAC_MDIO_Address register, to meet IEEE specifications. However, if your system supports higher clock frequencies on the MIM interface, there is a provision to select a different divider.

Table 28.7 lists the MDIO frame structure.

Table 28.7 MDIO Clause 45 frame structure

Field	Description
IDLE	The mdio line is in tri-state; there is no clock on MDC signal.
PREAMBLE	32 continuous bits of value 1
START	Start of packet is 00b
OPCODE	• 00b • 01b • 10b • 11b
PHY ADDR	5-bit address select for one of 32 PHYs
DEV ADDR	5-bit address select for one of 32 devices
TA	Turnaround • Z0b: Read and post-read increment address • 10b: Write and address MDIO accesses Where Z is the tri-state level
DATA/ADDRESS	16-bit value: For an address cycle (OPCODE = 00b), this frame contains the address of the register to be accessed on the next cycle. For the data cycle of a write frame, this field contains the data to be written to the register. For read or post-read increment address frames, this field contains the contents of the register read from the PHY. • In address and data write cycles, the GMAC drives the MDIO line during the transfer of these 16 bits. • In read and post-read increment address cycles, the PHY drives the MDIO line during the transfer of these 16 bits.

The frame structure for Clause 22 frames is also supported. The C45E bit in the MAC_MDIO_Address register can be programmed to enable Clause 22 or Clause 45 mode of operation. Table 28.8 lists the Clause 22 frame format.

Table 28.8 MDIO Clause 22 frame structure (1 of 2)

Field	Description
IDLE	The mdio line is in tri-state; there is no clock on MDC.
PREAMBLE	32 continuous bits of value 1

Table 28.8 MDIO Clause 22 frame structure (2 of 2)

Field	Description
START	Start of packet is 01b
OPCODE	• 01b for Write • 10b for Read
PHY ADDR	5-bit address select for one of 32 PHYs
DEV ADDR	5-bit address to select the register within each MMD
TA	Turnaround • Z0b: Read and post-read increment address • 10b: Write and address MDIO accesses Where Z is the tri-state level
DATA/ADDRESS	Any 16-bit value: • In a write operation, the GMAC drives MDIO. • In read operation, the PHY drives MDIO.

In addition to normal read and write operations, the SMA also supports post-read increment address while operating in Clause 45 mode.

28.5.2 GMII/MII Management Write Operation

After the Station Management Agent receives the PHY address and the write data from the MAC CSR module, the SMA starts a Write operation to the PHY registers.

Figure 28.11 shows the flow for a write operation from the SMA module to the PHY registers.

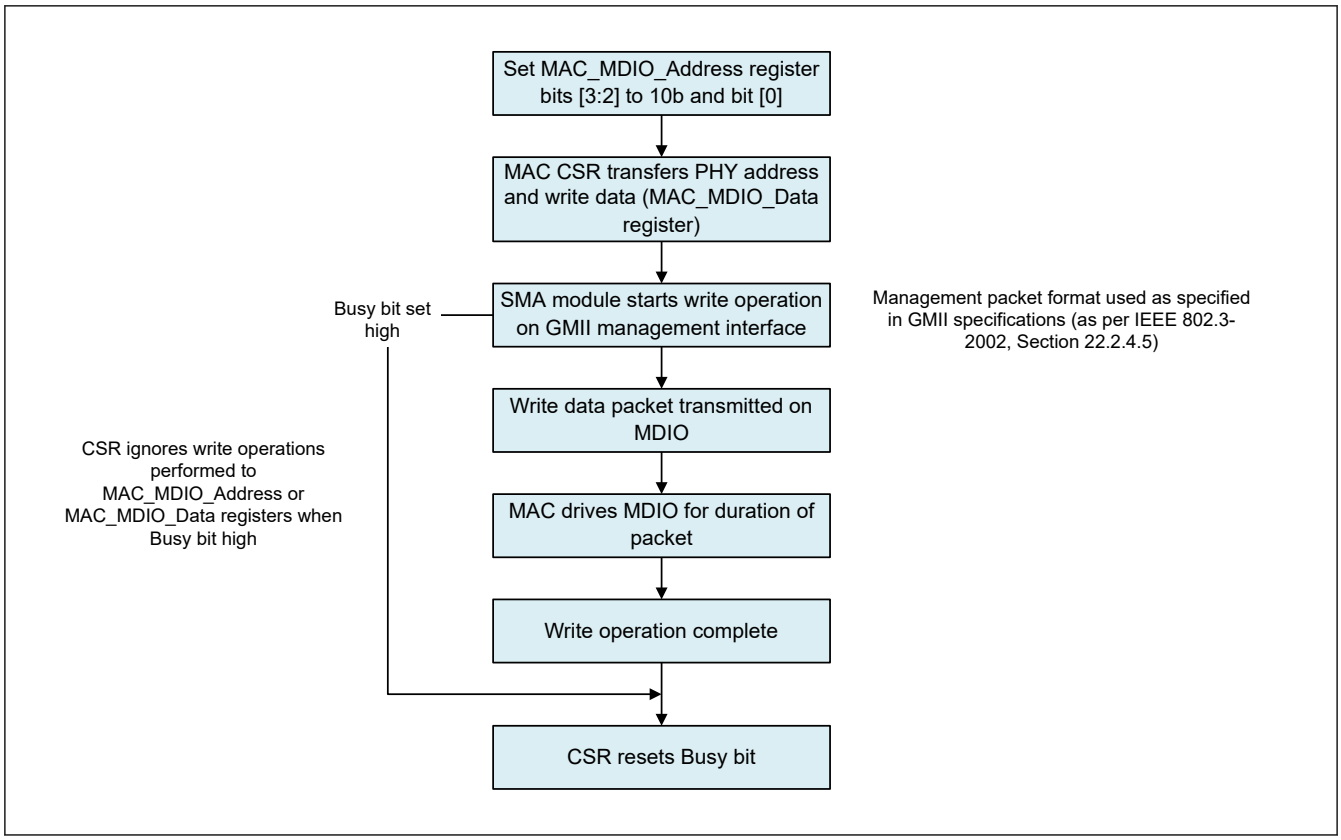


Figure 28.11 SMA write operation flow

When you set bits [3:2] to 01b and bit 0 in the MAC_MDIO_Address register, the MAC CSR module transfers the PHY address, the register address in PHY, and the write data (MAC_MDIO_Data register) to the SMA to initiate a Write operation into the PHY registers. At this point, the SMA module starts a Write operation on the GMII Management Interface using the Management Packet Format specified in the GMII specifications (as per IEEE 802.3-2002, Section 22.2.4.5). When the SMA module starts a Write operation, the write data packet is transmitted on the MDIO line. The MAC

drives the MDIO line for complete duration of the packet. The Busy bit is set high until the write operation is complete. The CSR ignores the Write operations performed to the MAC_MDIO_Address register or the MAC_MDIO_Data register during this period (the Busy bit is high). When the Write operation is complete, the SMA module indicates this to the CSR, and the CSR resets the Busy bit.

Table 28.9The packet format for the Write operation is shown in the following table:

Table 28.9 Packet format for write operation

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE
Z	1111…11	01	01	AAAAA	RRRRR	10	DDD…DDD	Z

28.5.3 GMII/MII Management Read Operation

After the Station Management Agent receives the PHY address and the register address in the PHY from the MAC CSR module, the SMA initiates a Read operation to the PHY registers.

The flow of a Management read operation is as follows:

1. Set bits [3:2] to 11b and bit 0 in the MAC_MDIO_Address register
2. The MAC CSR module transfers the PHY address and the register address in PHY to the SMA to initiate a Read operation in the PHY registers
3. The SMA module starts a Read operation on the GMII Management Interface using the Management Packet Format specified in the GMII specifications (as per IEEE 802.3-2002, Section 22.2.4.5).
4. When the SMA module starts a Read operation on the MDIO, the CSR ignores the Write operations to the MAC_MDIO_Address or MAC_MDIO_Data register during this period (the Busy bit is high) and the transaction is completed without any error on the MCI interface.
5. When the Read operation is complete, the SMA indicates this to the CSR.
6. The CSR resets the Busy bit and updates the MAC_MDIO_Data register with the data read from the PHY.

For more information about the communication from the application to the PHYs, see the Reconciliation Sublayer and Media Independent Interface Specifications sections of the IEEE 802.3z, 1000BASE Ethernet.

Table 28.10 lists the packet format for the read operation.

Table 28.10 Packet format for read operation

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE
Z	1111…11	01	10	AAAAA	RRRRR	Z0	DDD…DDD	Z

28.5.4 Preamble Suppression

The IEEE standard specifies 32-bit preamble (all-ones) for the MDIO frames. The GMAC provides controls to support preamble suppression. GMAC transmits MDIO frames with only 1 preamble bit. The preamble suppression can be enabled by setting PSE bit of MAC_MDIO_Address register.

28.5.5 Trailing Clocks and Back to Back transactions

The GMAC drives MDC clock on GMACn_MDC port for duration of the MDIO frame. There is no clock driven during the idle period. The trailing clocks feature can be used if the PHY needs the MDC clock to be active for some cycles after the MDIO frame. The NTC field in MAC_MDIO_Address register allows programming of trailing clocks from 0 to 7.

The GMAC supports Back to Back transactions which allows start of next MDIO frame even before the trailing clocks are completed for previous MDIO frame. This feature can be enabled by setting BTB bit in MAC_MDIO_Address register when trailing clocks feature is also enabled. When Back to Back transactions is enabled, the GMII Busy is cleared immediately after MDIO frame completion. This allows the software to issue the next command, which is executed by GMAC while trailing clocks are still on for the previous MDIO frame. When Back to Back transactions is not enabled, the GMII Busy is cleared after the trailing clocks are completed for MDIO frame.

28.5.6 Interrupt for MDIO Transaction Completion

GMAC can generate an interrupt on completion of MDIO read or write transactions. Therefore, the application need not poll the GMII Busy field of the MAC_MDIO_Address register to know the completion of MDIO commands.

28.6 Packet Filtering

28.6.1 Packet Filtering Sequence

Figure 28.12 shows the filtering sequence for Rx packets.

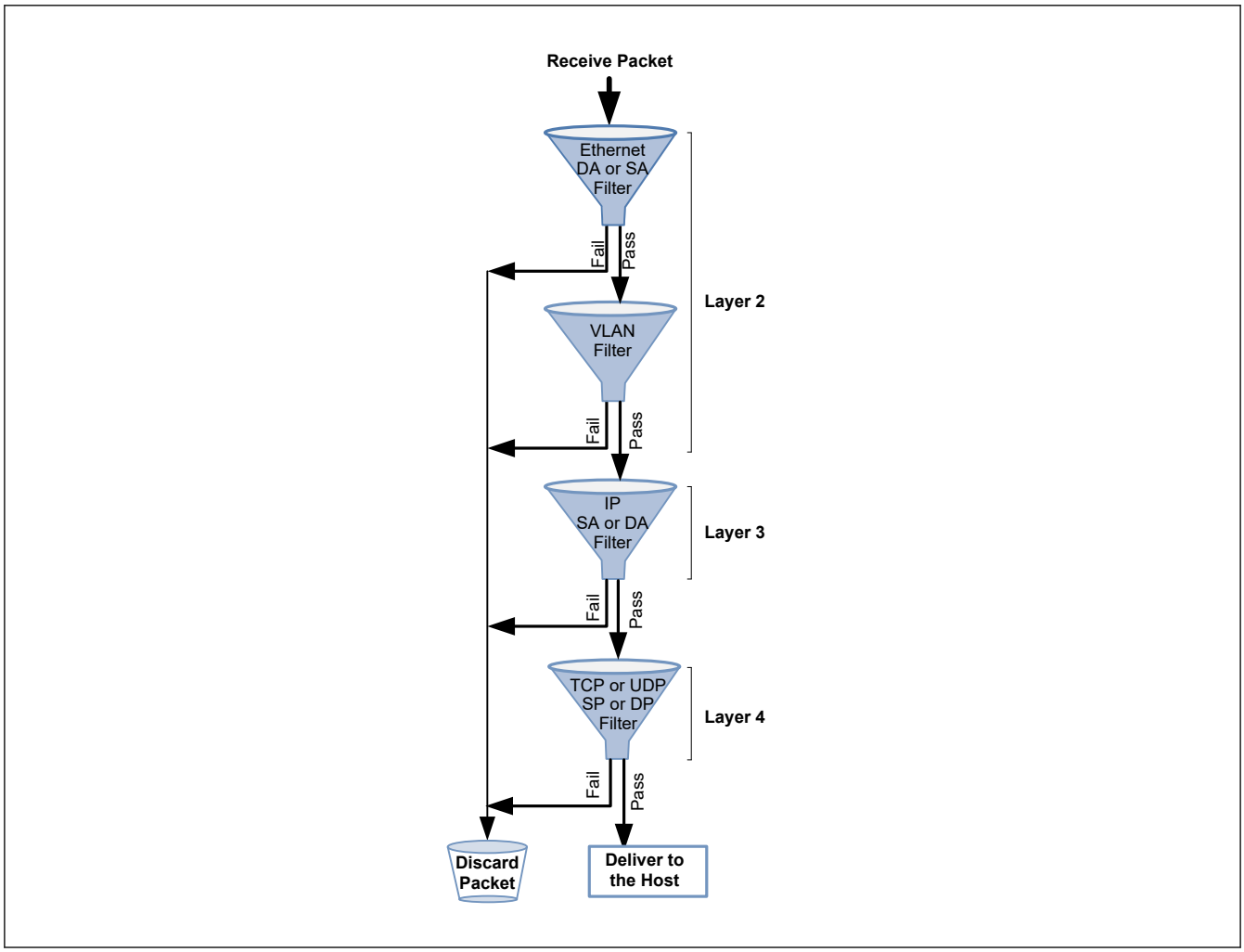


Figure 28.12 Packet filtering sequence

The sequence shown in Figure 28.12 is valid when all the filters (L2, VLAN, L3, L4) are active. If any of the Layer filters are not enabled, that filter is bypassed and the subsequent filter is applied. A packet that fails any of the filters is discarded. However, the discarded packet can be forwarded to the host based on the register control. For example, when bit RA of MAC_Packet_Filter register is set to 1, all discarded packets are forwarded to the host but with its packet status indicating the specific filter-failure. If RA = 0, bits VTFE and IPFE of MAC_Packet_Filter register controls if the packets that fail the VLAN filter and Layer 3-4 filter should be discarded or forwarded to the host.

28.6.2 Source Address or Destination Address Filtering

The Address Filtering Module of the MAC checks the source address (SA) and destination address (DA) fields of each incoming packet.

28.6.2.1 Unicast Destination Address Filtering

The MAC supports up to 32 MAC addresses for unicast perfect filtering. If perfect filtering is selected (HUC bit of MAC_Packet_Filter register is reset), the MAC compares all 48 bits of received unicast address with the programmed MAC address for any match. The default MacAddr0 is always enabled.

The MacAddr1 to MacAddr31 addresses are selected with an individual enable bit. You can mask each byte during comparison with corresponding received DA byte by setting the corresponding Mask Byte Control bit in the register. This enables group address filtering for the DA.

In hash filtering mode (when HUC bit is set), the MAC performs imperfect filtering for unicast addresses using a 256-bit Hash table. For hash filtering, the MAC uses the upper 8 bits CRC of the received destination address to index the content of the Hash table. A value of 0x00 selects bit 0 of selected register, and a value of 0xFF selects bit 255 of Hash Table register. If the corresponding bit (indicated by the 8-bit CRC) is set to 1, the unicast packet is considered to have passed the Hash filter; otherwise, the packet is considered to have failed the Hash filter.

28.6.2.2 Multicast Destination Address Filtering

To program the MAC to pass all multicast packets, set the PM bit in MAC_Packet_Filter register. If the PM bit is reset, the MAC performs the filtering for multicast addresses based on the HMC bit of the MAC_Packet_Filter register.

The multicast address is compared with the programmed MAC Destination Address registers (1–31). Group address filtering is also supported.

In Hash filtering mode, the MAC performs imperfect filtering using a 256-bit Hash table. The MAC uses the upper 6-bits CRC of received multicast address to index the content of the Hash table. A value of 0x00 selects bit 0 of selected register and a value of 0xFF selects bit 255 of the Hash Table register. If the corresponding bit is set to 1, the multicast packet is considered to have passed the Hash filter. Otherwise, the packet is considered to have failed the Hash filter.

28.6.2.3 Hash or Perfect Address Filtering

To configure the DA filter to pass a packet when its DA matches either the Hash filter or the Perfect filter, set the HPF bit and the corresponding HUC or HMC bits in MAC_Packet_Filter register. This is applicable to both unicast and multicast packets. If the HPF bit is reset, only one of the filters (Hash or Perfect) is applied to received packet.

28.6.2.4 Broadcast Address Filtering

The MAC does not filter any broadcast packets by default. To program the MAC to reject all broadcast packets, set the DBF bit in MAC_Packet_Filter register.

28.6.2.5 Unicast Source Address Filtering

The MAC can perform perfect filtering based on the source address field of received packets. By default, the MAC compares the SA field with the values programmed in the SA registers. You can configure the MAC Address registers[1–31] to use SA instead of DA for comparison by setting bit 30 of corresponding register.

The MAC also supports group filtering with SA. You can filter a group of addresses by masking one or more bytes of the address. The MAC drops the packets that fail the SA filter if the SAF bit is set in MAC_Packet_Filter register. Otherwise, the result of the SA filter is given as a status bit in the Receive Status word. When the SAF bit is set, the SA filter and DA filter result is AND'ed to decide whether the packet needs to be forwarded. This means that the packet is dropped if either filter fails. The packet is forwarded to the application only if the packet passes both filters in-order.

28.6.2.6 Inverse Filtering

For DA and SA filtering, you can invert the filter-match result at the final output by setting the DAIF and SAIF bits of MAC_Packet_Filter register. The DAIF bit is applicable for both Unicast and Multicast DA packets. The result of the unicast or multicast destination address filter is inverted in this mode. Similarly, when the SAIF bit is set, the result of unicast SA filter is reversed.

Table 28.11 and Table 28.12 summarize the DA and SA filtering based on the type of packets received.

Note: When the RA bit of MAC_Packet_Filter register is set, all packets are forwarded to the system along with the correct result of the address filtering in the Rx Status.

Table 28.11 Destination address filtering

Packet Type	PR	HPF	HUC	DAIF	HMC	PM	DBF	DA Filter Operation
Broadcast	1	x	x	x	x	x	x	Pass
	0	x	x	x	x	x	0	Pass
	0	x	x	x	x	x	1	Fail
Unicast	1	x	x	x	x	x	x	Pass all packets
	0	x	0	0	x	x	x	Pass on Perfect/Group filter match
	0	x	0	1	x	x	x	Fail on Perfect/Group filter match
	0	0	1	0	x	x	x	Pass on Hash filter match
	0	0	1	1	x	x	x	Fail on Hash filter match
	0	1	1	0	x	x	x	Pass on Hash or Perfect/Group filter match
	0	1	1	1	x	x	x	Fail on Hash or Perfect/Group filter match
Multicast	1	x	x	x	x	x	x	Pass all packets
	x	x	x	x	x	1	x	Pass all packets
	0	x	x	0	0	0	x	Pass on Perfect/Group filter match and drop Pause packets if PCF = 0x
	0	0	x	0	1	0	x	Pass on Hash filter match and drop Pause packets if PCF = 0x
	0	1	x	0	1	0	x	Pass on Hash or Perfect/Group filter match and drop Pause packets if PCF = 0x
	0	x	x	1	0	0	x	Fail on Perfect/Group filter match and drop Pause packets if PCF = 0x
	0	0	x	1	1	0	x	Fail on Hash filter match and drop Pause packets if PCF = 0x
	0	1	x	1	1	0	x	Fail on Hash or Perfect/Group filter match and drop Pause packets if PCF = 0x

Note: 'x' represents any value.

Table 28.12 Source address filtering

Packet Type	PR	SAIF	SAF	SA Filter Operation
Unicast	1	x	x	Pass all packets
	0	0	0	Pass status on Perfect or Group filter match but do not drop packets that fail
	0	1	0	Fail status on Perfect or Group filter match but do not drop packet
	0	0	1	Pass on Perfect or Group filter match and drop packets that fail
	0	1	1	Fail on Perfect or Group filter match and drop packets that fail

Note: 'x' represents any value.

28.6.3 VLAN Filtering

28.6.3.1 VLAN Tag Perfect Filtering

In VLAN tag perfect filtering, the MAC compares the VLAN tag of received packet and provides the VLAN packet status to the application. Based on the programmed mode, the MAC compares the lower 12 bits or all 16 bits of received VLAN tag to determine the perfect match.

If VLAN tag perfect filtering is enabled, the MAC forwards the VLAN-tagged packets along with VLAN tag match status and drops the VLAN packets that do not match. You can also enable the inverse matching for VLAN packets by setting the VTIM bit of MAC_VLAN_Tag register. In addition, you can enable processing of S-VLAN tagged packets along with the default C-VLAN tagged packets by setting the ESVL bit of MAC_VLAN_Tag register. For details, see section 28.12. TCP/IP Offloading Features. The VLAN packet status bit (Bit 10 of RDES0) indicates the VLAN tag match status for the matched packets.

Note: The source or destination address has precedence over the VLAN tag filters. This means that a packet that fails the source or destination address filter is dropped irrespective of the VLAN tag filter results. By default, the VLAN tag-based perfect filter is available.

28.6.3.2 VLAN Tag Hash Filtering

The 16-bit VLAN Hash Table is used for group address filtering based on the VLAN tag. The VLAN Tag Hash Filtering feature can be enabled using the VTHM (VLAN Tag Hash Table Match Enable) bit of the MAC_VLAN_Tag register.

The MAC provides VLAN tag hash filtering with a 16-bit Hash table.

The MAC performs the VLAN hash matching based on the VTHM of the MAC_VLAN_Tag register. If the VTHM bit is set, the most significant four bits of CRC-32 of VLAN tag are used to index the content of the MAC_VLAN_Hash_Table register. A value of 1 in the MAC_VLAN_Hash_Table register, corresponding to the index, indicates that the VLAN tag of the packet matched and the packet should be forwarded. A value of 0 indicates that VLAN-tagged packet should be dropped.

Note:
- The 16 or 12 bits of VLAN Tag are considered for CRC-32 computation based on ETV bit in MAC_VLAN_TAG register.
- When ETV bit is reset, most significant four bits of CRC-32 of VLAN Tag are inverted and used to index the content of MAC_VLAN_Hash_Table register.
- When ETV bit is set, most significant four bits of CRC-32 of VLAN Tag are directly used to index the content of MAC_VLAN_Hash_Table register.

The MAC also supports the inverse matching for VLAN packets. In the inverse matching mode, when the VLAN tag of a packet matches the perfect or hash filter, the packet should be dropped. If the VLAN perfect and VLAN hash match are enabled, a packet is considered as matched if either the VLAN hash or the VLAN perfect filter matches. When inverse match is set, a packet is forwarded only when both perfect and hash filters indicate mismatch.

Table 28.13 shows the different possibilities for VLAN matching and the final VLAN match status. When the RA bit of MAC_Packet_Filter register is set, all packets are received and the VLAN match status is indicated in the VF bit of RDES2 Normal Descriptor (Write-Back Format). When the RA bit is not set and the VTFE bit is set in MAC_Packet_Filter register, the packet is dropped if the final VLAN match status is Fail.

When VLAN VID is programmed to 0 in the VL field of MAC_VLAN_Tag register, all VLAN-tagged packets are considered as perfect matched but the status of the VLAN hash match depends on the VTHM and VTIM bits in MAC_VLAN_Tag register.

Table 28.13 VLAN match status

VID	VLAN Perfect Filter Match Result	VTHM bit	VLAN Hash Filter Match Result	VTIM bit	Final VLAN Match Status
VID = 0	Pass	0	x	x	Pass
	Pass	1	x	0	Pass
	Pass	1	Fail	1	Pass
	Pass	1	Pass	1	Fail
VID != 0	Pass	x	x	0	Pass
	Fail	0	x	0	Fail
	Fail	1	Fail	0	Fail
	Fail	1	Pass	0	Pass
	Fail	0	x	1	Pass
	Pass	x	x	1	Fail
	Fail	1	Pass	1	Fail
	Fail	1	Fail	1	Pass

Note: 'x' represents any value.

28.6.3.3 Double VLAN Processing

This feature is referred as the double VLAN tagging feature in which the MAC can process two VLAN tags, namely inner and outer. With this feature, the GMAC supports the following:

- Packet filtering and stripping based on any one of the two VLAN Tags in the Receive path. Stripping and providing up to two VLAN Tags in the Receive path as a part of the Receive status.

Table 28.14 lists the features supported by the MAC on the Receive side and the corresponding bits in the MAC_VLAN_Tag register.

Table 28.14 Double VLAN processing features in receive path

Feature	Description
Outer or inner VLAN tag-based filtering	The MAC can filter packets based on the outer or inner VLAN tag through the ERIVLT bit.
C-VLAN or S-VLAN tag-based filtering	The MAC can filter packets based on the C-VLAN or S-VLAN type based on the ERSVLM bit.
Outer and Inner VLAN Tag stripping	The MAC can strip the outer and inner VLAN Tags from received frame based on the EVLS and EIVLS bits.
16-bit outer and inner VLAN Tag and Type in Rx status	The MAC can provide the 16-bit outer and inner VLAN Tag and Type in the Rx status based on the EVLRXS and EIVLRXS bits, respectively.
Disabling or skipping checking of outer VLAN Tag type	The MAC can disable or skip checking of outer VLAN Tag type to match C-VLAN or S-VLAN based on the DOVLTC bit.

28.6.3.4 Extended Receive VLAN Filtering and Routing

The MAC Receiver can classify the received packets based on VLAN Tag and steer them to a specific Rx DMA channel. The MAC compares the received frame's VLAN Tag with all the enabled and relevant filters and provides a filtering result. If any of the perfect filters give a pass result and if the respective filter's DMA channel Number is enabled, the frame is routed to that DMA Channel.

In addition to filtering, routing can also be done. For details about routing, see Extended VLAN Based DMA Selection in (2) Dynamic (Per Packet) Mapping.

(1) Comparison Modes

For each VLAN Tag Filter, the application has the following comparison options:

- It can program the MAC to compare an outer VLAN Tag or an inner VLAN Tag with the programmed VID.
- It can choose if 12 or 16 bits of the VID field need to be compared.
- Type check can be disabled or enabled for each filter; if enabled, the application can choose if the VID comparison is for SVLAN or CVLAN type frames only.

For example, if a filter is enabled for 16 bits comparison, SVLAN Type, and Outer VLAN Tag, any single or double VLAN Tagged frames with Outer SVLAN Tags are compared with this filter, and a pass or fail result is obtained.

Note: The inner VLAN Tag comparison is applicable only if Double VLAN Tag processing is enabled through the parameter and the MAC_VLAN_Tag_Ctrl register bit.

(2) Filtering

When Extended RX VLAN Filtering & Routing feature is enabled, the application can enable both Perfect and Hash Filtering. The overall VLAN Filter Result is based on the perfect filter result and the Hash Filter result (if enabled). The filter result is passed to the application as part of the status bits.

Perfect Filtering is done based on the MAC_VLAN_Tag_Filter i registers. For each VLAN Tag Filter, the MAC compares the relevant VLAN Tag ID and gives a result. If any one of the VLAN Tag Filters gives a match, the frame is considered to have passed the VLAN Tag Filters. If the frame mismatches all the filters, the frame is considered to have failed the VLAN filter. This behavior is applicable only when the Inverse filtering is not enabled in the MAC_VLAN_Tag_Ctrl register.

If inverse filtering is enabled and the frame has mis-matched all the relevant filters then it is considered to have passed the VLAN filter. If the frame matches any one of the relevant filters then it is considered as a fail. If none of the enabled filters can perform a comparison or if none of the filters are enabled, then the frame is bypassed to the application.

The overall filter result and the values programmed in the VTFE and RA fields of the MAC Filter register determine if the frame must be dropped or forwarded to the application.

- When RA = 1 or VTFE = 0
  The MAC forwards the frame, irrespective of the filter result.
- When RA = 0 and VTFE = 1
  The MAC forwards the frame only if the VLAN Tag filter status is pass. After forwarding a frame to the application, the relevant filter result is indicated through the status bits.

28.6.3.5 VLAN Filter Status

The Extended Receive VLAN Filtering & Routing feature provides two status bits to indicate the comparison result of the VLAN tags.

By default, the MAC indicates the VLAN Filter Status through one bit in the status – VF in RDES2. When Extended RX VLAN Filtering & Routing is enabled, two status bits are used to indicate the comparison result of VLAN tags. The Outer VLAN Tag Filter Pass and Inner VLAN Tag Filter Pass bits are defined in the following positions. The status indicated through these bits is highly dependent on the how the fields are programmed:

In RDES2:

- Bit 15: Outer VLAN Tag Filter Status
- Bit 14: Inner VLAN Tag Filter Status

Outer VLAN Tag Filter Status (OTS)

- In perfect Filtering, without inverse filtering enabled, if this bit is set, it indicates that the frame's Outer VLAN Tag has matched one of the VLAN Tag Filters.
- If this bit is reset, it indicates that the frame's Outer VLAN Tag has either failed the relevant Outer VLAN Tag Filters or bypassed them.
- If none of the filters are enabled for Outer VLAN Tag Comparison, then this bit is reset.
- If Inverse Filtering is enabled and this bit is set, then the frame's VLAN Tag has passed all the relevant VLAN Tag Filters. If it is reset, then it has failed at least one of or bypassed all the Filters programmed for Outer VLAN Tag Comparison.
- This bit is valid for both Single and Double VLAN Tagged frames.

Inner VLAN Tag Filter Status (ITS)

- In perfect Filtering, without inverse filtering enabled, if this bit is set, it indicates that the frame's Inner VLAN Tag has matched one of the VLAN Tag Filters.
- If this bit is reset, it indicates that the frame's Inner VLAN Tag has either failed the relevant Inner VLAN Tag Filters or bypassed them.
- If none of the filters are enabled for Inner VLAN Tag Comparison, then this bit is reset.
- If Inverse Filtering is enabled and this bit is set, then the frame's VLAN Tag has passed all the relevant VLAN Tag Filters. If it is reset, then it has failed at least one of or bypassed all the Filters programmed for Inner VLAN Tag Comparison.
- This bit is valid for only Double VLAN Tagged frames, when Double VLAN Processing is enabled.

The application must look at the status bits and the programming to determine if the Frame has passed or failed the VLAN Filter.

Table 28.15 and Table 28.16 list the possible Filter combinations and the corresponding filter results. These tables explain the scenarios when Double VLAN Processing and Hash VLAN filter are enabled in the design.

Legend for the Table 28.15

- VTIM: VLAN Tag Inverse Match Enable - bit 17 in the VLAN_Tag_Ctrl register.
- HFO: Hash Filter enabled for Outer VLAN Tag Comparison - bit 25 and bit 27 in the MAC_VLAN_Tag_Ctrl register.
- HFI5: Hash Filter enabled for Inner VLAN Tag Comparison - bit 25 and bit 27 in the MAC_VLAN_Tag_Ctrl register.

- PFO: Perfect Filter comparison enabled for Outer VLAN Tag - Any of the MAC_VLAN_Tag_Filter i registers is enabled (bit 16 is set) and programmed for Outer VLAN Tag comparison (bit 20 is set to 0).
- PFI: Perfect Filter comparison enabled for Inner VLAN Tag - Any of the MAC_VLAN_Tag_Filter i Registers is enabled (bit 16 is set) and programmed for Inner VLAN Tag comparison (bit 20 is set to 1).
- OTS: Outer VLAN Tag Filter Status
- ITS: Inner VLAN Tag Filter Status

Table 28.15 lists the possible values of status bits (OTS and ITS) when at least one Perfect filter is enabled. Table 28.16 lists the possible values of status bits (OTS and ITS) when none of the perfect filters are enabled and only the VLAN Hash Filter is enabled.

Table 28.15 OTS and ITS bit values with at Least 1 perfect filter enabled

VTIM	HFO	HFI	PFO	PFI	OTS	ITS
0	0	0	0	1	0	1/0
0	0	0	1	0	1/0	0
0	0	0	1	1	1/0	1/0
0	1	0	1	1	1/0	1/0
0	1	0	1	0	1/0	0
0	1	0	0	1	1/0	1/0
0	0	1	1	1	1/0	1/0
0	0	1	1	0	1/0	1/0
0	0	1	0	1	0	1/0
1	0	0	0	1	0	1/0
1	0	0	1	0	1/0	0
1	0	0	1	1	1/0	1/0
1	1	0	1	1	1/0	1/0
1	1	0	1	0	1/0	0
1	1	0	0	1	1/0	1/0
1	0	1	1	1	1/0	1/0
1	0	1	1	0	1/0	1/0
1	0	1	0	1	0	1/0

Table 28.16 OTS and ITS bit Values with only VLAN Hash Filter enabled

VTIM	HFO	HFI	OTS	ITS
0	0	0	0	0
0	1	0	1/0	0
0	0	1	1/0	1/0
1	0	0	1/0	0
1	1	0	1/0	0
1	0	1	1/0	1/0

- When no perfect filters are enabled, any VLAN packet is considered to have bypassed the perfect filter.
- When VLAN Hash Filter is enabled for one of the Tags, the respective status bit depends on the result of the filter. Status bits are set to 0 when VLAN Hash Filter is not enabled.
- The value 1/0 for the ITS/OTS field indicate that the final result is dependent on the result of the enabled relevant filter.

Example 1:

The second row of Table 28.16 indicates that at least one Perfect Filter is enabled for Outer VLAN tag comparison and none of the filters are enabled for Inner VLAN tag comparison. Inverse VLAN Filtering is not enabled. The bit OTS is given as 1/0. If the received frame passes at the least one of the enabled Outer VLAN Tag filters then the bit is set to 1. If the frame does not pass any of the enabled Outer VLAN Tag filters, then the bit is set to 0.

Example 2:

Last Row of Table 28.16 indicates that Inverse Filtering is enabled, Hash Filter and at least one perfect filter is enabled for Inner VLAN Tag comparison, then if the received frame's Inner VLAN tag mismatches with both the Hash Filter and all the enabled Perfect filters, then the frame has the ITS bit set to 1. Otherwise, the bit is set to 0. OTS bit is set to 0 as comparison is not performed.

28.6.3.6 VLAN Stripping

Each of the VLAN Tags has individual control over stripping. The programming options of Always strip, never strip, strip on pass and strip on fail are available. Inner or Outer VLAN Tag Stripping is based on the pass or fail results of the individual tag. If a tag is bypassed by all the relevant filters, stripping is not applicable for the tag.

- If strip on Pass is enabled for the outer VLAN Tag, stripping is done only if the Outer VLAN tag has passed the relevant Filters. The Outer VLAN Tag Filter Result bit is set.
- If strip on Fail is enabled for the outer VLAN Tag, stripping is done only if the Outer VLAN Tag has failed relevant filters. The Outer VLAN Tag Filter Result Bit is reset.
- If the Outer VLAN tag of the received frame is bypassed by the entire filter (no comparison has been made), the tag is not stripped, though the Status Bit is still 0.
- As multiple filters are enabled, it is possible that the received VLAN frame matches more than one filters. The VLAN Tag's value is not always deterministic from the filter status bits.
- If the application strips the VLAN Tag based on the filter result, it might lose the VID. So, if stripping is enabled for any of the tags, the tag can be put in the status. The application must enable the respective "VLAN Tag in Status" bit [24] or bit [31] in the MAC_VLAN_Tag_Ctrl register.

28.6.3.7 VLAN Filter Fail Packets Queue

When VLAN filtering is enabled, the VLAN Filter Fail Packets can be routed to a programmable Queue (VFFQ) when RA = 1 or VTFE = 0 and the enable bit (VFFQE) for the queue is set.

1. RA and VTFE bits are implemented in the MAC_Packet_Filter register

2. VFFQ and VFFQE fields are implemented in the MAC_RxQ_Ctrl4 register

The packets that pass the VLAN filtering are routed based on the VLAN TAG priority field. The VLAN Tag priorities can be assigned to Rx Queues by programming the PSRQ field in the corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers. The packets that fail the VLAN filter are discarded if RA = 0 or VTFE = 1. However, when RA = 1 or VTFE = 0, the VLAN filter fail packets are still forwarded to the application. In such a scenario, when the VLAN Filter Fail Queue Enable (VFFQE) bit is set, the VLAN filter fail packets are forwarded to the Rx queue number programmed in the VFFQ bit. If VFFQE = 0, the Rx queue number is determined by the VLAN priority mapping as per the PSRQ fields.

Table 28.17 lists the Rx queue routing table for unicast tagged packets, with DA/SA Filter enabled.

Table 28.17 Rx queue routing table for unicast tagged packets (1 of 2)

RA	VFTE	SA/DA Filter Result	VLAN Filter Result	VFFQE	Queue Routing
x	x	Pass	Pass	x	PSRQ
0	0	Pass	Fail	0	PSRQ
0	0	Pass	Fail	1	VFFQ
0	x	Fail	x	x	DROPPED
0	1	Pass	Fail	x	DROPPED
1	x	Fail	x	0	UFFQ*1/PSRQ
1	x	Fail	x	1	UFFQ*1/PSRQ
1	x	Pass	Fail	0	PSRQ

Table 28.17 Rx queue routing table for unicast tagged packets (2 of 2)

RA	VFTE	SA/DA Filter Result	VLAN Filter Result	VFFQE	Queue Routing
1	x	Pass	Fail	1	VFFQ

Note: 'x' represents any value.
Note 1. When UFFQE is enabled else PSRQ.

28.6.4 Layer 3 and Layer 4 Filtering

The GMAC supports Layer 3 and Layer 4 based packet filtering. The Layer 3 filtering refers to the IP Source or Destination Address filtering in the IPv4 or IPv6 packets whereas Layer 4 filtering refers to the Source or Destination Port number filtering in TCP or UDP.

28.6.4.1 Description of Layer 3 and Layer 4 Filtering

When Layer 3 and Layer 4 filtering is enabled, the packets are filtered in the following way:

- Matched Packets: The MAC forwards the packets that match all enabled fields to the application along with the status. The MAC gives the matched field status only if the IPC bit of MAC_Configuration register is set and one of the following conditions is true:
  - All enabled Layer 3 and Layer 4 fields match
  - At least one of the enabled field matches and other fields are bypassed or disabled

  When multiple Layer 3 and Layer 4 filters are enabled, any filter match is considered as a match. If more than one filter matches, the MAC provides the status of the lowest filter where Filter 0 is the lowest filter and Filter 3 is the highest filter. For example, if Filter 0 and Filter 1 match, the MAC gives the status corresponding to filter 0.

Note: The source or destination address and VLAN tag filters (if enabled) have precedence over Layer 3 and Layer 4 filter. This means that a packet which fails the source or destination address or VLAN tag filter is dropped irrespective of the Layer 3 and Layer 4 filter results.

- Unmatched Packets: The MAC drops the packets that do not match any of the enabled fields. You can use the inverse match feature to block or drop a packet with specific TCP or UDP over IP fields and forward all other packets.
- When a packet is dropped, a partial packet with appropriate abort status may be received on MRI. This is because there is no buffer in the MAC receiver for storing the data until the TCP/IP fields are received and matched. However, the aborted or partial packets can be dropped in the MTL Rx FIFO. If the Rx FIFO operates in the Threshold (cut-through) mode and the threshold is programmed to a small value, such that packet transfer to application starts before the failed Layer 3 and Layer 4 filter results are available, the application may receive a partial packet with appropriate abort status.
- Non-TCP or UDP IP Packets: By default, all non-TCP or UDP IP packets are bypassed from the Layer 3 and Layer 4 filters. You can optionally program the MAC to drop all non-TCP or UDP over IP packets.

Layer 3 Filtering

The GMAC supports perfect matching or inverse matching for IP Source Address and Destination Address. In addition, you can match the complete IP address or mask the lower bits matching, that is, compare all bits of the address except the specified lower mask bits.

For IPv6 packets filtering, you can enable the last four data registers of a register set to contain the 128-bit IP Source Address or IP Destination Address. The IP Source Address or Destination Address should be programmed in the order defined in the IPv6 specification, that is, the first byte of the IP Source Address or Destination Address in the received packet is in the higher byte of the register and the subsequent registers follow the same order.

For IPv4 packet filtering, you can enable the second and third data registers of a register set to contain the 32-bit IP Source Address and IP Destination Address. The remaining two data registers are reserved. The IP Source Address or Destination Address should be programmed in the order defined in the IPv4 specification, that is, the first byte of IP Source Address and Destination Address in the received packet in the higher byte of the respective register.

Layer 4 Filtering

The GMAC supports perfect matching or inverse matching for TCP or UDP Source and Destination Port numbers. However, you can program only one type (TCP or UDP) at a time. The first data register contains the 16-bit Source

and Destination Port numbers of TCP or UDP, that is, the lower 16 bits for Source Port number and higher 16 bits for Destination Port number.

The TCP or UDP Source and Destination Port numbers should be programmed in the order defined in the TCP or UDP specification, that is, the first byte of TCP or UDP Source and Destination Port number in the received packet is in the higher byte of the register.

28.6.4.2 Layer 3 and Layer 4 Filters Registers

The MAC implements a set of registers for Layer 3 and Layer 4 based packet filtering. In a register set, there is a control register, such as MAC_L3_L4_CONTROLn (n = 0 to 7), to control the packet filtering. In addition, there are five address registers to program the Layer 3 and Layer 4 fields to be matched, such as:

- MAC_LAYER4_ADDRESSn (n = 0 to 7)
- MAC_LAYER3_ADDR0_REGn (n = 0 to 7)
- MAC_LAYER3_ADDR1_REGn (n = 0 to 7)
- MAC_LAYER3_ADDR2_REGn (n = 0 to 7)
- MAC_LAYER3_ADDR_REGn (n = 0 to 7)

You can configure the MAC to have up to four such independent set of registers.

28.7 IEEE 1588 Timestamp Support

The IEEE 1588 defines a Precision Time Protocol (PTP) which enables precise synchronization of time in measurement and control systems. This protocol enables heterogeneous systems that include clocks of varying inherent precision, resolution, and stability to synchronize. The protocol supports system-wide synchronization accuracy in the sub-microsecond range with minimal network and local clock computing resources.

The GMAC supports the IEEE 1588-2002 (version 1) and IEEE 1588-2008 (version 2). The IEEE 1588-2002 supports PTP transported over UDP/IP and IEEE 1588-2008 supports PTP transported over Ethernet. The GMAC provides programmable support for both standards.

The controller supports the following features:

- Provides an option to take snapshot of all packets or only PTP type packets
- Provides an option to take snapshot of only event messages
- Provides an option to take the snapshot based on the clock type: ordinary, boundary, end-to-end transparent, and peer-to-peer transparent
- Provides an option to select the node to be a master or slave for ordinary and boundary clock
- Identifies the PTP message type, version, and PTP payload in packets sent directly over Ethernet and sends the status

28.7.1 IEEE 1588 Timestamp Support

28.7.1.1 Clock Types

The GMAC supports the following clock types:

- Ordinary Clock
- Boundary Clock
- End-to-End Transparent Clock
- Peer-to-Peer Transparent Clock

(1) Ordinary Clock

The ordinary clock has a single PTP state and a single physical port. In a domain, an ordinary clock supports a single copy of the protocol.

The ordinary clock in a domain supports a single copy of the protocol. The ordinary clock has a single PTP state and a single physical port. In typical industrial automation applications, an ordinary clock is associated with an application device

such as a sensor or an actuator. In telecom applications, the ordinary clock can be associated with a timing demarcation device.

The ordinary clock can be a grandmaster or a slave clock. It supports the following features:

- Sends and receives PTP messages. The timestamp snapshot can be controlled as described in MAC_Timestamp_Control.
- Maintains the data sets such as timestamp values

Table 28.18 shows the messages for which you can take the timestamp snapshot on the receive side for master and slave nodes.

Table 28.18 Ordinary Clock: PTP Messages for Snapshot

Master	Slave
Delay_Req	SYNC

For an ordinary clock, you can take the snapshot of either of the following PTP message types: version 1 or version 2. You cannot take the snapshots for both PTP message types. You can take the snapshot by setting the TSVER2ENA bit and selecting the snapshot mode in MAC_Timestamp_Control.

(2) Boundary Clock

The boundary clock typically has several physical ports communicating with the network. The messages related to synchronization, master-slave hierarchy, and signaling terminate in the protocol engine of the boundary clock and such messages are not forwarded. The PTP message type status given by the MAC helps you to identify the type of message and take appropriate action.

The boundary clock is similar to the ordinary clock except for the following features:

- The clock data sets are common to all ports of the boundary clock.
- The local clock is common to all ports of the boundary clock.

(3) End-to-End Transparent Clock

The end-to-end transparent clock supports the end-to-end delay measurement mechanism between the slave clocks and the master clock. The end-to-end transparent clock forwards all messages like normal bridge, router, or repeater. The residence time of a PTP packet is the time taken by the PTP packet from the Ingress port to the Egress port.

The residence time of a SYNC packet inside the end-to-end transparent clock is updated in the correction field of the associated Follow_Up PTP packet before it is transmitted. Similarly, the residence time of a Delay_Req packet, inside the end-to-end transparent clock, is updated in the correction field of the associated Delay_Resp PTP packet before it is transmitted. Therefore, the snapshot must be taken at both Ingress and Egress ports only for the messages mentioned in Table 28.19. You can take the snapshot by setting the SNAPTYPSEL bits to 10 in the wMAC_Timestamp_Control register.

Table 28.19 End to End transparent clock: PTP messages for snapshot

PTP Messages
SYNC
Delay_Req

(4) Peer-to-Peer Transparent Clock

In the peer-to-peer transparent clock, the computation of the link delay is based on an exchange of Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages with the link peer.

The peer-to-peer transparent clock differs from the end-to-end transparent clock in the way it corrects and handles the PTP timing messages. In all other aspects, it is identical to the end-to-end transparent clock.

In the peer-to-peer transparent clock, the computation of the link delay is based on an exchange of Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages with the link peer. The residence time of the Pdelay_Req and the associated Pdelay_Resp packets is added and inserted into the correction field of the associated Pdely_Resp_Followup packet. Therefore, support for taking snapshot for the event messages related to Pdelay is added as shown in Table 28.20.

Table 28.20 Peer-to-Peer transparent clock: PTP messages for snapshot

PTP Messages
SYNC
Pdelay_Req
Pdelay_Resp

28.7.1.2 Delay Request-Response Mechanism

The system or network is classified into the master and slave nodes for distributing the timing and clock information.

The system or network is classified into the master and slave nodes for distributing the timing and clock information. Figure 28.13 shows the process that PTP uses for synchronizing a slave node to a master node by exchanging PTP messages.

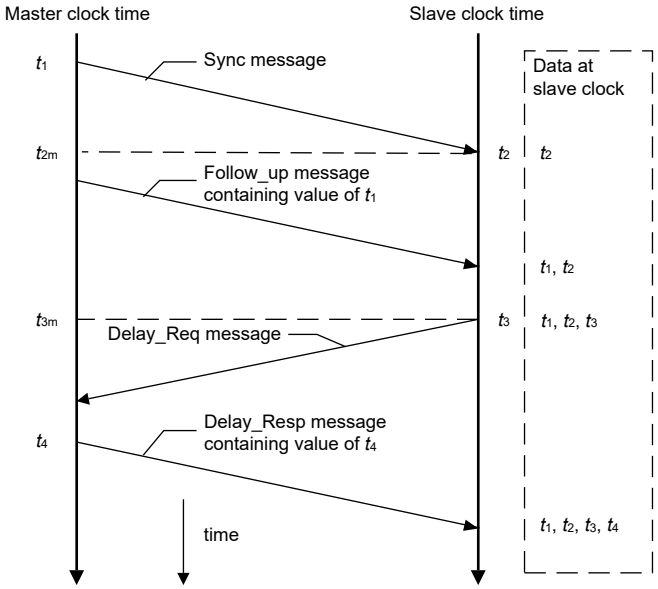


Figure 28.13 Networked time synchronization

As shown in Figure 28.13, PTP uses the following process:

1. The master broadcasts the PTP Sync messages to all its nodes. The Sync message contains the reference time information of the master. This message leaves the system of the master at t1. This time must be captured for Ethernet ports at GMII or MII.
2. The slave receives the Sync message and also captures the exact time, t2, using its timing reference.
3. The master sends a Follow_up message to the slave, which contains t1 information for later use.
4. The slave sends a Delay_Req message to the master and notes the exact time, t3, at which this packet leaves the GMII or MII interface.
5. The master receives the message, capturing the exact time t4, at which the message enters its system.
6. The master sends the t4 information to the slave in the Delay_Resp message.
7. The slave uses the four values of t1, t2, t3, and t4 to synchronize its local timing reference to the timing reference of the master.

Most of the PTP implementation is done in the software above the Ethernet/UDP layer. However, the hardware support is required to capture the exact time when specific PTP packets enter or leave the Ethernet port at the GMII or MII interface. This timing information must be captured and returned to the software for proper implementation of PTP with high accuracy.

28.7.1.3 Peer-to-Peer PTP Transparent Clock (P2P TC) Message Support

The IEEE 1588-2008 supports peer-to-peer PTP (Pdelay) message in addition to the SYNC, Delay Request, Follow-up, and Delay Response messages. Figure 28.14 shows the method to calculate the propagation delay in clocks supporting peer-to-peer path correction.

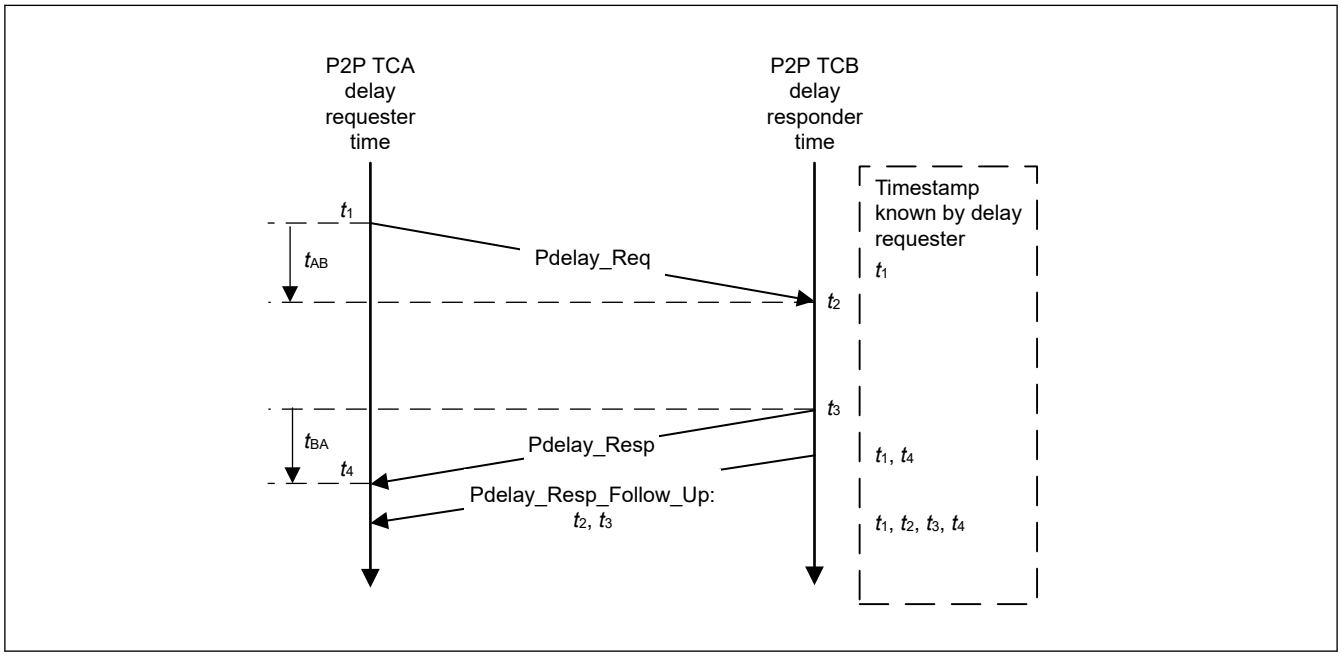


Figure 28.14 Propagation delay calculation in clocks supporting peer-to-peer path correction

As shown in Figure 28.14, the propagation delay is calculated in the following way:

1. Port 1 issues a Pdelay_Req message and generates a timestamp (t1) for the Pdelay_Req message.
2. Port 2 receives the Pdelay_Req message and generates a timestamp (t2) for this message.
3. Port 2 returns a Pdelay_Resp message and generates a timestamp (t3) for this message.
   To minimize errors because of any frequency offset between the two ports, Port 2 returns the Pdelay_Resp message as quickly as possible after the receipt of the Pdelay_Req message. Port 2 returns any one of the following:
   - Difference between the timestamps t2 and t3 in the Pdelay_Resp message
   - Difference between the timestamps t2 and t3 in the Pdelay_Resp_Follow_Up message
   - Timestamps t2 and t3 in the Pdelay_Resp and Pdelay_Resp_Follow_Up messages, respectively
4. Port 1 generates a timestamp (t4) on receiving the Pdelay_Resp message.
5. Port 1 uses all four timestamps to compute the mean link delay.

28.7.1.4 Timestamp Correction

According to the IEEE 1588 specification, a timestamp must be captured when the PTP message timestamp point (leading edge of the first bit of the octet immediately following the Start Frame Delimiter octet) crosses the boundary between the node and the network. As the MAC takes the timestamp at an internal point far from the actual boundary of the node and network, this captured timestamp is corrected/updated for the ingress/egress path latency (including the delay in the PHY layers). Further correction is done for the inaccuracies/errors introduced due to the clock (GMII Tx, Rx clock) being different at the capture point as compared to the PTP clock that is used to generate the time. The resultant CDC (Clock Domain Crossing) circuits add error depending on the clock period of the GMII and PTP clocks.

28.7.1.5 Ingress Correction

In the Receive side the timestamp captured at the internal snapshot point is delayed (later in time) as compared to the time at which that packet's SFD bit is received at the port's boundary. Therefore, the captured timestamp must be reduced by the ingress latency and the errors in CDC sampling. This correction value must be determined/calculated by the software and written into the MAC_Timestamp_Ingress_Corr_* registers.

The correction value consists of the following 3 components:

1. External latency in the PHY layer between boundary point and the input of the GMAC.
   If the PHY is compliant to the IEEE 802.3 Clause 45 MMD registers, it has a register indicating the maximum and minimum ingress latency. The software can read these registers and determine the average ingress latency in the PHY. Alternatively (if the PHY does not support these registers), the ingress latency must be determined from its datasheet or timing characteristics.
2. Internal latency from the input of the GMAC to the internal capture point
   The internal ingress latency can be read from the MAC_Ingress_Timestamp_Latency register. This is a read-only register and gives the latency in scaledNanoseconds format defined in IEEE 1588 Clause 5.3.2. The latency differs based on the active PHY interface (RGMII, RMII, so on) and the operating speed. Therefore, the software must read this register after any speed change in the MAC to determine the current internal latency.
3. CDC Synchronization

The CDC synchronization error is almost equal to 2 times the clock-period of the PTP clock (8 ns).

The values determined from these 3 components should be added by the software and must be written into the TSIC & TSICSNS fields of the MAC_Timestamp_Ingress_Corr_* registers.

Note: The value written to the register must be negative (two's complement), because it has to be subtracted from the captured timestamp. The MAC Receiver adds the value in this register to the captured timestamp and then gives the resultant value as the timestamp of the received packet.

When TSCTRLSSR bit in MAC_Timestamp_Control register is set, the nanoseconds field of the captured timestamp is in decimal format with a granularity of 1 ns. So the bit31 of TSIC must be set to 1 (for negative value) and bits [30:0] must be written with "10^9 - total ingress_correction_value[nanosecond part]" represented in binary. For example, if the required correction value is -5 ns, then the value is 0xBB9A_C9FB.

When TSCTRLSSR bit in MAC_Timestamp_Control register is reset, the nanoseconds field of the captured timestamp is in binary format with a granularity of ~0.466 ns. Therefore, bits [30:0] must be written with "2^{31} - total ingress_correction_value" represented in binary with bit [31] = 1.

28.7.1.6 Egress Correction

In the Transmit side the timestamp captured at the internal snapshot point is earlier (advanced in time) as compared to the time at which that packet's SFD bit is output at the port's boundary. Therefore, the captured timestamp must be compensated by the egress latency and the errors in CDC sampling. This correction value must be etermined/calculated by the software and written into the MAC_Timestamp_Egress_Corr_* registers.

The correction value consists of the following 3 components:

1. External latency in the PHY layer between the output of the GMAC and the boundary of the port and the network
   If the PHY is compliant to the IEEE 802.3 Clause 45 MMD registers, it has a register indicating the maximum and minimum egress latency. The software can read these registers and determine the average egress latency in the PHY. Alternatively (if the PHY does not support these registers), the egress latency must be determined from its datasheet or timing characteristics.
2. Internal latency from the internal capture point and the output of the GMAC
   This internal egress latency can be read from the MAC_Egress_Timestamp_Latency register. This is a read-only register and gives the latency in scaledNanoseconds format defined in IEEE 1588 Clause 5.3.2. The latency differs based on the operating speed. Therefore, the software must read this register after any speed change in the MAC, to determine the current internal latency.
3. CDC synchronization error

The CDC synchronization error value differs depending on the One-step timestamping mode. When One-step timestamping is enabled, the value = (1 × period of PTP_CLOCK (8 ns) + 4 × period of TX_CLK). Otherwise (Two-step timestamping mode), the value = −(2 × period of PTP_CLOCK (8 ns)).

28.7.1.7 PTP Processing and Control

Table 28.21 shows the common message header for the PTP messages. This format is taken from the IEEE 1588-2008.

Table 28.21 Message format defined in IEEE 1588-2008

Bits								Octet	Offset
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
Reserved				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
Reserved								1	5
flagField								2	6
correctionField								8	8
Reserved								4	16
sourcePortIdentity								10	20
sequenceId								2	30
controlField *1								1	32
logMessageInterval								1	33

Note 1. control Field is used in version 1. In version 2, message Type field is used for detecting different message types.

There are some fields in the Ethernet payload that you can use to detect the PTP packet type and control the snapshot to be taken. These fields are different for the following PTP packets:

- PTP packets over IPv4
- PTP frames over IPv6
- PTP packets over ethernet

(1) PTP Packets Over IPv4

Table 28.22 provides information about the fields that are matched to control snapshot for the PTP packets sent over UDP over IPv4 for IEEE 1588 version 1 and 2. The octet positions for the tagged packets are offset by 4. This is based on the IEEE 1588-2008, Annex D and the message format defined in Table 28.21.

Table 28.22 IPv4-UDP PTP packet fields required for control and status (1 of 2)

Field Matched	Octet Position	Matched Value	Description
MAC Packet Type	12, 13	0x0800	IPv4 datagram
IP version and Header Length	14	0x45	IP version is IPv4
Layer 4 Protocol	23	0x11	UDP
IP Multicast Address (IEEE 1588 version 1)	30, 31, 32, 33	0xE0, 0x00, 0x01, 0x81 (or 0x82 or 0x83 or 0x84)	Multicast IPv4 addresses allowed: • 224.0.1.129 • 224.0.1.130 • 224.0.1.131 • 224.0.1.132
IP Multicast Address (IEEE 1588 version 2)	30, 31, 32, 33	0xE0, 0x00, 0x01, 0x81 (Hex) 0xE0, 0x00, 0x00, 0x6B (Hex)	• PTP Primary multicast address: 224.0.1.129 • PTP Pdelay multicast address: 224.0.0.107
UDP Destination Port	36, 37	0x013F, 0x0140	• 0x013F: PTP event messages • 0x0140: PTP general messages
PTP Control Field (IEEE 1588 version 1)	74	0x00, 0x01, 0x02, 0x03, 0x04	• 0x00: SYNC • 0x01: Delay_Req • 0x02: Follow_Up • 0x03: Delay_Resp • 0x04: Management

Table 28.22 IPv4-UDP PTP packet fields required for control and status (2 of 2)

Field Matched	Octet Position	Matched Value	Description
PTP Message Type Field (IEEE 1588 version 2)	42 (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, 0xD	• 0x0: SYNC • 0x1: Delay_Req • 0x2: Pdelay_Req • 0x3: Pdelay_Resp • 0x8: Follow_Up • 0x9: Delay_Resp • 0xA: Pdelay_Resp_Follow_Up • 0xB: Announce • 0xC: Signaling • 0xD: Management
PTP Version	43 (nibble)	0x1 or 0x2	• 0x1: Supports PTP version 1 • 0x2: Supports PTP version 2

(2) PTP Frames Over IPv6

Table 28.23 provides information about the fields that are matched to control the snapshots for the PTP packets sent over UDP over IPv6 for IEEE 1588 version 1 and 2. The octet positions for the tagged packets are offset by 4. This is based on the IEEE 1588-2008, Annex D and the message format defined in Table 28.21.

Table 28.23 IPv6-UDP PTP packet fields required for control and status

Field Matched	Octet Position	Matched Value	Description
MAC Packet Type	12, 13	0x86DD	IP datagram
IP version	14 (bits [7:4])	0x6	IP version is IPv6
Layer 4 Protocol	20 *1	0x11	UDP
PTP Multicast Address	38 – 53	FF0x:0:0:0:0:0:0:181 (Hex) FF02:0:0:0:0:0:0:6B (Hex)	• PTP Primary multicast address: FF0x:0:0:0:0:0:0:0:181 (Hex) • PTP Pdelay multicast address: FF02:0:0:0:0:0:0:0:6B (Hex)
UDP Destination Port	56, 57 *1	0x013F, 0x0140	• 0x013F: PTP event messages • 0x0140: PTP general messages
PTP Control Field (IEEE 1588 version 1)	94 *1	0x00, 0x01, 0x02, 0x03, 0x04	• 0x00: SYNC • 0x01: Delay_Req • 0x02: Follow_Up • 0x03: Delay_Resp • 0x04: Management
PTP Message Type Field (IEEE 1588 version 2)	62 *1 (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, or 0xD	• 0x0: SYNC • 0x1: Delay_Req • 0x2: Pdelay_Req • 0x3: Pdelay_Resp • 0x8: Follow_Up • 0x9: Delay_Resp • 0xA: Pdelay_Resp_Follow_Up • 0xB: Announce • 0xC: Signaling • 0xD: Management
PTP Version	63 *1 (nibble)	0x1 or 0x2	• 0x1: Supports PTP version 1 • 0x2: Supports PTP version 2

Note 1. control Field is used in version 1. In version 2, message Type field is used for detecting different message types.

(3) PTP Packets Over Ethernet

Table 28.24 provides information about the fields that are matched to control the snapshots for the PTP packets sent over Ethernet for IEEE 1588 version 1 and 2. The octet positions for the tagged packets are offset by 4. This is based on the IEEE 1588-2008, Annex D and the message format.

Table 28.24 Ethernet PTP packet fields required for control and status

Field Matched	Octet Position	Matched Value	Description
MAC Destination Multicast Address *1	0 – 5	01-1B-19-00-00-00 01-80-C2-00-00-0E	All PTP messages can use any of the following multicast addresses *2: • 01-1B-19-00-00-00 • 01-80-C2-00-00-0E *3
MAC Packet Type	12, 13	0x88F7	PTP Ethernet packet
PTP Control Field (IEEE 1588 version 1)	46	0x00, 0x01, 0x02, 0x03, 0x04	• 0x00: SYNC • 0x01: Delay_Req • 0x02: Follow_Up • 0x03: Delay_Resp • 0x04: Management
PTP Message Type Field (IEEE 1588 version 2)	14 (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, or 0xD	• 0x0: SYNC • 0x1: Delay_Req • 0x2: Pdelay_Req • 0x3: Pdelay_Resp • 0x8: Follow_Up • 0x9: Delay_Resp • 0xA: Pdelay_Resp_Follow_Up • 0xB: Announce • 0xC: Signaling • 0xD: Management
PTP Version	15 (nibble)	0x1 or 0x2	• 0x1: Supports PTP version 1 • 0x2: Supports PTP version 2

Note 1. The unicast address match of destination addresses (DA), programmed in MAC address 0 to 31, is used if the TSENMACADDR bit of MAC_Timestamp_Control register is set.
Note 2. IEEE 1588-2008, Annex F
Note 3. The MAC does not consider the PTP version 1 messages with Peer delay multicast address (01-80-C2-00-00-0E) as valid PTP messages.

28.7.1.8 Transmit Path Functions

The MAC captures a timestamp when the Start Packet Delimiter (SFD) of a packet is sent on the GMII or MII interface. The packets, for which the timestamps has to be captured can be controlled on per-packet basis. Each Transmit packet can be marked to indicate whether a timestamp should be captured for it.

The MAC does not process the transmitted packets to identify the PTP packets. You need to specify the packets for which you want to capture timestamps. You can specify the packets by using the control bits in the Transmit descriptor. The MAC returns the timestamp to the software inside the corresponding Transmit descriptor, thus connecting the timestamp automatically to the specific PTP packet. The 64-bit timestamp information is written to the TDES0 and TDES1 fields. The TDES0 field holds the 32 least significant bits of the timestamp.

28.7.1.9 Receive Path Functions

The MAC can be programmed to capture the timestamp of all packets received on the GMII or MII interface or to process packets to identify the valid PTP messages. Use the following options of the MAC_Timestamp_Control register to control the snapshot of the time to be sent to the application:

- Enable snapshot for all packets
- Enable snapshot for IEEE 1588 version 1 or version 2 timestamp
- Enable snapshot for PTP packets transmitted directly over Ethernet or UDP-IP-Ethernet
- Enable timestamp snapshot for the received packet for IPv4 or IPv6
- Enable timestamp snapshot only for EVENT messages (SYNC, DELAY_REQ, PDELAY_REQ, or PDELAY_RESP)
- Enable the node to be a master or slave and select the snapshot type
  This feature controls the type of messages for which snapshots are taken.

Note: The GMAC also supports the PTP messages over VLAN packets.

Table 28.25 Timestamp snapshot dependency on register bits

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	PTP Messages
00b	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp
00b	0	1	SYNC
00b	1	1	Delay_Req
01b	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp, Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up
01b	0	1	SYNC, Pdelay_Req, Pdelay_Resp
01b	1	1	Delay_Req, Pdelay_Req, Pdelay_Resp
10b	x	x	SYNC, Delay_Req
11b	x	x	Pdelay_Req, Pdelay_Resp

The DMA returns the timestamp to the software inside the corresponding Receive Descriptor. The extended status, containing the timestamp message status and the IPC status, is written in normal descriptor RDES1 and the snapshot of the timestamp is written in RDES0 and RDES1 fields of context descriptor. The RDES0 field holds the 32 least significant bits of the timestamp.

28.7.2 IEEE 1588 System Time Source

To get a snapshot of the time, the MAC requires a reference system time in 64-bit format as defined in the IEEE 1588-2002. Reference system time is provided from Ethernet Switch module.

28.7.3 IEEE 1588 Auxillary Snapshot

28.7.3.1 IEEE 1588 Auxillary Snapshot

The auxiliary snapshot feature allows you to store a snapshot of the system time based on an external event. The event is considered to be the rising edge of the GMAC_PTPTRG or ELC signal.

Up to two auxiliary snapshot inputs and 4 depth of a single common auxiliary snapshot FIFO are available.

28.7.3.2 Description of IEEE 1588 Auxillary Snapshot

The snapshots taken for any input are stored in a common FIFO. The application can read the MAC_Timestamp_Status register to know the timestamp of which input is available for reading at the top of this FIFO.

The MAC stores these snapshots in a FIFO. Only 64-bits of the timestamp are stored in the FIFO. When a snapshot is stored, the MAC indicates this to the application with an interrupt. The value of the snapshot is read through a FIFO register access. If the FIFO becomes full and an external trigger to take the snapshot is asserted, a snapshot trigger-missed status (ATSSTM) is set in the MAC_Timestamp_Status register. This indicates that the latest auxiliary snapshot of the timestamp is not stored in the FIFO. The latest snapshot is not written to the FIFO when it is full.

When an application reads the 64-bit timestamp from the FIFO, the space becomes available to store the next snapshot. You can clear a FIFO by setting the ATSFC bit in MAC_Auxiliary_Control register. When multiple snapshots are present in the FIFO, the count is indicated in bits [27:25] of MAC_Timestamp_Status register.

28.7.4 PTP Timestamp Offload Function

28.7.4.1 Introduction to PTP Timestamp Offload Function

PTP Timestamp Offload Function is an optional feature that enables the automatic generation of specific PTP packets when the MAC is working as a specific node in the PTP network.

These packets may be generated periodically or triggered by the host software. In other modes, this feature can parse the incoming PTP packets on the receiver, and automatically generate and respond to the required PTP packets. This helps in offloading certain functions of a PTP node with better accuracy and lower latency of responses.

28.7.4.2 Description of PTP Offload Function

Based on the programmed mode, the MAC generates PTP Ethernet messages periodically or from the application, or based on reception of a particular PTP message.

Table 28.26 indicates the PTP message generation criteria.

Table 28.26 PTP message generation criteria

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	Mode	Criteria for Generation of PTP Messages	PTP Message Type Generated
00b	0	1	Ordinary or Boundary Slave	SYNC message reception	Delay_Req
00b	1	1	Ordinary or Boundary Master	Periodic or on trigger from application	SYNC
				Delay_Req message reception	Delay_Resp
01b	0	1	Transparent Slave	Periodic or on trigger from application	Pdelay_Req
				Pdelay_Req message reception	Pdelay_Resp
				SYNC message reception	Delay_Req
01b	1	1	Transparent Master	Periodic or on trigger from application	Pdelay_Req
				Pdelay_Req message reception	Pdelay_Resp
				Periodic or on trigger from application	SYNC
				Delay_Req message reception	Delay_Resp
11b	x	x	Peer-to-Peer Transparent	Periodic or on trigger from application	Pdelay_Req
				Pdelay_Req message reception	Pdelay_Resp
All other programming combinations are invalid for PTP Offload feature.					

Note: Clocks supporting peer delay mechanism must not generate delay request/delay response messages, according to IEEE 1588-2008 specification. However, the GMAC controller supports this for flexibility, with a programmable control bit (DRRDIS). You can use the DRRDIS bit to control the response generation for delay request/delay response message. For example, in transparent slave mode, delay request is generated in response to received sync only when the bit is reset.

For example, when the MAC is set as an Ordinary or Boundary Slave clock in the PTP network, it can respond to the reception of SYNC messages with an automatic generation and transmission of the corresponding Delay_Req message. Similarly, various other modes of operation are explained in Table 28.26.

The MAC supports the multicast communication model for the generation of SYNC and Pdelay_Req PTP messages. For instance, the Destination Address field of the generated PTP over Ethernet packet is the defined special multicast addresses (0x011B19000000 for all except peer delay mechanism messages and 0x0180C200000E for peer delay mechanism messages).

When the MAC responds to received SYNC, Delay_Req and Pdelay_Req PTP messages with special multicast destination address, it also uses the corresponding special multicast address in the DA field of the automatically generated Delay_Req, Delay_Resp, and Pdelay_Resp PTP messages, respectively.

When the MAC responds to received SYNC, Delay_Req and Pdelay_Req PTP messages with unicast destination address, it takes the SA field of the received packets and makes them as the DA field of the automatically generated Delay_Req, Delay_Resp, and Pdelay_Resp PTP messages, respectively.

At the same time, all the received PTP messages are forwarded to the application along with Rx status, indicating whether the response was generated by the MAC, if it satisfies the packet filtering logic of the MAC receiver

When the MAC automatically generates a PdelayReq or responds with a Delay_Req, the egress timestamp of these two PTP messages are provided in the Tx TS status (Tx Timestamp Status register and interrupt generated).

In addition to messageType and versionPTP fields match for basic PTP over Ethernet message detection, the following additional fields are matched to qualify the received PTP message type:

1. The domainNumber field is checked for a match against the value programmed in the CSR.
2. The twoStepFlag in flagField field is checked for one-step indication (0).

3. The transportSpecific field is checked for Default PTP over Ethernet (0x0) or 802.1AS mode (0x1) when enabled.

28.7.4.3 PTP Packet Generation

This section explains the format and content of the automatically generated PTP packets by the MAC when this mode is enabled. The template of the Common PTP Message Header is provided, as well as the detailed description of the fields of specific PTP packets generated.

Table 28.27 Common PTP message header and PTP message-specific fields

Filed	Description
messageType	The following encoded values are used for PTP message types: ● SYNC: 0x0 ● Delay_Req: 0x1 ● Pdelay_Req: 0x2 ● Pdelay_Resp: 0x3 ● Delay_Resp: 0x9
transportSpecific	The following transport protocol encoding is used: ● Default PTP over Ethernet: 0x0 ● 802.1AS mode: 0x1
versionPTP	It is always set to 2 because PTP version 2 is supported.
messageLength	There is no suffix supported, so this field contains the length of the PTP message that includes 34 byte PTP common header and the body specific to the message type. For SYNC and Delay_Req packets, this field contains 44, whereas for Delay_Resp, Pdelay_Req and Pdelay_Resp, it contains 54.
domainNumber	This contains the value from the MAC_PTO_Control register.
flagField	The following values are used: ● alternateMasterFlag (Octet 0 bit 0): 0 for SYNC and Delay_Resp ● twoStepFlag (Octet 0 bit 1): 0 for SYNC and Pdelay_Resp ● unicastFlag (Octet 0 bit 2): 0 for Multicast Address, 1 for Unicast Address
correctionField	For more information, see section 28.7.5.2. MAC Transmit PTP Mode.
sourcePortIdentity	This field takes the value programmed in the MAC_Source_Port_Identityx registers.
sequenceId	Pdelay_Resp and Delay_Resp use the same sequenceId field from received Pdelay_Req and Delay_Req PTP messages. For SYNC/Delay_Req, Pdelay_Req, a separate sequenceId counter is maintained. These sequenceId counters get incremented by 1 every time the corresponding message is generated and transmitted.
controlField	The following encoded values are used for controlField: ● SYNC: 0x00 ● Delay_Req: 0x01 ● Pdelay_Req: 0x02 ● Pdelay_Resp: 0x05 ● Delay_Resp: 0x03
logMessageInterval	● SYNC: This contains logSyncInterval from the corresponding MAC_Log_Message_Interval register. ● Delay_Resp: This contains the sum of DRSYNCR and logSyncInterval value taken from the MAC_Log_Message_Interval register for a multicast PTP message and 0x7F for unicast PTP message. ● Delay_Req, Pdelay_Req and Pdelay_Resp: 0x7F where logSyncInterval = log2 (Mean Value of Interval in seconds)
originTimestamp	This field is the captured egress timestamp for SYNC, Delay_Req, and Pdelay_Req PTP messages.
receiveTimestamp	For Delay_Resp PTP message, this is the ingress timestamp of the corresponding received Delay_Req PTP message.
requestingPortIdentity	For Delay_Resp and Pdelay_Resp PTP messages, this is the sourcePortIdentity field taken from the corresponding received Delay_Req and Pdelay_Req PTP messages.
requestReceiptTimestamp	For the Pdelay_Resp PTP message, this field is set to 0.

The MAC supports values of -15 to 15 for logSyncInterval fields, which translates to a range from 32.768 micro second (2^{-15}) to 2^{15} second. For a given value of log sync interval (N), the time interval between two SYNC packets is given by the following:

- $2^{(30+N)}$ ns, when N is negative (-1 to -15)
- 2^{N} seconds, when N is positive (0 to 15)

For example:

- When logSyncInterval is programmed to 1, the interval is 2^1; therefore, the SYNC message is sent once every 2 seconds.
- When logSyncInterval is programmed to -1, the interval is 2^{-1} = 0.536 seconds; therefore, the SYNC message is sent once every 536 milliseconds. The value is 0.536 seconds, because 2^{-30} = 1 ns.
- When logSyncInterval is programmed to -5, the interval is 2^{-5} = 33.55ms; therefore, the SYNC message is sent once every 33.55 milliseconds.

Note: The MAC uses the PTP system time to generate the intervals for periodic packet transmission. For negative values of log message interval programmed, the generated period may deviate from the value given by the equation $2^{(30+N)}$, because of the non-binary nature of the nanoseconds field of the system time.

28.7.5 One-Step Timestamp

28.7.5.1 Introduction to One-Step Timestamp

The GMAC supports the one-step timestamp feature. When the One step timestamp feature is enabled, the MAC identifies the offset in the packet and inserts the timestamp received from the application at that offset.

28.7.5.2 MAC Transmit PTP Mode

Depending upon the type of message and its mode, the MAC updates the following fields of Transmit PTP packets:

- correctionField in the PTP header of messages
- originTimestamp in SYNC, Delay_Req, and Pdelay_Req messages

Table 28.28 lists how the PTP mode is selected based on the settings of SNAPTYPSEL, TSMSTRENA, and TSEVNTENA bits of the MAC_Timestamp_Control register and the fields that are updated for the incoming PTP packets based on the message type in that mode, during the one-step timestamping operation.

Table 28.28 MAC transmit PTP mode and one-step timestamping operation (1 of 2)

Programming			Mode	Per Packet Control*1			Messages Processed on Tx
SNAPTYPSEL	TSMSTRENA	TSEVNTENA		TTSE*2	OSTC*3	TTS*4	
x	x	x	N/A	1	x	x	Timestamp is captured and returned to application
x	x	x	N/A	x	0	x	OST operation is not performed (PTP packet is not modified)
00b	x	0	End-to-end transparent	0	1	Ingress TS	Sync (correction field for residence time and Ingress Asym cor)
							Delay_Req (correction field for residence time and Egress Asym Cor)
00b	0	1	Ordinary or Boundary Slave	1	1	x	Delay_Req (originTimestamp field) Delay_Req (correction field for Egress Asym cor)
00b	1	1	Ordinary or Boundary Master	0	1	x	Sync (originTimestamp field) Sync (correction field for sub-nanosecond cor)
01b	x	0	End-to-End Transparent with support for peer delay mechanism	0	1	Ingress TS	Sync (correction field for residence time and Ingress Asym cor)
						Ingress TS	Pdelay_Req (correction field for residence time and Egress Asym Cor)
						Ingress TS	Pdelay_Resp (correction field for residence time and Ingress Asym Cor)

Table 28.28 MAC transmit PTP mode and one-step timestamping operation (2 of 2)

Programming			Mode	Per Packet Control*1			Messages Processed on Tx
SNAPTYPSEL	TSMSTRENA	TSEVNTENA		TTSE *2	OSTC*3	TTS*4	
01b	0	1	Ordinary or Boundary Slave with support for peer delay mechanism or Peer to Peer Transparent	0	1	Ingress TS	Sync (correction field for residence time and Ingress Asym cor) (applicable only for Peer to Peer transparent clock operation)
				1	1	x	Delay_Req (originTimestamp field) Delay_Req (correction field for Egress Asym cor)
				1	1	x	Pdelay_Req (originTimestamp field) Pdelay_Req (correction field for Egress Asym Cor)
				0	1	Ingress TS for Pdelay_Req	Pdelay_Resp (correction field for turnaround time and Ingress Asym Cor)
01b	1	1	Ordinary or Boundary Master with support for peer delay mechanism	0	1	x	Sync (originTimestamp field) Sync (correction field for sub-nanosecond cor)
				1	1	x	Pdelay_Req (originTimestamp field) Pdelay_Req (correction field for Egress Asym Cor)
				0	1	Ingress TS for Pdelay_Req	Pdelay_Resp (correction field for turnaround time and Ingress Asym Cor)
10b	x	x	End-to-End Transparent	0	1	Ingress TS	Sync (correction field for residence time and Ingress Asym cor)
						Ingress TS	Delay_Req (correction field for residence time and Egress Asym Cor)
11b	x	x	Peer-to-Peer Transparen	0	1	Ingress TS	Sync (correction field for residence time and Ingress Asym cor)
				1	1	x	Pdelay_Req (originTimestamp field) Pdelay_Req (correction field for Egress Asym Cor)
				0	1	Ingress TS for Pdelay_Req	Pdelay_Resp (correction field for turnaround time and Ingress Asym Cor)

Note 1. The per packet control values provided here are the recommended settings used by devices in typical PTP operation, for the programmed mode.
Note 2. TTSE represents TTSE bit of transmit descriptor. The TTSE function is independent of the OST function and the programmed operation mode for OST. The MAC captures and returns the timestamp when the TTSE bit is set.
Note 3. OSTC represents OSTC bit of transmit descriptor.
Note 4. TTS represents the timestamp value provided in the TTSH, TTSL fields of transmit descriptor.

28.7.5.3 Enabling One-Step Timestamp

You can enable the one-step timestamp feature for a packet by setting OSTC field in the descriptor. For updating the correction field in certain PTP packets, the ingress timestamp must be given in the TSSL and TSSH fields.

28.7.5.4 One-Step Time Stamping for PTP Over UDP

GMAC supports one-step timestamp operation for PTP messages sent over UDP/IPv4 and UDP/IPv6. Because origin timestamp and/or correction field (based on the PTP message type and the programmed PTP mode) in the PTP message are updated as part of the one-step timestamp operation, it also requires update to the checksum field of the PTP message sent over UDP/IP.

In GMAC, checksum is updated for a packet, in the MTL before the packet is sent to the MAC where the PTP message is updated for one step timestamp operation. PTP requires the timestamp to be captured while the packet (SFD of the packet) is on the line and one step timestamp requires the origin timestamp and/or correction field to be updated in the PTP header while the packet is being transmitted on the line. Because the PTP message is present in the UDP payload, it is not possible to update the checksum field in the UDP header for modifications made to the PTP message, because it is already transmitted by then. Instead, the following options are supported that are in line with specification mentioned in Annex D & E of IEEE 1588-2008:

- When PTP message is sent over UDP/IPv4, the UDP checksum may be set to 0. The application must set the UDP checksum to 0 and set CIC to 0 or 1 while providing the packet to the MAC and the MAC does not update checksum for changes in origin timestamp or correction field.
- When PTP message is sent over UDP/IPv6, a transmitting node extends the UDP payload of all PTP messages by 2 octets beyond the end of the PTP message. The contents of the UDP checksum field or the final 2 octets of the UDP payload might be modified to ensure that the UDP checksum remains intact after modification of PTP fields. The application must add the two extra bytes at the end of the PTP message while providing the packet to the MAC and the MAC updates these two bytes to keep the checksum correct.

A programmable option (CSC field in MAC_Timestamp_Control register) supports UDP with non-zero checksum for IPv4. When set, the two bytes at the end of the PTP message are updated to keep the UDP checksum correct, similar to PTP over UDP/IPv6. So, the application must add two bytes at the end of the PTP message sent to the GMAC and GMAC updates these two bytes to keep the checksum correct.

Updating the pad bytes to keep the checksum correct requires the old value of correction field and/or origin timestamp in the packet. To get better latency and area, the application can set the origin timestamp field to zero, for PTP message in particular modes in which GMAC overwrites the origin timestamp. For example, GMAC operating in Ordinary/Boundary master mode overwrites the origin timestamp field of the sync message, when OSTC is set. Because the sync message is originated by the port operating in Ordinary/Boundary master mode, it is possible for the application generating the message to set the origin timestamp field to zero.

Transmit checksum engine is automatically enabled when this feature is selected. GMAC uses the transmit checksum engine to identify the start of PTP message in the UDP/IP packet. Checksum engine has the capability to detect some kind of errors in the header (L3) and payload (L4). Also is not computed for some type of packets, which is indicated by "No" in Table 28.55. For one step, operation the packet is free of those errors and have "Yes/Not applicable" for the corresponding packet type in Table 28.55. If an error is reported by the checksum engine or there is a "No" in Table 28.55, for PTP messages sent over UDP/IP, no update is performed on the PTP message. See section 28.12.1. Transmit Checksum Offload Engine for details about the errors detected by transmit checksum engine.

Note:
- The packet is sent with specific IP address and UDP port numbers and does not have IPv4 options/IPv6 extension headers. Applications must take care of these requirements before giving the packet to the GMAC. GMAC does not validate them. As PTP is identified by specific value in the UDP destination port and as GMAC does not look into these fields, it assumes an IP packet with OSTC set as a PTP packet, and it is the responsibility of the application to set the OSTC bit only for IP packets which contains PTP messages.
- For PTP over UPD/IPv6 and PTP over UDP/IPv4 with CSC set, the GMAC does not verify the presence of the two pad bytes, instead it overwrites the two bytes after the PTP message.
- When Enable One step timestamp for PTP over UDP/IP feature is selected, Enable Transmit TCP/IP Checksum offload feature is automatically enabled and this is enabled only for Queue 0 in multiple transmit queue configurations. If the application requires OST support for PTP messages sent over UDP/IP for packets sent from queues other than 0, transmit checksum engine must be manually enabled for those transmit queues. If checksum engine is not enabled, no updates are done for PTP messages sent over UDP/IP even though OSTC is set.
- One-Step timestamp operation is not supported for tunneled packets

28.8 Multiple Channels and Queues Support

This chapter describes how the GMAC supports multiple queues and channels.

28.8.1 Multiple Queues and Channels Support in the Transmit Path

28.8.1.1 Introduction to Multiple Queues Support in the Transmit Path

The GMAC supports up to eight Transmit DMA and Transmit queues. The number of Tx DMA channels is equal to the number of Tx queues and is direct one-to-one mapping. This enables the interleaving of packet/data transfer of multiple Tx DMA on the host/system bus and each Tx Queue is reserved for the corresponding Tx DMA. Therefore, the integrity of packet transfer by each Tx DMA is maintained in the Tx queue and enables efficient utilization of the host bus for data transfers among multiple Tx DMA without depending on the packet length, Transmit buffer availability and so on.

If two Tx DMAs are allowed to transfer data into the same Tx queue, when a Tx DMA starts a packet transfer, other Tx DMA cannot transfer data to the same Tx queue unless the previous packet transfer is complete. This means that the second

DMA remains idle until the first packet transferred is complete, which is effectively a one-to-one mapping of the Tx DMA to the Tx queue.

28.8.1.2 Description of Fixed Priority Scheme

The fixed priority scheme is the default priority scheme for the DMA channels. In fixed priority scheme, the channel with highest priority always wins the arbitration when it requests the bus. Table 28.29 provides information about the priority levels of DMA channels.

Table 28.29 Fixed Priority Scheme for DMA Channels

Priority Level	Channel
0 (low)	Channel 0
1	Channel 1
2	Channel 2
3	Channel 3
4	Channel 4
5	Channel 5
6	Channel 6
7 (high)	Channel 7

The Fixed Priority scheme does not necessarily cause "starving" of lower priority channels for the AXI bus because of the capabilities of the AXI protocol and characteristics of the TxDMA, described as follows:

- Read command requests from a TxDMA for a burst of data transfers is driven on a separate AXI channel and gets executed in one clock cycle itself. The actual data transfer starts on a separate AXI channel only after the request is accepted and it takes multiple clock cycles to complete with relatively large read access latencies.
- AXI master is free to drive out subsequent requests even before the previous data transfers are completed.
- A TxDMA requests for another data transfer only after the data transfer for its previous request is completed and transferred to the MTL TxQ and space is available in TxQ to take in the next request.
- It is possible to program to control the maximum number of outstanding requests, Requested Burst sizes by DMA, and the burst sizes of each data transfer on AXI.

Considering the previous points, it is possible to program such that the requests of all the TxDMAs are placed and executed on the AXI bus before the data transfer of the first TxDMA is completed and it raises the next request to the arbiter.

Effectively, all the TxDMA channels get equal access to AXI bus and the corresponding TxQs get filled up at the same rate. The data gets read out from the TxQs as per the Traffic Class scheduler in the MTL. So at a steady state (assuming that AXI bandwidth available to fetch data from system memory is always more than the operating line rate), the bandwidth for each channel is controlled by the priorities and settings of the Traffic Scheduler in the MTL.

28.8.1.3 Description of Weighted Strict Priority

In Weighted Strict Priority (WSP), the weight corresponds to the number of burst transfers given to a channel in one arbitration cycle. The unused burst transfers of one or more channels are reallocated based on the priority. The channel with highest priority gets the unused burst transfer time before it is allocated to a channel with next highest priority.

The channel priorities are fixed. Channel 7 has the highest priority and Channel 0 has the lowest priority. When a channel uses the allocated burst transfers, the channel with next lower priority is processed. After processing the allocated bandwidth of all channels that had packets to transmit, any unused burst transfer time is allocated to the channel of the highest priority (if required), and then next highest priority (if required), and so on.

28.8.1.4 Description of Weighted Round Robin

GMAC supports a Weighted Round Robin (WRR) priority scheme, in which the weight can be programmed through the DMA_CHn_TX_CONTROL register. In Weighted Round Robin (WRR), all channels are serviced in round-robin order according to the weights settings. The TCW field of the MA_CHn_Tx_Control register provides the weight for each Transmit channel as shown in Table 28.30.

Table 28.30 Weight for DMA channels

TCW Field	Transmit Channel Weight
000b	1
001b	2
010b	3
011b	4
100b	5
101b	6
110b	7
111b	8

The configured weights correspond to the number of burst transfers given to a channel in one arbitration cycle. The unused or excess burst transfers are distributed equally to all channels.

28.8.2 Multiple Queues and Channels Support in the Receive Path

28.8.2.1 Introduction to Multiple Queues in the Receive Path

The GMAC supports up to eight Receive channels. The number of Rx channels is independent of the Rx queues.

28.8.2.2 Description of Multiple Channels and Queues in the Receive Path

The GMAC controller supports independent selection of the number of Rx DMA channels and the number of Rx Queues in the MTL Rx Buffer. Therefore, the packets from any Rx queue can be serviced by any of the Rx DMA. As all the packets of Rx Queues are stored in a shared RXFIFO buffer in MTL, the Rx scheduler selects the Rx queue whose packet data is to be transferred to the destination Rx DMA, which then forwards the packet data to the host memory. The sequence of events in time are as follows:

1. Each Rx DMA indicates whether it is ready to transfer data, that is, when the Rx DMA has fetched the descriptor and has a Receiver Buffer ready, to accept the packet data. Each Rx DMA also indicates the number of words that it can transfer, based on the space available in the Rx Buffer and the value programmed in the PBL field of its DMA_CHn_RX_CONTROL register.
2. The MTL Rx scheduler matches the number of bytes/packets available for transfer at the top of all Rx queues with the corresponding Rx DMA ready status and the number of bytes requested by it.
3. If there are multiple successful matches, the scheduler selects the Rx queue to be serviced, based on the arbitration schemes (Strict Priority or Weighted Strict Priority) decided by the RAA field of MTL_Operation_Mode register. In the Weighted Strict Priority scheme, the weights of each Rx queue is set in the RXQ_WEGT field of the corresponding MTL_RXQn_CONTROL register.
4. The MTL Rx controller fetches the data of the selected Rx queue from the shared RXFIFO memory and starts the transfer of the requested PBL number of bytes to the DMA block or until the end of packet and the corresponding Rx status is transferred (whichever is first).
5. After completing the current request, the scheduler repeats the process by starting a new arbitration cycle. However, it selects the same Rx queue for servicing, if the end of packet is not transferred and the RXQ_FRM_ARBIT field of the MTL_RXQn_CONTROL register is 1, for the Rx queue.

28.8.2.3 Rx Queue to DMA Mapping

The packets in the MTL Rx Queues can be routed to any one of the multiple DMA channels by programming the following registers:

- MTL_RxQ_DMA_Map0 register (for queues 0, 1, 2 and 3)
- MTL_RxQ_DMA_Map1 register (for queues 4, 5, 6 and 7)

The following types of Rx Queue to DMA mapping is possible through programming:

- Static Mapping

- Dynamic (Per Packet) Mapping

(1) Static Mapping

In Static Mapping mode, all packets of an Rx Queue are connected to a specific DMA channel.

In this mode, all of the packets of an Rx Queue are connected to a specific DMA channel. For example, all the packets from Rx Queue 0 can be routed to a DMA channel by programming Q0MDMACH (bits [3:0]) and Q0DDMACH (bit [7] = 0) of the MTL_RxQ_DMA_Map0 register.

Similarly, packets from other Rx Queues can be routed to any DMA channel by programming register fields corresponding to each Queue.

(2) Dynamic (Per Packet) Mapping

In Dynamic (per packet) Mapping mode, the destination DMA channel is decided by the MAC receiver for each packet.

In this mode, the destination DMA channel of a packet being read from a Rx Queue is not constant but decided independently for each packet. For example, if you set the Q1DDMACH bit of the MTL_RxQ_DMA_Map0 register, the static mapping is disabled for Rx Queue 1 and the value in Q1MDMACH is ignored. The destination DMA channel is decided by the MAC receiver for each packet, depending on the following in decreasing order of priority:

1. L3-L4 filter Based DMA Selection
   When the Enable Layer 3 and Layer 4 Packet Filter feature is enabled, the TCP/UDP and IP header fields of the received packet are matched against the corresponding values programmed and enabled for comparison in the MAC_L3_L4_Address_Control register. If the match is successful, the DMA channel number programmed in the DMCHN field of the MAC_L3_L4_Address_Control register is selected as the destination DMA channel number provided DMCHEN bit of the same register is set.
   If none of the L3-L4 Registers give a comparison match, GMAC proceeds to the next step.
2. Extended VLAN Based DMA Selection
   Extended routing is applicable only if the VLAN Filter has passed. Routing is only based on Perfect Filter Result. Each perfect filter has a DMA Channel Enable and a DMA Channel Number field which can be programmed. Routing is applicable for a filter, only if the DMA Channel Enable bit is set.
   The frame is routed to the smallest Matching Filter's DMA Channel provided it is enabled. If that filter's DMA Channel number is not enabled, the frame gets routed to Channel 0. For example, if a frame's VLAN tag matches filters 7, 3, and 1. Then the MAC checks if Filter 1's DMA Channel Number is enabled through programming. If yes, the frame gets routed to the programmed value; otherwise it gets routed to DMA. When the inverse filter is enabled; is routed to the least mismatched filter's DMA channel number provided it is enabled. If the DMA Channel enable bit is not set, then the frame is routed based on DA based addressing or to Channel 0.
   If Hash Filter is also enabled, it is used to determine the Filter result only. Routing still depends on the enabled Perfect Filters. If none of the perfect filters are enabled or if all of them are bypassed, the VLAN Filter based routing is not done. The frame is routed through DA Based addressing or to Channel 0. If all the perfect filters give a fail result and the Hash Filter has passed, VLAN Filter result is a pass. However, routing is based on DA Based Addressing or to Channel 0. The behavior is similar when inverse filtering is enabled.
3. Ethernet DA-Based DMA Selection
   The DA address of the received packet is compared against the programmed DA values in MAC Address Registers. If the address matches any of the programmed values, the corresponding DCS field (when enabled) determines the destination DMA channel number.

If none of the previous operations is able to make a successful match/decision, then the packet is routed to DMA Channel 0 by default.

(3) Broadcast/Multicast Packet Duplication

GMAC supports Broadcast/Multicast Packet Duplication feature to send the received Broadcast/Multicast packets to multiple DMA channels.

In a virtual system that supports multiple guest OS's (for instance, each OS owning a DMA), where each DMA could act as an independent Ethernet Node, there can be requirement to route the received multicast/broadcast packets to all/multiple DMA channels. This is implemented by having the MTL_RX_FIFO_read_controller to read the packet from the same queue multiple times and route the same packet to the specified multiple DMA channels.

Through software control, the DMA channel selection (DCS) field inside the MAC Address Registers are redefined to denote multiple DMA channels.

Note: Only the MAC_ADDRESSn_HIGH (n = 0 to 31) registers support Broadcast/Multicast packet duplication

The DCS field of the of MAC_ADDRESSn_HIGH (n = 0 to 31) registers determine the DMA channel number to which the received packet (that matches the MAC Address present in that register) must be routed.

For the MAC_ADDRESSn_HIGH (n = 0 to 31) registers, DCS is a programmable field that supports more than one DMA channel number to be selected.

The DCS field can hold any one of the following values:

- Binary value of the channel number when the PDC bit of the MAC_Ext_Configuration register is set to 0.
- One-hot value of the channel number when PDC bit of the MAC_Ext_Configuration register is set to 1.

To enable the Multicast/Broadcast packets to be routed to multiple DMA channels, use the DA based routing. Packet duplication is not supported in other DMA channel routing mechanisms such as Extended VLAN, or L3-L4 based routing. In other words, if the destination DMA channel is decided by the VLAN or L3-L4 filter result, it is routed only to the specific channel even if the DA of the packet matches the multicast address in the MAC_Address register.

The Packet Duplication feature is supported only on the highest MTL Queue configured. Therefore, packet duplication for Broadcast/Multicast packets requires that the MCBCQ field of MAC_RxQ_Ctrl1 register is programmed to the highest RxQ present in the configuration.

In a packet duplication, the MTL Read Controller sequentially transfers the same packet to all the required RxDMA. This packet is removed from the MTL FIFO only after the packet is transferred to all the selected DMA channels. Therefore this feature is suitable for smaller packet sizes or very low bandwidth traffic; to ensure that the MTL RxQ does not overflow due to the long period for which the packet remains in the queue during the packet duplication.

Note: When more than one bit is set for the DMA channel number, a packet that is not routed to the highest receive queue is routed to only one DMA channel as indicated by the lowest bit set in the DCS field of the MAC_ADDRESSn_HIGH (n = 1 to 31) register.

28.8.3 Priority Scheme for Tx DMA and Rx DMA

The GMAC DMA arbiter supports two types of arbitration: fixed priority and weighted round-robin.

The DMA arbiter performs the arbitration between the Tx and Rx paths of DMA channels for accessing descriptors and data buffers. The DMA arbiter supports two types of arbitration: fixed priority and weighted round-robin. The DA bit of the DMA_Mode register specifies the arbitration scheme (fixed or weighted round-robin) between the Tx and Rx DMA of a channel.

If you have enabled the Tx DMA and Rx DMA of a channel, you can specify which DMA gets the bus when the channel gets the control of the bus. You can set the priority between the corresponding Tx DMA and Rx DMA by using the TXPR field of the DMA_Mode register. For round-robin arbitration, you can use the PR field of the DMA_Mode register to specify the weighted priority between the Tx DMA and Rx DMA.

Table 28.31 provides information about the priority scheme between Tx DMA and Rx DMA.

Table 28.31 Priority scheme for Tx DMA and Rx DMA (1 of 2)

Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Priority Schemes
x	x	x	0	1	Rx always has priority over Tx
0	0	0	0	0	Tx and Rx have equal priority. Rx gets the access first on simultaneous requests
0	0	1	0	0	Rx has priority over Tx in ratio 2:1
0	1	0	0	0	Rx has priority over Tx in ratio 3:1
0	1	1	0	0	Rx has priority over Tx in ratio 4:1
1	0	0	0	0	Rx has priority over Tx in ratio 5:1
1	0	1	0	0	Rx has priority over Tx in ratio 6:1
1	1	0	0	0	Rx has priority over Tx in ratio 7:1
1	1	1	0	0	Rx has priority over Tx in ratio 8:1

Table 28.31 Priority scheme for Tx DMA and Rx DMA (2 of 2)

Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Priority Schemes
x	x	x	1	1	Tx always has priority over Rx
0	0	0	1	0	Tx and Rx have equal priority. Tx gets the access first on simultaneous requests
0	0	1	1	0	Tx has priority over Rx in ratio 2:1
0	1	0	1	0	Tx has priority over Rx in ratio 3:1
0	1	1	1	0	Tx has priority over Rx in ratio 4:1
1	0	0	1	0	Tx has priority over Rx in ratio 5:1
1	0	1	1	0	Tx has priority over Rx in ratio 6:1
1	1	0	1	0	Tx has priority over Rx in ratio 7:1
1	1	1	1	0	Tx has priority over Rx in ratio 8:1

Note: Bits 14, 13, 12, 11, and 1 are not valid if one of the paths (Transmit or Receive) is not present in a channel. The Transmit and Receive paths are always present in Channel 0.

28.8.4 Selection of Tag Priorities Assigned to Rx Queues

The GMAC controller supports assigning tag priorities to Rx Queues through programming registers.

The VLAN Tag priorities can be assigned to Rx Queues by programming the PSRQ field in the corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers. The bit corresponding to the VLAN Tag priority can be set in the PSRQ field for assigning that priority to the Rx Queue. For example, if you want to assign VLAN Tag priority of 3 to Rx Queue 0, set bit [3] in PSRQ field of MAC_RxQ_Ctrl2 register. The VLAN Tag priority assigned to particular Rx Queue must be unique, that is, more than one Rx Queue cannot be assigned the same VLAN Tag priority. However, more than one VLAN Tag priorities can be assigned to same Rx Queue.

The settings in the PSRQ field is used for VLAN Tagged Rx packet routing to Rx Queues. The received VLAN Tagged Rx packet is routed to Rx Queue that has the VLAN Tag priority match.

28.8.5 Distribution of Rx Packets from MAC to Rx Queues

The GMAC controller provides multiple Rx queues to support classification and distribution of ingress packets. The MAC receiver pushes the ingress packets to the Rx queues based on packet type (tagged, untagged, multicast, PTP, AVB, and so on), tag priorities, MAC packet filter results, preemptable or express packet, and so on. The distribution is controlled by the settings of the MAC_RxQ_Ctrl* and MAC_Packet_Filter register fields.

GMAC additionally supports distribution of ingress packets based on user-defined Type field of the ingress packets when TBRQE field (bit [29]) of the MAC_RxQ_Ctrl1 register is enabled. You can program up to 8 Type for matching, in the MAC_TMRQ_REGSn registers which is accessible through MAC_Indir_Access_Ctrl and MAC_Indir_Access_Data registers.

Table 28.32 shows MAC_TMRQ_REGSn register specification.

Table 28.32 MAC_TMRQ_REGSn register specification

Bit	Symbol	Function	R/W
15:0	TYP	Type field Value Specifies the type value of packet that needs to be compared with received Ethernet packet. This field is valid when programmed to a value greater than or equal to 0x0600.	R/W
18:16	TMRQ	Type Match Rx Queue Number Specifies the receive queue number to which the packet needs to be forwarded on detecting a type match.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	PFEX	Preemption or Express Packet This bit specifies if it is a preemption packet or express packet.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Based on the packet filter settings, the MAC receiver decides to either drop or forward the Ingress packet. For details, see section 28.6.1. Packet Filtering Sequence. However, the results of Layer 3 and Layer 4 filtering, mentioned in section

28.6.4. Layer 3 and Layer 4 Filtering are not considered for the distribution of ingress packets into the Rx queues. Only the results of DA/SA/VLAN filters mentioned in section 28.6.2. Source Address or Destination Address Filtering and section 28.6.3. VLAN Filtering are applicable.

Filtering of the broadcast packets is exclusively controlled by the DBF field, bit [5] of the MAC_Packet_Filter register. The filtering of multicast packets is controlled by the PM field, bit [4] of the MAC_Packet_Filter register, in addition to the DA/SA match.

By default, packets that fail any of the DA/SA filters, are dropped by the MAC Receiver. However, packets that fail the DA/SA/VLAN filter can still be forwarded to the application when the RA field of the MAC_Packet_Filter register is 1. Even when RA field is 0, VLAN filter fail packets can be forwarded if the VTFE field of the MAC_Packet_Filter register is 0.

The ingress packets go through the following operations in the decreasing order of priority.

1. RA = 0: Ingress packets are dropped if any one of the following condition is true.
   - DA/SA filter fail
   - VLAN filter fail and VTFE field of the MAC_Packet_Filter register is 1
2. RA = 1: Ingress packets that fail the DA/SA filter are also forwarded to the Rx queue as mentioned in Table 28.33.
3. RA = 1 and VTFE = 0: Ingress tagged packets that pass the DA/SA filter but fail the VLAN filter, are forwarded to the Rx queue as mentioned in Table 28.34.
4. Ingress packets that pass the DA/SA/VLAN filters are forwarded to the Rx queue
   - Table 28.35 shows the routing priority in the decreasing order when OMCBCQ field of the MAC_RxQ_Ctrl1 register is 0.
   - Table 28.36 shows the routing priority in the decreasing order when OMCBCQ field of the MAC_RxQ_Ctrl1 register is 1.

Table 28.33 Routing DA/SA failed packets to Rx queue

Packet Type	TSN Type	Destination Rx Queue Determined By
Broadcast/Multicast	Express	if MFFQE = 1, MFFQ; else if VFFQE = 1, VFFQ; (only for tagged packets) else RxQ0
Unicast	Express	if UFFQE = 1, UFFQ; else if VFFQE = 1, VFFQ; (only for tagged packets) else RxQ0
All	Preemptable	FPRQ field

Table 28.34 Routing DA/SA pass but VLAN filter fail packets to Rx queue

TSN Packet Type	Destination Rx Queue Determined By
Express	if VFFQE = 1, VFFQ; else RxQ0
Preemptable	RQ field

Table 28.35 Priority routing of filter pass packets when OMCBCQ = 0 (1 of 2)

Packet Type	TSN Type	Tagged	Destination Rx Queue Determined By
Broadcast/Multicast	Express	Yes	if MCBCQEN = 1, MCBCQ; else if TBRQE = 1, TMRQ; else PSRQ
		No	if MCBCQEN = 1, MCBCQ; else if TBRQE = 1, TMRQ; else UPQ
Unicast AVTP Control	Express	No	AVCPQ
		Yes	if TACPQE = 1, AVCPQ; else PSRQ
Unicast PTP over Ethernet	Express	No	PTPQ
		Yes	PTPQ or PSRQ based on the value of TPQC.

Table 28.35 Priority routing of filter pass packets when OMCBCQ = 0 (2 of 2)

Packet Type	TSN Type	Tagged	Destination Rx Queue Determined By
Remaining Unicast	Express	No	If TBRQE = 1, TMRQ; else UPQ
		Yes	If TBRQE = 1, TMRQ; else PSRQ
All Packets	Preemptable	No	If TBRQE = 1, TMRQ; else FPRQ
		Yes	If TBRQE = 1, TMRQ; else PSRQ

Table 28.36 Priority routing of filter pass packets when OMCBCQ = 1

Packet Type	TSN Type	Tagged	Destination Rx Queue Determined By
All AVTP Control	Express	No	AVCPQ
		Yes	if TACPQE = 1, AVCPQ; else PSRQ
All PTP over Ethernet	Express	No	PTPQ
		Yes	PTPQ or PSRQ based on TPQC
Remaining Broadcast/Multicast	Express	Yes	if MCBCQEN = 1, MCBCQ; else if TBRQE = 1, TMRQ; else PSRQ
		No	if MCBCQEN = 1, MCBCQ; else if TBRQE = 1, TMRQ; else UPQ
Remaining Unicast	Express	No	if TBRQE = 1, TMRQ; else UPQ
		Yes	if TBRQE = 1, TMRQ; else PSRQ
All Packets	Preemptable	No	if TBRQE = 1, TMRQ; else FPRQ
		Yes	if TBRQE = 1, TMRQ; else PSRQ

Note: 1. The fields mentioned in Table 28.33 through Table 28.36 belong to MAC_RxQ_Ctrl0, MAC_RxQ_Ctrl1, MAC_RxQ_Ctrl2, MAC_RxQ_Ctrl3, or MAC_RxQ_Ctrl4 registers.

2. If the RxQ pointed by the various fields in Table 28.33 through Table 28.36 is not enabled in the MAC_RxQ_Ctrl0 register as per the packet type, the packet is written to
   - RxQ0 for Express packets
   - FPRQ field for Preemptable packets

   If RxQ0/FPRQ is not enabled, the packet is dropped.

3. If the AV8021ASMEN field of the MAC_Timestamp_Control register is 1, MAC receiver checks only for untagged PTP packet. Tagged packet with PTP payload are considered as Remaining (Unicast/Broadcast/Multicast) tagged packets in Table 28.33 through Table 28.36.

28.8.6 Audio Video Bridging

28.8.6.1 Introduction to AV Feature

The GMAC supports the AV data transfer in 100 Mbps and 1000 Mbps modes. The AV feature enables transmission of time-sensitive traffic over bridged local area networks (LANs). The following standards define various aspects of the AV feature implementation:

- IEEE 802.1Qav-2009: Allows the bridges to provide time-sensitive and loss-sensitive real-time audio video data transmission (AV traffic). It specifies the priority regeneration and controlled bandwidth queue draining algorithms that are used in bridges and AV traffic sources.
- IEEE 802.1Qat-2009: Allows the network resources to be reserved for specific traffic streams traversing a bridged local area network.
- IEEE 802.1AS-2011: Specifies the protocol and procedures used to ensure that the synchronization requirements are met for time-sensitive applications such as audio and video across bridged and virtual-bridged LANs consisting of LAN media where the transmission delays are fixed and symmetrical. For example, IEEE 802.3 full-duplex links include the maintenance of synchronized time during normal operation followed by addition, removal, or failure of network components and network reconfiguration.

While the AVB standards were originally designed for Audio and Video over standard Ethernet, extending these to control streams would additionally benefit the industrial and automotive sectors. Time Sensitive Networking (TSN) is a set of standards developed by the Time Sensitive Networking Task Group with the intent to extend the future AVB enhancements to other sectors that could benefit.

Bridges are increasingly used to interconnect devices that support scheduled applications (for example, industrial automation, process control and vehicle control). The IEEE 802.1Qbv-2015 (Enhancements to Scheduling Traffic) provides performance assurances of latency and delivery variation to enable these applications in an engineered LAN while maintaining the existing guarantees for the credit-based shaper and best-effort traffic.

The IEEE 802.3br (Interspersing Traffic) and IEEE 802.1Qbu (Frame Preemption) enables the suspension of a large preemptable packet being transmitted by the MAC layer to allow one or more express packets to be transmitted before the transmission of the preemptable packet is resumed. This provides the capability to schedule express traffic packets with minimal delays/latencies or at predictable times with efficient utilization of the line bandwidth.

As per the IEEE specification for 802.1Qbv, 802.1Qbu, and 802.3br, the TSN features are supported only in the following modes:

- Full duplex
- 100 Mbps or higher link speed
- Pause or no flow control
- When Qbv schedule is enabled for time-sensitive traffic, Pause is likely to interfere with the Qbv schedules. So, it is recommended not to enable PAUSE when Qbv scheduler is enabled.
- Two or more Tx queues (for Qbv) configured and enabled
- One or more Express Queues and one or more Preemption queues on both transmit and receive (for Qbu/802.3br) configured and enabled.

28.8.6.2 Transmit Path Functions

When the GMAC controller supports the AV Bridging feature, you can enable and disable certain functions associated with the transmit path.

When you select the AV feature, the Transmit paths of Queue 0 and the highest enabled queue are enabled by default.

The Transmit path of Queue 0 supports the strict priority algorithm, and it is used for best-effort traffic. For a queue, the strict priority algorithm determines that a packet is available for transmission if the queue contains one or more packets. When the threshold mode for MTL Tx FIFO is enabled, the strict priority algorithm determines that a packet is available for transmission if the queue contains a partial packet of size equal to the programmed threshold limit.

The Transmit paths of additional queues support traffic management by using the credit-based shaper algorithm. For a queue, the credit-based shaper algorithm determines that a queue is available for transmission if the following conditions are true:

- The queue contains one or more packets.
- The credit for the queue is positive as per the algorithm.

You can disable the credit-based shaper algorithm for all queues or lower-priority queues. For example, you can either disable the credit-based shaper algorithm for Queue 1 and Queue 2 or only for Queue 1. You should not disable the credit-based shaper algorithm for Queue 2 and enable it for Queue 1. When you disable the credit-based shaper algorithm for a queue, the channel uses the default strict priority algorithm.

Each Transmit DMA has a separate descriptor chain for fetching the transmit data. The Transmit channel that gets the access to the system bus depends on the DMA arbiter.

The Transmit path has a shared FIFO (MTL layer) for each queue. The data fetched by the DMA is put in the respective part of the FIFO. The MTL Tx Queue Scheduler controls which part of the FIFO data is transmitted by the MAC. If the credit-based shaper algorithm is enabled for a queue, the corresponding queue is selected for transmission if the following conditions are true:

- If the packet is available in the channel and has a positive or zero credit.
- If the higher priority queue has no packet waiting in the FIFO.

If the credit-based shaper algorithm is disabled for all queues, the packet to be transmitted from a queue is selected based on the priority scheme described in Table 28.37.

Table 28.37 Weight for DMA channels

TCW Field	Transmit Channel Weight
000b	1
001b	2
010b	3
011b	4
100b	5
101b	6
110b	7
111b	8

28.8.6.3 Receive Path Functions

When the GMAC controller supports the AV Bridging feature, you can enable and disable certain functions associated with the receive path.

When you select the AV feature, the Receive path of Queue 0 is enabled by default. All traffic is received on this channel. You can enable the Receive paths of additional queues. By enabling the Receive paths of multiple channels, you can demultiplex the received data to send the packets into separate receive channels.

To differentiate between the AV and non-AV traffic, the MAC provides a status that indicates if it is an AV packet and its corresponding VLAN Priority tag value. This status is updated in the Extended Status field of the Receive descriptor as explained in section 28.17.5. Receive Descriptor. All received packets with the EtherType field of 0x22F0 are detected as AV packets. The AV packets can be of the following two types:

- AV data packets: The AV data packets are always tagged. The tagged AV data packets are received based on the programmed priority value. To specify the channel to which an AV packet with a given priority must be sent, program bits [15:8] in the Transmit Flow Control Register of the corresponding queue.
- AV control packets: The AV control packets can be either tagged or untagged. The untagged AV control packets are received on Queue 0 by default. To receive these packets on any other queue, you can program bits [2:0] of the MAC_RxQ_Ctrl1 register. Similar to the AV data packets, the tagged AV control packets are received based on the programmed priority value.

In addition to the AV packets, you can receive the untagged PTP packets on any queue. By default, the PTP packets (tagged or untagged) are received on Queue 0. To receive these packets on any other queue, you need to program bits [6:4] of the MAC_RxQ_Ctrl1 register.

28.8.6.4 Credit-Based Shaper Algorithm

The MTL Queue Scheduler uses the credit-based shaper algorithm to arbitrate the AV traffic in all queues and the legacy Ethernet traffic in Queue 0. You can program the additional queues to use the credit-based shaper algorithm.

The following sections provide information about how you can implement the credit-based shaper algorithm:

- Credit Value
- idleSlopeCredit and sendSlopeCredit Values
- Bandwidth Status

(1) Credit Value

The credit value is part of the credit-based sharper algorithm used by the MTL Queue Scheduler.

The credit value is accumulated every transmit clock cycle, that is, 40 ns for 100 Mbps and 8 ns for 1000 Mbps. The credit to be added or subtracted per cycle can be fractional based on the required idleSlope and sendSlope values, as described in Table 28.38.

Table 28.38 Credit value per transmit cycle example

Mode	Values	Description
100 Mbps	• portTransmitRate = 100 Mbps • idleSlope = 70 Mbps (assuming 70% bandwidth reserved for a higher priority traffic class) • sendSlope = 30 Mbps	• credit = 2.8 bits accumulates per cycle (40 ns) for the higher priority traffic class when best-effort packet is being transmitted. • credit = 1.2 bits drains per cycle (40 ns) when higher priority traffic class packet is being transmitted.

The DMA stores the queue traffic in the respective part of the Tx FIFO based on the slot number in the transmit descriptor (if enabled) or the bandwidth availability on the AMBA AXI application bus.

The credit for a queue builds up only when the packet is available but it cannot be transmitted because the MAC is sending a packet from another queue. The GMAC supports another mode in which the credit can build up in advance for a queue in which no packet is available in respective part of the FIFO. This enables sending a burst of high priority traffic in a queue as soon as the data is available. You can enable this mode with bit 3 of the CBS Control registers of corresponding queues.

When reset, the accumulated credit parameter in the credit-based shaper algorithm is set to zero if there is positive credit, and there is no packet to transmit in a queue. The credit does not accumulate when there is no packet waiting in a queue and other queues are transmitting. When set, the accumulated credit parameter in the credit-based shaper algorithm is not reset to zero if there is positive credit and no packet to transmit in a queue. The credit accumulates even when there is no packet waiting in a queue and other queues are transmitting.

(2) idleSlopeCredit and sendSlopeCredit Values

The idleSlopeCredit and sendSlopeCredit values are programmable.

The software must program the idleSlopeCredit and sendSlopeCredit values. The programmed values should be the credit accumulated or drained per clock cycle scaled by 1024, such as, 2.8 × 1024 = 2867 and 1.2 × 1024 = 1229. In addition, the software must program the hiCredit and loCredit values, scaled by 1024, to adjust for scaling of the idleSlopeCredit and sendSlopeCredit values. This means that if computed hiCredit and loCredit values are 12,000 bits and 3,036 bits respectively, the values to be programmed in the hiCredit and loCredit registers of the corresponding channel are 12000 × 1024 bits and two's complement of 3036 × 1024, respectively.

(3) Bandwidth Status

The hardware maintains the status of the actual bandwidth consumed by each higher priority queue in the CBS status registers. This enables the software to estimate the average bandwidth consumed by numerically higher traffic classes as compared to the reserved bandwidth.

The CBS status register gives the average number of bits transmitted during the previous programmed slot interval (1, 2, 4, 8, or 16 slots of 125 µs) in a queue. The status register is updated even if the credit-based shaper algorithm is not enabled for a queue. The number of slots over which the average bits transmitted per slot are computed is programmed in bits [6:4] of the CBS control register of the respective queue. For example, if you have programmed two slots, the average bits are computed over slot numbers 0–1, 2–3, 4–5, and so on.

The value programmed in the idleSlopeCredit register of a queue is proportional to the bandwidth reserved for the queue. The software can allocate any bandwidth that is not used by the higher priority queue to the reserved bandwidth of the lower priority queue.

A lower priority queue, which is using the credit-based shaper algorithm, cannot use the unused reserved bandwidth of any higher priority queue that is using the credit-based shaper algorithm. However, a lower priority queue, which is using the strict-priority algorithm, can use the unused reserved bandwidth of any higher priority queue that uses the credit-based shaper algorithm. For example, Queue 1 and Queue 2 use the credit-based shaper algorithm (with reserved bandwidth of 50% and 25%, respectively) and Queue 0 uses the strict-priority algorithm. If Queue 1 uses only 40% of reserved bandwidth, the remaining 10% is used by Queue 0. Queue 2 cannot exceed the reserved bandwidth of 25%.

28.8.6.5 Slot Number Function and Audio Video Bridging

(1) Introduction to Slot Number Function

When the AV feature is enabled, you can use the slot number function to schedule the data fetching from the system memory by the DMA. This feature is useful when the source AV data needs to be transmitted at specific intervals.

You can program the slot number at which the DMA should fetch the data from system memory in the Transmit Descriptor Word 3 (TDES3). This 4-bit field allows the application to schedule data up to 16 slots; a programmable slot interval that ranges from 1 µs to 4096 µs and granularity of 1 µs. This field is applicable only for the AV channels.

(2) Description of Slot Number Function

When the DMA fetches a Tx descriptor, it compares the slot number of the Tx descriptor with the internally generated reference slot interval. The slot interval is a counter that is updated based on a programmable slot interval (with range from 1 µs to 4096 µs) of the IEEE 1588 system time. In addition, the slot interval counter is initialized to zero when the seconds field of the system time is incremented, that is, the sub-second counter rolls over. The DMA fetches the data only if it matches the current slot or the next slot. The DMA remains in the descriptor fetch state till there is a match.

To enable the DMA to fetch the data only if it matches the current slot or the next two slots, you can program bit 1 of the Slot Function Control and Status register of the corresponding DMA channel.

Note: If the slot number in the descriptor is less than the reference slot number, the DMA takes it as a future slot.

You can enable the check for slot number by setting bit 0 of the Slot Function Control and Status register of corresponding DMA channel. When this check is not enabled, the packets are fetched immediately after the descriptor is read. In addition, bits [19:16] indicate the value of the reference slot number in DMA.

28.8.7 Queue Modes

You can enable a Tx queue for generic/AV traffic. The queues are enabled for generic queuing using WRR or WSP algorithms.

For enabling a Tx queue for AV, the queuing is based on the CBS or SP algorithms.

You can enable an Rx queue for generic, or AV traffic. A particular queue enabled for generic/AV based routing is determined by the RXQn_EN field of corresponding queue. The following list explains how Rx queues are enabled based on the selected features:

- Multiple Rx queues without AV feature
  All queues are enabled for generic queuing based on the VLAN Tag priority. The VLAN Tag priority should match the PSRQ field of the MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers. By default untagged packets are routed to Receive queue specified in UPQ field of MAC_RxQ_Ctrl1 register. Queue 0 is the default value of UPQ field. You can override the default value with any other value, for the UPQ field. The Rx packets can also be routed to a particular DMA channel based on the DCS field of perfectly-matched MAC Address register.
- Multiple Rx queues with AV feature
  AV feature is enabled for all selected Rx queues. The queuing is done based on the VLAN Tag priority for DCB data packets. The VLAN Tag priority should match the PSRQ field of the MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers. The Rx packets can also be routed to a particular DMA channel based on the DCS field of perfectly-matched MAC Address register. The AV control packets (tagged or untagged) are routed based on the AVCPQ field of the MAC_RxQ_Ctrl1 register. The PTP over Ethernet packets are routed based on the AVPTPQ field of MAC_RxQ_Ctrl1 register.

28.8.7.1 Programming Guidelines for Disabling Flow Control in AV Queues

The GMAC supports programming of same priorities for AV and DCB enabled Rx queues, when both coexist. For example, priority 1 can be programmed in PSRQ[n] field corresponding to Rx Queue enabled for AV in MAC_RxQ_Ctrl2 /3 register and same priority 1 can be programmed in PSRQ[n] field corresponding to Rx Queue enabled for DCB in MAC_RxQ_Ctrl2 /3 register, at the same time. When AV and DCB operation coexists, it requires that the Flow Control is disabled for AV enabled queues. See section 28.18.11.4. Disabling Flow Control for AV Enabled Queues.

28.8.8 Queue Priorities

You can program the priority of an Rx queue corresponding field of MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers. The priority should be assigned in the following order:

1. AV queue (high priority)
2. DCB queue (medium priority)
3. Best-effort queue (low priority)

28.9 Enhancements to Scheduled Traffic (EST)

The IEEE 802.1Qbv-2015 defines the schedule for each of the queues on every egress port which makes the implementation aware of traffic arrival schedule. This information can be used to block the lower priority traffic from transmission in this time window/slot. This ensures that scheduled traffic is forwarded from sender to receiver through all the network nodes with a deterministic delay.

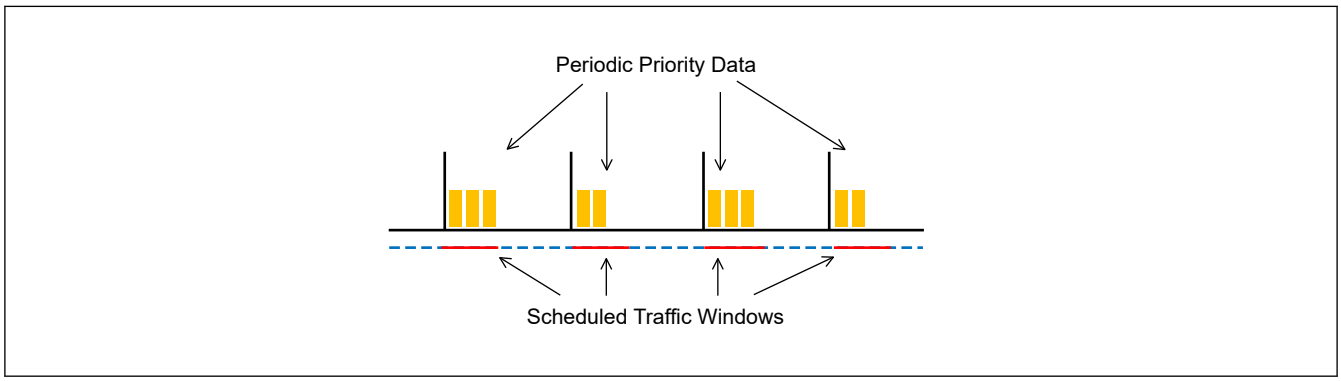


Figure 28.15 Weighted round robin example

One of the important requirements to achieve a low latency is to ensure there are no interfering frames during the scheduled windows that are reserved for high priority traffic. The use of scheduled traffic imposes limitations while starting a transmission.

As shown in Figure 28.16, if an interfering frame begins transmission just before the start of a reserved time period, it can extend transmission into the reserved window, and potentially interfere with higher priority traffic. Therefore, a guard band whose size is equal to the largest possible interfering frame is required before the window starts.

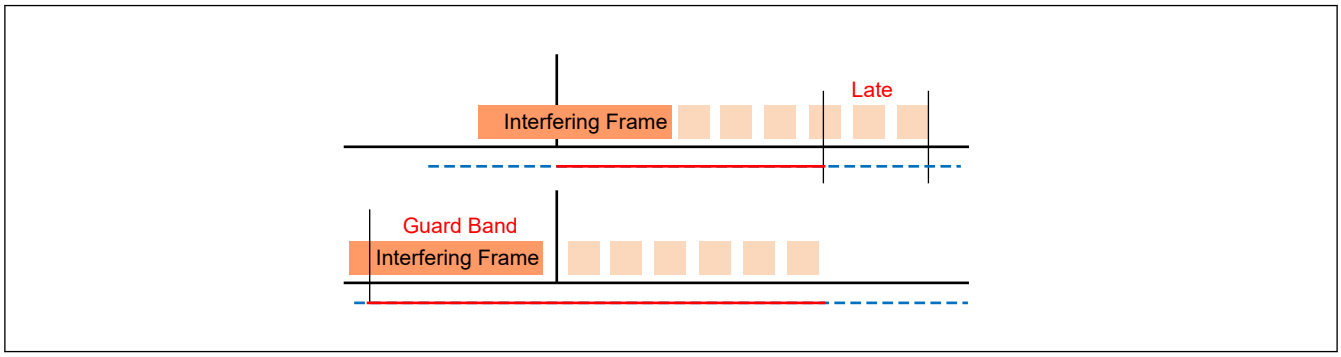


Figure 28.16 Implementing a guard band to avoid delays due to interfering frames

A larger guard band equates to a less efficient use of network bandwidth. However, with the implementation of IEEE802.1Qbu (frame preemption), this issue is addressed. Frame preemption breaks the interfering frame into smaller fragments. Therefore, the guard band needs to be only as large as the largest possible interfering fragment instead of the largest possible interfering frame.

During the guard band, only the frames that can complete the transmission of the entire frame before the next gate close event are permitted. This ensures that the high priority traffic can always start at the beginning of the window reserved for it.

28.9.1 Frequently Used Terms in EST

Following are some of the frequently used terms in the EST support and their definitions.

- Gate Control List
  The list in the hardware memory that holds the gate controls and the associated time intervals.
- Gate Controls
  For a given schedule (row in gate control list) there is a Gate Open (O) and Gate Close (C) state associated with each TC. The set of O or C values, whose width is same as configured TCs is called the Gate Controls. Example: CCOCOOCO means TC7 = C, TC6 = C, TC5 = O and so on.
- Time Interval

Time interval (in nano seconds) is a 24-bit field in the Gate Control list that indicates the time for which the associated gate controls are valid.

- Base Time Register
  Each Gate Control List is associated with a 64-bit Base Time Register that holds the start time (in PTP format) for the list.

28.9.2 Updates to the Transmit Scheduling to Support EST

To support EST, following updates are required to the transmit scheduling:

- Implementation of Gate Control List (GCL)
- Enforcing gate controls in the scheduler
- Accounting for Gate Open (O) duty cycle in the computation of idleSlope (CBS)

28.9.2.1 Implementation of Gate Control List (GCL)

Figure 28.17 shows block diagram from the IEEE 802.1Qbv spec that illustrates how the gate control list governs the gate close(C) and open(O) events based on the schedule provided for each event.

GCL has the following two parts:

- Time Interval
  Defines the time in nano seconds for which the gate controls are valid and should be applied before reading the next gate controls from the list. Width of 24 to represent a maximum of 16ms schedule interval respectively. Supports left shift of the time interval up to 7 bits to be able to apply a multiplication factor from 1 to 128 ns (in steps of 2^n). The maximum value (post shifting) of this field must be 999,999,999 ns.

- Gate Control
  Defines the Open (O) represented by logic 1 or Close (C) represented by logic 0 state for the gate of each TC.

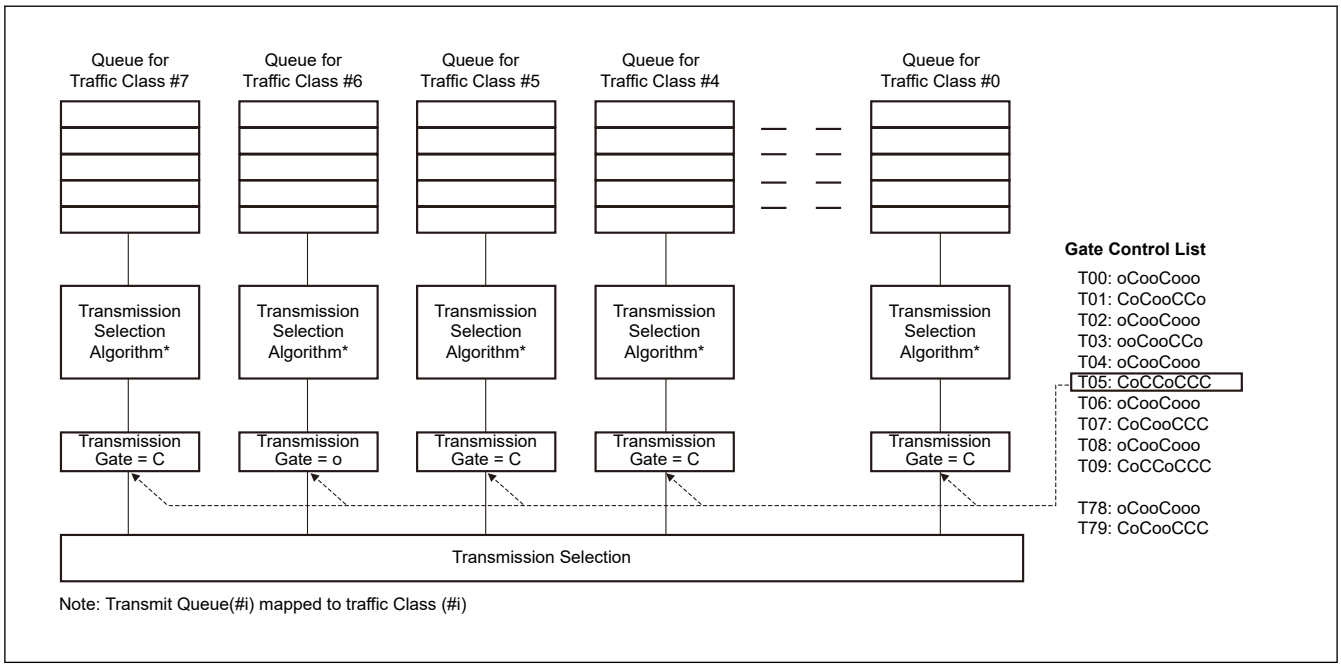


Figure 28.17 Block diagram from IEEE 802.1 Qbv Specification - GCL Governing Gate Close and Open Events

The implementation of GCL consists of the following two Gate Control Lists:

- HWOL - Hardware Owned List which is a list for hardware access.
- SWOL - Software Owned List which is a list for software access.

The access to these lists is mutually exclusive. The ownership to the list is set by hardware in the SWOL field of MTL_EST_Status register. Following table provides the implementation details of GCL.

Table 28.39 External memory used for holding the two gate-control lists

Gate Control (up to 8 bits)	Time Interval (ns) 16, 20, or 24 bits	
OOCCCCCC	T0	HWOL or SWOL
OOOOCCCC	T1	
CCCCOOCC	T2	
CCCCCCOO	T3	
I	I	
I	I	
OCOCOCOO		
OOOOCCCC	T last	
OOOOCCCC	T0	SWOL or HWOL
OOOOCCCC	T1	
OOCCCCCC	T2	
OOOOCCCC	T3	
I	I	
I	I	
OCOCCCCC		
OOCCCCCC	T last	

28.9.2.2 Registers Related to Gate Control List

Following are the set of 4 registers (one for each GCL) related to GCL. These registers are implemented through Indirect Addressing using the MTL_EST_GCL_Control and MTL_EST_GCL_Data registers.

1. 64-bit Base Time Register (BTR)
2. 40-bit Cycle Time (CTR)
3. m-bit Time Extension (TER)
4. 9-bit [Gate Control] List Length (LLR)

(1) Base Time Register (BTR)

Base Time Register is a 64-bit register that holds the start time to execute the GCL. The format of the BTR is same as the PTP format (upper 32-bits holds time in seconds and lower 32 bits hold time in nano seconds).

(2) Cycle Time Register (CTR)

Cycle Time Register is a 40-bit register that holds the time at which the execution of the GCL should be repeated. The Cycle Time Register consists of an 8-bit value in seconds, and a 32-bit value in nano seconds (similar to the PTP time format with truncated seconds register).

For a given Gate Control List the start time is "Base Time" + N × "Cycle Time" where N is an integer value representing the iteration number starting with 0 for first iteration. If the Gate Control List execution takes longer than the Cycle time, then the list is truncated at the Cycle Time and the subsequent loop begins at Cycle Time.

(3) Time Extension Register (TER)

Time Extension Register (TER) is a 31-bit register that holds the amount of time (in nano seconds) the current Gate Control List can be extended before switching to the new Gate Control List. This is useful to avoid small fragments of the current list before switching to a new list.

(4) List Length Register (LLR)

List Length Register (LLR) is an 9-bit register for a GCL depth of 512 that holds the integer value of the length of the GCL (that is the number of valid rows in each GCL). The processing of the GCL stops after the number of rows read equals to the LLR value.

28.9.3 Transmission Gating Implementation

A Bridge or an end station can be enhanced to allow transmission from each TC that is yet to be scheduled relative to a known timescale. To achieve this, a transmission gate is associated with each TC; the state of the transmission gate determines if the queued frames can be selected for transmission.

For a TC, the transmission gate can be in one of the following two states:

1. OPEN
   Queued frames are selected for transmission, in accordance with the definition of the transmission selection algorithm associated with the TC.
2. CLOSED
   Queued frames are not selected for transmission.

A frame on a traffic class queue is not available for transmission if the transmission gate is in the closed state or if there is insufficient time available to transmit the entirety of that frame before the next gate-close event associated with that queue.

The implementation has visibility to the current schedule of gate controls and the immediate next schedule of the gate controls. So the maximum Gate Open period does not exceed the sum of the two Time Intervals. This is because, a frame is selected for transmission only if the gate is currently Open and the duration of gate open interval is greater than the time taken to transfer the entire frame.

The implementation must know the frame size before the transmission, so that the transmission overruns can be avoided and only the frames that can complete are scheduled at all times. This can be achieved by

- Programming the OVHD field (bits [5:0]) of the MTL_EST_Ext_Control register with fixed overhead of the IPG or EIPG, preamble, and scheduler delay
- Automatic prediction of variable overhead due to the offloads enabled for a packet in the MAC transmitter. Consider the following while programming the EST related fields for correct scheduling of packets:
  - Speeds: 1Gpbs/100 Mbps
  - PTOV in ns: 9 PTP clock cycles
  - CTOV in ns: 3 PTP + 3 Transmit clock cycles
  - Per packet slot interval in ns:
  - (packet size + overhead + IPG or EIPG + preamble) × (time to transmit 1 byte)
  - overhead, scheduler delay in bytes is calculated as follows:
    (11 + X) Tx clock cycles
    where,
    If Tx COE is selected, X = 2. Otherwise, X = 0

The implementation adequately compensates for the implementation delays in the data transfer from the buffer to the line by offsetting the current time with all the relevant delays (provided by the CTOV field of MTL_EST_Control register). This ensures that the schedule provided is always accurately implemented at the line.

Note:
- An overhead is added to the packet size to account for the scheduler delay, IPG or EIPG, and Preamble overheads on the line. Therefore for a 128 byte frame to be transmitted, the Gate Open window should at least be able to accommodate 128 + overhead bytes.
- In 100 Mbps mode of operation, the rounding down error is about 0.4%. For example, the Gate open duration must be at least 1035 bytes for transmitting 1000 Byte Frames at a speed of 100 Mbps (31 bytes for scheduler delay, line overheads, and 4 Bytes for rounding down error).
- The minimum time interval for GCL entry should be equal to the time required to transmit fragment size + scheduler delay + IPG/EIPG + Preamble overheads.

28.9.4 Idle Slope Computation Updates

When EST is enabled, credit is accumulated only when the gate is open; therefore, the effective data rate of the idleSlope must be increased to reflect the duty cycle for the transmission gate associated with the queue.

The idleSlope is computed based on the GateOpenTime and OperCycleTime values. Program the idleSlope registers (implemented one per CBS enabled TC) based on the following equation. The existing MTL register has sufficient bit width to accommodate the new values for idleSlope.

idleSlope = (operIdleSlope(N) × OperCycle/GateOpenTime)

28.9.5 Operational Details of GCL

Set the switch to software List, the SSWL field of the MTL_EST_Control register, so that hardware can access the programmed gate control list. The first set of gate controls are applied when the current time is equal to the value in the Base Time Register (BTR) and is held until the programmed "Time Interval" value.

To avoid transmission overruns, one additional gate control event is always read ahead from the list. This enables the GCL to determine the next gate close events (if any) for the TCs that are open.

The scheduling based on the Gate Open state and Time Interval of only the current and subsequent schedule. An internal accumulator is used to add the time intervals when gate controls are applied. BTR + Accumulator holds the time at which the next set of gate controls are to be applied.

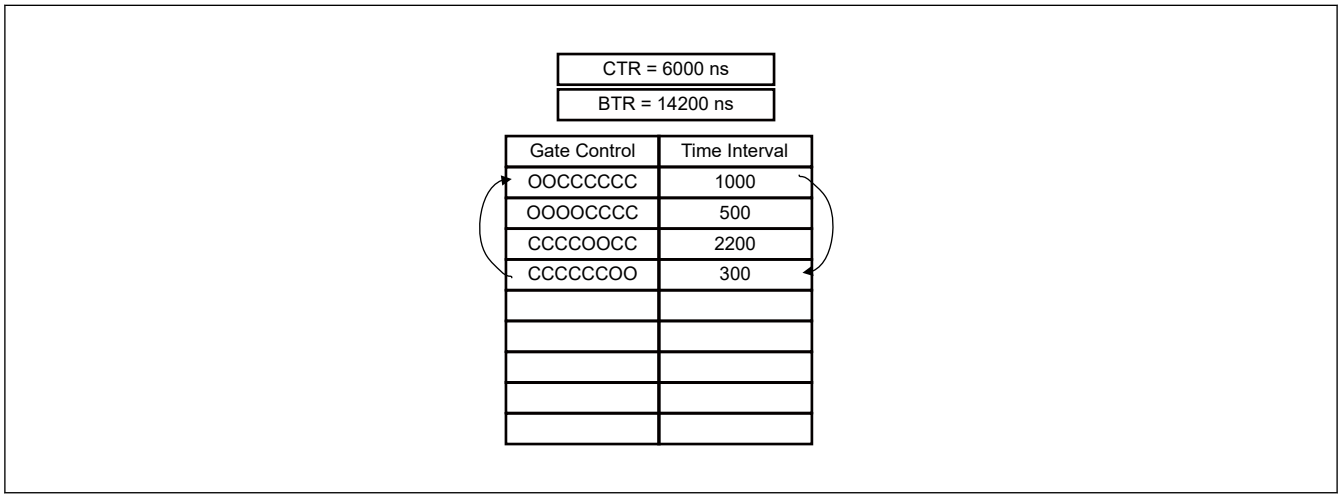


Figure 28.18 Block diagram from IEEE 802.1 Qbv Specification - GCL Governing Gate Close and Open Events

The GCL is read progressively from the first row adhering to the schedule. The read operations continue until the List Length (from LLR Register) is reached and the execution of the list restarts at BTR + CTR time. At this point the BTR is incremented by value in CTR to mark the beginning of a new cycle. In the absence of any gate controls, all the gates are deemed to be in Open state, during the execution of the list.

In cases where the execution time of the list is greater than the CycleTime, the list is truncated and the next iteration starts when the current time equals BTR + CTR.

Table 28.40 GCL and associated registers - BTR and CTR (1 of 2)

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
14200	OOCCCCCC	1000	14200
15200	OOOOCCCC	1500	14200
15700	CCCCOOCC	3700	14200
17900	CCCCCCOO	4000	14200
18200	OOOOOOOO	0	20200
20200	OOCCCCCC	1000	20200

Table 28.40 GCL and associated registers - BTR and CTR (2 of 2)

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
21200	OOOOCCCC	1500	20200

In the example in Table 28.40, the execution starts at 14200 and the first set of Gate Controls "OOCCCCCC" are applied immediately. As the time interval is 1000 ns the next set of gate controls are applied at 14200 (BTR) + 1000 (Accumulator) = 15200 ns as shown in Figure 28.19. As there are no gate controls available after the execution of the last gate control and before the next iteration of the loop, the gates are deemed to be in open state during that time period as depicted at time 18200 in Table 28.40.

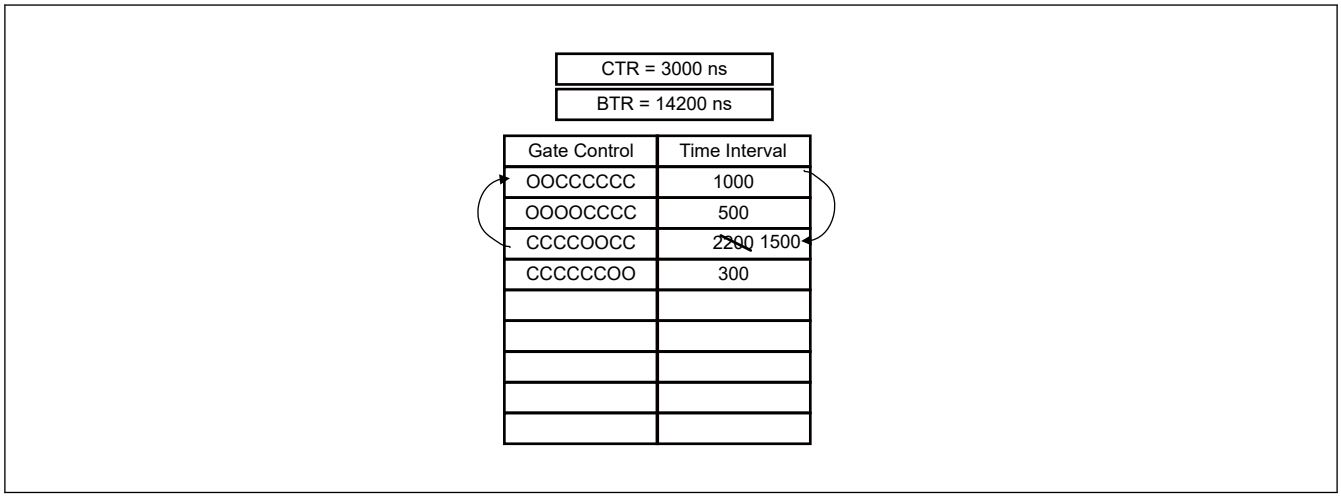


Figure 28.19 GCL and Associated Registers - BaseTime and CycleTime List Execution Time > CycleTime

As the list execution takes longer than the allocated CycleTime, the list is truncated and the list is started from the BTR + CTR as shown in Table 28.41.

Table 28.41 GCL and associated registers - BTR and CTR, execution time > cycle time

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
14200	OOCCCCCC	1000	14200
15200	OOOOCCCC	1500	14200
15700	CCCCOOCC	3700	14200
17200	OOCCCCCC	1000	17200
18200	OOOOCCCC	1500	17200
18700	CCCCOOCC	3700	17200

While applying the third set of gate controls, the BTR + CycleTime (17200) < BTR + Accumulator (17900), so the list is truncated and execution switches to a new iteration at 17200.

28.9.6 Installing a New GCL

When a new software programmed GCL is available and must be executed at the new BTR value, the switch to the new GCL can happen in one of the following ways.

- New Base Time aligned with current schedule
- New Base Time unaligned with current schedule

28.9.6.1 New Base Time Aligned with Current Schedule

If the choice of cycle time for the new gating cycle is unchanged from the cycle time for the current gating cycle, and if the BTR chosen for the new gating cycle (new BTR) is an integer multiple of the current cycle time (+ current BTR), then the new gating cycle exactly lines up with the old gating cycle, that is, the cycle start times for the new gating cycle is same as they would have been for the old configuration. This could be considered to be the ideal case and allows the new gating

cycle to be installed and executed with no timing issues. The implementation completes the execution of an iteration of the current list and switches to the new list at the beginning of the BaseTime listed in the new list.

If (New BaseTime ≥ Current Time) ConfigChangeTime = New BaseTimeElse If (New BaseTime < Current Time)

1. Set the BTRError
2. ConfigChangeTime = (New BaseTime + N × New CycleTime)

where N is the smallest integer for which the relation ConfigChangeTime ≥ CurrentTime and (N ≤ 8) is TRUE.

When N > 8 the hardware cannot auto recover and the loop count value in BTRError reporting is set to 1111b requiring the software to reprogram the New Base Time.

Figure 28.20 shows the installation of the new GCL along with the timelines.

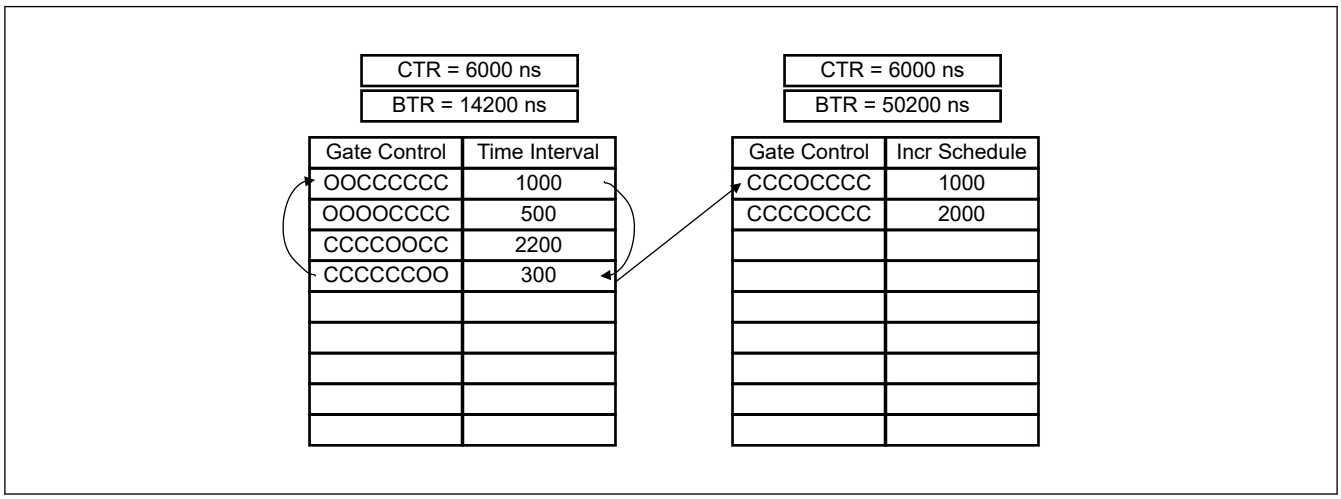


Figure 28.20 Switching to a new configuration that is aligned with the existing configuration

In the example, after the sixth iteration of the first GCL, the BTR values of the old and new GCL are equal. At that point the new GCL is processed as a natural extension to the existing GCL.

Table 28.42 GCL and associated registers - BTR and CTR

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
44200	OOCCCCCC	1000	44200
45200	OOOOCCCC	1500	44200
45700	CCCCOOCC	3700	44200
47900	CCCCCCOO	4000	44200
48200	OOOOOOOO	0	50200
50200	CCCOCCCC	1000	50200
51200	CCCCOCCC	3000	50200
53200	OOOOOOOO	0	56200

28.9.6.2 New Base Time Unaligned with Current Schedule

If new CycleTime differs from current CycleTime or New BaseTime is in the future and is not an integer multiple of current CycleTime, then the old and new cycles do not line up; when new BaseTime is reached (when the new configuration is installed and starts to execute), the last old cycle is normally truncated to start the first new cycle. This could be undesirable if it results in a very short last old cycle; arguably it would be better to simply extend the penultimate old cycle by that small amount, rather than starting a very short cycle. The Cycle Time Extension Register (related to the current configuration list) allows this extension of the last old cycle to be done in a defined way; if the last complete old cycle ends normally in less than current Cycle Time Extension (TER) ns before the new base time, then the last complete cycle before new BaseTime is reached is extended so that it ends at new BaseTime.

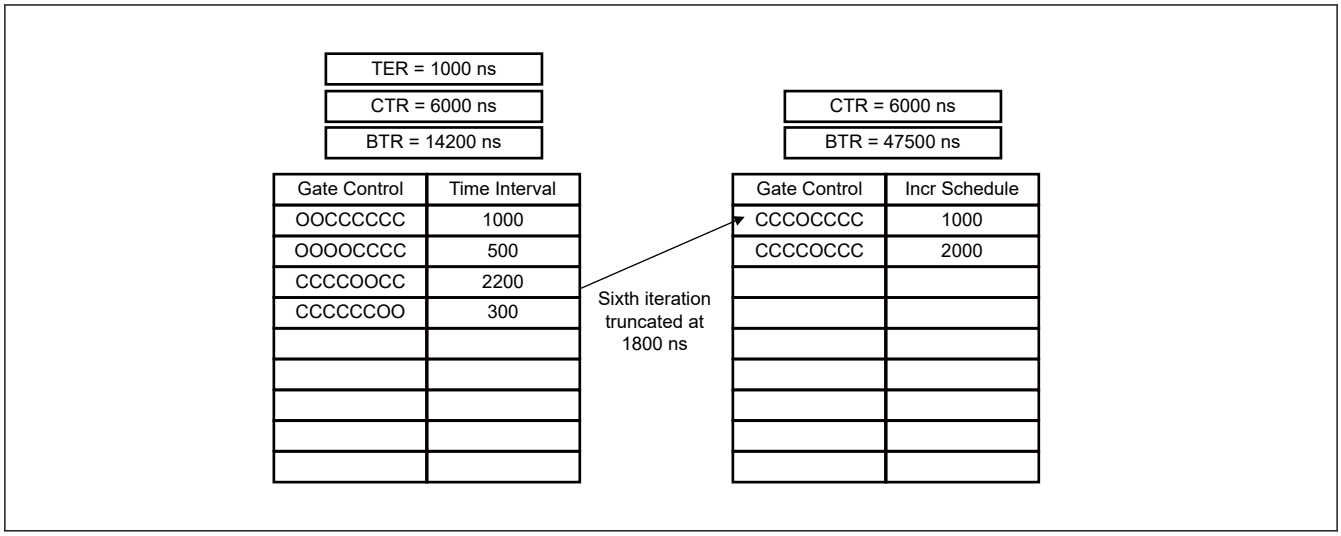


Figure 28.21 Switching to new list truncating the current list

At the end of the fifth iteration the Current time + Cycle time extension (TER) < New BTR so the sixth iteration of current configuration is started. During the sixth iteration of the current list when the new BTR value is smaller than the next schedule in the current list, it switches to the new list.

Table 28.43 Extending to new list by truncating the current list

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
44200	OOCCCCCC	1000	44200
45200	OOOOCCCC	1500	44200
45700	CCCCOOCC	3700	44200
47500	CCCOCCCC	4000	44200
48500	CCCCOCCC	0	50200
50500	OOOOOOOO	1000	50200

Following is an example where the current configuration list is extended instead of starting a new iteration as the extension time of 800 ns is less than the allowed cycle extension time (TER) of 1000 ns.

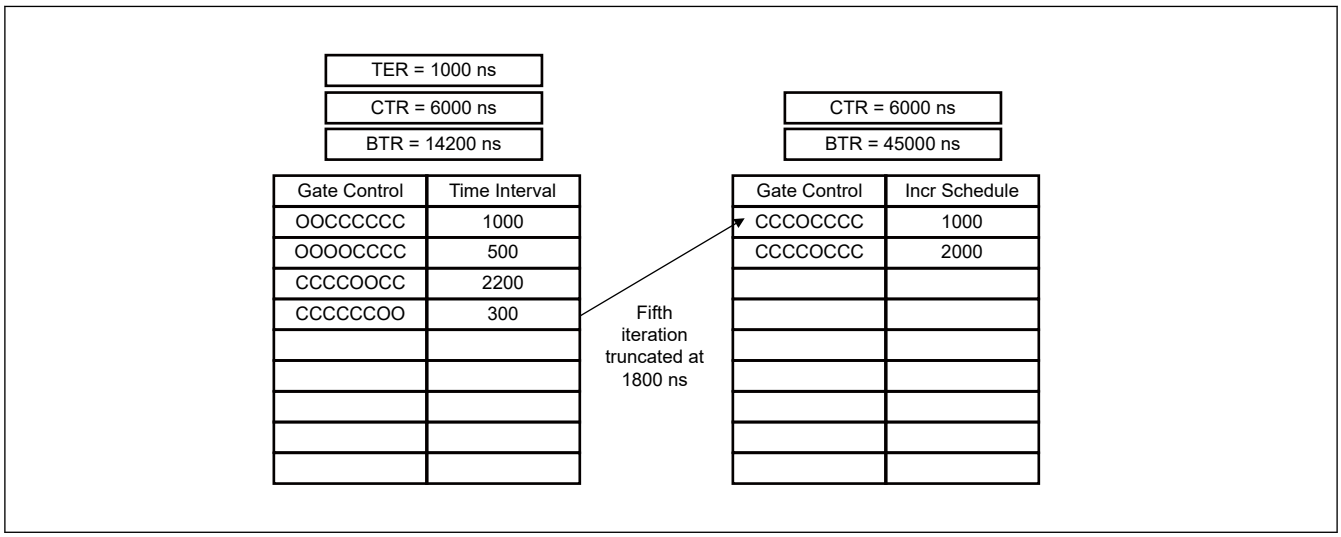


Figure 28.22 Switching to new list by extending the current list

Table 28.44 Switching to new list by extending the current list

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
38200	OOCCCCCC	1000	38200
39200	OOOOCCCC	1500	38200
39700	CCCCOOCC	3700	38200
41900	CCCCCCOO	4000	38200
42200	OOOOOOOO	0	44200
45000	CCCOCCCC	1000	45000
46000	CCCCOCCC	3000	45000
48000	OOOOOOOO	0	51000

28.9.7 Impact of Transmit Scheduling Algorithms on EST

When EST is used in isolation, the Gate Control List manages the final open/close state of the Queues along with the checks carried out by the Transmission Selection Algorithm in MTL. As the Gate Controls operate on a predefined repetitive schedule, it is recommended to use Strict Priority or Credit Based Shaper (CBS) scheduling algorithms.

The algorithm is applied only among the group of queues that open simultaneously. To ensure Queues whose gates are "Open" get priority, these algorithms are modified to treat "gate open" queues and "gate closed" queues as separate groups giving priority to gate open queues.

For example, consider 4 queues (Q3, Q2, Q1, Q0) with weights 4:3:2:1; Q3 & Q2 are in Open state in slot one slot, while Q1 & Q0 are in Open state in another slot. In this case, the scheduler works as follows:

1. In the first slot, the Q3 & Q2 are serviced in the ratio of 4:3 for the duration the slot is open.
2. In the second slot, the queues Q1 & Q0 are serviced in the ratio of 2:1.
3. Fresh arbitration is started every time a slot is opened.

In other words, the traffic does not get distributed in the intended ratio of 4:3:2:1; but as two groups with different ratios and only for the duration of the slot when the gates are open continuously.

28.10 Frame Preemption (FPE)

28.10.1 Overview of the Frame Preemption

The IEEE 802.1Qbv-2015 defines the schedule for each of the queues on every egress port which makes the implementation aware of traffic arrival schedule. This information can be used to block the lower priority traffic from transmission in this time window/slot. This ensures that scheduled traffic is forwarded from sender to receiver through all the network nodes with a deterministic delay.

One of the important requirements to achieve a low latency is to ensure there are no interfering frames during the scheduled windows that are reserved for high priority traffic. The use of scheduled traffic imposes limitations while starting a transmission.

If an interfering frame begins transmission just before the start of a reserved time period, it can extend transmission into the reserved window, and potentially interfere with higher priority traffic. Therefore, a guard band whose size is equal to the largest possible interfering frame is required before the window starts.

A larger guard band equates to a less efficient use of network bandwidth. However, with the implementation of IEEE802.1Qbu (frame preemption), this issue is addressed. Frame preemption breaks the interfering frame into smaller fragments. Therefore, the guard band needs to be only as large as the largest possible interfering fragment instead of the largest possible interfering frame.

During the guard band, only the frames that can complete the transmission of the entire frame before the next gate close event are permitted. This ensures that the high priority traffic can always start at the beginning of the window reserved for it.

Note: Internal loopback is not supported for preempted frames

28.10.2 Description of the Frame Preemption

Preemption allows one or more higher priority (express) frames to interrupt the transmission of a lower priority (preemptable) frame; the preemptable frame transmission is resumed and completed after the express frame transmission is complete. To support frame preemption, the following two abstractions of the MAC are used:

- A preemptable MAC, called pMAC, which carries the preemptable traffic.
- An express MAC, called eMAC, which carries the express traffic.

In the implementation, only parts of the MAC that holds the state during preemption is replicated and represented as pMAC and eMAC. The MAC uses the following two ways to puts on hold, the transmission of the preemptable traffic, in the presence of express traffic:

- The MTL scheduler interrupts the preemptable traffic that is currently being transmitted.
  When the preemption capability is active, the MAC interrupts the transmission and reception of preemptable frames. A preempted fragment can be followed by zero or more express frames, before the continuation fragments. The preemptable frame can be fragmented any number of times, however, the minimum final and non-final fragment length criterion must be is met.
  However, interleaving of more than 1 preemptable packet is not permitted. This implies that if a preemptable packet is fragmented by an express packet, another preemptable packet cannot be transferred until all the remaining fragments of the first preempted packet are transferred.
- The MTL scheduler prevents starting the transmission of preemptable traffic.
  When the preemption capability is inactive, the pMAC entity is disabled and only express traffic is transmitted or received.

Note: Queue0 can be an express queue when FPE is enabled and EST is disabled.

28.10.3 Enabling the Frame Preemption

Enable the frame preemption feature by setting EFPE field of the MAC_FPE_CTRL_STS register.

28.10.4 GCL Modification to Support FPE

In the EST-only configuration, the GCL entry has up to 24 bits of Time Interval and up to 8 high order bits representing the Gate Open/Close state requirements as shown in Table 28.45.

Table 28.45 Gate control list when FPE is disabled

Gate Control (up to 8 bits)	Time Interval(ns) (16, 20 , or 24 bits)
OOCCCCCC	T0
OOOOCCCC	T1
CCCCOOCC	T2
CCCCCCOO	T3
|	|
|	|
OCOCOCOO	T126
OOOOCCCC	T127

EST only supports SetGate operation, which implies that the gates are set to either open or close at a given time interval.

However, when both Frame Preemption (FPE) and EST are enabled, the GCL also supports Set-and-Hold-MAC and Set-and-Release-MAC operations, in addition to the SetGate operation.

To enable the support of hold and release operations, the format of the GCL is slightly changed while configuring and enabling the FPE. The first Queue (Q0) is always programmed to carry preemption traffic and therefore it is always Open. The bit corresponding to Q0 is renamed as Release/Hold MAC control.

The TX Queues whose packets are preemptable are indicated by setting the PEC field of the MTL_FPE_CTRL_STS register. The GCL bit of the corresponding Queue is ignored and considered as always "Open". So, even if the software writes a "0" ("C"), it is ignored for such queues.

Table 28.46 Gate control list when FPE is enabled

Gate Control (up to 7 bits)	Release/Hold Indication	Time Interval (ns) (16, 20 , or 24 bits)
CCCOOOO	0	T0
CCCCOOO	0	T1
OCCCOOO	1	T2
COCCOOO	1	T3
CCOCOOO	1	T4
CCCOOOO	0	T5
CCCCOOO	0	T6

When the Release/Hold bit transitions from a '0' to '1', a Set-and-Hold-MAC operation is performed. This marks the cease of the preemptable traffic. This is achieved by sending a Hold request to MTL "ha" ns in advance (where ha is the time interval mentioned in the Hold Advance (HADV) field of the MTL_FPE_Advance register).When the elease/Hold bit transitions from a '1' to '0' a Set-and-Release-MAC operation is performed. This marks the resuming of the preemptable traffic. This is achieved by sending a Release request to MTL "ra" ns in advance (where ra is the time interval mentioned in the Release Advance (RADV) field of the MTL_FPE_Advance register).The preemptable traffic is blocked for the time duration the Release/Hold bit is set to a '1' in the Gate Control List.

Note: HADV is the advance time to initiate the hold state. Ideally HADV should be programmed to a value of minimum fragment size, which should be smaller than the time interval (minimum fragment size + IPG + Preamble overheads in nanoseconds)

28.10.5 Impact of Preemption on CBS

In CBS, the definition of "Transmit" is as follows:

- TRUE for the duration of frame transmission from the queue;
- FALSE when frame transmission from the queue is complete.

When CBS algorithm is used along with frame preemption, the value of "Transmit" is TRUE only while the frame is being transmitted by the MAC. If the frame transmission is delayed or interrupted (for instance, a preemptable frame transmission is interrupted to allow the transmission of an express frame from a different queue, or the start of express frame is delayed because a preemptable frame is being transmitted) the value of "Transmit" is FALSE until transmission of the frame commences or is resumed.

Also, the value of "Transmit" is FALSE during the transmission of any overhead that is a consequence of frame preemption. For example, additional frame overhead (mCRC, Fragment Count) that is added to the preemptable frame.

At any given time, if there are no frames in the queue, and the value of Transmit is FALSE, and credit is positive value, the credit value is set to zero if here is no preemptable frame from the queue for which transmission is in progress but has been interrupted.

28.10.6 mPacket Format

When the preemption capability is active, MAC sends mPackets to the PHY. An mPacket can be one of the following:

1. A express packet
2. A preemptable packet
3. An initial fragment of a preemptable packet
4. A continuation fragment of a preemptable packet

Figure 28.23 shows the format of the mPacket. It contains an express packet, a complete preemptable packet or the initial fragment of a preemptable packet. Figure 28.23 (b) shows the format of an mPacket containing a continuation fragment of a packet.

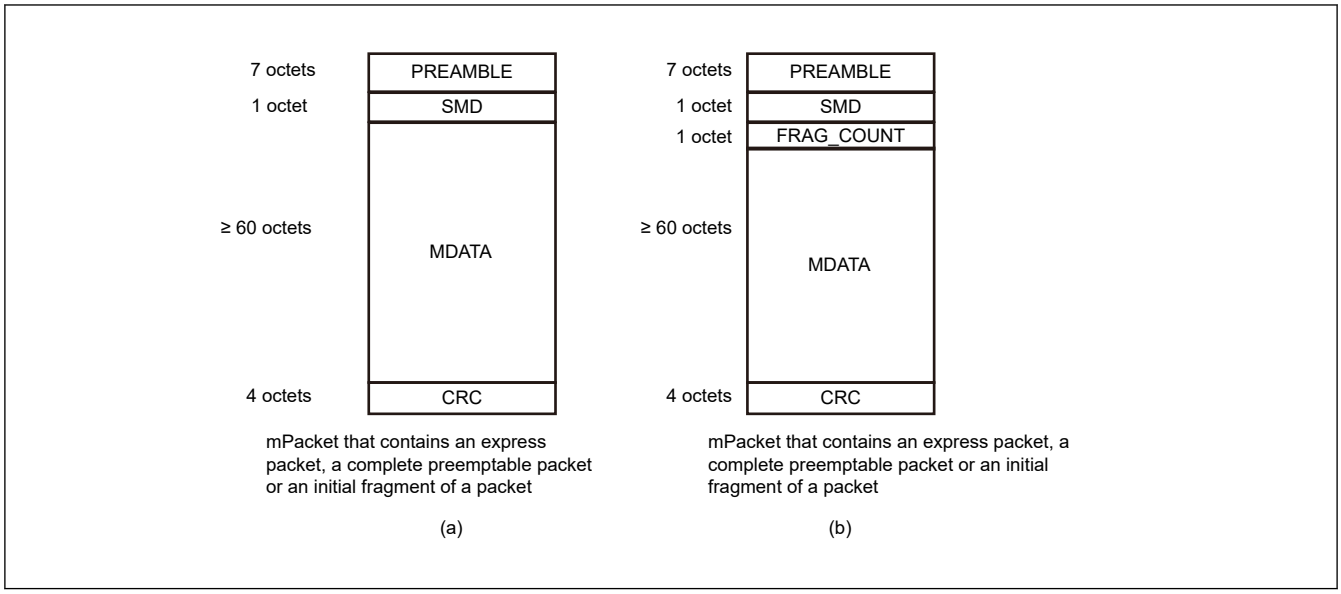


Figure 28.23 mPacket formats

Preamble

The preamble in the mPacket shown in Figure 28.23 (a) contains seven octets. The preamble in the mPacket shown in Figure 28.23 (b) contains six octets. Each octet contains the value of 0x55 (transmitted in order from left to right 10101010).

Start mPacket Delimiter (SMD)

The value of the SMD indicates whether the mPacket contains an express packet, the start of a preemptable packet (initial fragment or complete packet), or any of continuation fragments of a preemptable packet. Table 28.47 lists the valid SMD values.

Table 28.47 Possible SMD values of mPacket

mPacket Type	Notation	Frame Count	Value
verify packet	SMD-V	—	0x08
respond packet	SMD-R	—	0x19
express packet	SMD-E	—	0xD5
preemptable packet start	SMD-S0	0	0xE6
	SMD-S1	1	0x4C
	SMD-S2	2	0x7F
	SMD-S3	3	0xB3
continuation fragment	SMD-C0	0	0x61
	SMD-C1	1	0x52
	SMD-C2	2	0x9E
	SMD-C3	3	0x2A

frag_count

A frag_count is a modulo-4 counter that increments for each continuation fragment of the preemptable packet. The frag_count protects against mPacket reassembly errors by enabling detection of the loss of up to 3 packet fragments.

The frag_count field is present only in mPackets with SMD-C notation (continuation fragment). The frag_count is zero in the first continuation fragment of each preemptable packet.

Table 28.48 Possible frag_count values (1 of 2)

frag_count	Value
0	0xE6

Table 28.48 Possible frag_count values (2 of 2)

frag_count	Value
1	0x4C
2	0x7F
3	0xB3

mData

The contents of the packet from the MAC, starting with the first byte after the SFD to the last byte before the FCS are sent in the mData fields of one or more mPackets for that frame. The minimum size of the mData field is 60 bytes.

CRC

The CRC field contains a cyclic redundancy check (CRC) and has an indication of the final mPacket of a frame. In the final mPacket of a frame, the CRC field contains the last 4 octets of the MAC frame (the FCS field).

For other mPackets, the CRC field contains an mCRC value. The mCRC is calculated on the octets of the packet from the first octet of the frame (the octet following the SFD of preemption frames) to the last octet of the packet transmitted in that mPacket by performing an XOR of the calculated 32-bit CRC value of the fragment and a value of 0x0000FFFF.

Summary of Packet Formats

- Express Packet
  7 bytes of PREAMBLE, SMD-E, Data, and CRC
- Complete Preemptable Packet
  7 bytes of PREAMBLE, <Current Preemptable packet SMD>, Data, CRC
- Initial Fragment (non-final) of Preemptable Packet
  7 bytes of PREAMBLE, <Current Preemptable packet SMD>, Data, mCRC
- Continuation fragments (non-final) of Preemptable packet
  6 bytes of PREAMBLE, <Current Preemptable continuation fragment SMD>, <Current Preemptable continuation fragment FC>, Data, mCRC
- Final fragment of Preemptable packet
- 6 bytes of PREAMBLE, <Current Preemptable continuation fragment SMD>, <Current Preemptable continuation fragment FC>, Data, CRC

Note: When Frame Preemption is enabled, the MAC receiver communicates the exceptions related to received preempted fragments (incorrect fragment sequencing, missed fragment, and so on) to MTL layer by reporting it as CRC error. Software should not set the DCRCC (Disable CRC Checking for Received Packets) bit of MAC_Ext_Configuration register to 1; otherwise the MTL layer might forward the unintended preempted fragments to the application.

Table 28.49 Current and previous SMD values (1)

Previous Preemptable packet SMD	Current Preemptable packet SMD
SMD-S0	SMD-S1
SMD-S1	SMD-S2
SMD-S2	SMD-S3
SMD-S3	SMD-S0

Table 28.50 Current and previous SMD values (2) (1 of 2)

Previous Preemptable fragment SMD	Previous Preemptable fragment FC	Current Preemptable continuation fragment SMD	Current Preemptable continuation fragment FC
SMD-S0	NA	SMD-C0	FC0
SMD-S1	NA	SMD-C1	FC0
SMD-S2	NA	SMD-C2	FC0
SMD-S3	NA	SMD-C3	FC0
SMD-C0	FC0	SMD-C0	FC1

Table 28.50 Current and previous SMD values (2) (2 of 2)

Previous Preemptable fragment SMD	Previous Preemptable fragment FC	Current Preemptable continuation fragment SMD	Current Preemptable continuation fragment FC
SMD-C0	FC1	SMD-C0	FC2
SMD-C0	FC2	SMD-C0	FC3
SMD-C0	FC3	SMD-C0	FC0
SMD-C1	FC0	SMD-C1	FC1
SMD-C1	FC1	SMD-C1	FC2
SMD-C1	FC2	SMD-C1	FC3
SMD-C1	FC3	SMD-C1	FC0
SMD-C2	FC0	SMD-C2	FC1
SMD-C2	FC1	SMD-C2	FC2
SMD-C2	FC2	SMD-C2	FC3
SMD-C2	FC3	SMD-C2	FC0
SMD-C3	FC0	SMD-C3	FC1
SMD-C3	FC1	SMD-C3	FC2
SMD-C3	FC2	SMD-C3	FC3
SMD-C3	FC3	SMD-C3	FC0

28.10.7 Transmit Preemption

28.10.7.1 MTL Tx Preemption

To support preemption MAC should have more than 1 TX queue with at least 1 Queue designated as Express Queue. (and 1 queue designated as preemption queue). When FPE is enabled (setting EFPE = 1 in MAC_FPE_CTRL_STS register), the MTL preempts a preemptable frame, when a "hold" request is asserted (EST/Qbv configured and enabled) express frames are available for transmission, that is, frame is present in MTL FIFO and is qualified for arbitration, after ensuring that the minimum mPacket mData field size is met.

Therefore, preemption occurs only if at least 60 bytes of the preemptable frame have been transmitted and at least 64 bytes (including the frame CRC) remain to be transmitted.

The earliest starting position of preemption is controlled by the AFSZ field of the MTL_ FPE_CTRL_STS register. Preemption does not occur until at least 64 × (1 + AFSZ) - 4 bytes of the preemptable frame have been sent.

When preemption occurs, all the preemptable queues are blocked and only the express queues are allowed to arbitrate (if more than one express queue has traffic) and transmit.

Continuation fragment of the preempted frame is the first frame to be transmitted after "release" request is asserted (EST/Qbv configured and enabled) and all the express traffic transmission completes.

Note: All the PTP packets should be transmitted as express packets.

MTL communicates the following frame-type information to the MAC using a 2-bit preemption control signal on the MTI interface qualified by SoF and EoF.

- Express Frame
- Preemption Frame (Full or Fragment)
- Continuation Fragment (non-Final or Final)

Table 28.51 Preemption control values on MTI interface for various frame types (1 of 2)

Qualifier	Preemption Control Value	Frame Type
SoF	00b	Start Express
SoF	01b	Start Preemption

Table 28.51 Preemption control values on MTI interface for various frame types (2 of 2)

Qualifier	Preemption Control Value	Frame Type
SoF	10b	Continuation Fragment
SoF	11b	Reserved
EoF	00b	End of Frame
EoF	01b	End of Fragment

If the SOF and EOF are sent in the same cycle on the MTI interface, the mti_preemp_ctrl (Preemption Control Value) for EoF is taken as 00b, End of Frame.

MTL should wait for the previous fragment status to be received before resuming the continuation fragments of a preempted frame. Fragment status is described in detail in the Tx Fragment Status section

28.10.7.2 MAC Tx Preemption

MAC supports preemption by implementing the functionality needed to generate the mPackets as described in section 28.10.6. mPacket Format.

Based on the Preemption Control value received on the MTI interface (qualified with SoF and EoF), MAC determines the frame type (shown in Table 28.51) and generates mPackets accordingly.

When the preemption capability is active, MAC replaces the SFD of a preemption packet with an SMD-S value. A 2-bit rolling frame count is encoded in the SMD-S value.

The SMD-E value is the same as the SFD value so the SFD of an express packet does not need to be replaced.

28.10.7.3 Tx Fragment Status

MAC sends Tx fragment status to indicate successful transmission of fragmented mPackets.

In case of a transmission error (underflow, jabber, and so on) the frame status is sent (and not a fragment status) with an error indication along with all other relevant status fields. In case of receiving an error status for a transmitted fragment, the MTL drops the remaining fragments and does not send any more continuation fragments.

28.10.8 Receive Preemption

28.10.8.1 MAC Receive Preemption

When FPE is enabled, the MAC Receiver passes the incoming packets and differentiates between Express packets and preemptable packets. An SMD containing an SMD-E indicates express packet, and SMD containing an SMD-S indicates the first mPacket of a preemptable packet.

If an mPacket containing an SMD-S is received when MAC has not completed receiving the previous preempted packet, MAC sets a CRC Error status for the previously received partial packet.

When an SMD-S is detected, MAC records the frame count indicated by the SMD and then begins sending data on the MRI interface.

The MAC checks the last four bytes of the mPacket . If the last four bytes of the mPacket do not match CRC, that indicates the end of the packet with or without a CRC error as per the CRC check result. If the last four octets of the mPacket match, that indicates that the packet was preempted.

An SMD containing an SMD-C indicates an mPacket that continues the data for a preempted packet. Upon receiving an SMD value of SMD-C, MAC checks the following:

1. A preempted packet is in progress
2. The frame count indicated by the SMD matches the frame count of the packet in progress
3. The frag_count value indicates the next fragment count.

If any of these checks fail, the mPacket is discarded and MAC sets a CRC Error status for the partially received packet.

If all the checks pass, the next fragment count is incremented modulo 4.

When a packet is preempted, the MAC saves the state of the partially received packet (filter check status, timestamp, length fields and so on) and can process any received Express packets before the continuation fragment is received.

The MAC Receiver sends a "dummy status" for all the mPacket fragments successfully received and sends the Rx status with the final fragment. If an error is detected during any of the fragments the Rx status is sent and the fragment is marked as final fragment. All subsequent continuation fragments received for this packet are dropped in the MAC.

The MAC communicates the following frame type information to the MTL using a 2-bit preemption control signal on the MRI interface qualified by SOF and EOF.

1. Express Frame
2. Preemption Frame (Full or Fragment)
3. Continuation Fragment (non-Final or Final)

Table 28.52 Preemption control values on MRI interface for various frame types

Qualifier	Preemption Control Value	Frame Type
SoF	00b	Start Express
SoF	01b	Start Preemption
SoF	10b	Continuation Fragment
SoF	11b	Reserved
EoF	00b	End of Frame
EoF	01b	End of Fragment

If the SOF and EOF are sent in the same cycle on the MRI interface, Preemption Control Value for EoF is taken as '0' indicating End of Frame.

28.10.8.2 Data Alignment

When a received frame cannot be fragmented on any byte boundary, MAC retains the unaligned bytes of data in the previous fragment and resends them with the next fragment as shown in Figure 28.24.

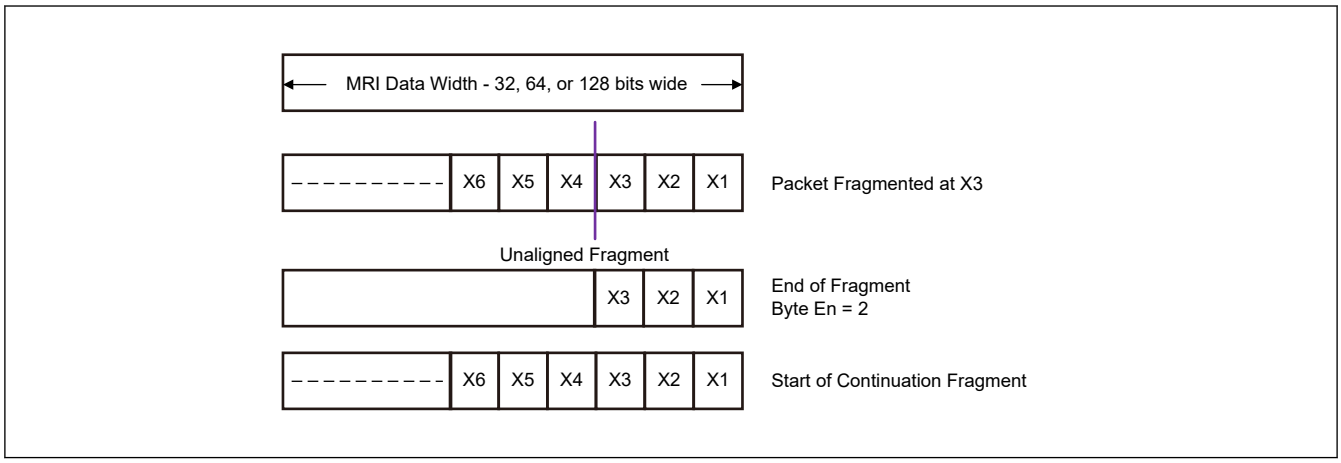


Figure 28.24 Data alignment feature of MAC

MTL can choose to ignore the partial data (based on Byte Enable value) that comes with End of Fragment and instead use the "data width aligned" data received in the first beat of the next continuation fragment.

28.10.8.3 MTL Receive Preemption

The MTL Rx must have at least 2 Rx queues to support FPE function as the preemptable packets and the express packets must be routed to separate Rx queues. The destination Rx queue of a received packet is controlled by MAC_RxQ_Ctrl[n] registers. Program these registers such that Preemptable traffic and express traffic are not routed to the same Rx queue. As the queue mapping for tagged packets is based on VLAN user-priority field, this implies that priority of preemptable and express packets are mutually exclusive. In other words, packets of a certain priority (traffic class) are either express or preemptable but cannot be both.

For the preemptable frames, in addition to the PSRQ (Priority Selected in Rx queue) based queue routing, a programmable "Frame Preemption Residue Queue" (FPRQ) is supported to route all other Preemption Packets received (Untagged, SA/DA or VLAN Filter Fails forwarded due to RA being set or VTFE being reset).

Table 28.53 Queue routing for various preemption packet types

Preemption Packet Type	Queue Routing
AV Tagged Packets passing the SA/DA and VLAN filters	PSRQ
DCB/Generic Tagged Packets passing the SA/DA and VLAN filters	PSRQ
Tagged Packets failing the SA/DA and VLAN filters without setting the Receive All (RA = 0) or setting VTFE = 1	Dropped
All other packets OR when RA = 1	FPRQ

The express frames use Queue 0 as a default queue. So, in case of filter failures, Queue0 must not be used for receiving preemption frames.

When a fragment is received, it is pushed into its destination Rx queue. When this packet is preempted, and is followed by an express packet, it is pushed/stored in a different Rx queue. The MTL saves the context of the preempted frame, and processes one or more express frames. When the continuation fragment arrives, it restores the saved context and pushes the remainder of the fragmented frame into its Rx queue.

28.10.8.4 MTL Receive Arbitration

On the ARI Interface, data is fetched from the MTL Rx FIFO based on the arbitration selected in the MTL_Operation_Mode register. Frame based arbitration can be used only when all the MTL Preemption Queues operate in Store and Forward mode. Otherwise, there is loss of bandwidth on the ARI interface as all the fragments of a preemptable packet are not available. Therefore, express packets received between the fragments are blocked until all the fragments are received and transferred; this defeats the purpose of express traffic.

When operating in either Threshold (cut through) or Store and forward modes of operation, PBL based arbitration is recommended over Frame based arbitration. In case of PBL based arbitration, the watermark check is always performed and the arbitration/transfer of data in terms of chunks of "PBL" size of data which is almost similar to the concepts of "fragments". Therefore the express queue packets get blocked for less time as well as the ARI interface transfers the data without loss of efficiency.

28.10.9 Received Preemption Frames Support in Offload Engines

When Frame Preemption is enabled, all the packets processed by the Offload Engines (ARP, PTO) and PMT, PTP, and PAUSE packets must be received as express/normal packets. These offload engine logic and functions do not recognize preemption frames.

28.10.10 Verify and Respond mPackets

When FPE function is present, the MAC can receive and detect the Verify and Respond mPackets, even when FPE is not enabled by software. When MAC detects valid Verify/Respond mPackets, it notifies the software by setting the RVER and RRSP fields of MAC_FPE_CTRL_STS register respectively. Optionally an interrupt can be generated. As such packets have empty (all-zero) data payload, they are dropped inside the MAC and not forwarded to the MRI.

Software can set the SVER and SRSP fields of MAC_FPE_CTRL_STS register to request MAC to transmit Verify and Respond mPackets respectively. Upon successful transmission of these frames, MAC clears the SVER/SRSP bits and sets the TVER & TRSP fields of MAC_FPE_CTRL_STS register. Optionally an interrupt can be generated when these events occur.

Note: 1. For the preemption frames received with CRC error indication, ignore the following status fields:
- Packet length
- Dribble error indication
- Runt frame indication

2. When FPE is configured, Forward Undersized Good Packets (FUP) bit of MTL_RxQ_Operation_Mode register should not be set.

3. When FPE is configured, do not disable CRC checking on receive. This is required because, internally, FPE error fragments are identified using CRC checks. For details, see the DCRCC field in the MAC_Ext_Configuration register.
4. Express and preemption traffic must be routed to different DMAs.

28.10.11 Frame Preemption and MMC Counter and Interrupt Registers

The following MMC counters and associated Interrupt registers are instantiated/present in the MAC.

Table 28.54 MMC counters and associated interrupt registers

Frame Assembly Error Counter	Description	Associated MMC Counter
Frame Assembly Error Counter	A 32-bit counter that provides the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value.	MMC_Rx_Packet_Assembly_Err_Cntr
Frame SMD Error Counter	A 32-bit counter that provides the number of received MAC frames rejected due to arriving with an SMD-C when there was no preceding preempted frame	MMC_Rx_Packet_SMD_Err_Cntr
Frame Assembly OK Counter	A 32-bit counter that provides the number of MAC frames that were successfully reassembled	MMC_Rx_Packet_Assembly_Ok_Cntr
MAC Rx Fragment Counter	A 32-bit counter that provides the number of additional mPackets received due to preemption	MMC_Rx_FPE_Fragment_Cntr
MAC Tx Fragment Counter	A 32-bit counter that provides the number of additional mPackets transmitted due to preemption	MMC_Tx_FPE_Fragment_Cntr
Hold Request Counter	A 32-bit counter that maintains the count of number of times a hold request is given to MAC	MMC_Tx_Hold_Req_Cntr

28.10.11.1 Additional Registers Associated With MMC Interrupts

Following are the additional registers associated with MMC interrupts for the MMC error counters:

- MMC_FPE_Tx_Interrupt
- MMC_FPE_Tx_Interrupt_Mask
- MMC_Rx_Interrupt
- MMC_FPE_Rx_Interrupt_Mask

28.11 Time-Based Scheduling

Time-based scheduling feature is suitable for traffic whose periodicity and rate are predictable. This is an optional feature. To improve the quality-of-service of such traffic,

- The transmit DMA fetches the packet from the host memory for transmission at designated time. This helps the software to setup the Transmit descriptors in advance even before packet is ready/available. It reduces the overhead on the software and avoids constant monitoring of the time and preparing descriptors just in time when the packet is targeted to be transmitted.
- The MAC transmits the packet only at the designated/pre-determined time even if the packets are fetched in advance. This helps in maintaining a constant transmission rate that can be consumed by the receiver station; therefore avoiding congestion and excessive buffering in the network.

28.11.1 Description of Time-Based Scheduling

The time-based scheduling feature supports fetching and launching an Ethernet packet at (or after) a pre-determined time. The time-based scheduling is supported only in the following modes:

- Full duplex mode
- Link speed is 100Mbps or higher

Note: • In EST enabled configurations the ESTM mode bit setting impacts the time-based scheduling feature only when the DUT processes the GCL list.

- Slot number is meaningful only after GCL is active. So, start the GCL list before enabling the Time-Based Scheduling (TBS) and Enhancements to Scheduled Traffic (EST) features. If GCL is not started, all gates are deemed open and if packets do not make progress, TxQ creates back pressure.

28.11.2 Definitions Used

Following are the definitions specific to the time-based scheduling feature:

- Launch Time
  The time beyond which MTL can schedule the packet for transmission.
- Fetch Time
  The time beyond which the Tx DMA can schedule a packet-fetch from the host memory.
- Expiry Time
  The time beyond which the packet is dropped by MTL.

28.11.2.1 Launch Time

The launch time is specified in the Enhanced Normal Descriptor. For details, see section 28.17.6. Enhanced Descriptor for Time-Based Scheduling section 28.17.6. Enhanced Descriptor for Time-Based Scheduling.

When the launch time is specified, it is valid only for the specific frame that is scheduled for transmission. The launch time is reset after the transmission of one frame.

Following are the two formats of the launch time.

- Normal/Absolute Format
- EST/Offset Format

Normal/Absolute Format

In this format, the Launch time is an absolute time value at which the packet is launched for transmission.

The launch time is interpreted in the Normal/Absolute format if the ESTM bit of the MTL_TBS_CTRL register is set to 0.

The launch time is a 32-bit value, where most-significant 8-bits represent the time in seconds and the rest 24 bits represent the time in 256 ns. The launch time is compared against the IEEE 1588 based System/PTP Time (bits [39:8]) and rolls over after 256 seconds.

The maximum value of the lower 32-bits of System Time is 999,999,999 decimal (0x3B9AC9FF) and it wraps to 0 when reaching this value (representing a full second). Therefore, the maximum value of the lower 24 bits of the Launch Time (after multiplying by 256) must be 0x3B9AC9.

As the maximum value of Launch time is 256 seconds,

- Launch Time is greater than current System Time when its value is between System Time[39:8] and System Time[39:8] + 128 sec
- Launch Time is less than current System Time when its value is between "System Time[39:8] + 128 sec" and "System Time[39:8] + 256 sec", because this is a modulo 256 computation.

EST/Offset Format

In this format, the Launch time is a offset value relative to the time indicated by the Base Time Register (BTR) of the GCL list provided in the GCL Slot Number (GSN). The value in the BTR is always updated to the start time of the current loop of the GCL.

For each packet, the GSN value and the Launch Time value are specified in the Enhanced Transmit Descriptor.

The launch time offset is a 32-bit value; the upper 8-bit represent the time in seconds and the rest 24 bits represent the time in 256 ns, which is added to the BTR value corresponding to GCL Slot Number. The value of the Launch Time Offset must be smaller than the value of the Cycle Time (specified in the CTR register that is implemented when EST is enabled).

If the CTR is greater than or equal to 1s, the maximum value of the lower-24 bit of Launch Time offset should be 999,999,999 decimal (0x3B9AC9FF).

In this format, the launch time is a 64-bit value, which is interpreted as,

Launch Time = Launch GSN BTR[63:0] + (Launch Time Offset[31:0] << 8), which is compared with the System Time [63:0].

GSN BTR is the Base Time value at which the Launch GSN loop started execution.

GCL Slot Number (GSN)

A modulo 16 counts of the GCL loop count is implemented and known as GSN. The count is incremented for every new GCL loop; installation of new GCL list does not impact the count. The current GCL slot number can be obtained by reading the CGSN field of the MTL_EST_Status register.

The maximum value of GSN is 15. Therefore, GSN values between Current GSN (CGSN field of MTL_EST_Status register) and CGSN + 8 represent current or future slots; all other GSN values are interpreted as elapsed slots or past slots. So, for the correct interpretation of time, GSN value must be between CGSN and CGSN+8, for correct interpretation of time.

28.11.2.2 Launch Expiry Time

Normal/Absolute Mode

In the normal mode, when the LEOV (Launch Expiry Offset Valid bit of MTL_TBS_CTRL register) is set, the Launch Expiry Offset (LEOS field of MTL_TBS_CTRL register) determines the maximum amount of time a frame is eligible for launch, starting from the time the frame becomes eligible for launch.

The Launch Expiry Offset is a 24-bit value defined in 256 ns units, with a maximum possible value of 999,999,999 ns (0x3B9AC9FF).

Launch Expiry Time = (Launch Time[39:8] + LEOS[32:8]) × 256 ns

The packet with a specific Launch time is considered eligible for transmission when the Launch time is less than the System Time and (if LEOV is set) the System Time is less than the Launch Expiry time.

When the System time is greater than the Launch Expiry time the frame is categorized as expired and is dropped from the MTL FIFO.

Note: 1. For correct interpretation and meaningful operation, the Fetch, Launch, and Launch Expiry time should never be set to a value larger than Current System Time + 128 sec; such a value is interpreted as time that has already elapsed.

2. In full duplex mode, the frames dropped from the MTL FIFO have Error Summary (bit 15) and Excessive Deferral (bit 3) of TxStatus set.

EST/Offset Mode

In the EST mode of operation, when the LEOV field of the MTL_TBS_CTRL register is set, the launch expiry GSN offset (LEGOS field of the MTL_TBS_CTRL register) and Launch Expiry Offset (LEOS field of MTL_TBS_CTRL register) determine the maximum amount of time a frame remains eligible for launch, starting from the time the frame becomes eligible for launch.

Launch Expiry Offset = (LEGOS: LEOS)

LEGOS holds the GSN offset (multiples of CTR time) and LEOS holds maximum value of CTR (sub CTR values) in ns.

Launch Expiry offset is a 24-bit value defined in the units of 256 ns, with a maximum possible value of the smaller of 999,999,999ns or CTR-1 ns.

The Launch Expiry GSN is computed as follows:

Launch Expiry GSN = (Launch GSN + LEGOS + CCMA)

Where,

CCMA is the CTR Carry due to Modulo addition. This value is 1 if ((Launch Time Offset + LEOS) << 8) is equal to or greater than CTR. When CCMA = 1,

Launch Expiry Offset = (Launch Time Offset+ LEOS) - CTR

When CCMA = 0,

Launch Expiry Offset = (Launch Time Offset + LEOS).

Launch Expiry Time = Launch Expiry GSN BTR[63:0] + Launch Expiry Time Offset.

When LEOV is not set, Launch Expiry Time is not checked.

When LEOV is set, and

- the system time is greater than the Launch Expiry time, the frame is dropped from MTL FIFO. The frame is considered as expired.
- the launch time is smaller than the System Time and Launch Expiry Time is greater than system time, the frame is considered eligible for launch.

Note: 1. Max value of LEGOS is 7. This implies that when LEOV is set, the frame has a maximum life time of < 8 GCL loop iterations after it becomes eligible for Launch.
2. The slot number of the First GCL List executed each time after EST is enabled, is zero.

28.11.2.3 Fetch Time

The Fetch Time can also be indicated for each packet by the software. This is done by setting FTOV field of DMA_TBS_CTRL register.

If the FTOV filed is not set, the Fetch Time Offset is not valid and the DMA fetches packets without any time constraints.

The fetch time accounts for all possible delays in the DMA fetch operation and ensures that the frame is present in the MTL FIFO before the launch time.

Normal/Absolute Mode

In the Normal mode fetch time derived/calculated by reducing the time specified in the Fetch Time Offset (FTOS) field of DMA_TBS_CTRL register from the given Launch time.

Normal mode of operation, the Fetch Launch Time is computed as:

Fetch Time[39:8] = (Launch Time[39:8] - FTOS[31:8])

The Fetch time is 32-bits and is compared against System Time[39:8] to determine the Eligible for fetching the frame:

- The Fetch time is defined as "greater" than System Time if the Fetch Time is in the range of "System Time[39:8]" and "System Time[39:8] + 128 sec". The frame is considered as not-eligible for fetch.
- The Fetch time is defined as "smaller" than the System Time if the Fetch time is in the range of "System Time[39:8] + 128 sec" and "System Time[39:8] + 256 sec". The frame is considered as eligible for fetch.

This is a modulo 256 computation.

EST/Offset Mode

In the EST/Offset mode, the Fetch GSN Offset (FGOS field of MTL_TBS_CTRL register) provides the Slot Number offset to be deducted from the Launch GSN. In this case the FTOS value should be the smaller of 999,999,999ns or CTR-1 ns.

If (Launch Time Offset ≥ FTOS):

Fetch Time Offset = ((Launch Time Offset - FTOS) × 256 ns)

CBFS(CTR Borrow for Fetch Subtraction) = 0

If (Launch Time Offset < FTOS)

Fetch Time Offset = CTR + ((Launch Time Offset - FTOS) × 256 ns)

CBFS (CTR Borrow for Fetch Subtraction) = 1

The Fetch GSN is computed as follows:

Fetch GSN = Launch GSN - FGOS - CBFS

Fetch Time = Fetch GSN BTR[63:0] + Fetch Time Offset

The frame is marked eligible for DMA fetch when the fetch time is smaller than the System Time.

Note: Max value of FGOS is 7. This implies that when FTOV is set, the frame can be fetched at a maximum of <8 GCL loop iterations before it becomes eligible for Launch.

DMA Operations Sequence

Following is the sequence of operation when FTOV = 1.

1. Fetch the first Enhanced Normal Descriptor (FD is set).
2. If LTV is set and Fetch enabled, compute the Fetch Time based on the Launch Time. Wait for the System Time to be greater than Fetch Time.
3. Read the Frame (Data) from the host memory and transfer to MTL FIFO.
4. Close the Normal Descriptor.
5. Fetch the next Normal Descriptor (if the previous Descriptor was not the Last).
6. Repeat steps 4 to 6, until the last descriptor of the frame (LD is set).

After the Last Descriptor of a frame, the next enhanced normal descriptor should be programmed with a new Launch Time and with LTV bit set. Otherwise, the subsequent frames are processed without any time restrictions.

Control Word (MTL)

Time-based scheduling ensures that a packet is allowed to arbitrate for transmission only if the System Time is greater than the Launch Time.

In MTL,

- Launch Time and GSN, use the same field that is used to hold the 64-bit timestamp value for OSTC (One Step Timestamping Correction). If the LTV bit is set and OSTC bit is not set, the Time Stamp field is interpreted as Launch Time. Launch Time uses the fields meant for OSTC TimeStamp[39:8] and GSN uses OSTC TimeStamp[43:40] field.

When the MTL TXFIFO Read Controller reads the first control word, it checks bit-31 to determine if the frame has a valid Launch Time. If bit-31 is set, it reads the subsequent control words to get the Launch Time.

Impact on Transmission Selection Algorithm

Time-Based Scheduler does not directly influence the Transmission Selection Algorithm but has an indirect influence as it determines if a queue is eligible to participate in the scheduling.

If a frame of a specific queue has a valid Launch Time, the Time Based Scheduler ensures that the frame participates in Transmit Scheduling (TRC Scheduler) only after the Launch time has elapsed.

The gating done by Time-Base scheduler is only for the Scheduling (picking frame for Transmission). It might not alter any other characteristics and behavior of the frame. For example, the Frame waiting for the Launch time to lapse, continues to gain credits in CBS, and the CBS credits are not lost as the frame is marked as available but not ready. Strict Priority or CBS are recommended Transmit Scheduling algorithms for the TBS implementation because, scheduling for TBS enabled queues are more predictable.

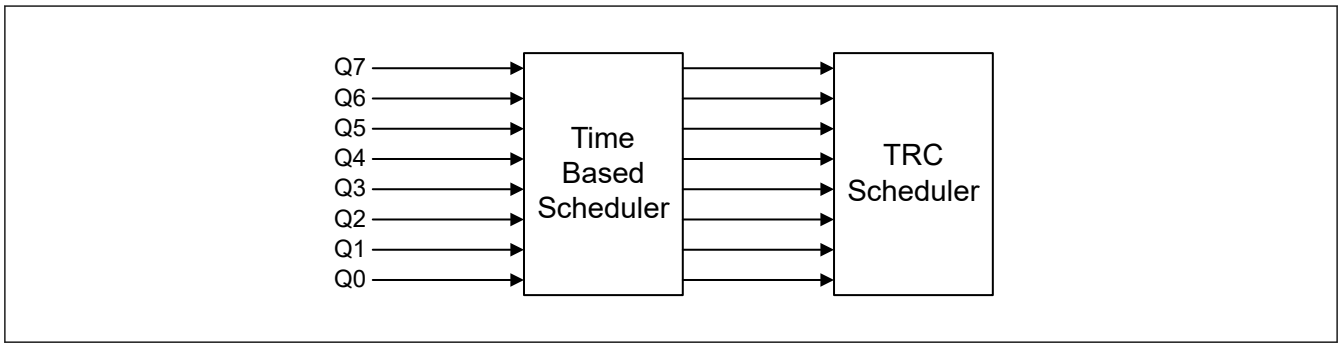


Figure 28.25 Time Based Scheduler Implementation

28.11.3 Time-Based Scheduling Flow

Figure 28.26 shows the time-based scheduling flow.

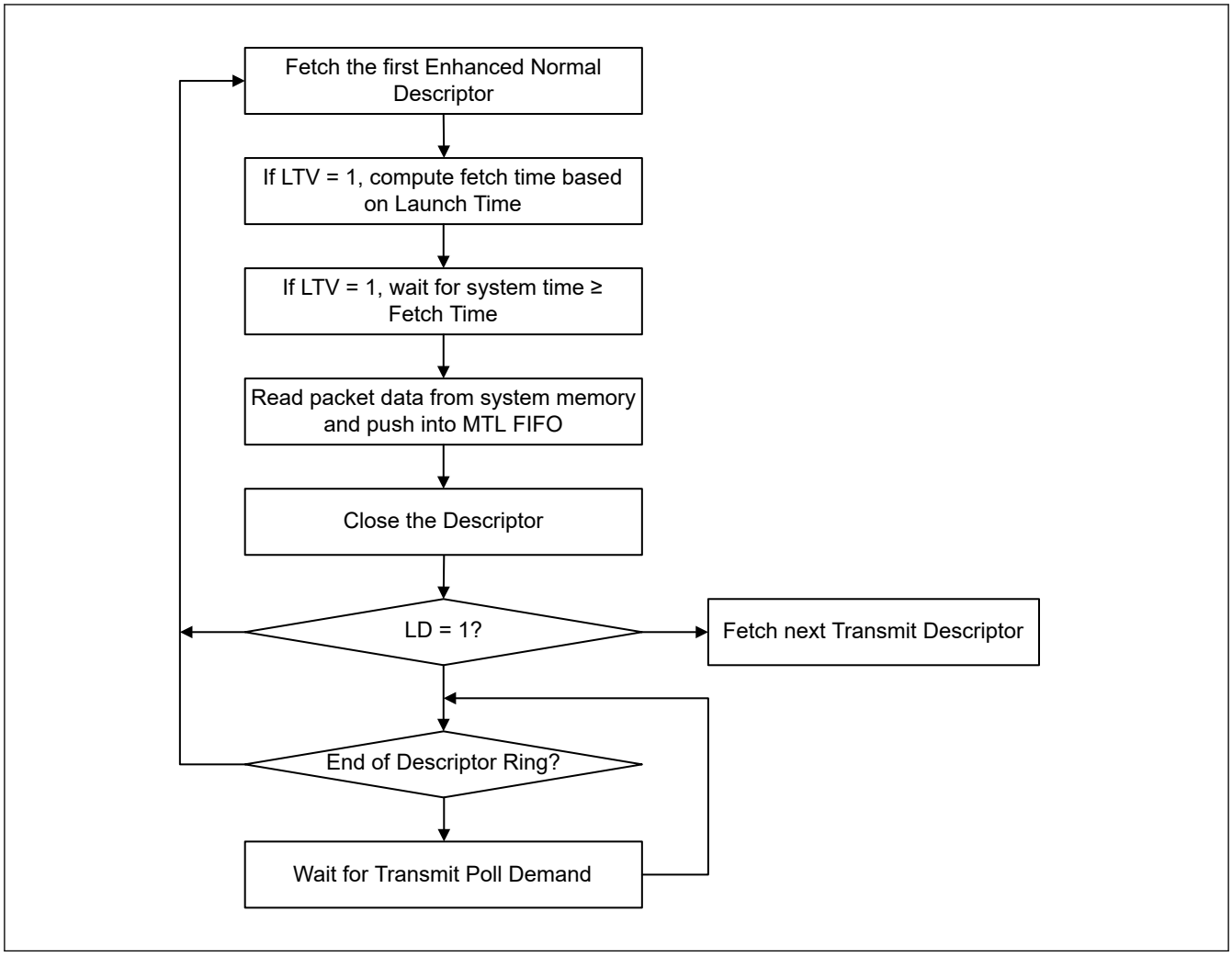


Figure 28.26 Time-based scheduling flow

28.12 TCP/IP Offloading Features

Communication protocols such as TCP and UDP implement checksum fields, which help determine the integrity of data transmitted over a network. The most widespread use of Ethernet is to encapsulate TCP and UDP over IP datagrams. Therefore, the MAC has an optional Checksum Offload Engine (COE) to support checksum calculation and insertion in the Transmit path, and error detection in the Receive path.

The TCP Segmentation Offload (TSO) engine is useful in offloading the TCP segmentation functions to the hardware.

28.12.1 Transmit Checksum Offload Engine

28.12.1.1 Overview of Transmit Checksum Offload Engine

The MAC has a Checksum Offload Engine (COE) to support checksum calculation and insertion in the Transmit path, using which, the software can offload the task of checksum insertion to the hardware. In the transmit path MAC calculates the checksum and inserts it in the Tx packet. This feature helps in reducing the load on the software and can improve the overall throughput of the system.

28.12.1.2 Description of Transmit Checksum Offload Engine

The checksum offload engine module supports two types of checksum calculation and insertion. The checksum engine can be controlled for each packet by setting the CIC bits (TDES3 bits [17:16]).

Note: The checksum for TCP, UDP, or ICMP is calculated over a complete packet, and then inserted into its corresponding header field. Because of this requirement, when this function is enabled, the Tx FIFO automatically operates in the store-and-forward mode even if the GMAC is configured for Threshold (cut-through) mode.

Note: You must make sure that the Tx FIFO is deep enough to store a complete packet before that packet is transferred to the MAC transmitter. The reason being that when space is not available to accept the programmed burst length of data, then the MTL Tx FIFO starts reading to avoid dead-lock. In such a case, the COE fails as the start of the packet header is read out before the payload checksum can be calculated and inserted. Therefore, you must enable the checksum insertion only in the packets that are less than the number of bytes, given by the following equation:

Packet size < TxQSize - (PBL + 5) × (132/8),

Where,

- TxQSize is indicated by TQS field of MTL_TXQn_OPERATIOn_MODE register.
- PBL = TxPBL field in the DMA_CHn_TX_CONTROL register.

(1) IP Header Checksum Engine

In IPv4 datagrams, the integrity of the header fields is indicated by the 16-bit Header Checksum field (the eleventh and twelfth bytes of the IPv4 datagram). The COE detects an IPv4 datagram when the Type field of Ethernet packet has the value 0x0800 and the Version field of IP datagram has the value 0x4. The checksum field of the input packet is ignored during calculation and replaced with the calculated value.

Note: IPv6 headers do not have a checksum field. Therefore, the COE does not modify the IPv6 header fields.

The result of this IP header checksum calculation is indicated by the IP Header Error status bit in the Transmit status (bit 0 in Table 28.75). This status bit is set whenever the values of the Ethernet Type field and the Version field of IP header are not consistent, or when the Ethernet packet does not have enough data, as indicated by the IP header Length field. In other words, this bit is set when an IP header error is asserted under the following circumstances:

- For IPv4 datagrams:
  - The received Ethernet type is 0x0800, but the Version field of IP header is not equal to 0x4.
  - The IPv4 Header Length field indicates a value less than 0x5 (20 bytes).
  - The total packet length is less than the value given in the IPv4 Header Length field.
- For IPv6 datagrams:
  - The Ethernet type is 0x86dd but the IP header Version field is not equal to 0x6.
  - The packet ends before the IPv6 header (40 bytes) or extension header (as given in the corresponding Header Length field in an extension header) is completely received.

(2) TCP/UDP/ICMP Checksum Engine

The TCP/UDP/ICMP Checksum Engine processes the IPv4 or IPv6 header (including extension headers) and determines whether the encapsulated payload is TCP, UDP, or ICMP. The checksum is calculated for the TCP, UDP, or ICMP payload and inserted into its corresponding field in the header. The Tx COE can work in the following two modes:

- The TCP, UDP, or ICMPv6 pseudo-header is not included in the checksum calculation and is assumed to be present in the Checksum field of the input packet. This engine includes the Checksum field in the checksum calculation, and then replaces the Checksum field with the final calculated checksum.
- The engine ignores the Checksum field, includes the TCP, UDP, or ICMPv6 pseudo-header data into the checksum calculation, and overwrites the checksum field with the final calculated value.

Note: For ICMP-over-IPv4 packets, the Checksum field in the ICMP packet must always be 0x0000 in both modes, because pseudo-headers are not defined for such packets. If it does not equal 0x0000, an incorrect checksum may be inserted into the packet.

The result of this operation is indicated by the Payload Checksum Error status bit in the Transmit Status vector (bit 12 in Table 28.75). This engine sets the Payload Checksum Error status bit when it detects that the packet has been forwarded to the MAC Transmitter engine in the store-and-forward mode without the end of packet (EOP) being written to the FIFO, or when the packet ends before the number of bytes indicated by the Payload Length field in the IP Header is received. When the packet is longer than the indicated payload length, the COE ignores them as stuff bytes, and no error is reported. When

this engine detects the first type of error, it does not modify the TCP, UDP, or ICMP header. For the second error type, it still inserts the calculated checksum into the corresponding header field.

Table 28.55 describes the functions supported by Transmit Checksum Offload engine based on the packet type. When the MAC does not insert the checksum, it is indicated as "No" in the table.

Note: You should not enable checksum insertion for IPv4 or IPv6 packets that are greater than the frame size constraint specified in section 28.12.1.2. Description of Transmit Checksum Offload Engine because it might result in incorrect checksum insertion or unexpected behavior.

Table 28.55 Transmit checksum offload engine functions for different packet types

Packet Type	Hardware IP Header Checksum Insertion	Hardware TCP/UDP Checksum Insertion
Non-IPv4 or IPv6 packet	No	No
IPv4 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in MAC) but less than or equal to the frame size constraint specified in section 28.12.1.2. Description of Transmit Checksum Offload Engine.	Yes	Yes
IPv6 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in MAC) but less than or equal to the frame size constraint specified in section 28.12.1.2. Description of Transmit Checksum Offload Engine.	Not Applicable	Yes
IPv4 with TCP, UDP, or ICMP	Yes	Yes
IPv4 packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers: • Hop-by-hop options (in IPv6 main header) • Hop-by-hop options (in IPv6 extension header) • Destinations options • Routing (with segment left 0) • Routing (with segment left > 0) • TCP, UDP, or ICMP • Authentication • Any other next header field in main or extension headers	 • Not applicable • Not applicable • Not applicable • Not applicable • Not applicable • Not applicable • Not applicable • Not applicable	 • Yes • No • Yes • No • No • Yes • No • No
IPv4 packet has TCP header with Options fields	Yes	Yes
IPv4 Tunnels: • IPv4 packet in an IPv4 tunnel • IPv6 packet in an IPv4 tunnel	 • Yes (IPv4 tunnel header) • Yes (IPv4 tunnel header)	 • No • No
IPv6 Tunnels: • IPv4 packet in an IPv6 tunnel • IPv6 packet in an IPv6 tunnel	 • Not applicable • Not applicable	 • No • No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Yes	Yes
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Not applicable	Yes
IPv4 frames with security features (such as encapsulated security payload)	Yes	No
IPv6 frames with security features (such as encapsulated security payload)	Not applicable	No

28.12.2 Receive Checksum Offload Engine

28.12.2.1 Overview of Receive Checksum Offload Engine

GMAC provides the Checksum Offload Engine that is used to detect any error in an IPv4 or IPv6 packet in the receive path. The MAC verifies the checksum field of the received packet with the internally calculated checksum and provides the status.

28.12.2.2 Description of Receive Checksum Offload Engine

The Receive Checksum Offload engine is used to detect errors in IP packets by calculating the header checksum and further matching it with the received header checksum. This engine also identifies a TCP, UDP, or ICMP payload in received IP packets and calculates the checksum of such payloads appropriately.

Here, both IPv4 and IPv6 packet in the received Ethernet packets are detected and processed for data integrity. The MAC receiver identifies IPv4 or IPv6 packets by checking for value 0x0800 or 0x86DD, respectively, in the Type field of the received Ethernet packet. This identification is applicable to single VLAN-tagged packets. It is also applicable to double VLAN-tagged packets when the Enable Double VLAN Processing option is selected and the EDVLP bit of the MAC_VLAN_Tag register is set.

The Rx COE calculates the IPv4 header checksums and checks that they match the received IPv4 header checksums. The result of this operation (pass or fail) is given to the RFC module for insertion into the receive status word. The IP Header Error bit is set for any mismatch between the indicated payload type (Ethernet Type field) and the IP header version, or when the received packet does not have enough bytes, as indicated by the Length field of the IPv4 header (or when fewer than 20 bytes are available in an IPv4 or IPv6 header).

Packets with TCP/IP errors (header or payload) are dropped in MTL when DIS_TCP_EF bit of the MTL_RXQn_OPERATIOn_MODE register is reset and FEP bit is set.

This engine also identifies a TCP, UDP, or ICMP payload in the received IP datagrams (IPv4 or IPv6) and calculates the checksum of such payloads properly, as defined in the TCP, UDP, or ICMP specifications. This engine includes the TCP, UDP, or ICMPv6 pseudo-header bytes for checksum calculation and checks whether the received checksum field matches the calculated value. The result of this operation is given as a Payload Checksum Error bit in the receive status word. This status bit is also set if the length of the TCP, UDP, or ICMP payload does not match the expected payload length given in the IP header.

Table 28.56 describes the functions supported by the Rx COE based on the packet type. When the payload of an IP packet is not processed (indicated as "No" in the table), the information (whether the checksum engine is bypassed or not) is given in the receive status.

Note: The MAC does not append any payload checksum bytes to the received Ethernet packets.

Table 28.56 Receive checksum offload engine functions for different packet types (1 of 2)

Packet Type	Hardware IP header checksum checking	Hardware TCP/UDP/ICMP checksum checking
Non-IPv4 or IPv6 packet	No	No
IPv4 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in the MAC)	Yes	Yes
IPv6 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in the MAC)	Not Applicable	Yes
IPv4 with TCP, UDP, or ICMP	Yes	Yes
IPv4 header's protocol field contains a protocol other than TCP, UDP, or ICMP	Yes	No
IPv4 packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers: • Hop-by-hop options (in IPv6 main header) • Hop-by-hop options (in IPv6 extension header) • Destinations options • Routing (with segment left 0) • Routing (with segment left > 0) • TCP, UDP, or ICMP • Any other next header field in main or extension headers	 • Not applicable • Not applicable • Not applicable • Not applicable • Not applicable • Not applicable • Not applicable	 • Yes • No • Yes • Yes • No • Yes • No
IPv4 packet has TCP header with Options fields	Yes	Yes
IPv4 Tunnels: • IPv4 packet in an IPv4 tunnel • IPv6 packet in an IPv4 tunnel	 • Yes (IPv4 tunnel header) • Yes (IPv4 tunnel header)	 • No • No

Table 28.56 Receive checksum offload engine functions for different packet types (2 of 2)

Packet Type	Hardware IP header checksum checking	Hardware TCP/UDP/ICMP checksum checking
IPv6 Tunnels: • IPv4 packet in an IPv6 tunnel • IPv6 packet in an IPv6 tunnel	 • Not applicable • Not applicable	 • No • No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Yes	Yes
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Not applicable	Yes
IPv4 frames with security features (such as encapsulated security payload)	Yes	No
IPv6 frames with security features (such as encapsulated security payload)	Not applicable	No

28.12.2.3 Enabling the Receive Checksum Offload Engine

Set the IPC bit of the MAC_Configuration register.

28.12.2.4 Header-Payload Split

The MAC is capable of identifying the boundary between header and payload of the received packet and store them into separate buffers. This feature is useful for multiple reasons:

- Header buffers can be located in faster memory/cache while Payload buffers are located in slower DRAM memory. This enables the software stack to process the headers faster
- Buffers used for payload can be directly forwarded to the application layer without the need to copy the payload into application buffer
- In case Large Receive Offload (LRO) function is implemented in software/driver layer, it is easy to link the buffers of multiple and contiguous payload data and form a bigger packet with a new header. This reduces the number of packets forwarded to the upper layer and thus improves system software performance

The GMAC supports multiple methods and packet types for header-payload splitting. This is controlled by the setting the SPLM field in MAC_Ext1_cfg register and described in Table 28.57.

Table 28.57 Split header support depending on the packet type

Packet Type	SPLM	Description
TCP or UDP packet	00b (L3/L4 Split)	The DMA writes the Ethernet header + IP header + TCP or UDP header into the header buffer.
IP packet (not TCP/UDP)		The DMA writes the Ethernet header + IP header into the header buffer.
Non-IP packet		The DMA does not split the header and payload
Any packet	01b (L2 Split)	The DMA writes the Ethernet header • For AV type packets using AV type Split Offset (SAVO) when split AV Enable (SAVE) field is set to 1. • For non-AV type packets using SPLOFST
IP packet	10b (Combination of L2 or L3/L4 Split)	L3/L4 split
Non-IP packet		L2 split
NA	11b	Reserved

Note:
- L3/L4 split is applicable for IP packets that are either untagged or VLAN stripped. If VLAN tag is retained in the packet forwarded to the DMA, L3-L4 split is not performed. When SPLM field is set to 2, L2 Split is performed for IP packets which are VLAN tagged.
- For AV packets, program the SAVE and SAVO fields of the MAC_Ext1_Cfg register to specify an alternative L2 split value.

The IP header includes IPv4 options in case of a IPv4 packet, and IPv6 extension headers in case of IPv6 frames. The points at which the L3/L4 header is split are shown in Figure 28.27.

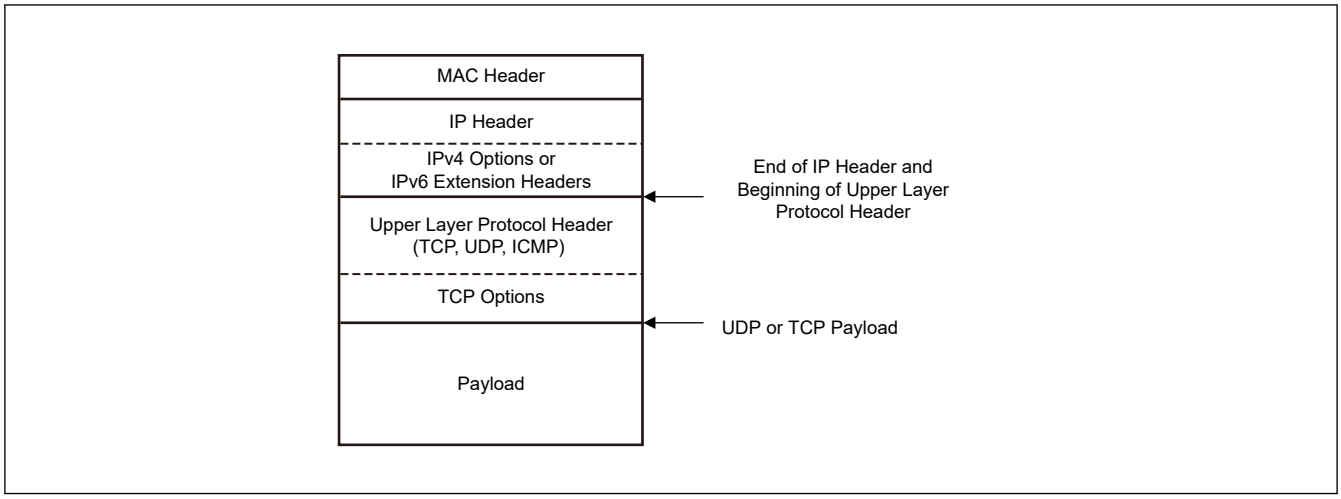


Figure 28.27 L3/L4 header split points

Figure 28.28 shows the L2 Split boundary. In this mode, the header payload boundary is determined for received packets at a fixed length offset starting from the Length/Type field, as indicated by the SPLOFST field of the MAC_Ext_Cfg1 register. This mode is applicable for all received packets when SPLM = 01b and only for non-IP packets when SPLM = 10b. For details, see section 28.3. Register Descriptions.

Table 28.58 shows the split offset for various packet types, when SPLOFST field is set to 2.

The split offset reference is always from the beginning of LT field, whether the VLAN tag is present or stripped.

For details about the descriptor structure for the split header feature, see section 28.17.3. Descriptor Structure for Split Header Support .

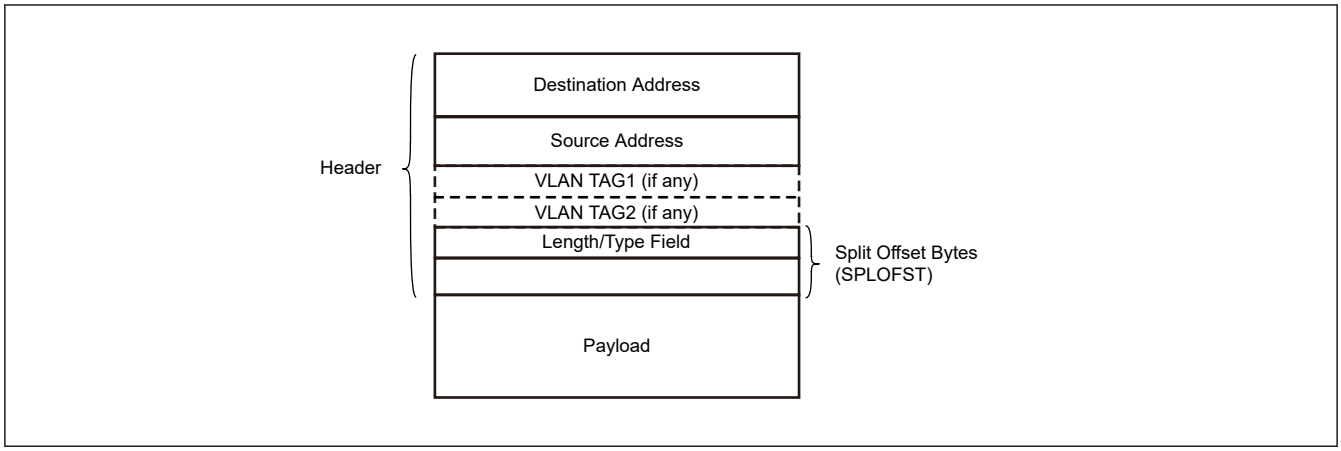


Figure 28.28 Header split

Table 28.58 Split offset calculation example: SPLOFST = 2

Packet Type	Split Offset (From the Beginning of LT Field)
Untagged packet	DA + SA + LT = 14
Single VLAN tagged packet with VTAG stripped	
Double VLAN tagged packet with both VTAGs stripped	
Single VLAN tagged packet with no stripping	DA + SA + VTAG + LT = 18
Double VLAN tagged packet with only one VTAG stripped	
Double VLAN tagged packet with no stripping	DA + SA +2 × VTAG + LT = 22

28.12.3 Using the IPv4 ARP Offload Engine

28.12.3.1 Introduction to IPv4 ARP Offload Engine

GMAC supports the Address Recognition Protocol (ARP) Offload for IPv4 packets. This feature allows the processing of the IPv4 ARP request packet in the receive path and generating corresponding ARP response packet in the transmit path. GMAC generates the ARP reply packets for appropriate ARP request packets. The ARP packet for IPv4 is L2 layer packet with Length/Type of 0x0806.

28.12.3.2 Description of IPv4 ARP Offload Engine

The ARP offloading process is as follows:

1. The MAC receiver gets an ARP request if the Target Protocol Address of request matches the IPv4 address programmed in the L3 register of the MAC.
2. The MAC generates an ARP reply packet.
3. The MAC copies the Sender Hardware Address field in the ARP request to the following fields:
   - DA field of the Ethernet packet header
   - Target Hardware Address field of the ARP reply packet
4. The MAC copies the Sender Protocol Address field in the ARP request to the Target Protocol Address field in the ARP reply packet.
5. The MAC places its MAC address in the following fields:
   - SA field of the Ethernet packet header
   - Sender Hardware Address field of the ARP reply packet
6. The MAC copies the Target Protocol Address field in the ARP request to the Sender Protocol Address field in the ARP reply packet.
7. The MAC sets the opcode field in ARP reply packet to 2 indicating ARP reply.
8. The MAC recalculates the CRC and performs padding for generated ARP reply packet.
9. The MAC transmitter sends the ARP reply.

The MAC processes only one ARP request at a time. It does not store the fields of multiple ARP requests. If the MAC receives an ARP request when it is already processing an earlier ARP request, the MAC does not generate the ARP reply for new ARP request. The MAC forwards the new ARP request packet to application with ARP Reply Not Generated (Bit 34) status bit set. However, in power-down mode, if the MAC receives an ARP request when it is already processing an earlier ARP request, the MAC drops the new ARP request.

If the Disable CRC check bit of the MAC Extension Configuration bit is set, then the MAC does not check for valid CRC of a ARP request Packet. It can generate an ARP response packet if the other conditions are valid.

The ARP request Packet must always have a valid CRC.

Note: When the received ARP request is less than 64 bytes packet length, GMAC does not send a ARP response. It is treated as normal packet and forwarded to the application based on the GMAC filter settings.

28.13 Power Management and Energy Efficient Ethernet

28.13.1 Overview of Low-Power Management

The GMAC supports the following techniques to save power.

- Magic Packet
- Remote Wakeup

The Magic Packet and Remove Wakeup techniques are used to save power in the host system when it is idle (Sleep mode) and has to be woken up only at the reception of specific packets from the Ethernet network. In the Sleep mode, the power to the host logic along with majority of the GMAC (except the MAC receiver logic), can be shut down. On receiving the

specific packets from the network, the MAC provides the trigger to restore the power to the host system and come back to normal state.

The (EEE) mode is compliant with the IEEE 802.3az-2010 standard. It is primarily targeted to save power in the Ethernet port when there is no traffic on the line. In this mode, the host indicates to the far-end that it does not have any packets to transmit for near future and puts the transmitter port (MAC Controller, PCS and PHY layers) into low-power mode. Similarly, the Receiver port can also be put into low-power mode when the far-end host indicates that it does not have any traffic to transfer. This allows significant saving of power in the Ethernet port (mainly in the PHY) with intermittent and bursty traffic profile. The triggering of entry and exit out of the EEE mode is controlled by the MAC and is supported within the GMAC.

Simultaneous operation of the EEE mode along with any or both the other power saving modes is also supported in GMAC.

28.13.1.1 Description of Magic Packet Mode

This section describes the power saving through magic packet detection.

Note:
- The magic packet feature is based on the Magic Packet technology white paper from Advanced Micro Device (AMD).
- The watchdog timeout limit for a magic packet is 2,048 bytes irrespective of the value programmed in WD bit of MAC_Configuration register and PWE bit in MAC_Watchdog_Timeout register.
- The value programmed in DCRCC bit of MAC_Ext_Configuration register is applicable to a magic packet.
- When you enable the low-power mode in the PMT block, the MAC drops all received packets and does not forward any packet to the MTL Rx Queue or the application. The MAC comes out of the low-power mode when a remote wake-up packet is received and the corresponding detection is enabled.

In the Magic Packet based power saving is a mode, the reception of valid magic packet by MAC receiver triggers an exit from low-power mode. The MAC enters power saving mode when PWRDWN bit of MAC_PMT_Control_Status register is programmed to 1. Exit from the magic packet based low-power mode is enabled by setting the MGKPKTEN bit of MAC_PMT_Control_Status register to 1.

The magic packet contains a unique pattern at any offset after the Destination address, Source address, and Length/Type fields. In addition to the unique pattern matching, the MAC receiver also checks for the following, to detect the received packet as a valid magic packet:

- The packet must be addressed to it (Destination Address of the received packet should perfect match the MAC_Address0_High and MAC_Address0_Low registers) or with multicast/broadcast address
- The packet must not have length error, FCS error, dribble bit error, GMII error, and collision
- The packet must not be runt (length including Ethernet header and FCS is at least 64 bytes)

The content of the unique pattern in magic packet is described as

- 6 bytes of all-ones (0xFFFFFFFFFFFF) called the synchronization stream. There can be more than six bytes of 0xFF, but last 6 are considered.
- The synchronization stream is immediately followed by 16 repetitions of Destination address field of the packet (MAC Address (MAC_Address0_High and MAC_Address0_Low registers) or multicast/broadcast address)
- No break or interruption between synchronization stream and first repetition of Destination address field or within its 16 repetitions

If the MAC address of a node is 0x001122334455, the MAC scans for the following data sequence:

Destination Address Source Address Length/Type…… FF FF FF FF FF FF

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

…CRC

28.13.1.2 Description of Remote Wakeup Packet Mode

This section describes the Remote Wakeup Packet based power saving mode.

Note:
- The remote wake-up packet feature is implemented based on the Device Class Power Management Reference Specification and various implementation-specific white papers.
- The watchdog timeout limit for a magic packet is 2,048 bytes irrespective of the value programmed in WD bit of MAC_Configuration register and PWE bit in MAC_Watchdog_Timeout register.
- The value programmed in DCRCC bit of MAC_Ext_Configuration register is applicable to a magic packet.
- When you enable the low-power mode in the PMT block, the MAC drops all received packets and does not forward any packet to the MTL Rx Queue or the application. The MAC comes out of the low-power mode when a remote wake-up packet is received and the corresponding detection is enabled.

In the Remote Wakeup Magic Packet based power saving mode, the reception of expected remote wakeup packet by MAC receiver triggers the exit from low-power mode. The MAC enters power saving mode when PWRDWN bit of MAC_PMT_Control_Status register is programmed to 1. Exit from the remote wakeup based low-power mode is enabled by programming RWKPKTEN bit of MAC_PMT_Control_Status register to 1.

The MAC implements a filter lookup table (programmed through MAC_RWK_Packet_Filter register) in which CRC, offset, and byte mask of the pattern embedded in remote wakeup packet and the filter operation commands are programmed.

The pattern embedded in the remote wakeup packet is located at any offset after the Destination address and Source address fields. In addition to the CRC match for the pattern, the MAC receiver also checks the following, to detect the received packet as a valid remote wakeup packet:

- The packet must be addressed to it (Destination Address of the received packet should perfect match the MAC_Address0_High and MAC_Address0_Low registers) or with multicast/broadcast address.
- The packet must not have length error, FCS error, dribble bit error, GMII error, and collision.
- The packet must not be runt (length including Ethernet header and FCS is at least 64 bytes).

When a valid remote wakeup packet is received, the MAC receiver sets the RWKPRCVD bit in MAC_PMT_Control_Status register and triggers the interrupt on GMAC_PMT. The PMTIS bit in MAC_Interrupt_Status register is set when power-gating is not enabled in low-power mode. An interrupt is triggered to the application via GMAC_SBD when interrupt is enabled (PMTIE bit in MAC_Interrupt_Enable register is set) and CSR clock is not gated off in low-power mode.

Remote Wake-Up Packet Filters

When Remote Wakeup based power saving mode is selected, it allows selection of 4 Remote Wakeup Filters. The Remote Wakeup Filters structure is shown in Table 28.59.

Table 28.59 Remote wake-up packet filter register

wkupfmfilter_reg0	Filter 0 Byte Mask							
wkupfmfilter_reg1	Filter 1 Byte Mask							
wkupfmfilter_reg2	Filter 2 Byte Mask							
wkupfmfilter_reg3	Filter 3 Byte Mask							
wkupfmfilter_reg4	RSVD	Filter 3 Command	RSVD	Filter 2 Command	RSVD	Filter 1 Command	RSVD	Filter 0 Command
wkupfmfilter_reg5	Filter 3 Offset		Filter 2 Offset		Filter 1 Offset		Filter 0 Offset	
wkupfmfilter_reg6	Filter 1 CRC-16				Filter 0 CRC-16			
wkupfmfilter_reg7	Filter 3 CRC-16				Filter 2 CRC-16			

The Remote Wakeup Filters are arranged in blocks of 4 filters each and each such block have eight 32-bit wide registers, that is, wkuppktfilter_reg0-7. The fields of Remote Wakeup Filter are as follows:

Filter i Byte Mask

The filter i byte mask register defines the bytes of the packet that are examined by filter i (0, 1, 2 and 3) to determine whether or not a packet is a wake-up packet.

- The MSB (31st bit) must be zero.

- Bit j[30:0] is the byte mask.
- If bit j (byte number) of the byte mask is set, the CRC block processes the Filter i Offset + j of the incoming packet; otherwise Filter i Offset + j is ignored

Filter i Command

The 4-bit filter i command controls the filter i operation.

- Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet
- Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC-16 value. Bit 2, along with bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2".
- Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. The details are as follows:
  - The And_Previous bit setting is applicable within a set of 4 filters.
  - Setting of And_Previous bit of filter that is not enabled has no effect, that is setting And_Previous bit of lowest number filter in the set of 4 filters has no effect. For example, setting of And_Previous bit of Filter 0 has no effect.
  - If And_Previous bit is set for filter to form AND chained filter, the AND chain breaks at the point any filter is not enabled. For example: If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set) but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result is considered. If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result ANDed with Filter 3 result is considered. If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 2 is not enabled (bit 0 in Filter 2 command is reset), then since setting of Filter 2 And_Previous bit has no effect only Filter 1 result ORed with Filter 3 result is considered.
  - If filters chained by And_Previous bit setting have complementary programming, then a frame may never pass the AND chained filter. For example, if Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 1 Address_Type bit is set (bit 3 in Filter 1 command is set) indicating multicast detection and Filter 2 Address_Type bit is reset (bit 3 in Filter 2 command is reset) indicating unicast detection or vice versa, a remote wakeup frame does not pass the AND chained filter as a remote wakeup frame cannot be of both unicast and multicast address type.
- Bit 0 is the enable for filter i. If bit 0 is not set, filter i is disabled.

Filter i Offset

The filter i offset register defines the offset (within the packet) from which the filter i examines the packets.

- This 8-bit pattern-offset is the offset for the filter i first byte to be examined.
- The minimum allowed offset is 12, which refers to the 13th byte of the packet.
- The offset value 0 refers to the first byte of the packet.

Filter i CRC-16

The filter i CRC-16 register contains the CRC-16 value calculated from the pattern and the byte mask programmed in the Remote Wakeup filter register.

- The 16-bit CRC calculation uses the following polynomial:
  $G(x) = x^{16} + x^{15} + x^2 + 1$
- Each mask, used in the hash function calculation, is compared with a 16-bit value associated with that mask. Each filter has the following:
  - 32-bit Mask: Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1, the corresponding byte is taken into the CRC-16 calculation.
  - 8-bit Offset Pointer: Specifies the byte to start the CRC-16 computation. The pointer and the mask are used together to locate the bytes to be used in the CRC-16 calculations.

The Remote Wakeup Filter registers are implemented as 8 indirect access registers (wkuppktfilter_reg[n]) and accessed by application through MAC_RWK_Packet_Filter register. When the Remote Wakeup Filters are to be programmed, the entire set of wkuppktfilter_reg registers must be written. The wkuppktfilter_reg register is programmed by sequentially writing the eight register values in MAC_RWK_Packet_Filter register for wkuppktfilter_reg0, wkuppktfilter_reg1, …, wkuppktfilter_reg7 respectively. The wkuppktfilter_reg register is read in a similar way. The MAC updates the wkuppktfilter_reg register current pointer value in RWKPTR field of MAC_PMT_Control_Status register.

Note:
- If the MAC_RWK_Packet_Filter register is accessed in byte or half-word mode, the internal counter to access the appropriate wkuppktfilter_reg is incremented when CPU accesses Lane 3.
- When MAC_RWK_Packet_Filter register is written, the content is transferred from CSR clock domain to PHY receive clock domain after the write operation, there should not be any further write to the MAC_RWK_Packet_Filter register until the first write is updated in PHY receive clock domain. Otherwise, the second write operation does not get updated to the PHY receive clock domain. Therefore, the delay between two writes to the MAC_RWK_Packet_Filter register should be at least 4 cycles of the PHY receive clock.

28.13.1.3 PMT Interrupt

The PMT interrupt is asserted when a valid remote wake-up packet is received. In addition to GMAC_SBD, the GMAC_PMT signal is asserted. The GMAC_PMT signal, synchronous to the Rx clock domain, is provided so that you can stop the application clock when the MAC is in the power-down mode. As the GMAC_PMT signal is generated in the PHY Rx clock domain, it is not cleared immediately when the MAC_PMT_Control_Status register is read.

This is because the resultant clear signal has to cross to the PHY Rx clock domain, and then clear the interrupt source. This delay is at least 4 clock cycles of Rx clock and can be significant when the GMAC is operating in the 10 Mbps mode. When the application clears the PWRDWN bit in Remote Wake-Up Packet Detection register, the MAC comes out of the power-down mode, but this event does not generate the PMT interrupt.

28.13.2 Energy Efficient Ethernet (EEE)

EEE is an operational mode that enables the IEEE 802.3 Media Access Control (MAC) sub layer along with a family of physical layers to operate in the Low-Power Idle (LPI) mode. The EEE operational mode supports the IEEE 802.3 MAC operation at 100 Mbps, 1000 Mbps, and 10 Gbps. The GMAC supports the IEEE 802.3az-2010 for EEE.

The LPI mode allows power saving by switching off the parts of the communication device functionality when there is no data to be transmitted and received. The systems on both sides of the link can disable some functionalities to save power during the periods of low-link utilization. The MAC controls whether the system should enter or exit the LPI mode and communicates this to the PHY.

The EEE specifies the capabilities negotiation methods that the link partners can use to determine whether EEE is supported, and then select the set of parameters that are common to both devices.

28.13.2.1 Transmit Path Functions

The transmit path functions include tasks that the MAC must perform to make the PHY to enter the LPI state.

In the transmit path, the software must set the LPIEN bit of the MAC_LPI_Control_Status register to indicate to the MAC to stop transmission and initiate the LPI protocol. The MAC completes the transmission in progress, generates its transmission status, and starts transmitting the LPI pattern instead of the IDLE pattern if the link status has been up continuously for a period specified in the LPI LS TIMER field of MAC_LPI_Control_Status register. The PHY Link Status bit of the MAC_LPI_Control_Status register indicates the link status of the PHY.

Note:
- According to the standard (IEEE 802.3az-2010), the PHY must not stop the TxCLK clock during the LPI state in the MII (10 or 100) mode. However, the MAC can stop the GTX_CLK clock during the LPI state in the GMII (1000) mode.
- According to the standard (IEEE 802.3az-2010), the LPI mode is supported only in the full-duplex mode. Therefore, you must not enable the LPI mode when the MAC Transmitter is configured for the half-duplex mode.

To make the PHY enter the LPI state, the MAC performs the following tasks:

1. De-asserts TX_EN
2. Asserts TX_ER
3. Sets TXD[3:0] to 0x1 (for 100 Mbps) or TXD[7:0] to 0x01 (for 1,000 Mbps)

The MAC maintains the same state of the TX_EN, TX_ER, and TXD signals for the entire duration during which the PHY remains in the LPI state.

4. Updates the status (TLPIEN bit of MAC_LPI_Control_Status register) and generates an interrupt

To bring the PHY out of the LPI state, that is, when the software resets the LPIEN bit, the MAC performs the following tasks:

1. Stops transmitting the LPI pattern and starts transmitting the IDLE pattern
2. Starts the LPI TW TIMER
   The MAC cannot start the transmission until the wake-up time specified for the PHY expires. The auto-negotiated wake-up interval is programmed in the TWT field of the MAC_LPI_Timers_Control register.
3. Updates the LPI exit status (TLPIEX bit of the MAC_LPI_Control_Status register) and generates an interrupt

Note:
- If the MAC is in the Tx LPI mode and the Tx clock is stopped, the application should not write to CSR registers that are synchronized to Tx clock domain.
- If the MAC is in the LPI mode and the application issues a soft reset or hard reset, the MAC transmitter comes out of the LPI mode.

28.13.2.2 Automated Entry/Exit of LPI mode in Transmit Path

The MAC transmitter can be programmed to enter and exit LPI IDLE mode automatically based on whether it is IDLE for a specific period of time or has a packet to transfer. These modes are enabled and controlled by MAC_LPI_Control_Status register.

When LPITXA (bit [19]) and LPITXEN (bit [16]) of MAC_LPI_Control_Status register are set, the MAC transmitter enters LPI IDLE state when the MAC transmit path (including the MTL layers and DMA layers) are idle. The MAC transmitter exits the LPI IDLE state and clear the LPITXEN bit as soon as any of functions in the TX path (DMA, MTL or MAC) becomes non-idle due to initiation of a packet transfer.

In addition, when bit [20] (LPIATE) is also set, the MAC transmitter enters LPI IDLE state only if the Transmit path remains in idle state (no activity) for the time period indicated by the value in MAC_LPI_Entry_Timer. In this mode also, the MAC transmitter exits the LPI IDLE state as soon as any of the functions becomes non-idle. However, the LPITXEN bit is not cleared but remains active so that reentry to LPI IDLE state is possible without any software intervention when the MAC becomes idle again.

When both LPIATE and LPITXA bits are cleared, you can directly control the entry and exit of LPI IDLE state by programming the LPITXEN bit.

28.13.2.3 Receive Path Functions

The receive path functions include the tasks that the PHY and MAC must perform when the PHY receives signals from the link partner to exit the LPI state.

In the receive path, when the PHY receives the signals from the link partner to enter into the LPI state, the PHY and MAC perform the following tasks:

1. The PHY asserts RX_ER
2. The PHY sets RXD[7:0] to 0x01
3. The PHY de-asserts RX_DV
   The PHY maintains the same state of the RX_ER, RXD, and RX_DV signals for the entire duration during which it remains in the LPI state.
4. The MAC updates the RLPIEN bit of the MAC_LPI_Control_Status register and immediately generates an interrupt

Note:
- If the LPI pattern is detected for a very short duration (that is, less than two cycles of Rx clock), the MAC does not enter the Rx LPI mode.
- If the duration between end of the current Rx LPI pattern and start of the next Rx LPI pattern, is very short (that is, less than two cycles of Rx clock), then the MAC exits and again enters the Rx LPI mode. The MAC does not give the Rx LPI Exit and Entry interrupts

When the PHY receives signals from the link partner to exit the LPI state, the PHY and MAC perform the following tasks:

1. The PHY de-asserts RX_ER and returns to a normal inter-packet state.

2. The MAC updates the RLPIEX bit of the MAC_LPI_Control_Status register and generates an GMAC_SBD interrupt immediately. The GMAC_LPI interrupt is also asserted.

Figure 28.29 shows the behavior of RX_ER, RX_DV, and RXD[3:0] signals during the LPI mode transitions.

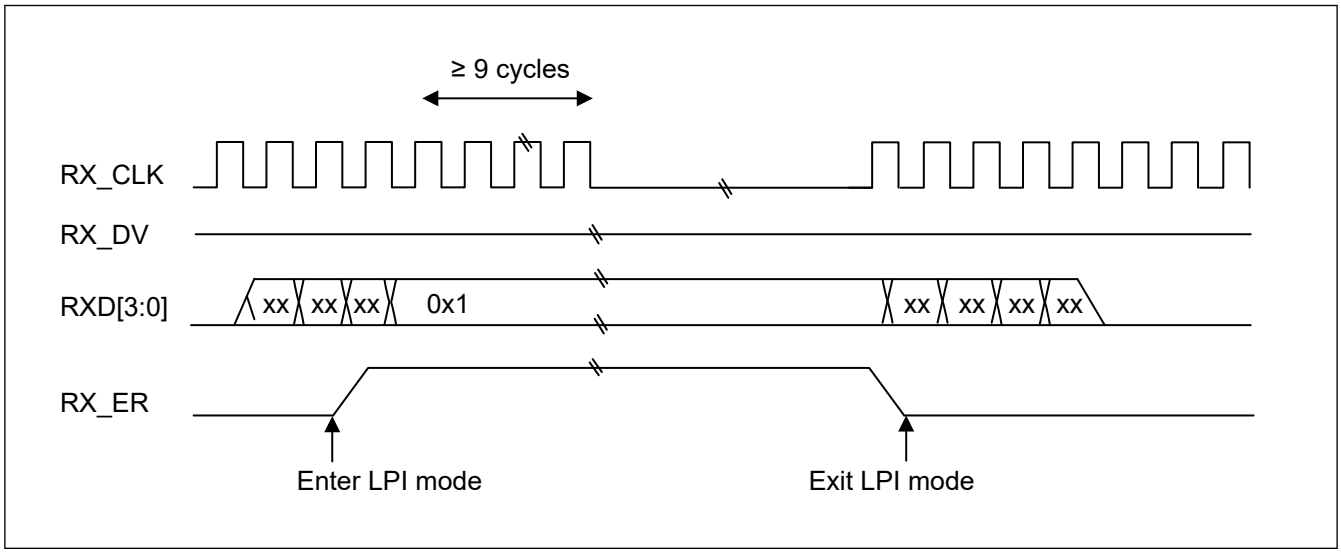


Figure 28.29 LPI transitions (receive)

Note:
- If the RX_CLK_stoppable bit (in the PHY register written through MDIO) is asserted when the PHY is indicating LPI to the MAC, the PHY may halt the RX_CLK at any time more than nine clock cycles after the start of the LPI state as shown in Figure 28.29.
- If the MAC is in the LPI mode and the application issues a soft reset or hard reset, the MAC receiver comes out of the LPI mode during reset. If the LPI pattern is still received after the reset is de-asserted, the MAC receiver again enters the LPI state.
- If the RX clock is stopped in the RX LPI mode, the application should not write to the CSR registers that are being synchronized to the RX clock domain.
- When the PHY sends the LPI pattern, if EEE feature is enabled, the MAC automatically enters the LPI state. There is no software control to prevent the MAC from entering the LPI state.

28.13.2.4 EEE Registers

The following are the registers related to Energy Efficient Ethernet features:

- MAC_LPI_Control_Status
- MAC_LPI_Timers_Control
- MAC_LPI_Entry_Timer

For detailed description of these registers, see section 28.3. Register Descriptions.

28.13.2.5 LPI Timers

The transmitter maintains the LPI LS TIMER, LPI TW TIMER, and LPI AUTO ENTRY TIMER timers.

Following are the LPI timers that are loaded with the respective values from the MAC_LPI_Timers_Control and MAC_LPI_Entry_Timer registers:

- LPI LS TIMER
  The LPI LS TIMER counts, in milliseconds, the time expired since the link status is up. You can enable the monitoring of the LUD bit of the MAC_PHYIF_Control_Status register by setting the PLSEN bit of MAC_LPI_Control_Status register.
  The link status is indicated to the MAC by the LNKSTS bit of the MAC_PHYIF_Control_Status register or the value programmed by the software in the PLS bit of the MAC_LPI_Control_Status register. If the link status is not available in the MAC_PHYIF_Control_Status, the software should get the PHY link status by reading the PHY register and accordingly update the PLS bit.

This timer is cleared every time the link goes down. It starts to increment when the link is up again and continues to increment until the value of the timer becomes equal to the terminal count. Once the terminal count is reached, the timer remains at the same value as long as the link is up. The terminal count is the value programmed in bits25:16] of the MAC_LPI_Timers_Control register. The GMII interface does not assert the LPI pattern unless the terminal count is reached. This ensures a minimum time for which no LPI pattern is asserted after a link is established with the remote station. This period is defined as 1 second in the IEEE 802.3-az-2010. The LPI LS TIMER is 10-bit wide. Therefore, the software can program up to 1023 milliseconds.

- LPI TW TIMER
  The LPI TW TIMER counts, in microseconds, the time expired since the de-assertion of LPI. The terminal count should be programmed in bits [15:0] of LPI_Timers_Control register. The terminal count of the timer is the value of resolved Transmit TW that is the auto-negotiated time after which the MAC can resume the normal transmit operation. After exiting the LPI mode, the MAC resumes its normal operation after the TW timer reaches the terminal count.
  The MAC supports the LPI TW TIMER in units of microsecond. The LPI TW TIMER is 16-bit wide. Therefore, the software can program up to 65535 micro seconds.
- LPI AUTO ENTRY TIMER
  This timer counts in steps of eight microseconds, the time for which the MAC transmit path has to remain in idle state (no activity), before the MAC Transmitter enters the LPI IDLE state and starts transmitting the LPI pattern. This timer is enabled when LPITE bit in MAC_LPI_Control_Status register is set.

Note: Program the PLS bit of MAC_LPI_Control_Status to 0 before switching between the GMII and MII modes. This resets the internal timers. If the mode is changed after the LPI LS TIMER or LPI TW TIMER starts, the change in the Tx clock frequency can result in incorrect timeout.

28.13.2.6 LPI Interrupt

The MAC generates the LPI interrupt when the Tx or Rx side enters or exits the LPI state. The interrupt GMAC_SBD is asserted when the LPI interrupt status is set. The LPI interrupt can be cleared by reading the MAC_LPI_Control_Status register.

When the MAC exits the Rx LPI state, then in addition to the GMAC_SBD, the GMAC_LPI is asserted. You can use the GMAC_LPI signal to trigger the external clock-gating circuitry to restore the application clock to the MAC. The GMAC_LPI signal, synchronous to the Rx clock domain, is provided so that you can stop the application clock when the MAC is in the LPI state.

The GMAC_LPI signal is generated in the Rx clock domain. It may not be cleared immediately after the MAC_LPI_Control_Status register is read. This is because the clear signal, generated in CSR clock domain, has to cross the Rx clock domain, and then clear the interrupt source. This delay is at least four clock cycles of Rx clock and can be significant when the GMAC is operating in the 10Mbps mode.

28.14 MAC Management Counters

GMAC supports a set of registers known as MAC Management Counters that store statistics on the received and transmitted packets.

28.14.1 MAC Management Counters

The GMAC supports storing the statistics about the received and transmitted packets in registers that are accessible through the application.

The counters in the MAC Management Counters (MMC) module can be viewed as an extension of the register address space of the CSR module. The MMC module maintains a set of registers for gathering statistics on the received and transmitted packets. The register set includes a control register for controlling the behavior of the registers, two status registers containing interrupts generated (receive and transmit), and Interrupt Enable registers (receive and transmit). These registers are accessible from the Application. Each register is 32-bits wide. 8-, 16- and 32-bit accesses are allowed as long as the address is word-aligned. The MMCs are accessed using transactions, in the same way the CSR address space is accessed.

The MMC counters are free running. There is no separate enable for the counters to start. If a particular MMC counter is present in the RTL, it starts counting when corresponding packet is received or transmitted. The Receive MMC counters are updated for packets that are passed by the Address Filter (AFM) block. The statistics of packets, dropped by the AFM

module, are not updated unless they are runt packets of less than 6 bytes (DA bytes are not received fully). To get statistics of all packets, set bit 0 in the MAC_Packet_Filter register.

The MMC module gathers statistics on encapsulated IPv4, IPv6, TCP, UDP, or ICMP payloads in received Ethernet packets.

28.14.2 Address Assignments

The MMC registers follow a certain naming convention and their descriptions use certain terminologies that you should know.

The MMC register naming conventions are as follows.

- "tx" as a prefix or suffix indicates counters associated with transmission
- "rx" as a prefix or suffix indicates counters associated with reception
- "_g" as a suffix indicates registers that count only good packets
- "_gb" as a suffix indicates registers that count packets regardless of whether they are good or bad.

The following definitions define the terminology used in MMC register descriptions.

- Transmitted packets are considered "good" if transmitted successfully. In other words, a transmitted packet is good if the packets transmission is not aborted because of any of the following errors:
  - Jabber Timeout
  - No Carrier or Loss of Carrier
  - Late Collision
  - Packet Underflow
  - Excessive Deferral
  - Excessive Collision
- Received packets are considered "good" if none of the following errors exists:
  - CRC error
  - Runt packet (shorter than 64 bytes)
  - Alignment error (in 10/100 Mbps only)
  - Length error (non-Type packet only)
  - Out of Range (non-Type packet only, longer than 1518 bytes)
  - GMII_RXER Input error
- The maximum transmit frame size depends on the frame type, as follows:
  - Untagged frame maxsize = 1,518
  - VLAN Frame maxsize = 1,522
  - Jumbo Frame maxsize = 9,018
  - JumboVLAN Frame maxsize = 9,022
- The maximum receive packet size depends on the packet type and control bits (JE, S2KP, GPSLCE and EDVLP), as shown in the Table 28.60.

Table 28.60 Size of the maximum receive packet (1 of 2)

JE	S2KP	GPSLCE	EDVLP	Untagged Frame maxsize in bytes	Single VLAN Frame maxsize in bytes	Double VLAN Frame maxsize in bytes
1	x	x	1	9018	9022	9026
0	1	x	x	2000	2000	2000
0	0	1	1	GPSL	GPSL+4	GPSL+8
0	0	0	1	1518	1522	1526
1	x	x	0	9018	9022	9022

Table 28.60 Size of the maximum receive packet (2 of 2)

JE	S2KP	GPSLCE	EDVLP	Untagged Frame maxsize in bytes	Single VLAN Frame maxsize in bytes	Double VLAN Frame maxsize in bytes
0	0	1	0	GPSL	GPSL+4	GPSL+4
0	0	0	0	1518	1522	1522

Note: • The MMC counters registers at the following offset addresses are of type Read-Only and have the default value of 0:
  – 0x0714 to 0x0778
  – 0x0780 to 0x07E4
  – 0x0810 to 0x0844
  – 0x0850 to 0x0884

28.15 Flow Control

28.15.1 Transmit Flow Control

The Transmit Flow Control involves transmitting Pause packets in full-duplex mode and backpressure in half-duplex mode to control the flow of packets from the remote end.

28.15.1.1 Flow Control in Full-Duplex Mode

GMAC supports full-duplex flow control operations.

In full-duplex mode, the GMAC uses the following packet type for flow control:

- IEEE 802.3x Pause packets

(1) Pause Packet Structure

Table 28.61 describes the fields of a Pause packet.

Table 28.61 Pause packet fields

Field	Description
DA	Contains the special multicast address
SA	Contains the MAC address 0
Type	Contains 8808
MAC Control opcode	Contains 0001 for IEEE 802.3x Pause Control packets
PT	Contains Pause time specified in the PT field of the MAC_Q0_Tx_Flow_Ctrl register

(2) Pause Packet Control

When the FCB bit is set, the MAC generates and transmits a single Pause packet. If the FCB bit is set again after the Pause packet transmission is complete, the MAC sends another Pause packet irrespective of whether the pause time is complete or not. To extend the pause or terminate the pause prior to the time specified in the previously-transmitted Pause packet, the application should program the Pause Time register with appropriate value and then again set the FCB bit.

Similarly, when the hw_flowctrl is asserted, the MAC generates and transmits a single Pause packet. If the hw_flowctrl remains asserted at a configurable number of slot times before the Pause time runs out, the MAC transmits a second Pause packet. This process is repeated as long as the hw_flowctrl remains active. If the hw_flowctrl goes inactive prior to the sampling time, the MAC transmits a Pause packet with zero Pause time (if the DZPQ bit in MAC_Q0_Tx_Flow_Ctrl register is set to 0) to indicate to the remote end that the Receive buffer is ready to receive new data packets.

28.15.1.2 Flow Control in Half-Duplex Mode

In half-duplex mode, the MAC uses the deferral mechanism for the flow control (backpressure). When the application requests to stop receiving packets, the MAC sends a JAM pattern of 32 bytes when it senses a packet reception, provided the transmit flow control is enabled. This results in a collision and the remote station backs off. If the application requests a

packet to be transmitted, it is scheduled and transmitted even when the backpressure is activated. If the backpressure is kept activated for a long time (and more than 16 consecutive collision events occur), the remote stations abort the transmission because of excessive collisions.

Table 28.62 describes the flow control in the Tx path for Queue 0 based on the setting of the following bits:

- EHFC bit of MTL_RxQ0_Operation_Mode register
- TFE bit of MAC_Q0_Tx_Flow_Ctrl register
- DM bit of MAC_Configuration register

Flow control is similar for all queues.

Table 28.62 Tx MAC flow control

EHFC	TFE	DM	Description
x	0	x	The MAC transmitter does not perform the flow control or backpressure operation.
0	1	0	The MAC transmitter performs back-pressure when bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set.
1	1	0	The MAC transmitter performs back-pressure when bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set. In addition, the MAC Tx performs back-pressure when Rx Queue level crosses the threshold set by bits [10:8] of MTL_RxQ0_Operation_Mode register.
0	1	1	The MAC transmitter sends the Pause packet when bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set.
1	1	1	The MAC transmitter sends the Pause packet when bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set. In addition, the MAC Tx sends a Pause packet when Rx Queue level crosses the threshold set by bits [10:8] of MTL_RxQ0_Operation_Mode register.

28.15.1.3 Enabling Transmit Flow Control

To independently enable Transmit Flow Control for each Rx queue, set the TFE bit in the MAC_Q0_Tx_Flow_Ctrl register.

28.15.1.4 Triggering Transmit Flow Control

The Transmit Flow Control involves transmitting Pause packets in full-duplex mode and backpressure in half-duplex mode to control the flow of packets from the remote end. The application can request the MAC to send a Pause packet or initiate back-pressure by using either of the following methods:

- Software Trigger: In this method, the application sets the FCB_BPA bit in the corresponding MAC_Q0_Tx_Flow_Ctrl register.
- Hardware Trigger: In this method, the application triggers the flow control by asserting the flow control based on the Rx Queue (hw_flowctrl).

The hw_flowctrl is triggered based on the following:

- Rx Queue Threshold: The flow control operation of the MAC is enabled when the EHFC bit of corresponding MTL_RXQn_OPERATIOn_MODE register is set. The flow control signal to the MAC is asserted when the fill level of the Rx queue crosses the threshold configured in RFA field of MTL_RXQn_OPERATIOn_MODE register. This flow control signal is de-asserted when the fill-level of the queue is lower than the threshold configured in the RFD field. The hardware flow control generated based on the Rx Queue threshold crossing condition is applicable only when the Rx queue size is 4,096 bytes or more.

Note: The RFA sets the threshold at which the flow control is triggered and the MAC schedules to send a PAUSE frame to be sent by the local transmitter.

The remote transmitter can continue to send packets until it receives this PAUSE packet. So the RXQ needs space to take in this data even after the flow-control is triggered, to avoid overflow and loss of packets.

The max-space required that is calculated theoretically can be 2 max-sized packets + a little more assuming that the read from the RXFIFO is not occurring during this whole time.

Theoretically, this makes RFA = 4 or more, which implies that RxQ must have at the least 4 KB size.

Practically, the system works with RFA = 2 (Full – 2 KB, with very little probability of overflow).

If you have FIFO size less than 4K, with RFA = 2, the flow control is triggered with 1000B in the RxQ, which is less than max packet size.

The RFA field MTL_RXQn_OPERATIOn_MODE register is used for activating the Flow Control.

These bits control the threshold (fill-level of Rx queue) at which the flow control is activated.

For Example: If your RXQ size is 2K and you set the RFA to 1K; as soon as 1K is received, Flow Control is activated. If you send a packet of 1500 bytes (which is allowed as per spec), flow-control is triggered even before you receive a complete packet. This reduces the throughput considerably.

If your RXQ size is 4K, you can set the RFA to 2K, so that you are able to receive the whole packet.

28.15.2 Receive Flow Control

In the Receive path, the Flow Control is functional only in the full-duplex mode. If any Pause packet is received in the half-duplex mode, the packet is considered as a normal control packet.

Note: Receive pause packets should have a frame size of 64 bytes.

28.15.2.1 Description of Receive Flow Control

The Receive Flow Control is implemented by the MAC based on the bit value of the respective register, and the destination address and different fields of the received packet.

Table 28.63 describes the flow control in the Rx path based on the setting of the following bits:

- RFE bit of MAC_Rx_Flow_Ctrl register
- DM bit of MAC_Configuration register

Table 28.63 Rx MAC flow control

RFE	DM	Description
0	x	The MAC receiver does not detect the received Pause packets.
1	0	The MAC receiver does not detect the received Pause packets but recognizes such packets as Control packets.
1	1	The MAC receiver detects or processes the Pause packets and responds to such packets by stopping the MAC transmitter.

If RFE bit is enabled, the MAC transmitter is blocked when the 802.3x Pause packet is received.

The following list describes the Rx flow control:

1. The MAC checks the destination address of the received Pause packet for either of the following:
   - Multicast destination address: The DA matches the unique multicast address specified for the control packet (0x0180C2000001).
   - Unicast destination address: The DA matches the content of the MAC Address Register 0 and the UP bit of MAC_Rx_Flow_Ctrl register is set.
     If the UP bit is set and the MAC processes Pause packets with unicast destination address in addition to the unique multicast address.
2. The MAC decodes the following fields of the received packet:
   - Type field: This field is checked for 0x8808.
   - Opcode field: This field is checked for 0x0001 (Pause packet).
   - Pause Time field: The Pause time (for Pause packet) is captured to determine the time for which transmitter needs to be blocked.
3. If the byte count of the status indicates 64 bytes and there is no CRC error, the MAC transmitter does the following:
   - For 802.3x Pause packets, the MAC pauses the transmission of any data packet for the duration of the decoded Pause Time value multiplied by the slot time (64 byte times).
4. The MAC transfers the received control packet to the application based on the setting of the PCF field in MAC_Packet_Filter register.

If subsequent Pause packets are received before the earlier Pause Time expires, the MAC updates the Pause Timer with new value.

28.15.2.2 Enabling Receive Flow Control

To enable Pause Flow Control, set the RFE bit in the MAC_Rx_Flow_Ctrl register.

28.16 Loopback Mode

The MAC supports Loopback of transmitted packets to its receiver.

28.16.1 Guidelines for Using Loopback Mode

The following are some guidelines for using the loopback mode:

- Enable loopback only with the full-duplex mode. In half-duplex mode, the carrier sense signal (crs) or collision (col) signal inputs get sampled which may result into issues such as packet dropping.
- If the loopback mode is enabled without connecting a PHY chip, you should externally generate the Tx and Rx clocks and provide these clocks to the MAC.
- Do not loop back big packets. Big packets may get corrupted in the loopback FIFO.

The Transmit and Receive clocks can have an asynchronous timing relationship. Therefore, an asynchronous FIFO is used to make the loopback path of the TXD to the Receive path. The asynchronous FIFO is 10-bits (6-bits in 10/100 Mbps mode) wide to accommodate TXD, TXEN, and TXER. The depth of FIFO is five in 1000 Mbps mode and nine in 10/100 Mbps mode. The FIFO is free-running to write on the write clock and read on every read clock.

At the start of each packet read out of the FIFO, the Write and Read pointers get re-initialized to have an offset of 2 (4 in 10/100 Mbps mode). This avoids overflow or underflow during a packet transfer. This also ensures that the overflow or underflow occurs only during the IPG period between the packets. The FIFO depth of five or nine is sufficient to prevent data corruption for packet sizes up to 9,022 bytes with a difference of 200 ppm between (G)MII Transmit and Receive clock frequencies. Therefore, bigger packets should not be looped back because they may get corrupted in this loopback FIFO.

At the end of every received packet, the Receive Protocol Engine module generates received packet status and sends it to the Receive Packet Controller module. The control, missed packet, and filter fail status are added to the Receive status in the Receive Packet Controller module.

The MAC does not process ARP or PMT packets that are looped back.

28.16.2 Enabling Loopback Mode

To enable loopback feature, program the LM bit of the MAC_Configuration register.

You can enable loopback for all PHY interfaces. The data is always looped back through internal asynchronous FIFO on to the internal Receive MII or GMII interface, irrespective of which PHY interface is selected. The loopback data is also passed through the corresponding transmit PHY interface block, on to the Ethernet line.

Note:
- During loopback, the data/packet is reflected on gmii_txd signal.
- Preemption is not supported in loopback mode.

28.17 Descriptors

28.17.1 Overview

The DMA in the Ethernet subsystem transfers data based on a linked list of descriptors. The application creates the descriptors in the system memory. The GMAC supports the following two types of descriptors:

- Normal Descriptor: Normal descriptors are used for packet data and to provide control information applicable to the packets to be transmitted.
- Context Descriptor: Context descriptors are used to provide control information applicable to the packet to be transmitted.

Each normal descriptor contains two buffers and two address pointers. These buffers enable the adapter port to be compatible with various types of memory management schemes.

Note: There is no limit for the number of descriptors that can be used for a single packet.

28.17.2 Descriptor Structure

The GMAC supports the ring structure for DMA descriptor as shown in Figure 28.30.

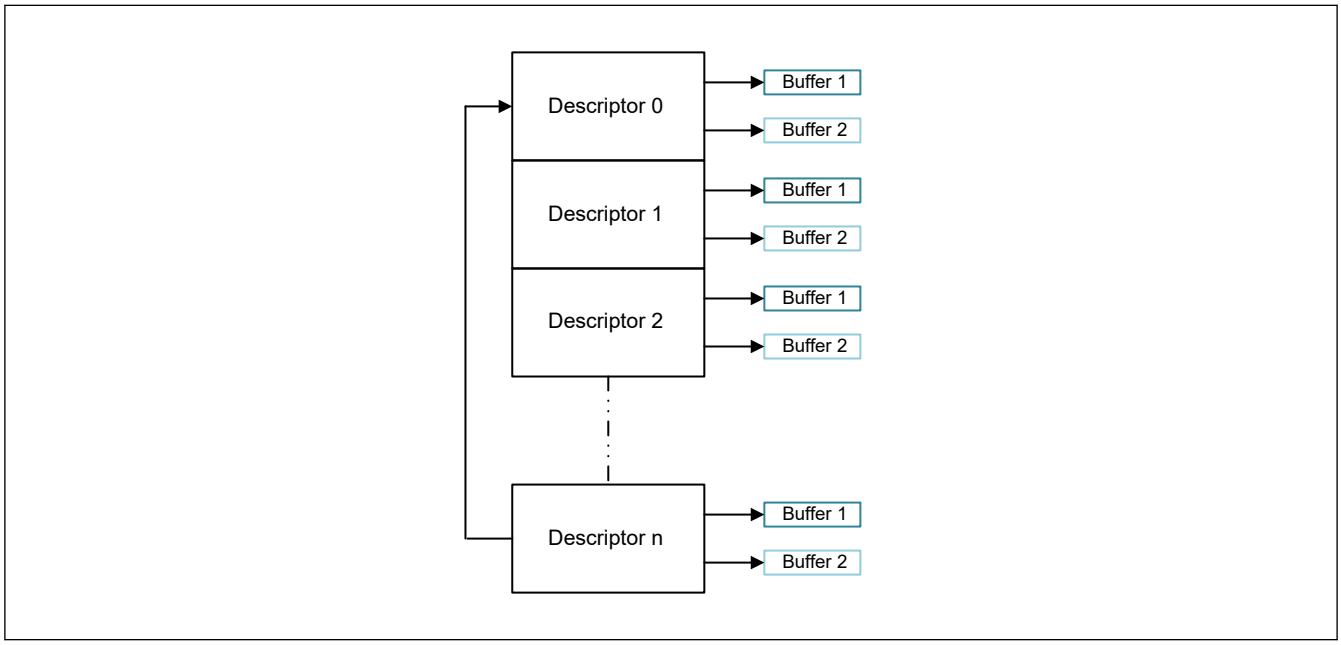


Figure 28.30 Descriptor ring structure

In Ring structure, descriptors are separated by the Word, DWord, or LWord number programmed in the DSL field of the DMA_CHn_CONTROL register. The application needs to program the total ring length, that is, the total number of descriptors in ring span in the following registers of a DMA channel:

- Transmit Descriptor Ring Length Register (DMA_CHn_TXDESC_RING_LENGTH)
- Receive Control 2 Register (DMA_CHn_RX_CONTROL2)

The Descriptor Tail Pointer Register contains the pointer to the descriptor address (N). The base address and the current descriptor pointer decide the address of the current descriptor that the DMA can process. The descriptors up to one location less than the one indicated by the descriptor tail pointer (N – 1) are owned by the DMA. The DMA continues to process the descriptors until the following condition occurs:

Current Descriptor Pointer == Descriptor Tail Pointer;

The DMA goes into the Suspend mode when this condition occurs. The application must perform a write to the Descriptor Tail pointer register and update the tail pointer so that the following condition is true:

Current Descriptor Pointer < Descriptor Tail Pointer;

The DMA automatically wraps around the base address when the end of ring is reached, as shown in Figure 28.31.

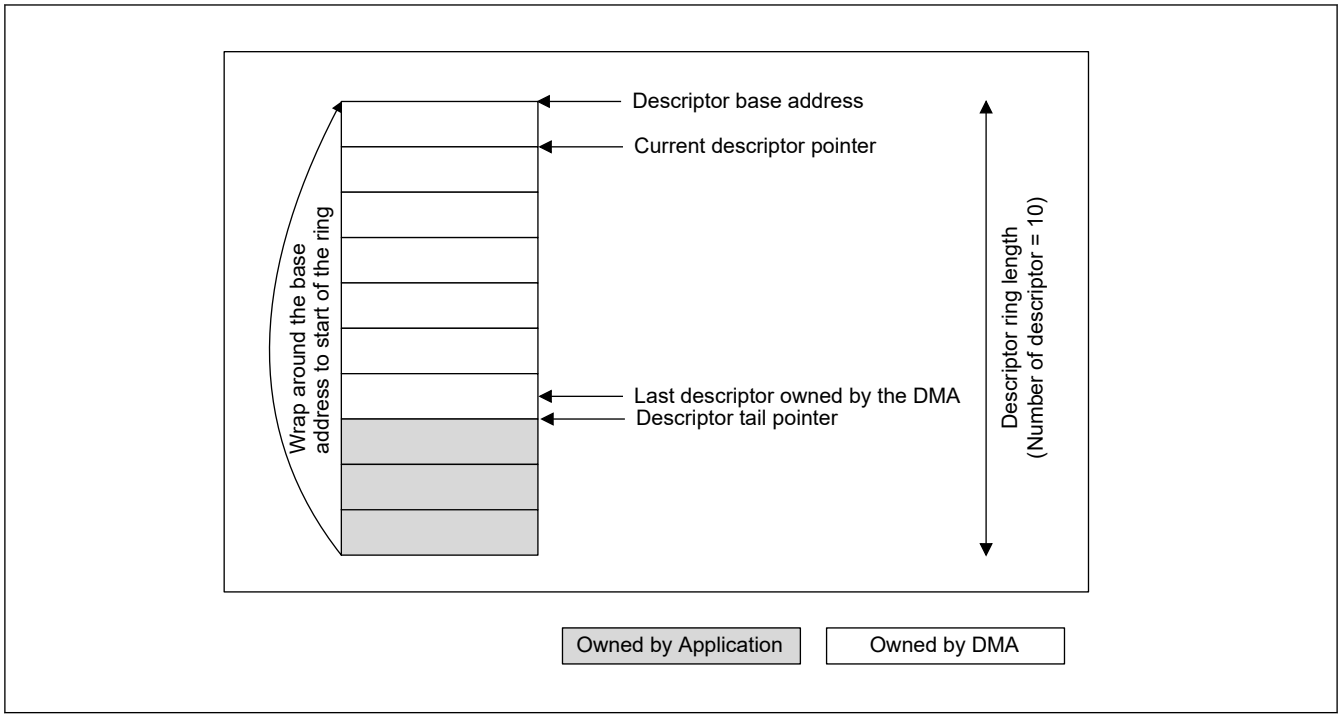


Figure 28.31 DMA descriptor ring

For descriptors owned by the application, the OWN bit of DES3 is reset to 0. For descriptors owned by the DMA, the OWN bit is set to 1. If the application has only one descriptor in the beginning, the application sets the last descriptor address (tail pointer) to Descriptor Base Address + 1. The DMA processes the first descriptor and then waits for the application to advance the tail pointer.

28.17.3 Descriptor Structure for Split Header Support

GMAC supports splitting the incoming receive packet header such that the header and the payload are stored in different buffers (buffer-1 and buffer-2) in the system memory. For details about splitting header on receive packets, see section 28.12.2.4. Header-Payload Split.

Figure 28.32 shows the descriptor structure without the Split Header feature. Figure 28.33 shows the descriptor structure with the Split Header feature.

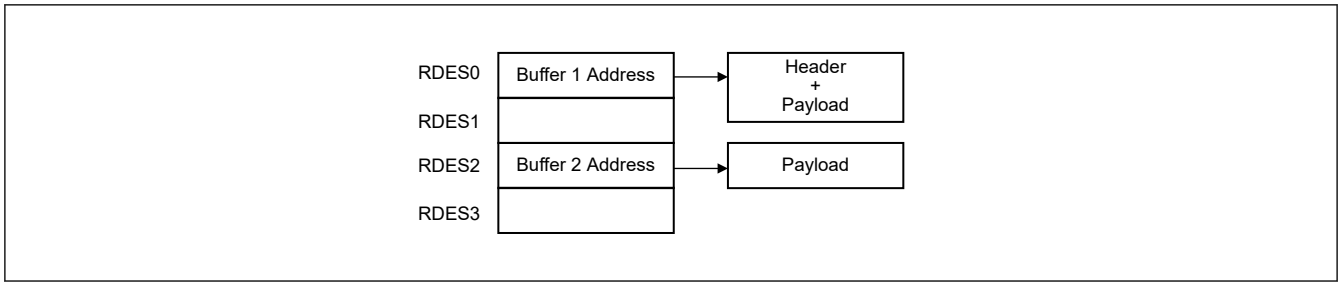


Figure 28.32 Descriptors without split header feature

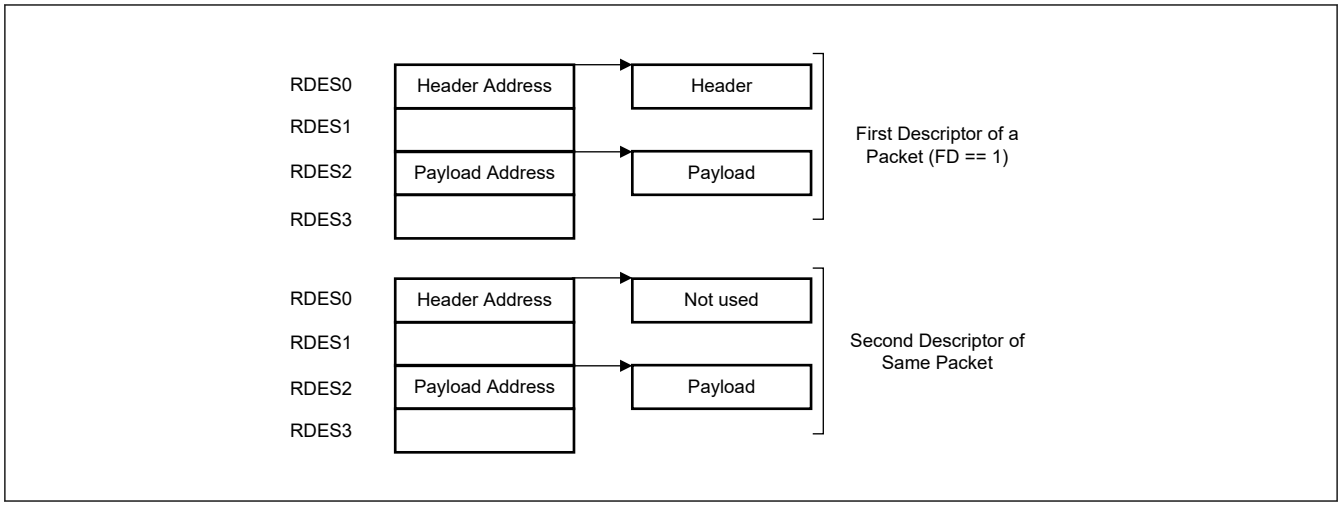


Figure 28.33 Descriptors with split header feature

The DMA writes the header of the received packet by using the header address to which the RDES0 in the first descriptor is pointing (FD bit of RDES3 is set). The DMA writes the payload of the received packet into the buffer address to which the RDES2 is pointing. For subsequent descriptors (FD is set to 0), the address to which RDES0 (Header address) is pointing is not used. The payload is written only to buffers to which the RDES2 (payload address) is pointing.

The HDSMS field of the MAC_Rx_Configuration register indicates maximum header size allowed for splitting the header data in the received packet based on the value programmed. When the ARBS field of the DMA_CHn_RX_CONTROL register is programmed to a non-zero value, ARBS field indicates Receive Buffer size for buffer-1 and the RBSZ field indicates receive buffer size for buffer-2. The ARBS field can be set to 0.

The value in the HDSMS field should be greater than or equal to the Buffer-1 size indicated by ARBS field for all the channels.

Table 28.64 lists how DMA processes a packet based on the packet type.

Table 28.64 Split header support depending on the packet type

Packet Type	SPLM	Description
TCP or UDP packet	00b (L3/L4 Split)	The DMA writes the Ethernet header + IP header + TCP or UDP header into the header buffer.
IP packet (not TCP/UDP)		The DMA writes the Ethernet header + IP header into the header buffer.
Non-IP packet		The DMA does not split the header and payload
Any packet	01b (L2 Split)	The DMA writes the Ethernet header based on split offset (SPLOFST) into the header buffer.
IP packet	10b (Combination of L2 or L3/L4 Split)	L3/L4 split
Non-IP packet		L2 split
NA	11b	Reserved

The IP header includes IPv4 options in case of IPv4 packet and IPv6 extension headers in case of IPv6 frames. The points at which the header is split are shown in Figure 28.34.

Note:
- The VLAN tag stripping must be set for the split function. For instance, the DMA separates the header and payload of an untagged packet only. So, when a tagged packet is received, the program GMAC such that the VLAN tags are deleted/stripped from the received packets.
- L3/L4 split is applicable for IP packets that are either untagged or VLAN stripped. If VLAN tag is retained in the packet forwarded to the DMA, L3-L4 split is not performed. However, if SPLM field is set to 2, L2 split is performed for VLAN tagged IP packets.
- For AV packets, to specify an alternative L2 split value, use the SAVE and SAVO fields of the MAC_Ext1_Cfg register.
- L2 Split is not supported for packets with variable preamble

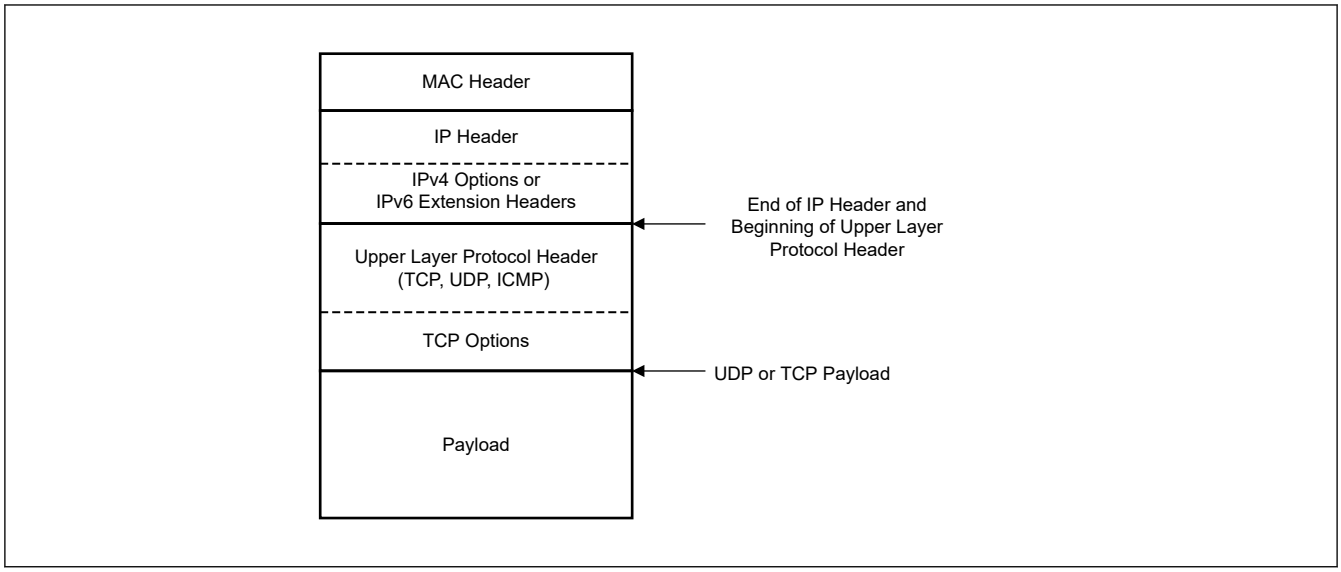


Figure 28.34 Header split points

Table 28.65 lists the header length availability in RDES2.

Table 28.65 Header Length Availability in RDES2

First Descriptor (FD) Value	Last Descriptor (LD) Value	Header Length Availability in RDES2
0	1	Header length not available
1	0	HL[9:0] for all packets
1	1	For IP packets, HL[9:0] For non-IP packets, HL[9:2]

The DMA writes the header length in RDES2 of the first receive descriptor (RDES3[29] (FD bit) is set) for the packet. The packet length is written in RDES3 of the last receive descriptor (RDES3[28] (LD bit) set).

The buffer length for the payload is set by the driver through the RBSZ field in the corresponding DMA Channel Register 2 (Receive Control Register).

The DMA fills receive buffers fully in all except the last descriptor. The header length is taken to be the value based on the bits programmed in the HDSMS field (bits [22:20]) of MAC_Extended_Configuration register.

28.17.4 Transmit Descriptor

The DMA in GMAC requires at least one descriptor for a transmit packet. In addition to two buffers, two byte-count buffers, and two address pointers, the transmit descriptor has control fields which can be used to control the MAC operation on per-transmit packet basis. The Transmit Normal descriptor has two formats: Read format and Write-Back format.

28.17.4.1 Transmit Normal Descriptor (Read Format)

Table 28.66 shows the Read Format for a Transmit normal descriptor. Table 28.67 through Table 28.70 describe the read format for the Transmit Normal Descriptors: TDES0, TDES1, TDES2, and TDES3.

Table 28.66 Transmit descriptor read format

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	Buffer 1 Address																															
TDES1	Buffer 2 Address																															
TDES2	IOC	TTSE	Buffer 2 Length														RES	Buffer 1 Length															
TDES3	OWN	CTXT	FD	LD	OPC		Reserved			SLOTNUM or THL				TSE	CIC		Reserved															

Table 28.67 TDES0 normal descriptor (read format)

Bit	Name	Description
31:0	BUF1AP	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1.

Table 28.68 TDES1 normal descriptor (read format)

Bit	Name	Description
31:0	BUF2AP	Buffer 2 Address Pointer This bit indicates the physical address of Buffer 2 when a descriptor ring structure is used. There is no limitation for the buffer address alignment.

Table 28.69 TDES2 normal descriptor (read format)

Bit	Name	Description
31	IOC	Interrupt on Completion This bit controls the setting of TI and ETI status bits in the DMA_CHn_STATUS register. When ETIC = 1 and TDES2[LD] = 0, this bit sets the ETI bit. When TDES3[LD] = 1, this bit sets the TI status bit.
30	TTSE	Transmit Timestamp Enable This bit enables the IEEE1588 time stamping for Transmit packet referenced by the descriptor, if TSE bit is not set.
29:16	B2L	Buffer 2 Length The driver sets this field. When set, this field indicates Buffer 2 length.
15:14	—	Reserved. The write value should be 0.
13:0	B1L	Buffer 1 Length For Header length only bits [9:0] are taken. The size 13:0 is applicable only when interpreting buffer 1 length.

Table 28.70 TDES3 normal descriptor (read format) (1 of 2)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit after it completes the transfer of data given in the associated buffer(s).
30	CTXT	Context Type This bit should be set to 0 for normal descriptor.
29	FD	First Descriptor When this bit is set, it indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor When this bit is set, it indicates that the buffer contains the last segment of the packet. When this bit is set, the B1L or B2L field should have a non-zero value.

Table 28.70 TDES3 normal descriptor (read format) (2 of 2)

Bit	Name	Description
27:26	CPC	CRC Pad Control This field controls the CRC and Pad Insertion for Tx packet. This field is valid only when the first descriptor bit (TDES3[29]) is set. The following list describes the values of bits [27:26]: ● 00b: CRC and Pad Insertion The MAC appends the cyclic redundancy check (CRC) at the end of the transmitted packet of length greater than or equal to 60 bytes. The MAC automatically appends padding and CRC to a packet with length less than 60 bytes. ● 01b: CRC Insertion (Disable Pad Insertion) The MAC appends the CRC at the end of the transmitted packet but it does not append padding. The application should ensure that the padding bytes are present in the packet being transferred from the Transmit Buffer, that is, the packet being transferred from the Transmit Buffer is of length greater than or equal to 60 bytes. ● 10b: Disable CRC Insertion The MAC does not append the CRC at the end of the transmitted packet. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. ● 11b: CRC Replacement The MAC replaces the last four bytes of the transmitted packet with recalculated CRC bytes. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. This field is valid only for the first descriptor. Note: When the TSE bit is set, the MAC ignores this field because the CRC and pad insertion is always done for segmentation.
25:23	—	Reserved. The write value should be 0.
22:19	SLOTNUM or THL	SLOTNUM: Slot Number Control Bits in AV Mode These bits indicate the slot interval in which the data should be fetched from the corresponding buffers addressed by TDES0 or TDES1. When the Transmit descriptor is fetched, the DMA compares the slot number value in this field with the slot interval maintained in the RSN field DMA_CHn_SLOT_FUNCTION_CONTROL_STATUS. It fetches the data from the buffers only if a value matches. These bits are valid only for the AV channels. THL: TCP/UDP Header Length If the TSE bit is set, this field contains the length of the TCP/UDP header. The minimum value of this field must be 5 for TCP header. The value must be equal to 2 for UDP header. This field is valid only for the first descriptor.
18	—	Reserved. The write value should be 0.
17:16	CIC	Checksum Insertion Control These bits control the checksum calculation and insertion. The following list describes the bit encoding: ● 00b: Checksum Insertion Disabled. ● 01b: Only IP header checksum calculation and insertion are enabled. ● 10b: IP header checksum and payload checksum calculation and insertion are enabled, but pseudo-header checksum is not calculated in hardware. ● 11b: IP Header checksum and payload checksum calculation and insertion are enabled, and pseudo-header checksum is calculated in hardware. This field is valid only for the first descriptor.
15:0	—	Reserved. The write value should be 0.

28.17.4.2 Transmit Normal Descriptor (Write-Back Format)

The write-back format of the Transmit Descriptor includes timestamp low, timestamp high, OWN, and Status bits.

The write-back format is applicable only for the last descriptor of the corresponding packet. The LD bit (TDES3[28]) is set in the descriptor where the DMA writes back the status and timestamp information for the corresponding Transmit packet.

Table 28.71 shows the write-back format of the Transmit Descriptor. Table 28.72 through Table 28.75 describe the write-back format for the Transmit Normal Descriptors: TDES0, TDES1, TDES2, and TDES3.

Table 28.71 Transmit descriptor write-back format (1 of 2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	Timestamp Low																															
TDES1	Timestamp High																															
TDES2	Reserved																															

Table 28.71 Transmit descriptor write-back format (2 of 2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES3	OWN	CTXT	FD	LD	Reserved				DE	Reserved					TTSS	RES		JT	FF	PCE	LoC	NC	LC	EC	Collision Count				ED	UF	DB	IHE

Table 28.72 TDES0 normal descriptor (write-back format)

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for the corresponding Transmit packet. The DMA writes the timestamp only if TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and the Timestamp status (TTSS) bit is set.

Table 28.73 TDES1 normal descriptor (write-back format)

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High The DMA updates this field with the most significant 32 bits of the timestamp captured for corresponding transmit packet. The DMA writes the timestamp only if the TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and Timestamp status (TTSS) bit is set.

Table 28.74 TDES2 normal descriptor (write-back format)

Bit	Name	Description
31:0	—	Reserved.

Table 28.75 TDES3 normal descriptor (Write-Back Format) (1 of 2)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the GMAC DMA owns the descriptor. The DMA clears this bit when it completes the packet transmission. After the write-back is complete, this bit is set to 0.
30	CTXT	Context Type This bit should be set to 0 for Normal descriptor.
29	FD	First Descriptor This bit indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor This bit is set to 1 for last descriptor of a packet. The DMA writes the status fields only in the last descriptor of the packet.
27:24	—	Reserved
23	DE	Descriptor Error When this bit is set, it indicates that the descriptor content is incorrect. The DMA sets this bit during write-back while closing the descriptor. Descriptor Errors can be: ● Incorrect sequence from the context descriptor. For example, a location after the first descriptor for a packet. ● All 1s ● CTXT is set to 1 along with LD or FD bits set to 1. Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA sets the TI bit in the DMA_CHn_STATUS register. Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.
22:18	—	Reserved
17	TTSS	Tx Timestamp Status This status bit indicates that a timestamp has been captured for the corresponding transmit packet. When this bit is set, TDES0 and TDES1 have timestamp values that were captured for the Transmit packet. This field is valid only when the Last Segment control bit (TDES3 [28]) in a descriptor is set. This bit is valid only when IEEE1588 timestamping feature is enabled; otherwise, it is reserved.

Table 28.75 TDES3 normal descriptor (Write-Back Format) (2 of 2)

Bit	Name	Description
16:15	—	Reserved
14	JT	Jabber Timeout This bit indicates that the MAC transmitter has experienced a jabber time-out. This bit is set only when the JD bit of the MAC_Configuration register is not set.
13	FF	Packet Flushed This bit indicates that the DMA or MTL flushed the packet because of a software flush command given by the CPU.
12	PCE	Payload Checksum Error This bit indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either because of insufficient bytes, as indicated by the Payload Length field of the IP Header or the MTL starting to forward the packet to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet packet being transmitted to avoid deadlock, the MTL starts forwarding the packet when the FIFO is full, even in the store-and-forward mode. This error can also occur when Bus Error is detected during packet transfer.
11	LoC	Loss of Carrier This bit indicates that Loss of Carrier occurred during packet transmission (that is, the carrier sense signal (CRS) was inactive for one or more transmit clock periods during packet transmission). This is valid only for the packets transmitted without collision and when the MAC operates in the half-duplex mode.
10	NC	No Carrier This bit indicates that the carrier sense signal (CRS) form the PHY was not asserted during transmission.
9	LC	Late Collision This bit indicates that packet transmission was aborted because a collision occurred after the collision window (64 byte times including Preamble in MII mode and 512 byte times including Preamble and Carrier Extension in GMII mode). This bit is not valid if Underflow Error is set.
8	EC	Excessive Collision This bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the MAC_Configuration register, this bit is set after first collision and the transmission of the packet is aborted.
7:4	CC	Collision Count This 4-bit counter value indicates the number of collisions occurred before the packet was transmitted. The count is not valid when the EC bit is set.
3	ED	Excessive Deferral This bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000 Mbps mode or Jumbo Packet enabled mode) if DC bit is set in the MAC_Configuration register. When TBS is enabled in full duplex mode and this bit is set, it indicates that the frame has been dropped after the expiry time has reached.
2	UF	Underflow Error This bit indicates that the MAC aborted the packet because the data arrived late from the system memory. The underflow error can occur because of either of the following conditions: ● The DMA encountered an empty Transmit Buffer while transmitting the packet ● The application filled the MTL Tx FIFO slower than the MAC transmit rate The transmission process enters the suspended state and sets the underflow bit corresponding to a queue in the MTL_Interrupt_Status register.
1	DB	Deferred Bit This bit indicates that the MAC deferred before transmitting because of presence of carrier. This bit is valid only in the half-duplex mode.
0	IHE	IP Header Error When IP Header Error is set, this bit indicates that the Checksum Offload engine detected an IP header error. If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload. In full duplex mode, when EST/Qbv is enabled and this bit is set, it indicates the frame drop status due to Frame Size error or Schedule Error.

28.17.4.3 Transmit Context Descriptor

The context descriptor is used to provide the timestamps for one-step timestamp correction, VLAN Tag ID for VLAN insertion feature.

The Transmit Context descriptor can be provided any time before a packet descriptor. The context is valid for the current packet and subsequent packets. The context descriptor is used to provide the timestamps for one-step timestamp correction and VLAN Tag ID for VLAN insertion feature. Write back is done on a context descriptor only to reset the OWN bit.

Note: The VLAN Tag IDs and MSS values, provided by the application in a context descriptor with their corresponding Valid bits set, are stored internally by the DMA. When the outer or inner VLAN tag is provided with the Valid bit set, the DMA always passes the last valid VLAN tag to the MTL. The application cannot invalidate the valid VLAN tag stored by the DMA. The VLAN tag is inserted or replaced based on the control inputs provided for the packet. The Inner VLAN Tag Control input is used only for the next packet that immediately follows the context descriptor. The application must provide a context descriptor before the normal descriptor of each packet for which the DMA should use the inner VLAN Tag control input.

Table 28.76 shows the format of the Transmit Context descriptor. Table 28.77 through Table 28.80 describe the format for the Transmit Context Descriptors: TDES0, TDES1, TDES2, and TDES3.

Table 28.76 Transmit context descriptor format

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	Timestamp Low																															
TDES1	Timestamp High																															
TDES2	Reserved																															
TDES3	OWN	CTXT	RES		OSTC	TCV	RES		CDE	Reserved																						

Table 28.77 TDES0 context descriptor

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low For one-step correction, the driver can provide the lower 32 bits of timestamp in this descriptor word. The DMA uses this value as the low word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

Table 28.78 TDES1 context descriptor

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High For one-step correction, the driver can provide the upper 32 bits of timestamp in this descriptor. The DMA uses this value as the high word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

Table 28.79 TDES2 context descriptor

Bit	Name	Description
31:0	—	Reserved

Table 28.80 TDES3 context descriptor (1 of 2)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the GMAC DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit immediately after the read.
30	CTXT	Context Type This bit should be set to 1 for Context descriptor.
29:28	—	Reserved
27	OSTC	One-Step Timestamp Correction Enable When this bit is set, the DMA performs a one-step timestamp correction with reference to the timestamp values provided in TDES0 and TDES1.
26	TCV	One-Step Timestamp Correction Input or MSS Valid When this bit and the OSTC bit are set, it indicates that the Timestamp Correction input provided in TDES0 and TDES1 is valid.

Table 28.80 TDES3 context descriptor (2 of 2)

Bit	Name	Description
25:24	—	Reserved
23	CDE	Context Descriptor Error When this bit is set, it indicates that the descriptor content is incorrect. The DMA sets this bit during write-back while closing the context descriptor. Context Descriptor Errors can be: ● Incorrect sequence from the context descriptor. For example, a location before the first descriptor for a packet. ● All 1s. ● CD, LD, and FD bits set to 1. Note 1: When the Context Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA sets the TI bit in the DMA_CHn_STATUS register Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent. Note3: This error is categorized as an abnormal event; recovery is only by issuing a software reset (DMA stopping-reconfiguring-restarting recovery mechanism is not supported)
22:0	—	Reserved

28.17.5 Receive Descriptor

The DMA in GMAC attempts to read a descriptor only if the Tail Pointer is different from the Base Pointer or current pointer. It is recommended to have a descriptor ring with a length that can accommodate at least two complete packets received by the MAC. Otherwise, the performance of the DMA is impacted greatly because of the unavailability of the descriptors. In such situations, the RxFIFO in MTL becomes full and starts dropping packets.

The following Receive Descriptors are present:

- Normal descriptors
- Context descriptors

All RX descriptors are prepared by the software and given to the DMA as "Normal" Descriptors with the content as shown in Receive Normal Descriptor (Read Format). The DMA reads this descriptor and after transferring a received packet (or part of) to the buffers indicated by the descriptor, the Rx DMA closes the descriptor with the corresponding packet status. The format of this status is given in the section 28.17.5.2. Receive Normal Descriptor (Write-Back Format).

For some packets, the normal descriptor bits are not enough to write the complete status. For such packets, the RX DMA writes the extended status to the next descriptor (without processing or using the Buffers/ Pointers embedded in that descriptor). The format and content of the descriptor write back is described in section 28.17.5.3. Receive Context Descriptor.

28.17.5.1 Receive Normal Descriptor (Read Format)

The read format for a Receive Normal descriptor is made up of a header or Buffer 1 address, reserved field, payload or Buffer 2 or Next Descriptor address, a 30-bit reserved filed, OWN bit, and an interrupt bit.

Table 28.81 shows the Read Format for a Receive Normal Descriptor.

Table 28.81 Receive normal descriptor read format

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDES0	Header or Buffer 1 Address																															
RDES1	Reserved																															
RDES2	Payload or Buffer2 or Next Descriptor Address																															
RDES3	OWN	IOC	Reserved				BUF2V	BUF1V	Reserved																							

Note: In the Receive Descriptor (Read Format), if the Buffer Address field is all 0s, GMAC does not transfer data to that buffer and skips to the next buffer or next descriptor.

Table 28.82 RDES0 normal descriptor (read format)

Bit	Name	Description
31:0	BUF1AP	Header or Buffer 1 Address Pointer When the SPH bit of DMA_CHn_CONTROL register of a channel is reset, these bits indicate the physical address of Buffer 1. When the SPH bit is set, these bits indicate the physical address of Header Buffer where the Rx DMA writes the L2/L3/L4 header bytes of the received packet. The application can program a byte-aligned address for this buffer which means that the LS bits of this field can be non-zero. However, while transferring the start of packet, the DMA performs a Write operation with RDES0[3:0] as zero. However, the packet data is shifted as per actual offset as given by buffer address pointer. If the address pointer points to a buffer where the middle or last part of the packet is stored, the DMA ignores the offset address and writes to the full location as indicated by the data-width.

Table 28.83 RDES1 normal descriptor (read format)

Bit	Name	Description
31:0	—	Reserved

Table 28.84 RDES2 normal descriptor (read format)

Bit	Name	Description
31:0	BUF2AP	Payload or Buffer 2 Address or Next Descriptor Pointer These bits indicate the physical address of Buffer 2. When the SPH bit of the DMA_CHn_CONTROL register is set, the buffer address pointer must be bus width-aligned, that is, BUF2AP[3:0] = 0x0. LSBs are ignored internally. When the SPH bit of the DMA_CHn_CONTROL register is reset, there is no limitations on the RDES2 value. However, the RxDMA uses the LS Bits of the pointer address only while transferring the start bytes of a packet. If the BUF2AP is giving the address of a buffer in which the middle or last part of a packet is stored, the DMA ignores BUF2AP[3:0] and writes to the complete location.

Table 28.85 RDES3 normal descriptor (read format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the GMAC DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: • The DMA completes the packet reception • The buffers associated with the descriptor are full
30	IOC	Interrupt Enabled on Completion When this bit is set, an interrupt is issued to the application when the DMA closes this descriptor.
29:26	—	Reserved
25	BUF2V	Buffer 2 Address Valid When this bit is set, it indicates to the DMA that the buffer 2 address specified in RDES2 is valid. The application must set this bit so that the DMA can use the address, to which the Buffer 2 address in RDES2 is pointing, to write received packet data.
24	BUF1V	Buffer 1 Address Valid When set, this indicates to the DMA that the buffer 1 address specified in RDES1 is valid. The application must set this value if the address pointed to by Buffer 1 address in RDES1 can be used by the DMA to write received packet data.
23:0	—	Reserved

28.17.5.2 Receive Normal Descriptor (Write-Back Format)

Table 28.86 shows the write-back format for a Receive Normal descriptor.

Table 28.86 Receive normal descriptor (write-back format) (1 of 2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDES0	Inner VLAN Tag																Outer VLAN Tag															
RDES1	OAM Sub-Type Code or MAC Control Packet Opcode																TD	TSA	PV	PFT	PTP Message Type				IPCE	IPCB	IPV6	IPV4	IPHE	Payload Type		

Table 28.86 Receive normal descriptor (write-back format) (2 of 2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDES2	L3L4FM			L4FM	L3FM	MAC Address Match or Hash Value								HF	DAF	SAF	OTS	ITS	Reserved			ARPNR	L3/L4 Header Length									
RDES3	OWN	CTXT	FD	LD	RS2V	RS1V	RS0V	CE	GP	RWT	OE	RE	DE	Length / Type Field			ES	Packet Length														

Table 28.87 RDES0 normal descriptor (write-back format)

Bit	Name	Description
31:16	IVT	Inner VLAN Tag This field contains the Inner VLAN tag of the received packet if the RS0V bit of RDES3 is set. This is valid when VLAN tag stripping are enabled.
15:0	OVT	Outer VLAN Tag This field contains the Outer VLAN tag of the received packet if the RS0V bit of RDES3 is set.

Table 28.88 RDES1 normal descriptor (write-back format) (1 of 2)

Bit	Name	Description
31:16	OPC	OAM Sub-Type Code, or MAC Control Packet opcode OAM Sub-Type Code If bits [18:16] of RDES3 are set to 111b, this field contains the OAM sub-type and code fields. MAC Control Packet opcode If bits [18:16] of RDES3 are set to 110b, this filed contains opcode.
15	TD	Timestamp Dropped This bit indicates that the timestamp was captured for this packet but it got dropped in the MTL Rx FIFO because of overflow.
14	TSA	Timestamp Available When Timestamp is present, this bit indicates that the timestamp value is available in a context descriptor word 1 (RDES1) and word 0 (RDES0). This is valid only when the Last Descriptor bit (RDES3 [28]) is set. The context descriptor is written in the next descriptor just after the last normal descriptor for a packet.
13	PV	PTP Version This bit indicates that the received PTP message has the IEEE 1588 version 2 format. When this bit is reset, it indicates the IEEE 1588 version 1 format.
12	PFT	PTP Packet Type This bit indicates that the PTP message is sent directly over Ethernet.
11:8	PMT	PTP Message Type These bits are encoded to give the type of the message received: ● 0x0: No PTP message received ● 0x1: SYNC (all clock types) ● 0x2: Follow_Up (all clock types) ● 0x3: Delay_Req (all clock types) ● 0x4: Delay_Resp (all clock types) ● 0x5: Pdelay_Req (in peer-to-peer transparent clock) ● 0x6: Pdelay_Resp (in peer-to-peer transparent clock) ● 0x7: Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock) ● 0x8: Announce ● 0x9: Management ● 0xA: Signaling ● 0xB–0xE: Reserved ● 0xF: PTP packet with Reserved message type
7	IPCE	IP Payload Error When this bit is set, it indicates either of the following: ● The 16-bit IP payload checksum (that is, the TCP, UDP, or ICMP checksum) calculated by the MAC does not match the corresponding checksum field in the received segment. ● The TCP, UDP, or ICMP segment length does not match the payload length value in the IP Header field. ● The TCP, UDP, or ICMP segment length is less than minimum allowed segment length for TCP, UDP, or ICMP. Bit 15 (ES) of RDES3 is not set when this bit is set.
6	IPCB	IP Checksum Bypassed This bit indicates that the checksum offload engine is bypassed.

Table 28.88 RDES1 normal descriptor (write-back format) (2 of 2)

Bit	Name	Description
5	IPV6	IPv6 header Present This bit indicates that an IPV6 header is detected. When the SPH bit of the DMA_CHn_CONTROL register is set, the IPV6 header is available in the header buffer area to which RDES0 is pointing.
4	IPV4	IPV4 Header Present This bit indicates that an IPV4 header is detected. When the SPH bit of the DMA_CHn_CONTROL register is set, the IPV4 header is available in the header buffer area to which RDES0 is pointing.
3	IPHE	IP Header Error When this bit is set, it indicates either of the following: ● The 16-bit IPv4 header checksum calculated by the MAC does not match the received checksum bytes. ● The IP datagram version is not consistent with the Ethernet Type value. ● Ethernet packet does not have the expected number of IP header bytes. This bit is valid when either bit 5 or bit 4 is set.
2:0	PT	Payload Type These bits indicate the type of payload encapsulated in the IP datagram processed by the Receive Checksum Offload Engine (COE): ● 000b: Unknown type or IP/AV payload not processed ● 001b: UDP ● 010b: TCP ● 011b: ICMP ● 110b: AV Tagged Data Packet ● 111b: AV Tagged Control Packet ● 101b: AV Untagged Control Packet ● 100b: IGMP if IPV4 Header Present bit is set else DCB (LLDP) Control Packet If the COE does not process the payload of an IP datagram because there is an IP header error or fragmented IP, it sets these bits to 000b.

Table 28.89 RDES2 normal descriptor (write-back format) (1 of 2)

Bit	Name	Description
31:29	L3L4FM	Layer 3 and Layer 4 Filter Number Matched These bits indicate the number of the Layer 3 and Layer 4 Filter that matched the received packet: ● 000b: Filter 0 ● 001b: Filter 1 ● 010b: Filter 2 ● 011b: Filter 3 ● 100b: Filter 4 ● 101b: Filter 5 ● 110b: Filter 6 ● 111b: Filter 7 This field is valid only when bit 28 or bit 27 is set high. When more than one filter matches, these bits give the number of lowest filter.
28	L4FM	Layer 4 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 4 Port Number fields. This status is given only when one of the following conditions is true: ● Layer 3 fields are not enabled and all enabled Layer 4 fields match ● All enabled Layer 3 and Layer 4 filter fields match When more than one filter matches, this bit gives the layer 4 filter status of filter indicated by bits [31:29].
27	L3FM	Layer 3 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 3 IP Address fields. This status is given only when one of the following conditions is true: ● All enabled Layer 3 fields match and all enabled Layer 4 fields are bypassed ● All enabled filter fields match When more than one filter matches, this bit gives the layer 3 filter status of filter indicated by bits [31:29].
26:19	MADRM	MAC Address Match or Hash Value When the HF bit is reset, this field contains the MAC address register number that matched the Destination address of the received packet. This field is valid only if the DAF bit is reset. When the HF bit is set, this field contains the hash value computed by the MAC. A packet passes the hash filter when the bit corresponding to the hash value is set in the hash filter register.
18	HF	Hash Filter Status When this bit is set, it indicates that the packet passed the MAC address hash filter. Bits [26:19] indicate the hash value.

Table 28.89 RDES2 normal descriptor (write-back format) (2 of 2)

Bit	Name	Description
17	DAF	Destination Address Filter Fail When this bit is set, it indicates that the packet failed the DA Filter in the MAC.
16	SAF	Source Address Filter Fail When this bit is set, it indicates that the packet failed the SA Filter in the MAC.
15	OTS	VLAN Filter Status The bit is redefined as Outer VLAN Tag Filter Status (OTS). This bit is valid for both Single and Double VLAN Tagged frames
14	ITS	Inner VLAN Tag Filter Status (ITS) This bit is valid only for Double VLAN Tagged frames. For more information, see the Filter Status topic.
13:11	—	Reserved
10	ARPNR	ARP Reply Not Generated When this bit is set, it indicates that the MAC did not generate the ARP Reply for received ARP Request packet. This bit is set when the MAC is busy transmitting ARP reply to earlier ARP request (only one ARP request is processed at a time).
9:0	HL	L3/L4 Header Length This field contains the length of the header of the packet split by the MAC at L3 or L4 header boundary as identified by the MAC receiver. This field is valid only when the first descriptor bit is set (FD = 1). The header data is written to the Buffer 1 address of corresponding descriptor. If header length is zero, this field is not valid. It implies that the MAC did not identify and split the header.

Table 28.90 RDES3 normal descriptor (write-back format) (1 of 2)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the GMAC DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: ● The DMA completes the packet reception ● The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor When this bit is set, it indicates that the current descriptor is a context type descriptor. The DMA writes 0 to this bit for normal receive descriptor. When CTXT and FD bits are used together, {CTXT, FD} ● 00b: Intermediate Descriptor ● 01b: First Descriptor ● 10b: Reserved ● 11b: Descriptor Error (due to all 1s) Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data. Receive DMA sets the CDE field of the DMA_CHn_STATUS register but does not the RI field even when IOC field is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	FD	First Descriptor When this bit is set, it indicates that this descriptor contains the first buffer of the packet. If the size of the first buffer is 0, the second buffer contains the beginning of the packet. If the size of the second buffer is also 0, the next descriptor contains the beginning of the packet. See the CTXT bit description for details of using the CTXT bit and FD bit together.
28	LD	Last Descriptor When this bit is set, it indicates that the buffers to which this descriptor is pointing are the last buffers of the packet.
27	RS2V	Receive Status RDES2 Valid When this bit is set, it indicates that the status in RDES2 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
26	RS1V	Receive Status RDES1 Valid When this bit is set, it indicates that the status in RDES1 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
25	RS0V	Receive Status RDES0 Valid When this bit is set, it indicates that the status in RDES0 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.

Table 28.90 RDES3 normal descriptor (write-back format) (2 of 2)

Bit	Name	Description
24	CE	CRC Error When this bit is set, it indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received packet. This field is valid only when the LD bit of RDES3 is set.
23	GP	Giant Packet When this bit is set, it indicates that the packet length exceeds the specified maximum Ethernet size of 1518, 1522, or 2000 bytes (9018 or 9022 bytes if jumbo packet enable is set). Note: Giant packet indicates only the packet length. It does not cause any packet truncation.
22	RWT	Receive Watchdog Timeout When this bit is set, it indicates that the Receive Watchdog Timer has expired while receiving the current packet. The current packet is truncated after watchdog timeout.
21	OE	Overflow Error When this bit is set, it indicates that the received packet is damaged because of buffer overflow in Rx FIFO. Note: This bit is set only when the DMA transfers a partial packet to the application. This happens only when the Rx FIFO is operating in the threshold mode. In the store-and-forward mode, all partial packets are dropped completely in Rx FIFO.
20	RE	Receive Error When this bit is set, it indicates that the RXER signal is asserted while the RXDV signal is asserted during packet reception. This error also includes carrier extension error in the GMII and half-duplex mode. Error can be of less or no extension, or error (rxd != 0f) during extension.
19	DE	Dribble Bit Error When this bit is set, it indicates that the received packet has a non-integer multiple of bytes (odd nibbles). This bit is valid only in the MII Mode.
18:16	LT	Length/Type Field This field indicates if the packet received is a length packet or a type packet. The encoding of the 3 bits is as follows: ● 000b: The packet is a length packet ● 001b: The packet is a type packet. ● 011b: The packet is a ARP Request packet type ● 100b: The packet is a type packet with VLAN Tag ● 101b: The packet is a type packet with Double VLAN Tag ● 110b: The packet is a MAC Control packet type ● 111b: The packet is a OAM packet type ● 010b: Reserved
15	ES	Error Summary When this bit is set, it indicates the logical OR of the following bits: ● RDES3[24]: CRC Error ● RDES3[19]: Dribble Error ● RDES3[20]: Receive Error ● RDES3[22]: Watchdog Timeout ● RDES3[21]: Overflow Error ● RDES3[23]: Giant Packet This field is valid only when the LD bit of RDES3 is set.
14:0	PL	Packet Length These bits indicate the byte length of the received packet that was transferred to system memory (including CRC). This field is valid when the LD bit of RDES3 is set and Overflow Error bits are reset. The packet length also includes the two bytes appended to the Ethernet packet when IP checksum calculation is enabled and the received packet is not a MAC control packet. This field is valid when the LD bit of RDES3 is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current packet.

28.17.5.3 Receive Context Descriptor

This descriptor is read-only for the application. Only the DMA can write to this descriptor. The context descriptor provides information about the extended status related to the last received packet. The bit 30 of RDES3 indicates the context type descriptor.

Table 28.91 shows the format for a Receive Context Descriptor.

Table 28.91 Receive context descriptor format

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDES0	Timestamp Low																															
RDES1	Timestamp High																															
RDES2	Reserved																															
RDES3	OWN	CTXT	DE	Reserved																												

Table 28.92 RDES0 context descriptor

Bit	Name	Description
31:0	RTSL	Receive Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for corresponding receive packet. When this field and the RTSH field of RDES1 show all-ones value, the timestamp must be considered as corrupt.

Table 28.93 RDES1 context descriptor

Bit	Name	Description
31:0	RTSH	Receive Packet Timestamp High The DMA updates this field with most significant 32 bits of the timestamp captured for corresponding receive packet. When this field and the RTSL field of RDES0 show all-ones value, the timestamp must be considered as corrupt.

Table 28.94 RDES2 context descriptor

Bit	Name	Description
31:0	—	Reserved

Table 28.95 RDES3 context descriptor

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: ● The DMA completes the packet reception ● The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor When this bit is set, it indicates that the current descriptor is a context descriptor. The DMA writes 1 to this bit for context descriptor. When CTXT and DE bits are used together, {CTXT, DE} ● 00b: Reserved ● 01b: Reserved ● 10b: Context Descriptor ● 11b: Descriptor Error Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data. Receive DMA sets the CDE bit in DMA_CHn_STATUS register but does not set the RI field even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	DE	Descriptor Error See the CTXT bit description for details of using the DE bit along with CTXT bit.
28:0	—	Reserved

28.17.6 Enhanced Descriptor for Time-Based Scheduling

The time-based scheduling feature needs Enhanced Descriptors (that are 32 bytes) to be enabled on all the DMA channels that intend to use the feature (by setting the EDSE bit of DMA_CHn_TX_CONTROL register).

The structure of 32-byte Descriptor for the Context and the Normal Descriptor in Read and Write formats are described in the following sections:

28.17.6.1 Enhanced Normal Descriptor - Read

Table 28.96 Enhanced normal descriptor - read

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETDES4	LTV	Reserved																			GCL Slot Number				Launch Time [31:24]							
ETDES5	Launch Time [23:0]																								Reserved							
ETDES6	Reserved																															
ETDES7	Reserved																															
TDES0	Header or Buffer 1 Address																															
TDES1	Buffer 2 Address																															
TDES2	IOC	TTSE/TMWD	Buffer 2 Length														VTIR		Header or Buffer 1 Length													
TDES3	OWN	CTXT	FD	LD	OPC		SAIC		SLOTNUM or THL					TSE	CIC		RES	Frame Length													
	TCP Payload Length																															

Table 28.97 ETDES4 enhanced normal descriptor - read

Bit	Name	Description
31	LTV	Launch Time Valid indicates the Launch Time (LT) and GSN fields present in the descriptor are valid. The LTV must be set only if the FD bit of the descriptor is not set.
30:12	—	Reserved
11:8	GSN	GCL slot number associated with the packet
7:0	LT	Launch Time [31:24] associated with the packet

Table 28.98 ETDES5 enhanced normal descriptor - read

Bit	Name	Description
31:8	LT	Launch Time [23:0] associated with the packet
7:0	—	Reserved

Table 28.99 ETDES6 enhanced normal descriptor - read

Bit	Name	Description
31:0	—	Reserved

Table 28.100 ETDES7 enhanced normal descriptor - read

Bit	Name	Description
31:0	—	Reserved

For details of the other fields, see section 28.17.4.1. Transmit Normal Descriptor (Read Format).

28.17.6.2 Enhanced Normal Descriptor - Write

Table 28.101 Enhanced normal descriptor - write (1 of 2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETDES4	LTV	Reserved																			GCL Slot Number				Launch Time [31:24]							
ETDES5	Launch Time [23:0]																								Reserved							
ETDES6	Reserved																															

Table 28.101 Enhanced normal descriptor - write (2 of 2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETDES7	Reserved																															
TDES0	Timestamp Low																															
TDES1	Timestamp High																															
TDES2	Reserved																															
TDES3	OWN	CTXT	FD	LD	Reserved				DE	Reserved					TTSS	EUE	ES	JT	FF	PCE	LoC	NC	LC	EC	Collision Count				ED	UF	DB	IHE

For details of LTV, GSN and LT fields, see section 28.17.6.1. Enhanced Normal Descriptor - Read.

For details of the other fields, see section 28.17.4.2. Transmit Normal Descriptor (Write-Back Format).

Note:
- In both the write back formats, no modifications can be done to the extended 16 bytes. The rest of the 16 bytes (TDES0 to TDES3) are written back as per the previous 16 bytes Descriptor format.
- Fetch time when enabled overrides the AV slot function
- When an unaligned new GCL list (with a different CTR) is installed, it is recommended not to have traffic during the installation of the new list. Any traffic during the switching of the lists might have unpredictable behavior regarding Fetch, Launch, and Launch expiry as the CTR and BTR values get updated while the frame is being processed

28.17.6.3 Enhanced Context Descriptor - Read

Table 28.102 Enhanced context descriptor - read

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETDES4	Reserved																															
ETDES5	Reserved																															
ETDES6	Reserved																															
ETDES7	Reserved																															
TDES0	Timestamp Low																															
TDES1	Timestamp High																															
TDES2	Inner VLAN Tag																RES		Maximum Segment Size													
TDES3	OWM	CTXT	RES		OSTC	TCMSSV	RES		CDE	RES			IVTIR		IVLTV	VLTV	VLAN Tag															

For details of the other fields, see section 28.17.4.3. Transmit Context Descriptor.

28.17.6.4 Enhanced Context Descriptor - Write

Table 28.103 Enhanced context descriptor - write

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETDES4	Reserved																															
ETDES5	Reserved																															
ETDES6	Reserved																															
ETDES7	Reserved																															
TDES0	Reserved																															
TDES1	Reserved																															
TDES2	Reserved																															
TDES3	OWN	CTXT	Reserved						CDE	Reserved																							

For details of the other fields, see section 28.17.4.3. Transmit Context Descriptor.

28.18 Programming

28.18.1 Initializing DMA

Complete the following steps to initialize the DMA:

1. Provide a software reset. This resets all of the MAC internal registers and logic (bit-0 of DMA_Mode).
2. Wait for the completion of the reset process (poll bit 0 of the DMA_Mode, which is only cleared after the reset operation is completed).
3. Program the following fields to initialize the DMA_SysBus_Mode register:
   (a) AAL
   (b) Fixed burst or undefined burst
   (c) OSR_LMT
   (d) If fixed length value is enabled, select the maximum burst length possible on the AXI Bus (bits [3:1])
4. Create a descriptor list for transmit and receive. In addition, ensure that the descriptors are owned by DMA (set bit 31 of descriptor TDES3/RDES3). For more information about descriptors, see section 28.17. Descriptors.
5. Program the Transmit and Receive Ring length registers (DMA_CHn_TXDESC_RING_LENGTH (n = 0 to 7) and DMA_CHn_RxDesc_Ring_Length (n = 0 to 7). The ring length programmed must be at least 4.
   The descriptor address from the start to the end of the ring must not cross the 4 GB boundary.
6. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (DMA_CHn_TXDESC_LIST_ADDRESS (n = 0 to 7) and DMA_CHn_RXDESC_LIST_ADDRESS (n = 0 to 7)). Also, program transmit and receive tail pointer registers indicating to the DMA about the available descriptors (DMA_CHn_TXDESC_TAIL_POINTER (n = 0 to 7) and DMA_CHn_RXDESC_TAIL_POINTER (n = 0 to 7)).
7. Program the settings of the following registers for the parameters like maximum burst-length (PBL) initiated by DMA, descriptor skip lengths, OSP in case of Tx DMA, RBSZ in case of Rx DMA, and so on:
   - DMA_CHn_CONTROL (n = 0 to 7)
   - DMA_CHn_TX_CONTROL (n = 0 to 7)
   - DMA_CHn_RX_CONTROL (n = 0 to 7)
8. Enable the interrupts by programming the DMA_CHn_INTERRUPT_ENABLE (n = 0 to 7) register.
9. Start the Receive and Transmit DMAs by setting SR (bit 0) of the DMA_CHn_RX_CONTROL (n = 0 to 7) and ST (bit 0) of the DMA_CHn_TX_CONTROL (n = 0 to 7) register.
10. Repeat steps 4 to 9 for all the Tx DMA and Rx DMA channels selected in the hardware.

28.18.2 Initializing MTL Registers

The Transaction Layer (MTL) registers must be initialized to establish the transmit and receive operating modes and commands.

Complete the following steps to initialize the MTL registers:

1. Program the Tx Scheduling (SCHALG) and Receive Arbitration Algorithm (RAA) fields in MTL_Operation_Mode to initialize the MTL operation in case of multiple Tx and Rx queues.
2. Program the Receive Queue to DMA mapping in MTL_RxQ_DMA_Map0 and MTL_RxQ_DMA_Map1 registers.
3. Program the following fields to initialize the mode of operation in the MTL_TxQ0_Operation_Mode register.
   (a) Transmit Store and Forward (TSF) or Transmit Threshold Control (TTC) in case of threshold mode
   (b) Transmit Queue Enable (TXQEN) to value 10b to enable Transmit Queue0
   (c) Transmit Queue Size (TQS)
4. Program the following fields to initialize the mode of operation in the MTL_RxQ0_Operation_Mode register:
   (a) Receive Store and Forward (RSF) or RTC in case of Threshold mode
   (b) Flow Control Activation and De-activation thresholds for MTL Receive FIFO (RFA and RFD)
   (c) Error Packet and undersized good Packet forwarding enable (FEP and FUP)

(d) Receive Queue Size (RQS)

5. Repeat previous two steps for all MTL Tx and Rx queues.

28.18.3 Initializing MAC

The MAC configuration registers establish the operating mode of the MAC. These registers must be initialized before initializing the DMA.

The following MAC Initialization operations can be performed after DMA initialization. If the MAC initialization is completed before the DMA is configured, enable the MAC receiver (last step in the following sequence) only after the DMA is active. Otherwise, received frames fill the Rx FIFO and overflow.

1. Provide the MAC address registers: MAC_Address0_High and MAC_Address0_Low. If more than one MAC address is enabled, program the MAC addresses appropriately.
2. Program the following fields to set the appropriate filters for the incoming frames in the MAC_Packet_Filter register:
   (a) Receive All
   (b) Promiscuous mode
   (c) Hash or Perfect Filter
   (d) Unicast, multicast, broadcast, and control frames filter settings
3. Program the following fields for proper flow control in the MAC_Q0_Tx_Flow_Ctrl register:
   (a) Pause time and other Pause frame control bits
   (b) Transmit Flow control bits
   (c) Flow Control Busy
4. Program the MAC_Interrupt_Enable register, as required.
5. Program the appropriate fields in the MAC_Configuration register. For example: Inter-packet gap while transmission and jabber disable.
6. Set bit 0 and 1 in the MAC_Configuration register to start the MAC transmitter and receiver.

To support Jumbo Transmit/Receive packets, follow these steps:

- In the MAC_configuration register
  1. Set JE field to 1
  2. Set JD and WD fields to 0
     To avoid giant packet error reporting
  3. Set GPSLCE field to 1
  4. Set GPSL field of the MAC_Ext_Configuration register to a value > 9026

To support Transmit/Receive packets, up to 16K, follow these steps:

- In the MAC_configuration register,
  1. Set JD and WD fields to 1
     To avoid giant packet error reporting
  2. Set GPSLCE field to 1
  3. Set GPSL field of the MAC_Ext_Configuration register to 16383

28.18.4 Performing Normal Receive and Transmit Operation

During normal operation of the GMAC, normal and transmit interrupts are read, descriptors polled, the DMA is suspended (if it does not own descriptors), and values of current host transmitter or receiver descriptor pointers are read for debugging.

For normal operation, complete the following steps:

1. For normal transmit and receive interrupts, read the interrupt status. Then, poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).

2. Set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
3. If the descriptors are not owned by the DMA (or no descriptor is available), the DMA goes into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and writing the descriptor tail pointer to Tx/Rx tail pointer register (DMA_CHn_TXDESC_TAIL_POINTER and DMA_CHn_RXDESC_TAIL_POINTER).
4. The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (DMA_CHn_CURRENT_APP_TXDESC and DMA_CHn_CURRENT_APP_RXDESC).
5. The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (DMA_CHn_CURRENT_APP_TXBUFFER and DMA_CHn_CURRENT_APP_RXBUFFER).

28.18.5 Stopping and Starting Transmission

Complete the following steps to pause the transmission for some time. The steps are provided for Channel 0.

1. Disable the Transmit DMA by clearing bit 0 (ST) of DMA_CHn_TX_CONTROL (n = 0 to 7) register.
2. Wait for any previous frame transmissions to complete. You can check this by reading the appropriate bits of MTL_TxQ0_Debug register (TRCSTS is not 01b and TXQSTS = 0).
3. Disable the MAC transmitter and MAC receiver by clearing bit (RE) and bit 1(TE) of the MAC_Configuration register.
4. Disable the Receive DMA, after making sure that the data in the Rx FIFO is transferred to the system memory (by reading the appropriate bits of MTL_TxQ0_Debug register, PRXQ = 0 and RXQSTS = 00b).
5. Make sure that both Tx Queue and Rx Queue are empty (TXQSTS is 0 in MTL_TxQ0_Debug register and RXQSTS is 0 in MTL_RxQ0_Debug register).
6. To restart the operation, first start the DMAs, and then enable the MAC Transmitter and Receiver.

Note:
- Do not change the configuration (such as duplex mode, speed, port, or loop back) when the MAC is actively transmitting or receiving. These parameters are changed by software only when the MAC transmitter and receiver are not active.
- Similarly, do not change the DMA-related configuration when Transmit and Receive DMA are active.

28.18.6 Programming Guidelines for Switching to New Descriptor List in RxDMA

Switching to a new descriptor list is different in the Rx DMA compared to the Tx DMA. Switching to a new descriptor list is permitted when the RxDMA is in SUSPEND state, as clarified by the following points:

- Generally, RxDMA prepares the descriptors in advance.
- If the RxDMA goes to SUSPEND due to descriptors not being available, a major failure occurs (software is not able to free the filled-up descriptors/buffers). If this issue is not rectified immediately, frames are lost because of an RxFIFO Overflow. Therefore, the software is allowed to create a new descriptor list and program the RxDMA to start using it immediately, without going into STOP state.

28.18.7 Programming Guidelines for Multi-Channel Multi-Queuing

28.18.7.1 Transmit

1. Program the Transmit queue size in the TQS field of MTL_TXQn_OPERATIOn_MODE register. Based on the value programmed in the TQS field, the size of the queue is determined. In the Transmit operation, the number of channels is equal to the number of the queues. Due to this reason, the Channel-to-Queue mapping is fixed.
2. For a queue to be used, enable the ST bit of DMA_CHn_TX_CONTROL register and corresponding TXQEN in MTL_TxQn_Operation Mode register.
3. Program the scheduling method in the SCHALG field of MTL_Operation_Mode register.
4. In case of CBS algorithm in AVB queues, the MTL_TXQn_ETS_CONTROL, MTL_TXQn_SENDSLOPECREDIT, MTL_TXQn_HICREDIT and MTL_TXQn_LOCREDIT registers need to be programmed as required.

28.18.7.2 Receive

1. Program the Receive queue size in the RQS field of MTL_RXQn_OPERATIOn_MODE Register. Based on the value programmed in RQS field, the size of the queue is determined.
2. Enable SR bit of statically or dynamically mapped DMA_CHn_RX_CONTROL Register and corresponding RXQnEN in MAC_RxQ_Ctrl0 register.
3. The MAC routes the Rx packets to the Rx Queues based on following packet types:
   (a) PTP Packets: Based on the programming of PTPQ in MAC_RxQ_Ctrl1 register.
   (b) AV Untagged Control packets: Based on the programming of AVCPQ in MAC_RxQ_Ctrl1 register.
   (c) VLAN Tag Priority field in VLAN Tagged packets: Program PSRQ7-0 of the MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 Register for the routing of tagged packets based on the USP (user Priority) field of the received packets to the Rx Queues 0 to 7.
   (d) The AV tagged control and data packets are also routed based on PSRQ field of the MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers.
   The priorities set in PSRQ7-0 should be unique.
4. If multiple RX DMA channels are enabled, the following programming should be done for proper arbitration and mapping:
   (a) Program the RAA field of MTL_Operation_Mode register to select the arbitration algorithm to decide which RxQ is read out from the RxFIFO memory.
   (b) Program the MTL_RXQn_CONTROL register to decide the weights and the packet arbitration for each RxQ.
   (c) If static mapping is programmed in MTL_RxQ_DMA_Map0/1 register (QnDDMACH is reset to 0), bits QnMDMACH and others need to be programmed to select the channel for which each queue is mapped.
   (d) Set QnDDMACH bit in MTL_RxQ_DMA_Map0/1 register to select dynamic mapping of packets in each RxQueue.
   (e) In dynamic channel mapping, the routing of a packet to a specific RxDMA channel is decided by the value of DCS field in the lowest MAC Address register.

28.18.7.3 Programming Guidelines for Recovering from DMA Channel Failure

When the DMA channel issues a bus error, follow these steps to recover from the failure.

(1) Recovering from the Receive DMA Channel Failure

Follow these steps if you get bus error in the Receive DMA channel:

1. Set the RPF bit to 1. This flushes all the packets one after the other. This step is optional. However, setting this bit prevents HOL (head-of-line) blocking in the Rx queues when packets sent to the RxDMA are stopped due to bus error.
2. Re-program the specific Registers of the DMA channel.
3. Start the DMA channel.

(2) Recovering from the Transmit DMA Channel Failure

1. Stop the specific DMA channel, even if it is in active state.
2. Flush the corresponding MTL queue.
3. Re-program the specific Registers of the DMA channel.
4. Start the DMA channel.

Note: Due to the known limitations in the design, reprogramming the DMA channel registers might not always be successful in recovering from a Bus Error. If GMAC is not fully functional after reprogramming the DMA, as a workaround, issue a Soft Reset to recover from the Bus Error.

28.18.7.4 Programming Guidelines for Disabling Receive Queue

Follow these steps to disable specific receive queue

1. Disable the receiver by setting the RE field of the MAC_Configuration register to 0.
2. Read the MAC_Debug register (ensure that RPESTS and RFCFCSTS fields are 0) and MTL_RXQn_DEBUG registers of all the Rx queues.
3. When MTL_RXQn_DEBUG registers are read as 0, disable the intended Rx queue by programming the corresponding fields in the MAC_RxQ_Ctrl0 register.
4. Enable the receiver by setting the RE field of MAC_Configuration register to 1.

28.18.8 Programming Guidelines for GMII Link State Transitions

28.18.8.1 Transmit and Receive Clocks Running when Link Down

Complete the following steps when the link is down but the Transmit and Receive clocks are running.

1. Disable the Transmit DMA by clearing bit 0 (ST) of DMA_CHn_TX_CONTROL (n = 0 to 7) Register.
2. Disable the MAC receiver by clearing bit 2 (RE) of MAC_Configuration register.
3. Wait for any previous frame transmissions to complete. You can check this by reading the appropriate bits of MTL_TxQ0_Debug register (TRCSTS is not 01). Or, Flush the Tx FIFO for faster empty operation.
4. Disable the MAC transmitter by clearing bit 1(TE) of the MAC_Configuration register.
5. Make sure that both Tx Queue and Rx Queue are empty (TXQSTS is 0 in MTL_TxQ0_Debug register and RXQSTS is 0 in MTL_RxQ0_Debug register).
6. After the link is up, read the PHY registers to know the latest configuration and accordingly program the MAC registers.
7. Restart the operation by starting the Tx DMA, and then enabling the MAC Transmitter and Receiver. You do not need to disable the Rx DMA. As the Receiver is disabled, the FIFO does not get any data in the Rx FIFO.

28.18.8.2 Transmit and Receive Clocks Stopped when Link Down

Complete the following steps when the link is down and the Transmit and Receive clocks are stopped:

1. Disable the MAC Transmitter and Receiver by clearing bits RE and TE of MAC_Configuration register. This does not take effect immediately as the clocks are absent.
2. Wait until the link is up and the clocks are restored.
3. Wait for the completion of the transfer of any partial frame, if any, at time of stopping of Transmit/Receive clock. This can be checked by reading the MAC_Debug register (should be all-zero). Some old packets may still remain in the TXFIFO as the MAC Transmitter is stopped.
4. Read the PHY registers to know the latest operating mode and accordingly program the MAC registers.
5. Restart the MAC Transmitter and Receiver by setting RE and TE bits.

28.18.9 Programming Guidelines for IEEE 1588 Timestamping

28.18.9.1 Initialization Guidelines for System Time Generation

You can enable the timestamp feature by setting bit 0 of the MAC_Timestamp_Control register. However, it is essential that the timestamp counter be initialized after this bit is set. Complete the following steps during GMAC initialization:

1. Mask the Timestamp Trigger interrupt by clearing the bit 12 of MAC_Interrupt_Enable register.
2. Set bit 0 of MAC_Timestamp_Control register to enable timestamping.
3. If one-step timestamping is enabled
   (a) To enable one-step timestamping, program bit 27 of the TDES3 Context Descriptor.
   (b) Program registers MAC_Timestamp_Ingress_Asym_Corr and MAC_Timestamp_Egress_Asym_Corr to update the correction field in PDelay_Req PTP messages.
4. Enable the MAC receiver and transmitter for proper timestamping.

Note: If timestamp operation is disabled by clearing bit 0 of MAC_Timestamp_Control register, repeat all these steps to restart the timestamp operation.

28.18.10 Programming Guidelines for PTP Offload Feature

28.18.10.1 Programming Guidelines to Enable Automatic Periodic Generation of PTP Sync Messages

Follow these steps to enable automatic periodic generation of PTP sync messages:

1. Program SNAPTYPSEL, TSMSTRENA, and TSEVNTENA fields of MAC_Timestamp_Control register to 0, 1, and 1 respectively, to configure the node as Ordinary or Boundary Master (1, 1, and 1 for Transparent Master).
2. Program the PTOEN bit and DN field of MAC_PTO_Control register to enable PTP Offload feature and domain Number to send in egress PTP Sync message.
3. Program the ASYNCEN bit of MAC_PTO_Control register to enable periodic generation of PTP Sync messages.
4. Program the 80-bit Source Port Identity in MAC_Source_Port_Identity0, MAC_Source_Port_Identity1 and MAC_Source_Port_Identity2 registers to send in egress PTP Sync message.
5. Program the LSI field of MAC_Log_Message_Interval register to program the periodicity of the PTP Sync messages. For example, value of 1 corresponds to 2^1 which translates to PTP Sync message every 2 seconds and value of 0xFF (twos complement of -1) corresponds to 2^{-1} which translates to PTP Sync message every 0.536 seconds.
6. Program the TSIE bit of MAC_Interrupt_Enable register to enable generation of Timestamp Interrupt.
7. Wait for GMAC_SBD interrupt generated on setting of TXTSSIS bit in MAC_Timestamp_Status register, which indicates that the timestamp for PTP Sync message is captured in MAC_Tx_Timestamp_Status_Seconds and MAC_Tx_Timestamp_Status_Nanoseconds registers.

28.18.10.2 Programming Guidelines to Enable Periodic Generation of PTP Pdelay_Req Messages

Follow these steps to enable automatic periodic generation of PTP Pdelay_Req messages

1. Program SNAPTYPSEL, TSMSTRENA and TSEVNTENA fields of MAC_Timestamp_Control register to 1, 0, and 1 respectively to configure the node as Transparent Slave (1, 1, and 1 for Transparent Master OR 3, X, and X for Peer-to-Peer Transparent).
2. Program the PTOEN bit and DN field of MAC_PTO_Control register to enable PTP Offload feature and domain Number to send in egress PTP Pdelay_Req message.
3. Program the APDREQEN bit of MAC_PTO_Control register to enable periodic generation of PTP Pdelay_Req messages.
4. Program the 80-bit Source Port Identity in MAC_Source_Port_Identity0, MAC_Source_Port_Identity1 and MAC_Source_Port_Identity2 registers to send in egress PTP Pdelay_Req message.
5. Program the LMPDRI field of MAC_Log_Message_Interval register to program the periodicity of the PTP Pdelay_Req messages.
   For example, value of 1 corresponds to 2^1 which translates to PTP Pdelay_Req message every 2 seconds and value of 0xFF (twos complement of -1) corresponds to 2^{-1} which translates to PTP Pdelay_Req message every 0.536 seconds.
6. Program the TSIE bit of MAC_Interrupt_Enable register to enable generation of Timestamp Interrupt.
7. Wait for GMAC_SBD interrupt generated on setting of TXTSSIS bit in MAC_Timestamp_Status register, which indicates that the timestamp for PTP Sync message is captured in MAC_Tx_Timestamp_Status_Seconds and MAC_Tx_Timestamp_Status_Nanoseconds registers.

28.18.10.3 Programming Guidelines to Enable Generation of PTP Response Messages for Ordinary or Boundary Master Mode

Follow these steps to enable generation of PTP response messages for Ordinary or Boundary Master mode (Periodic PTP Sync messages generated and PTP Delay_Resp message generated in response to PTP Delay_Req message):

1. Program SNAPTYPSEL, TSMSTRENA and TSEVNTENA fields of MAC_Timestamp_Control register to 0, 1, and 1 respectively.

2. Program the PTOEN bit and DN field of MAC_PTO_Control register to enable PTP Offload feature and domain Number to match with ingress PTP Delay_Req message and send in egress PTP Delay_Resp message.
3. Program the 80-bit Source Port Identity in MAC_Source_Port_Identity0, MAC_Source_Port_Identity1 and MAC_Source_Port_Identity2 registers to match with ingress PTP Delay_Req message and send in egress PTP Delay_Resp message.
4. Program the DRSYNCR and LSI fields in MAC_Log_Message_Interval register, the sum of both these fields is updated in logMinDelayReqInterval field of PTP Delay_Resp message.

28.18.10.4 Programming Guidelines to Enable Generation of PTP response messages for Ordinary or Boundary Slave Mode

Follow these steps to enable generation of PTP response messages for Ordinary or Boundary Slave mode (PTP Delay_Req message generated in response to PTP Sync message):

1. Program SNAPTYPSEL, TSMSTRENA and TSEVNTENA fields of MAC_Timestamp_Control register to 0, 0, and 1 respectively.
2. Program the PTOEN bit and DN field of MAC_PTO_Control register to enable PTP Offload feature and domain Number to match with ingress PTP Sync message and send in egress PTP Delay_Req message.
3. Program the 80-bit Source Port Identity in MAC_Source_Port_Identity0, MAC_Source_Port_Identity1 and MAC_Source_Port_Identity2 registers to match with ingress PTP Sync message and send in egress PTP Delay_Req message.
4. Program the DRSYNCR field in MAC_Log_Message_Interval register to indicate one PTP Delay_Req message is generated in response to how many received PTP Sync messages.

28.18.10.5 Programming Guidelines to Enable Generation of PTP response messages for Transparent Slave Mode

Follow these steps to enable generation of PTP response messages for Transparent Slave mode (PTP Delay_Req message generated in response to PTP Sync message, PTP Pdelay_Resp message generated in response to PTP Pdelay_Req message and Periodic PTP Pdelay_Req messages generated)

1. Program SNAPTYPSEL, TSMSTRENA and TSEVNTENA fields of MAC_Timestamp_Control register to 1, 0, and 1 respectively.
2. Program the PTOEN bit and DN field of MAC_PTO_Control register to enable PTP Offload feature and domain Number to match with ingress PTP Sync or Pdelay_Req message and send in egress PTP Delay_Req or Pdelay_Resp or Pdelay_Req message.
3. Program the 80-bit Source Port Identity in MAC_Source_Port_Identity0, MAC_Source_Port_Identity1 and MAC_Source_Port_Identity2 registers to match with ingress PTP Sync or Pdelay_Req message and send in egress PTP Delay_Req or Pdelay_Resp or Pdelay_Req message.
4. Program the DRSYNCR and LMPDRI fields in MAC_Log_Message_Interval register to indicate one PTP Delay_Req message is generated in response to how many received PTP Sync messages and periodicity of the PTP Pdelay_Req messages.
5. Program the TSIE bit of MAC_Interrupt_Enable register to enable generation of Timestamp Interrupt.
6. Wait for GMAC_SBD interrupt generated on setting of TXTSSIS bit in MAC_Timestamp_Status register, which indicates that the timestamp for PTP Sync message is captured in MAC_Tx_Timestamp_Status_Seconds and MAC_Tx_Timestamp_Status_Nanoseconds registers for egress PTP Pdelay_Req and Pdelay_Resp messages.

28.18.10.6 Programming Guidelines to Enable Generation of PTP response messages for Transparent Master Mode

Follow these steps to enable generation of PTP response messages for Transparent Master mode (PTP Delay_Resp message generated in response to PTP Delay_Req message, PTP Pdelay_Resp message generated in response to PTP Pdelay_Req message and Periodic PTP Pdelay_Req or Sync messages generated):

1. Program SNAPTYPSEL, TSMSTRENA and TSEVNTENA fields of MAC_Timestamp_Control register to 1, 1, and 1 respectively.

2. Program the PTOEN bit and DN field of MAC_PTO_Control register to enable PTP Offload feature and domain Number to match with ingress PTP Delay_Req or Pdelay_Req message and send in egress PTP Delay_Resp or Pdelay_Resp or Pdelay_Req or Sync message.
3. Program the 80-bit Source Port Identity in MAC_Source_Port_Identity0, MAC_Source_Port_Identity1 and MAC_Source_Port_Identity2 registers to match with ingress PTP Delay_Req or Pdelay_Req message and send in egress PTP Delay_Resp or Pdelay_Resp or Pdelay_Req or Sync message.
4. Program the DRSYNCR, LSI and LMPDRI fields in MAC_Log_Message_Interval register, the sum of DRSYNCR and LSI is updated in logMinDelayReqInterval field of PTP Delay_Resp message and periodicity of the PTP Sync or Pdelay_Req messages.
5. Program the TSIE bit of MAC_Interrupt_Enable register to enable generation of Timestamp Interrupt.
6. Wait for GMAC_SBD interrupt generated on setting of TXTSSIS bit in MAC_Timestamp_Status register, which indicates that the timestamp for PTP Sync message is captured in MAC_Tx_Timestamp_Status_Seconds and MAC_Tx_Timestamp_Status_Nanoseconds registers for egress PTP Sync, Pdelay_Req and Pdelay_Resp messages.

28.18.10.7 Programming Guidelines to Enable Generation of PTP response messages for Peer-to-Peer Transparent Mode

Follow these steps to enable generation of PTP response messages for Peer-to-Peer Transparent mode (PTP Pdelay_Resp message generated in response to PTP Pdelay_Req message and Periodic PTP Pdelay_Req messages generated):

1. Program the SNAPTYPSEL, TSMSTRENA and TSEVNTENA fields of MAC_Timestamp_Control register to 3, X, and X respectively.
2. Program the PTOEN bit and DN field of MAC_PTO_Control register to enable PTP Offload feature and domain Number to match with ingress PTP Pdelay_Req message and send in egress PTP Pdelay_Resp message.
3. Program the 80-bit Source Port Identity in MAC_Source_Port_Identity0, MAC_Source_Port_Identity1 and MAC_Source_Port_Identity2 registers to match with ingress PTP Pdelay_Req message and send in egress PTP Pdelay_Resp message.
4. Program the LMPDRI field in MAC_Log_Message_Interval register to indicate periodicity of the PTP Pdelay_Req messages
5. Program the TSIE bit of MAC_Interrupt_Enable register to enable generation of Timestamp Interrupt
6. Wait for GMAC_SBD interrupt generated on setting of TXTSSIS bit in MAC_Timestamp_Status register, which indicates that the timestamp for PTP Sync message is captured in MAC_Tx_Timestamp_Status_Seconds and MAC_Tx_Timestamp_Status_Nanoseconds registers for egress PTP Pdelay_Req and Pdelay_Resp messages.

28.18.11 Programming Guidelines for AV Feature

After you enable the AV feature in the GMAC controller, there are several programming tasks to be followed:

- Initializing the DMA
- Enabling slot number checking
- Enabling average bits per slot reporting
- Disabling flow control for AV-enabled queues (receive flow control)

28.18.11.1 Initializing the DMA in Audio Video Feature

The first step in programming the AV feature is to initialize the DMA.

1. Provide a software reset to reset all internal registers and logic (bit 0 in DMA_Mode register) of GMAC.
2. Wait for the completion of the reset process. Poll bit 0 of the DMA_Mode register, which is cleared only after the reset operation is completed.
3. Program the fields to initialize the DMA register by setting the values in DMA_Mode register.
4. Create a proper descriptor list for transmit and receive. In addition, ensure that the DMA owns the Transmit and Receive descriptors. When OSF mode is used, at least two TX descriptors are required. For more information about descriptors, see section 28.17. Descriptors.

5. Make sure that your software creates three or more different transmit or receive descriptors in the list before reusing any of the descriptors.
6. Program the Transmit and Receive Ring length registers (DMA_CHn_TXDESC_RING_LENGTH (n = 0 to 7) and DMA_CHn_RX_CONTROL2 (n = 0 to 7)). The ring length programmed must be at least 4.
7. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (DMA_CHn_TXDESC_LIST_ADDRESS (n = 0 to 7), DMA_CHn_RXDESC_LIST_ADDRESS (n = 0 to 7)). In addition, you must program the Transmit and Receive tail pointer registers indicating to the DMA about the available descriptors (DMA_CHn_TXDESC_TAIL_POINTER (n = 0 to 7) and DMA_CHn_RXDESC_TAIL_POINTER (n = 0 to 7)).
8. Program the following fields to initialize the mode of operation in the MTL_TxQ0_Operation_Mode register:
   (a) Transmit Store and Forward (TSF)
   (b) Transmit Threshold Control (TTC)
   (c) Transmit Queue Enable (TXQEN) to value 10b to enable Transmit Queue0
   (d) Transmit Queue Size (TQS)
9. Enable the interrupts by programming the DMA_CHn_INTERRUPT_ENABLE (n = 0 to 7) register.
10. Repeat steps 4 through 9 for all additional channels of AV feature.
11. Program the CBS control register, idleSlope, sendSlope, hiCredit, and loCredit registers of the AVQueues
12. Start the Receive and Transmit DMA by setting bit 0 of the DMA_CHn_TX_CONTROL (n = 0 to 7) register and bit 0 of the DMA_CHn_RX_CONTROL (n = 0 to 7) register.

28.18.11.2 Enabling Slot Number Checking

These steps should be completed after step 11 and before step 12 of section 28.18.1. Initializing DMA.

You can use the slot number check feature to specify the intervals at which the DMA Channels mapped to AV Queues fetch the frames from the AXI bus. This feature is useful for a uniform and periodic transfer of the AV traffic from the host memory. Complete the following steps to enable the slot number checking:

1. Enable timestamping by following the steps described in section 28.18.9.1. Initialization Guidelines for System Time Generation.
2. Make sure that the SLOTNUM field (bits 22:19) of TDES3 Normal Descriptor (Read Format) contains a valid slot number. You can read the current reference slot number from the DMA_CHn_SLOT_FUNCTION_CONTROL_STATUS (n = 0 to 7).
3. Set bit 0 (ESC) of the Slot Function Control and Status register of a channel to enable the slot number checking.

28.18.11.3 Enabling Average Bits Per Slot Reporting

You can enable the ability to report the average bits that are transmitted in a slot.

The CBS Status register of the additional AV channels provides information about the average bits that are transmitted in a slot. The software can asynchronously read this register to retrieve information about the average bits transmitted per slot. Complete the following steps to enable average bits per slot reporting:

1. Enable timestamping by following the steps described in the section 28.18.9.1. Initialization Guidelines for System Time Generation.
2. Program bits [6:4], SLC, of the MTL_TXQn_ETS_CONTROL register of a channel with the number of slots over which the average transmitted bits per slot need to be computed.
3. Enable bit 9(ABPSSIE) of the MTL_Qn_INTERRUPT_CONTROL_STATUS register of a channel to generate the average bits per slot interrupt.
   - The frequency of this interrupt depends on the value programmed in step 2. For example, when you program value 0 in the SLC field, the interrupt is generated at every 125 micro-second.
   - When not required, you can disable this interrupt to stop the interrupt flooding.
4. Read bits [16:0], ABS, from the MTL_TXQn_ETS_STATUS register of a channel on each interrupt.

- The software can read the ABS bits in polling mode even if the ABPSIE bit is not enabled. When high, bit 1 (ABPSIS) of the MTL_TXQn_ETS_STATUS register indicates that a new value is updated in the ABS field.

28.18.11.4 Disabling Flow Control for AV Enabled Queues

(1) Receive Flow

Program the EHFC (Enable Hardware Flow Control) bit of corresponding Rx Queue's MTL_RXQn_OPERATIOn_MODE register to 0.

28.18.12 Programming Guidelines for Energy Efficient Ethernet

During the GMAC initialization, after you enable the (EEE) you can program how to enter and exit Tx LPI mode:

28.18.12.1 Entering and Exiting the Tx LPI Mode

EEE enables the IEEE 802.3 Media Access Control (MAC) sub layer along with a family of Physical layers to operate in the Low-Power Idle (LPI) mode. In the Transmit path, the software must set the LPIEN bit of the MAC_LPI_Control_Status register to indicate to the MAC to stop transmission and initiate the LPI protocol.

Follow these steps while initializing GMAC:

1. Read the PHY register through the MDIO interface, check if the remote end has the EEE capability, and then negotiate the timer values.
2. Program the PHY registers through the MDIO interface (including the RX_CLK_stoppable bit that indicates to the PHY whether to stop Rx clock in LPI mode.)
3. Program bits [25:16] and bits [15:0] in MAC_LPI_Timers_Control register.
4. Read the link status of the PHY chip by using the MDIO interface and update bit 17 of MAC_LPI_Control_Status accordingly. This update should be done whenever the link status in the PHY chip changes.
5. Program the MAC_1US_Tic_Counter as per the frequency of the clock used for accessing the CSR slave port.
6. Program the MAC_LPI_Entry_Timer register (LPIET) with the IDLE time for which the MAC should wait before entering the LPI state on its own.
7. Set LPITE and LPITXA (bits [20:19]) of MAC_LPI_Control_Status register to enable the auto-entry into LPI and auto-exit of MAC from LPI state.
8. Program the MAC_1US_Tic_Counter as per the frequency of the clock used for accessing the CSR slave port.
9. Program the MAC_LPI_Entry_Timer register (LPIET) with the IDLE time for which the MAC should wait before entering the LPI state on its own.
10. Set LPITE and LPITXA (bits [20:19]) of MAC_LPI_Control_Status register to enable the auto-entry into LPI and auto-exit of MAC from LPI state.
11. Set bit 16 of MAC_LPI_Control_Status register to make the MAC Transmitter enter the LPI state. The MAC enters the LPI mode after completing all scheduled packets and remains IDLE for the time indicated by LPIET. It sets the TLPIEN (bit [0]) after entry to LPI state.
12. When a packet is scheduled for transmission (when the TxDMA comes out of IDLE state or when a packet is presented at ATI or MTI interface), the MAC Transmitter exits LPI state automatically. It waits for TWT time before setting the TLPIEX interrupt status bit and then resume the packet transmission.
13. MAC Transmitter re-enters LPI state if it remains IDLE for LPIET time and sets the TLPIEN bit and the entry-exit cycle continues.
14. Reset LPITXEN in case the application wants to over-ride the auto-entry/exit modes and make the MAC Transmitter exit the LPI state directly.

Note: To ensure MAC enter the LPI state only after completion of transmission of all the queued frames in the Tx FIFO, you should set bit 19 in MAC_LPI_Control_Status register.

28.18.13 Programming Guidelines for Header Payload Split Receive

The Header-Payload Split support is an optional feature that can be enabled by

1. Setting the SPH field in the DMA_CHn_CONTROL register
2. Selecting the Header-Payload split option by setting the SPLM field of MAC_Ext_Cfg1 register.

The DMA can process the header and payload of the received packets separately.

Note: In case of L3/L4 header-payload split, if the IPv4 header length or IPv6 payload length fields are corrupt, the split might not always happen at the correct boundary.

Note:
- When the SPH field in the DMA_CHn_CONTROL register is set to 1, Buffer 1 of only the first descriptor is used with buffer size indicated by the HDSMS filed of MAC_Ext_Configuration.
- Buffer 2 of the first descriptor and subsequent descriptors are used with buffer size indicated by the RBSZ field of the DMA_CHn_RX_CONTROL register.
- When the HL field of the first descriptor (RDES2) is zero, it indicates packet is not split into header and payload.
- When the HL field of the first descriptor (RDES2) is non-zero, it indicates that the packet is split into header and payload.

28.18.14 Programming Guidelines for VLAN Filtering on Receive

Complete the following steps to program VLAN filtering on receive:

1. Program MAC_VLAN_Tag register for the following bit to select the filtering method:
   (a) ETV: Enable 12-bit VLAN Tag Comparison or 16-bit VLAN Tag comparison.
   (b) VTHM: VLAN Tag Hash Table Match Enable.
   (c) ERIVLT: Enable inner VLAN Tag or outer VLAN Tag (to enable the inner or outer VLAN Tag filtering, Double VLAN Processing should enabled by setting EDVLP)
   (d) ERSVLM: Enable Receive S-VLAN Match or C-VLAN match (for S-VLAN processing to be enabled, set ESVL)
   (e) DOVLTC: Ignores VLAN Type for Tag Match
   (f) VTIM: to enable VLAN Tag Inverse Match instead of the normal VLAN Tag matching
2. Program VL field of MAC_VLAN_Tag register for the 12-bit or 16-bit VLAN tag.
3. If Hash filtering of VLAN tag is enabled, program MAC_VLAN_Hash_Table register. When ETV bit is reset, upper 4 bits of the calculated CRC-32 of VLAN Tag are inverted and used to index the content of MAC_VLAN_Hash_Table register. When ETV bit is set, upper 4 bits of calculated CRC-32 of VLAN Tag are used to index the content of MAC_VLAN_Hash_Table register. For example, when ETV bit is set, a hash value of 1000b selects bit 8 of the VLAN Hash table. When ETV bit is reset a hash value of 1000b selects bit 7 of the VLAN Hash table.

28.18.15 Programming Guidelines for Extended VLAN Filtering and Routing on Receive

For the indirect access of the per VLAN Tag registers, follow these steps:

- Write
  1. Write the required data into the MAC_VLAN_Tag_Data register.
  2. Program the OFS field of MAC_VLAN_Tag_Ctrl register with the required Filter Register offset and command type to the CT field. For a write command, set this bit to 0.
  3. Write 1 to the OB field and wait until the OB bit is reset before performing the next write. This is to ensure that the appropriate VLAN Tag Filter Register has been programmed.
- Read
  1. Program the OFS field of the MAC_VLAN_Tag_Ctrl register with the required register's offset and command type to the CT field. For a read command, set this bit to 1.
  2. Write 1 to the OB field and wait till the OB bit is reset. The appropriate VLAN Tag Filter register value is available in the MAC_VLAN_Tag_Data register.

28.18.16 Programming Guidelines for EST

Program the gate control values and time intervals in the Software Owned Gate Control List (SWOL) along with the other EST related registers that are described in section 28.3. Register Descriptions, to appropriate values. The following sub-sections provide step by step details for programming the GCL and the other EST related registers.

28.18.16.1 Programming the GCL and GCL Linked Registers

Follow these steps to program the Gate Control List (GCL) and the four other registers implemented per GCL.

1. The GCL and the four other GCL-linked registers should be accessed through indirect addressing using the MTL_EST_GCL_Control and MTL_EST_GCL_Data registers. The SWOL field of the MTL_EST_Status register indicates if GCL0 or GCL1 is owned by software.
2. To program the GCL, write the 32-bit write data to MTL_EST_GCL_Data register. Then program MTL_EST_GCL_Control register to write the write address and other control information.
3. In the MTL_EST_GCL_Data register, Write Data consists of up to 8 bits (configurable) of Gate Controls and up to 24 bits (configurable) of Time Interval. Gate Close is indicated by programming a '0' and Gate Open is indicated by programming a '1'. For a 4-TC and 20-bit Time Interval Configuration, the data width is 24-bits and the remaining 8-bits are reserved/read-only. The data should be written in the following format.
   {0x00, TC3, TC2, TC1, TC0, 20-bit Time Interval} where TCx = 0 or 1.
4. MTL_EST_GCL_Control register: Program the SRWO bit to 1 (to start a Write Op) and program the address and R/W fields appropriately.
5. Poll and check for the SRWO bit to be cleared by hardware to indicate the completion of the previous operation before initiating a new R/W operation using the same indirect addressing mode.
6. Repeat steps 3, 4, 5 until the programming of the GCL is completed.
7. Using the indirect addressing method, program the BTR, CTR, TER and LLR registers. Set the GCRR bit in the MTL_EST_GCL_Control register appropriately. The GCRR bit interprets the address field as belonging to these registers (instead of the GCL).

After programming of the GCL and the related registers, program the MTL_EST_Control register to allow hardware to own and process the GCL.When the List Length (as indicated in LLR) is 1, the associated Time Interval should be smaller than the value of the Cycle Time Register. Otherwise, an error is reported (as detailed in the Error handling section) as a single set of gate controls add no value in the TSN context.

Note: The time unit in all the GCL related registers is seconds and nano-seconds. In cases where internally generated PTP System Time is used, the nano-seconds field must be programmed to use the Digital Rollover mode. The TSCTRLSSR field of MAC_Timestamp_Control must be set to 1.

28.18.16.2 Programming the EST Registers

After completing the steps mentioned in section 28.18.16.1. Programming the GCL and GCL Linked Registers, program the MTL_EST_Control register.

1. Set the Current Time Offset Value and Time Internal Left Shift fields of the MTL_EST_Control register appropriately. Also, set the Enable EST (EEST) and Switch to SWOL (SWOL) bits.
2. This enables GMAC to own and process the new GCL and switch to the new GCL at the BTR value. If enabled, GMAC generates an interrupt when the switch to the new list occurs.
3. Software must address any other interrupts received during the hardware execution of the GCL.

Set the SSWL field in the MTL_EST_Control register to handoff to the controller.

The SSWL is reset/cleared by the controller when it successfully switches to the new list. The controller also flips the SWOL bit to indicate the new GCL that the software owns.

To install a new GCL, program the GCL it owns (indicated by SWOL bit) as described in section 28.18.16.1. Programming the GCL and GCL Linked Registers and then program the MTL_EST_Control register as described. Ensure that the new BTR is set to an appropriate value to avoid BTRError that might need software intervention in some cases.

To avoid transmission overruns, the packet length (frame size) information should be available at all times. Therefore, program the packet length in the first descriptor of every Tx frame.

28.18.17 Programming the Launch Time in Time-Based Scheduling

The Launch Time is programmed in the Enhanced Normal Transmit Descriptors and is driven as a control word in MTL as follows:

The OSTC and Launch Time features are mutually exclusive and should not be used together; in case of a conflict (if OSTC = LTV = 1 in MTL), priority is given to OSTC and the Launch Time is ignored.

If a context descriptor is received with a valid OSTC values immediately before receiving a first normal descriptor with LTV bit set, then the LTV is ignored.

28.19 Usage Notes

28.19.1 Known Issue and Workaround

There are known issues in GMAC. Please consider workaround when using related function.

Table 28.104 Known issue and workaround (1 of 2)

No.	Issue	Workaround
1	The EST (Enhancements to Scheduled Traffic) scheduler switches to the next Gate Control List (GCL) after executing the current Gate-Control List (GCL) regardless of the difference between the Cycle Time Register (CTR) value, and sum of the Base Time Register (BTR) and Time Intervals (TI) of the GCL rows whose execution is complete. If the GCL execution takes longer than the cycle time, the GCL is truncated at the cycle time, and the subsequent loop begins at BTR + N × Cycle Time, where N represents the iteration number, an integer. However, in the following situations, the GCL incorrectly updates the internal BTR twice. As a result, the GCL skips the execution of the next GCL loop: ● CTR value is lesser or greater than GCL loop execution time. ● The difference between the CTR and the sum of the BTR and Time Intervals of completely executed GCL rows is less than 8 PTP clock periods (8 × 8 ns = 64 ns).	Program the CTR, BTR, and Time Intervals of the GCL rows such that the difference between the CTR and the sum of the BTR and time intervals of fully executed GCL rows must be greater than 8 PTP clock periods (64 ns). Alternatively, the CTR must be equal to the sum of the BTR and Time Intervals of the fully executed GCL rows.
2	The DMA fetches and processes descriptors up to the one located prior to the Tail Pointer (DMA_CHn_TxDesc_Tail_Pointer and DMA_CHn_RxDesc_Tail_Pointer). It generates Buffer Unavailable Interrupt, and enters suspend state. When the software updates the Tail Pointer, the DMA exits the suspend state, fetches, and processes further descriptors if the Tail Pointer is moved ahead. Otherwise, the DMA remains in suspend state and generates Buffer Unavailable Interrupt to indicate to the software that there are still no descriptors to fetch and process, even after the Tail Pointer update. This prompts the software to take corrective action. However, when Tail Pointer is updated with same address, the DMA incorrectly exits suspend state, fetches and processes descriptor pointed by Tail Pointer and beyond. This continues if the DMA finds a 1 in the OWN bit position in the descriptors yet to be set or made available to the DMA for processing. This results in unintended packets to be transmitted.	Do not program the Tail Pointer with same value, especially when the Buffer Unavailable Interrupt is due to the DMA hitting the Tail Pointer limit.
3	When the Enhancements to Scheduling Traffic (EST) feature is enabled, an extraneous IPG (that is, more than the programmed minimum IPG) is observed even when back-to-back packets are available for scheduling. The waiting period until the current packet is completely forwarded to the MAC transmitter for scheduling the next eligible packet results in an extraneous IPG of approximatively two clock cycles of the MAC transmitter clock, converted to bit times based on the operating speed.	The software must program the Time Interval (TI) to accommodate the extraneous IPG based on the number of packets that are expected to be scheduled. For example, one-Gbps speed mode, the TI must be increased by at least a value that is equal to (clock period (8 ns) × extra IPG (2)). That is, by 16 ns per data packet.

Table 28.104 Known issue and workaround (2 of 2)

No.	Issue	Workaround
4	When Credit Based Shaper (CBS) is enabled for a Transmit Queue (TXQ), the packet that is available in a TXQ is scheduled for transmission when the TXQ accumulates zero or positive credit. When the current credit is negative or when a packet is available in the Transmit Queue and is waiting for the opportunity to be scheduled for transmission, the MAC increments the credit. Equally, during the actual packet transfer, the MAC decrements the credit. The rate at which the MAC increments or decrements the credit depends on the percentage of the total bandwidth that is reserved for the TXQ. As per the IEEE 802.1Qav standard, the MAC must decrement the credit also when the packet overheads are transmitted. These overheads include the preamble bytes before the start of packet, CRC/FCS bytes, and the minimum 12-byte Inter Packet Gap (IPG) after the end of the packet data transfer. However, the MAC decrements the credit only up to the last byte of the packet data (that is, the last byte of the Frame Check Sequence (FCS) transfer) and increments the credit during the subsequent nominal IPG period that is associated with that packet.	You must compute the additional/extra bandwidth that the TXQ consumes due to the defective algorithm. For this, you must use the below formula and example, substituting the values with the average length of the packets that are transmitted for the use case. Then, you must program the parameters that determine the fractional bandwidth for this TXQ to reflect the lesser percentage, such that the effective programmed fractional bandwidth is closer to the desired one. Formula: Additional/ Extra bandwidth = ((# of packets × 12 bytes) / (Total number of bytes that is transmitted in that window, including the preamble bytes of each packet)) x Fractional bandwidth that is programmed for the TXQ. Example: If you program a 30% bandwidth for a TXQ which transfers a stream of 100 packets of 128 bytes length, the additional consumed bandwidth is equal to (30% × (100 × 12) / (100 × (8 + 128))) ≈ 2.65%. Effectively, this TXQ then receives 32.65% of total bandwidth instead of the programmed 30%.

29. Ethernet Switch (ETHSW)

29.1 Overview

With the ethernet switch function, this LSI enables to build the network topology such as line and ring type without external switching hub. This switch has also TSN (Time Sensitive Networking) capability like 802.1AS-Rev, 802.1Qbv, and 802.1Qbu/802.3br. Additionally, four pulse generators are attached to the switch to generate arbitrary pulse synchronous to network time and to capture the network time. Use of the ethernet switch and mode setting are controlled by the related registers. The specifications of the ethernet switch are summarized in following table.

Table 29.1 Ethernet Switch Specifications

Parameter	Description
Function	• Three external ports and one internal port • Integrated 10/100/1000 MACs with flexible rate PHY interface up to 1 Gbps • 3 Gbps non-blocking switching capacity • Hardware switching look-up providing a learning capacity of up to 4K MAC addresses. • VLAN table for 32 VLANs • VLAN manipulation functions on receive (VLAN insert) and transmit (VLAN removal/overwrite) • Priority Classification based on VLAN priorities, Ipv4 TOS/Ipv6 COS with programmable mapping, frame type and freely configurable frame pattern matching. • Support for 802.3br frame preemption on all Ethernet ports (management port not supported) at 10/100/1000 M speeds. • Configurable Time Multiplexed (TDMA) output queue scheduler supporting real-time network infrastructures using time slots for bandwidth reservation enabling deterministic delays • Standard Frame size support (1536) or extended frame sizes up to 1700 bytes or Jumbo frames up to 10 KB (depends on memory availability). • Configurable cut-through or store-and-forward forwarding individually per port • Hardware learning/aging without software involvement • MAC per port statistics • MAC optional support for half-duplex in 10/100 Mbps • MAC supporting Energy-Efficient Ethernet (EEE) with automatic or application controlled power state transitioning and signaling. • Switch statistics for total frames processed/discarded • Support selectable managed or unmanaged mode of operation with self-initialization. • Hardware allows for implementation of Rapid Spanning Tree Protocol (RSTP) Software. • Hardware allows for use of the switch in IEEE 1588 environments • Integrated MACs with timestamping capabilities; • IEEE 1588 1-step correction field updates on transmitted frames • Independent two timer module are available for timestamping and time for TDMA. • Frame pre-emption (802.1Qbu / 802.3br) • Cyclic Queuing and Forwarding (802.1Qch) • Support forwarding rule for the TSN-IA profile • Support a MAC source address filtering function • Exclusive forwarding of BPDU frames to/from management port. • Mirroring function to allow copying of frames to management port and others • Flexible frame snooping function to allow copying of frames to management port • Supports configurable VLAN switching such as flooding, when MAC address lookup to be omitted • Support Multicast, Broadcast with flooding control to avoid unnecessary duplication of frames • Programmable Multicast destination port mask (within lookup table entries) to restrict frame duplication for individual multicast addresses • Can be used in managed implementations with bridge protocol frame support (e.g. Spanning Tree protocols) or unmanaged implementations • Device Level Ring (DLR) hardware support option to implement beacon based nodes without burdening application processor with beacon frame processing. • Parallel Redundancy Protocol (PRP) support option with automatic and transparent redundant frame management. • Integrated Hub function for half-duplex repeater network infrastructure use. • Receive pattern match function for executing tasks based on specific frame reception. • Ingress filtering and frame header manipulation (active stream identification, flow metering) with Enhanced Frame Parser • 4 units Additional Timer function (Synchronous pulse can be generated at arbitrary timing based on switch timer time). • Event link (output) (Ethernet Switch Timer pulse event, Ethernet Switch TDMA timer event) • Module-stop function (Module-stop state can be set to reduce power consumption)

Figure 29.1 is a block diagram of the Ethernet interface.

Gray modules and registers are described in section 27, Ethernet Subsystem, section 28, Ethernet MAC (GMAC), and section 30, EtherCAT Slave Controller (ESC). See the related sections regarding the other registers and modules which are needed to operate the ethernet switch.

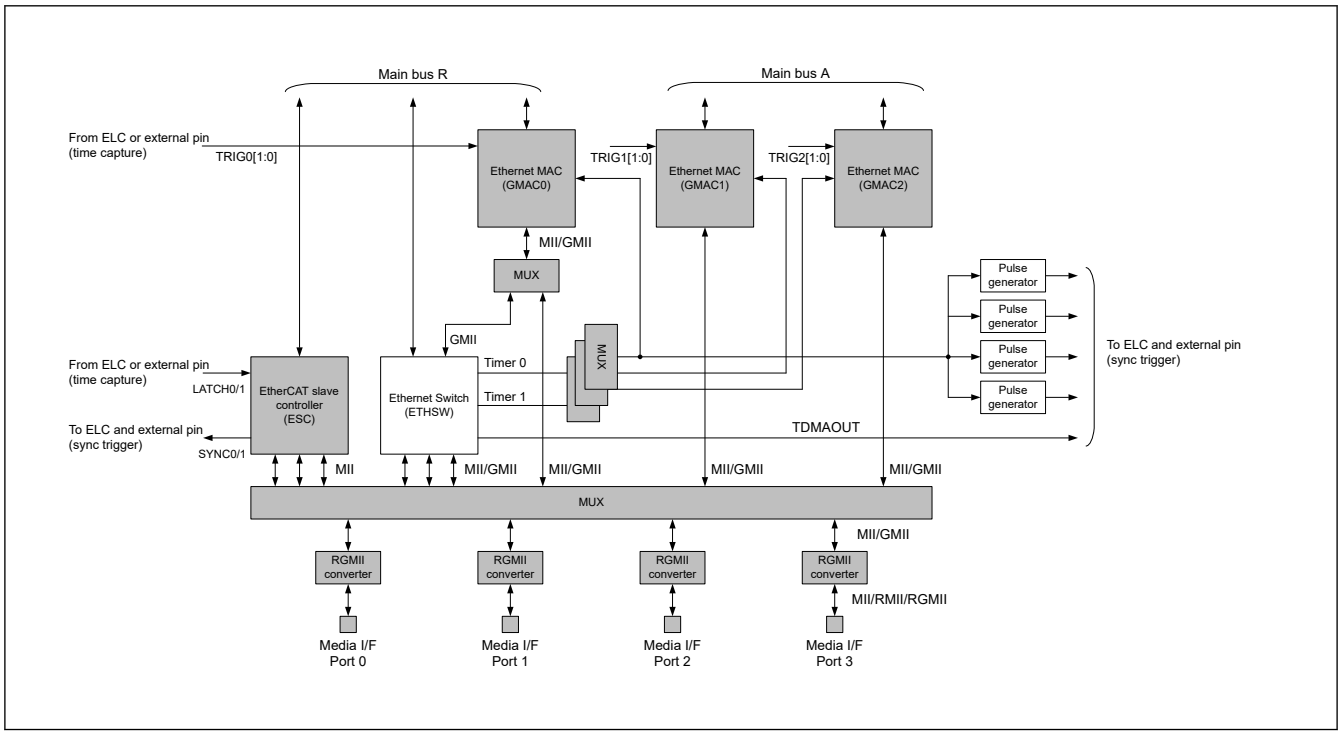


Figure 29.1 Block diagram of the ethernet interface

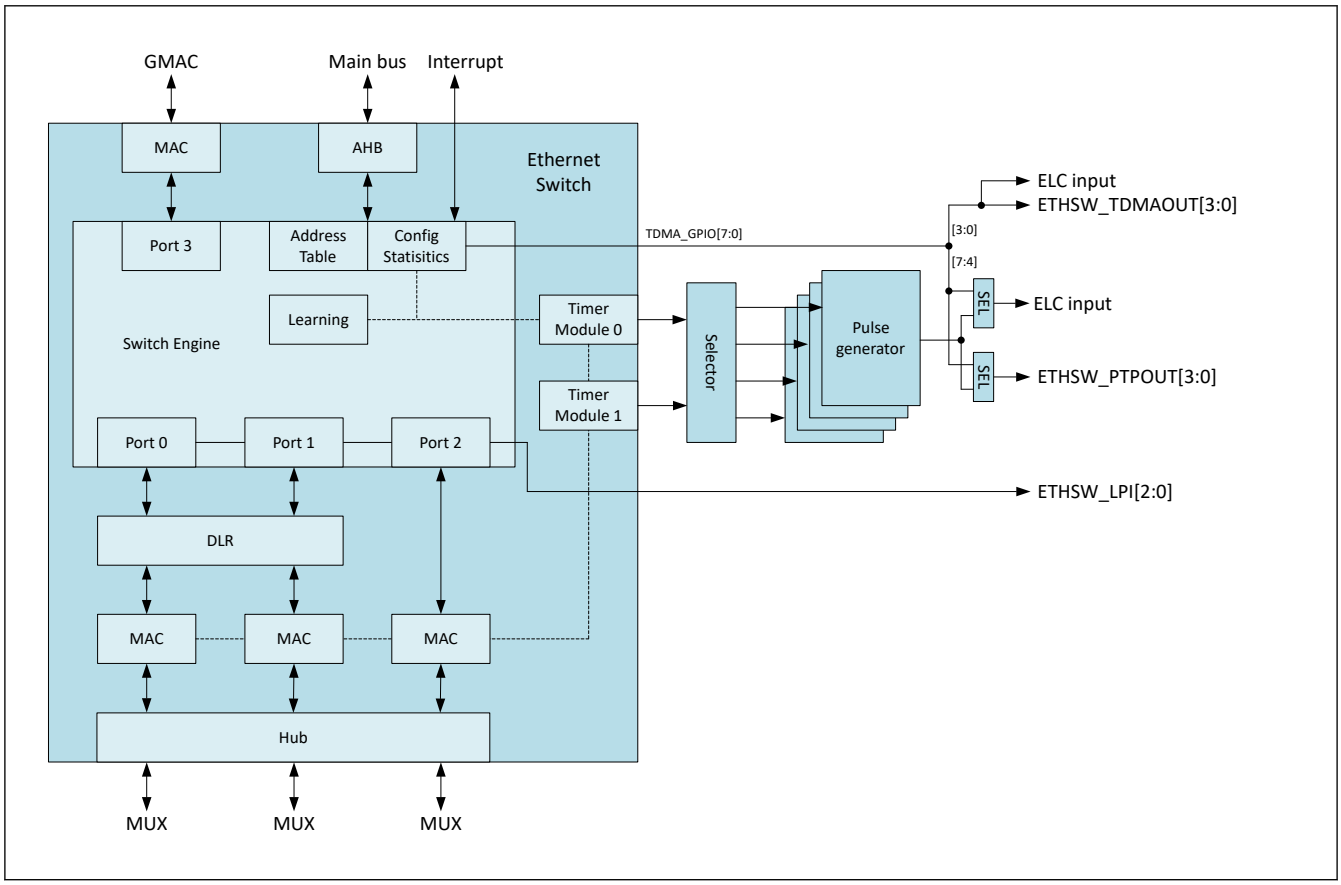


Figure 29.2 Ethernet switch overview

Table 29.2 Input/Output Pins of the ETHSW

Pin Name	I/O	Description
ETHSW_LPI0	Output	Port 0 MAC status indicating that it is currently receiving Low-Power-Idle sequences from the PHY
ETHSW_LPI1	Output	Port 1 MAC status indicating that it is currently receiving Low-Power-Idle sequences from the PHY
ETHSW_LPI2	Output	Port 2 MAC status indicating that it is currently receiving Low-Power-Idle sequences from the PHY
ETHSW_PTPOUT0 to ETHSW_PTPOUT3	Output	Ethernet Switch timer pulse output or TDMA timer output (TDMA_GPIO[4:7])
ETHSW_TDMAOUT0 to ETHSW_TDMAOUT3	Output	Ethernet Switch TDMA timer output (TDMA_GPIO[0:3])
ETHSW_PHYLINK0 to ETHSW_PHYLINK2	Input	Ethernet switch PHY link status input
ETHSW_MDC	Output	Management data clock output
ETHSW_MDIO	I/O	Management data I/O

29.2 Register Map

Table 29.3 ETHSW register map (1 of 12)

Address	Register symbol	Register name	Write protect
0x8012_0000	REVISION	Switch Core Version Register	—
0x8012_0004	SCRATCH	Scratch Register	—
0x8012_0008	PORT_ENA	Port Enable Register	—
0x8012_000C	UCAST_DEFAULT_MASK0	Unicast Default Mask Register 0	—
0x8012_0010	VLAN_VERIFY	Verify VLAN Domain Register	—
0x8012_0014	BCAST_DEFAULT_MASK0	Broadcast Default Mask Register 0	—
0x8012_0018	MCAST_DEFAULT_MASK0	Multicast Default Mask Register 0	—
0x8012_001C	INPUT_LEARN_BLOCK	Input Learning Block Register	—
0x8012_0020	MGMT_CONFIG	Management Configuration Register	—
0x8012_0024	MODE_CONFIG	Mode Configuration Register	—
0x8012_0028	VLAN_IN_MODE	VLAN Input Manipulation Mode Register	—
0x8012_002C	VLAN_OUT_MODE	VLAN Output Manipulation Mode Register	—
0x8012_0030	VLAN_IN_MODE_ENA	VLAN Input Mode Enable Register	—
0x8012_0034	VLAN_TAG_ID	VLAN Tag ID Register	—
0x8012_0038	BCAST_STORM_LIMIT	Broadcast Storm Protection Register	—
0x8012_003C	MCAST_STORM_LIMIT	Multicast Storm Protection Register	—
0x8012_0040	MIRROR_CONTROL	Port Mirroring Configuration Register	—
0x8012_0044	MIRROR_EG_MAP	Port Mirroring Egress Port Definition Register	—
0x8012_0048	MIRROR_ING_MAP	Port Mirroring Ingress Port Definition Register	—
0x8012_004C	MIRROR_ISRC_0	Ingress Source MAC Address for Mirror Filtering Register 0	—
0x8012_0050	MIRROR_ISRC_1	Ingress Source MAC Address for Mirror Filtering Register 1	—
0x8012_0054	MIRROR_IDST_0	Ingress Destination MAC Address for Mirror Filtering Register 0	—
0x8012_0058	MIRROR_IDST_1	Ingress Destination MAC Address for Mirror Filtering Register 1	—

Table 29.3 ETHSW register map (2 of 12)

Address	Register symbol	Register name	Write protect
0x8012_005C	MIRROR_ESRC_0	Egress Source MAC Address for Mirror Filtering Register 0	—
0x8012_0060	MIRROR_ESRC_1	Egress Source MAC Address for Mirror Filtering Register 1	—
0x8012_0064	MIRROR_EDST_0	Egress Destination MAC Address for Mirror Filtering Register 0	—
0x8012_0068	MIRROR_EDST_1	Egress Destination MAC Address for Mirror Filtering Register 1	—
0x8012_006C	MIRROR_CNT	Mirror Filtering Count Value Register	—
0x8012_0070	UCAST_DEFAULT_MASK1	Unicast Default Mask Register 1	—
0x8012_0074	BCAST_DEFAULT_MASK1	Broadcast Default Mask Register 1	—
0x8012_0078	MCAST_DEFAULT_MASK1	Multicast Default Mask Register 1	—
0x8012_007C	PORT_XCAST_MASK_SEL	Port Mask Select Register	—
0x8012_0088	QMGR_ST_MINCELLS	Minimum Memory Cell Statistics Register	—
0x8012_0094	QMGR_RED_MIN4	RED Minimum Threshold Register	—
0x8012_0098	QMGR_RED_MAX4	RED Maximum Threshold Register	—
0x8012_009C	QMGR_RED_CONFIG	RED Configuration Register	—
0x8012_00A0	IMC_STATUS	Input Memory Controller Status Register	—
0x8012_00A4	IMC_ERR_FULL	Input Port Memory Full and Truncation Indicator Register	—
0x8012_00A8	IMC_ERR_IFACE	Input Port Memory Error Indicator Register	—
0x8012_00AC	IMC_ERR_QOFLOW	Output Port Queue Overflow Indicator Register	—
0x8012_00B0	IMC_CONFIG	Input Memory Controller Configuration Register	—
0x8012_00B4	IMC_ERR_ALLOC	Input Port Error Indicator Register	—
0x8012_00C0 + 0x4 × n	GPARSERn (n = 0 to 3)	[n+1]th Parser of 1st Block	—
0x8012_00D0 + 0x4 × n	GARITHn (n = 0 to 3)	Snoop Configuration for Arithmetic [n+1]th Stage of 1st Block	—
0x8012_00E0 + 0x4 × (n - 4)	GPARSERn (n = 4 to 7)	[n-3]th Parser of 2nd Block	—
0x8012_00F0 + 0x4 × (n - 4)	GARITHn (n = 4 to 7)	Snoop Configuration for Arithmetic [n-3]th Stage of 2nd Block	—
0x8012_0100 + 0x4 × n	VLAN_PRIORITYn (n = 0 to 3)	VLAN Priority Register n	—
0x8012_0140 + 0x4 × n	IP_PRIORITYn (n = 0 to 3)	IP Priority Register n	—
0x8012_0180 + 0x4 × n	PRIORITY_CFGn (n = 0 to 3)	Priority Configuration Register n	—
0x8012_01B8	PRIORITY_TYPE1	Priority Type Register 1	—
0x8012_01BC	PRIORITY_TYPE2	Priority Type Register 2	—
0x8012_01C0	MGMT_ADDR0_lo	Lower MAC Address 0 for Bridge Protocol Frame Register	—
0x8012_01C4	MGMT_ADDR0_hi	Higher MAC Address 0 for Bridge Protocol Frame Register	—
0x8012_01C8	MGMT_ADDR1_lo	Lower MAC Address 1 for Bridge Protocol Frame Register	—
0x8012_01CC	MGMT_ADDR1_hi	Higher MAC Address 1 for Bridge Protocol Frame Register	—
0x8012_01D0	MGMT_ADDR2_lo	Lower MAC Address 2 for Bridge Protocol Frame Register	—
0x8012_01D4	MGMT_ADDR2_hi	Higher MAC Address 2 for Bridge Protocol Frame Register	—
0x8012_01D8	MGMT_ADDR3_lo	Lower MAC Address 3 for Bridge Protocol Frame Register	—

Table 29.3 ETHSW register map (3 of 12)

Address	Register symbol	Register name	Write protect
0x8012_01DC	MGMT_ADDR3_hi	Higher MAC Address 3 for Bridge Protocol Frame Register	—
0x8012_01E0	SRCFLT_ENA	MAC Source Address Filtering Enable Register	—
0x8012_01E4	SRCFLT_CONTROL	MAC Source Address Filtering Control Register	—
0x8012_01E8	SRCFLT_MACADDR_LO	Lower MAC Filtering Address Register	—
0x8012_01EC	SRCFLT_MACADDR_HI	Higher MAC Filtering Address Register	—
0x8012_01FC	PHY_FILTER_CFG	Debounce Filter Configuration Register	—
0x8012_0200 + 0x4 × n	SYSTEM_TAGINFOn (n = 0 to 3)	One VLAN ID Field n for VLAN Input Manipulation	—
0x8012_0240 + 0x4 × n	AUTH_PORTn (n = 0 to 3)	Port n Authentication Control and Configuration	—
0x8012_0280 + 0x4 × n	VLAN_RES_TABLEn (n = 0 to 31)	32 VLAN Domain Entries	—
0x8012_0300	TOTAL_DISC	Discarded Frame Total Number Register	—
0x8012_0304	TOTAL_BYT_DISC	Discarded Frame Total Bytes Register	—
0x8012_0308	TOTAL_FRM	Processed Frame Total Number Register	—
0x8012_030C	TOTAL_BYT_FRM	Processed Frame Total Bytes Register	—
0x8012_0340	IALK_CONTROL	IA Lookup Function Enable Register	—
0x8012_0344	IALK_OUI	IA Frames MAC Address OUI Register	—
0x8012_0348	IALK_ID_MIN	Minimum Value ID MAC Address Register	—
0x8012_034C	IALK_ID_MAX	Maximum Value ID MAC Address Register	—
0x8012_0350	IALK_ID_SUB	Offset Value ID MAC Address Register	—
0x8012_0354	IALK_ID_CONFIG	Configures Lookup Response Unknown IDs Register	—
0x8012_0358	IALK_VLAN_CONFIG	Configure Lookup Response Unknown VLAN Register	—
0x8012_035C	IALK_TBL_ADDR	IA Lookup Database Address Register	—
0x8012_0360	IALK_TBL_DATA	IA Lookup Database Data Register	—
0x8012_0380 + 0x4 × n	IALK_VLANIDn (n = 0 to 3)	IA Lookup VLANIDn Register	—
0x8012_03C0 + 0x4 × n	IMC_QLEVEL_Pn (n = 0 to 3)	Port n Queued Frame Count Register	—
0x8012_0400	LK_CTRL	Learning/Lookup Function Global Configuration Register	—
0x8012_0404	LK_STATUS	Status Bits and Table Overflow Counter Register	—
0x8012_0408	LK_ADDR_CTRL	Address Table Transaction Control and Read/Write Address	—
0x8012_040C	LK_DATA_LO	Lower 32-Bit Data of Lookup Memory Entry	—
0x8012_0410	LK_DATA_HI	Higher 25-Bit Data of Lookup Memory Entry	—
0x8012_0414	LK_DATA_HI2	Higher2 12-Bit Data of Lookup Memory Entry	—
0x8012_0418	LK_LEARNCOUNT	Learned Address Count Register	—
0x8012_041C	LK_AGETIME	Period of the Aging Timer	—
0x8012_0480	MGMT_TAG_CONFIG	Management Tag Configuration Register	—
0x8012_0504	TSM_CONFIG	Timestamping Control Module Configuration Register	—
0x8012_0508	TSM_IRQ_STAT_ACK	Interrupt Status/Acknowledge Register	—
0x8012_050C	PTP_DOMAIN	Domain Number of PTP Frame	—
0x8012_0540 + 0x10 × n	PEERDELAY_Pn_T0 (n = 0 to 3)	Port n Peer Delay Value for Timer 0	—

Table 29.3 ETHSW register map (4 of 12)

Address	Register symbol	Register name	Write protect
0x8012_0544 + 0x10 × n	PEERDELAY_Pn_T1 (n = 0 to 3)	Port n Peer Delay Value for Timer 1	—
0x8012_05C0	TS_FIFO_STATUS	Transmit Timestamp FIFO Status Register	—
0x8012_05C4	TS_FIFO_READ_CTRL	Transmit Timestamp FIFO Read Control Register	—
0x8012_05C8	TS_FIFO_READ_TIMESTAMP	32-bit Timestamp Value Read from FIFO	—
0x8012_0600	INT_CONFIG	Interrupt Enable Configuration Register	—
0x8012_0604	INT_STAT_ACK	Interrupt Status/ACK Register	—
0x8012_0680 + 0x20 × n	ATIME_CTRLn (n = 0, 1)	Timer n Control Register	—
0x8012_0684 + 0x20 × n	ATIMEn (n = 0, 1)	Timer n Count Register	—
0x8012_0688 + 0x20 × n	ATIME_OFFSETn (n = 0, 1)	Timer n Offset Register	—
0x8012_068C + 0x20 × n	ATIME_EVT_PERIODn (n = 0, 1)	Timer n Periodic Event Register	—
0x8012_0690 + 0x20 × n	ATIME_CORRn (n = 0, 1)	Timer n Correction Period Register	—
0x8012_0694 + 0x20 × n	ATIME_INCn (n = 0, 1)	Timer n Increment Register	—
0x8012_0698 + 0x20 × n	ATIME_SECn (n = 0, 1)	Timer n Seconds Time Register	—
0x8012_069C + 0x20 × n	ATIME_OFFS_CORRn (n = 0, 1)	Timer n Offset Correction Counter Register	—
0x8012_0700	MDIO_CFG_STATUS	MDIO Configuration and Status Register	—
0x8012_0704	MDIO_COMMAND	MDIO PHY Command Register	—
0x8012_0708	MDIO_DATA	MDIO Data Register	—
0x8012_0800 + 0x400 × n	REV_Pn (n = 0 to 3)	Port n MAC Core Revision	—
0x8012_0808 + 0x400 × n	COMMAND_CONFIG_Pn (n = 0 to 3)	Port n Command Configuration Register	—
0x8012_080C + 0x400 × n	MAC_ADDR_0_Pn (n = 0 to 2)	Port n MAC Address Register 0	—
0x8012_0810 + 0x400 × n	MAC_ADDR_1_Pn (n = 0 to 2)	Port n MAC Address Register 1	—
0x8012_0814 + 0x400 × n	FRM_LENGTH_Pn (n = 0 to 3)	Port n Maximum Frame Length Register	—
0x8012_0818 + 0x400 × n	PAUSE_QUANT_Pn (n = 0 to 3)	Port n MAC Pause Quanta	—
0x8012_081C + 0x400 × n	MAC_LINK_QTRIG_Pn (n = 0 to 2)	Port n Trigger Event Configuration Register	—
0x8012_0830 + 0x400 × n	PTPCLOCKIDENTITY1_Pn (n = 0 to 2)	Port n PTP Clock Identity 1 Register	—
0x8012_0834 + 0x400 × n	PTPCLOCKIDENTITY2_Pn (n = 0 to 2)	Port n PTP Clock Identity 2 Register	—
0x8012_0838 + 0x400 × n	PTPAUTORESPONSE_Pn (n = 0 to 2)	Port n PTP Auto Response Register	—
0x8012_0840 + 0x400 × n	STATUS_Pn (n = 0 to 3)	Port n Status Register	—

Table 29.3 ETHSW register map (5 of 12)

Address	Register symbol	Register name	Write protect
0x8012_0844 + 0x400 × n	TX_IPG_LENGTH_Pn (n = 0 to 3)	Port n Transmit IPG Length Register	—
0x8012_0848 + 0x400 × n	EEE_CTL_STAT_Pn (n = 0 to 2)	Port n MAC EEE Functions Control and Status	—
0x8012_084C + 0x400 × n	EEE_IDLE_TIME_Pn (n = 0 to 2)	Port n EEE Idle Time Register	—
0x8012_0850 + 0x400 × n	EEE_TWSYS_TIME_Pn (n = 0 to 2)	Port n EEE Wake Up Time Register	—
0x8012_0854 + 0x400 × n	IDLE_SLOPE_Pn (n = 0 to 3)	Port n MAC Traffic Shaper Bandwidth Control	—
0x8012_0858 + 0x400 × n	CT_DELAY_Pn (n = 0 to 2)	Port n Cut-Through Delay Indication Register	—
0x8012_085C + 0x400 × n	BR_CONTROL_Pn (n = 0 to 2)	Port n 802.3br Frame Configuration Register	—
0x8012_0868 + 0x400 × n	AFRAMESTRANSMITTEDOK_Pn (n = 0 to 3)	Port n MAC Transmitted Valid Frame Count Register	—
0x8012_086C + 0x400 × n	AFRAMESRECEIVEDOK_Pn (n = 0 to 3)	Port n MAC Received Valid Frame Count Register	—
0x8012_0870 + 0x400 × n	AFRAMECHECKSEQUENCEERRORS_Pn (n = 0 to 3)	Port n MAC FCS Error Frame Count Register	—
0x8012_0874 + 0x400 × n	AALIGNMENTERRORS_Pn (n = 0 to 3)	Port n MAC Alignment Error Frame Count Register	—
0x8012_0878 + 0x400 × n	AOCTETSTRANSMITTEDOK_Pn (n = 0 to 3)	Port n MAC Transmitted Valid Frame Octets Register	—
0x8012_087C + 0x400 × n	AOCTETSRECEIVEDOK_Pn (n = 0 to 3)	Port n MAC Received Valid Frame Octets Register	—
0x8012_0880 + 0x400 × n	ATXPAUSEMACCTRLFRAMES_Pn (n = 0 to 3)	Port n MAC Transmitted Pause Frame Count Register	—
0x8012_0884 + 0x400 × n	ARXPAUSEMACCTRLFRAMES_Pn (n = 0 to 3)	Port n MAC Received Pause Frame Count Register	—
0x8012_0888 + 0x400 × n	IFINERRORS_Pn (n = 0 to 3)	Port n MAC Input Error Count Register	—
0x8012_088C + 0x400 × n	IFOUTERRORS_Pn (n = 0 to 3)	Port n MAC Output Error Count Register	—
0x8012_0890 + 0x400 × n	IFINUCASTPKTS_Pn (n = 0 to 3)	Port n MAC Received Unicast Frame Count Register	—
0x8012_0894 + 0x400 × n	IFINMULTICASTPKTS_Pn (n = 0 to 3)	Port n MAC Received Multicast Frame Count Register	—
0x8012_0898 + 0x400 × n	IFINBROADCASTPKTS_Pn (n = 0 to 3)	Port n MAC Received Broadcast Frame Count Register	—
0x8012_089C + 0x400 × n	IFOUTDISCARDS_Pn (n = 0 to 3)	Port n MAC Discarded Outbound Frame Count Register	—
0x8012_08A0 + 0x400 × n	IFOUTUCASTPKTS_Pn (n = 0 to 3)	Port n MAC Transmitted Unicast Frame Count Register	—
0x8012_08A4 + 0x400 × n	IFOUTMULTICASTPKTS_Pn (n = 0 to 3)	Port n MAC Transmitted Multicast Frame Count Register	—
0x8012_08A8 + 0x400 × n	IFOUTBROADCASTPKTS_Pn (n = 0 to 3)	Port n MAC Transmitted Broadcast Frame Count Register	—
0x8012_08AC + 0x400 × n	ETHERSTATSDROPEVENTS_Pn (n = 0 to 3)	Port n MAC Dropped Frame Count Register	—

Table 29.3 ETHSW register map (6 of 12)

Address	Register symbol	Register name	Write protect
0x8012_08B0 + 0x400 × n	ETHERSTATSOCTETS_Pn (n = 0 to 3)	Port n MAC All Frame Octets Register	—
0x8012_08B4 + 0x400 × n	ETHERSTATSPKTS_Pn (n = 0 to 3)	Port n MAC All Frame Count Register	—
0x8012_08B8 + 0x400 × n	ETHERSTATSUNDERSIZEPKTS_Pn (n = 0 to 3)	Port n MAC Too Short Frame Count Register	—
0x8012_08BC + 0x400 × n	ETHERSTATSOVERSIZEPKTS_Pn (n = 0 to 3)	Port n MAC Too Long Frame Count Register	—
0x8012_08C0 + 0x400 × n	ETHERSTATSPKTS64OCTETS_Pn (n = 0 to 3)	Port n MAC 64 Octets Frame Count Register	—
0x8012_08C4 + 0x400 × n	ETHERSTATSPKTS65TO127OCTETS_Pn (n = 0 to 3)	Port n MAC 65 to 127 Octets Frame Count Register	—
0x8012_08C8 + 0x400 × n	ETHERSTATSPKTS128TO255OCTETS_Pn (n = 0 to 3)	Port n MAC 128 to 255 Octets Frame Count Register	—
0x8012_08CC + 0x400 × n	ETHERSTATSPKTS256TO511OCTETS_Pn (n = 0 to 3)	Port n MAC 256 to 511 Octets Frame Count Register	—
0x8012_08D0 + 0x400 × n	ETHERSTATSPKTS512TO1023OCTETS_Pn (n = 0 to 3)	Port n MAC 512 to 1023 Octets Frame Count Register	—
0x8012_08D4 + 0x400 × n	ETHERSTATSPKTS1024TO1518OCTETS_Pn (n = 0 to 3)	Port n MAC 1024 to 1518 Octets Frame Count Register	—
0x8012_08D8 + 0x400 × n	ETHERSTATSPKTS1519TOXOCTETS_Pn (n = 0 to 3)	Port n MAC Over 1519 Octets Frame Count Register	—
0x8012_08DC + 0x400 × n	ETHERSTATSJABBERS_Pn (n = 0 to 3)	Port n MAC Jabbers Frame Count Register	—
0x8012_08E0 + 0x400 × n	ETHERSTATSFRAGMENTS_Pn (n = 0 to 3)	Port n MAC Fragment Frame Count Register	—
0x8012_08E8 + 0x400 × n	VLANRECEIVEDOK_Pn (n = 0 to 3)	Port n MAC Received VLAN Tagged Frame Count Register	—
0x8012_08F4 + 0x400 × n	VLANTRANSMITTEDOK_Pn (n = 0 to 3)	Port n MAC Transmitted VLAN Tagged Frame Count Register	—
0x8012_08F8 + 0x400 × n	FRAMESRETRANSMITTED_Pn (n = 0 to 3)	Port n MAC Retransmitted Frame Count Register	—
0x8012_0900 + 0x400 × n	STATS_HIWORD_Pn (n = 0 to 3)	Port n MAC Statistics Counter High Word Register	—
0x8012_0904 + 0x400 × n	STATS_CTRL_Pn (n = 0 to 3)	Port n MAC Statistics Control Register	—
0x8012_0908 + 0x400 × n	STATS_CLEAR_VALUELO_Pn (n = 0 to 3)	Port n MAC Statistics Clear Value Lower Register	—
0x8012_090C + 0x400 × n	STATS_CLEAR_VALUEHI_Pn (n = 0 to 3)	Port n MAC Statistics Clear Value Higher Register	—
0x8012_0910 + 0x400 × n	ADEFERRED_Pn (n = 0 to 3)	Port n MAC Deferred Count Register	—
0x8012_0914 + 0x400 × n	AMULTIPLECOLLISIONS_Pn (n = 0 to 3)	Port n MAC Multiple Collision Count Register	—
0x8012_0918 + 0x400 × n	ASINGLECOLLISIONS_Pn (n = 0 to 3)	Port n MAC Single Collision Count Register	—
0x8012_091C + 0x400 × n	ALATECOLLISIONS_Pn (n = 0 to 3)	Port n MAC Late Collision Count Register	—
0x8012_0920 + 0x400 × n	AEXCESSIVECOLLISIONS_Pn (n = 0 to 3)	Port n MAC Excessive Collision Count Register	—

Table 29.3 ETHSW register map (7 of 12)

Address	Register symbol	Register name	Write protect
0x8012_0924 + 0x400 × n	ACARRIERSENSEERRORS_Pn (n = 0 to 3)	Port n MAC Carrier Sense Error Count Register	—
0x8012_2000 + 0x400 × n + 0x028 × s	Pn_QSTMACUs (n = 0 to 2, s = 0 to 7)	Qci Stream Filter Table MAC Address Upper Part	—
0x8012_2004 + 0x400 × n + 0x028 × s	Pn_QSTMACDs (n = 0 to 2, s = 0 to 7)	Qci Stream Filter Table MAC Address Downer Part	—
0x8012_2008 + 0x400 × n + 0x028 × s	Pn_QSTMAMUs (n = 0 to 2, s = 0 to 7)	Qci Stream Filter Table MAC Address Mask Upper Part	—
0x8012_200C + 0x400 × n + 0x028 × s	Pn_QSTMAMDs (n = 0 to 2, s = 0 to 7)	Qci Stream Filter Table MAC Address Mask Downer Part	—
0x8012_2010 + 0x400 × n + 0x028 × s	Pn_QSFTVLs (n = 0 to 2, s = 0 to 7)	Qci Stream Filter Table VLAN	—
0x8012_2014 + 0x400 × n + 0x028 × s	Pn_QSFTVLMs (n = 0 to 2, s = 0 to 7)	Qci Stream Filter Table VLAN Mask	—
0x8012_2018 + 0x400 × n + 0x028 × s	Pn_QSFTBLs (n = 0 to 2, s = 0 to 7)	Qci Stream Filter Table SDU/Gate/Meter ID	—
0x8012_201C + 0x400 × n + 0x028 × s	Pn_QSMFCs (n = 0 to 2, s = 0 to 7)	Qci Stream Match Packet Count	—
0x8012_2020 + 0x400 × n + 0x028 × s	Pn_QMSPPCs (n = 0 to 2, s = 0 to 7)	Qci MSDU Passed Packet Count	—
0x8012_2024 + 0x400 × n + 0x028 × s	Pn_QMSRPCs (n = 0 to 2, s = 0 to 7)	Qci MSDU Reject Packet Count	—
0x8012_21E8 + 0x400 × n	Pn_QSEIS (n = 0 to 2)	Qci Stream Filter Error Interrupt Status (SDU Oversize)	—
0x8012_21EC + 0x400 × n	Pn_QSEIE (n = 0 to 2)	Qci Stream Filter Error Interrupt Enable	—
0x8012_21F0 + 0x400 × n	Pn_QSEID (n = 0 to 2)	Qci Stream Filter Error Interrupt Disable	—
0x8012_2200 + 0x400 × n	Pn_QGMOD (n = 0 to 2)	Qci Gate Mode Register	—
0x8012_2204 + 0x400 × n	Pn_QGPPC (n = 0 to 2)	Qci Gate (All) Passed Packet Count Port n	—
0x8012_2208 + 0x400 × n + 0x008 × g	Pn_QGDPCg (n = 0 to 2, g = 0 to 7)	Qci Gate g Dropped Packet Count Port n	—
0x8012_2244 + 0x400 × n	Pn_QGEIS (n = 0 to 2)	Qci Gate Error Interrupt Status	—
0x8012_2248 + 0x400 × n	Pn_QGEIE (n = 0 to 2)	Qci Gate Error Interrupt Enable	—
0x8012_224C + 0x400 × n	Pn_QGEID (n = 0 to 2)	Qci Gate Error Interrupt Disable	—
0x8012_2250 + 0x400 × n + 0x014 × m	Pn_QMDESCm (n = 0 to 2, m = 0 to 7)	Qci Port n Flow Meter m Descriptor Register	—

Table 29.3 ETHSW register map (8 of 12)

Address	Register symbol	Register name	Write protect
0x8012_2254 + 0x400 × n + 0x014 × m	Pn_QMCBSCm (n = 0 to 2, m = 0 to 7)	Qci Meter CBS Configuration Port n, Meter m	—
0x8012_2258 + 0x400 × n + 0x014 × m	Pn_QMCIRCm (n = 0 to 2, m = 0 to 7)	Qci Meter CIR Configuration n m	—
0x8012_225C + 0x400 × n + 0x014 × m	Pn_QMGPCm (n = 0 to 2, m = 0 to 7)	Qci Meter Green Packet Count	—
0x8012_2260 + 0x400 × n + 0x014 × m	Pn_QMRPCm (n = 0 to 2, m = 0 to 7)	Qci Meter Red Packet Count	—
0x8012_22F0 + 0x400 × n	Pn_QMEC (n = 0 to 2)	Qci Meter Enable Configuration	—
0x8012_22F4 + 0x400 × n	Pn_QMEIS (n = 0 to 2)	Qci Meter Error Interrupt Status	—
0x8012_22F8 + 0x400 × n	Pn_QMEIE (n = 0 to 2)	Qci Meter Error Interrupt Enable	—
0x8012_22FC + 0x400 × n	Pn_QMEID (n = 0 to 2)	Qci Meter Error Interrupt Disable	—
0x8012_2300 + 0x400 × n	Pn_PCP_REMAP (n = 0 to 2)	Port n VLAN Priority Code Point (PCP) Remap	—
0x8012_2304 + 0x400 × n	Pn_VLAN_TAG (n = 0 to 2)	Port n VLAN TAG Information for Priority Regeneration	—
0x8012_2308 + 0x400 × n	Pn_VLAN_MODE (n = 0 to 2)	Port n VLAN Mode	—
0x8012_230C + 0x400 × n	Pn_VIC_DROP_CNT (n = 0 to 2)	Port n VLAN Ingress Check Drop Frame Counter	—
0x8012_2328 + 0x400 × n	Pn_LOOKUP_HIT_CNT (n = 0 to 2)	Port n DST Address Lookup Hit Counter	—
0x8012_232C + 0x400 × n	Pn_ERROR_STATUS (n = 0 to 2)	Port n Frame Parser Runtime Error Status	—
0x8012_2330 + 0x400 × n	Pn_ERROR_MASK (n = 0 to 2)	Port n Frame Parser Runtime Error Mask	—
0x8012_23C0	CHANNEL_STATE	Enable/Disable State of Ingress Channels	—
0x8012_23C4	CHANNEL_ENABLE	Enable Operation of Channel	—
0x8012_23C8	CHANNEL_DISABLE	Disable and Reset Operation of Channel	—
0x8012_23CC + 0x004 × n	ASI_MEM_WDATAn (n = 0 to 3)	Memory Write Data Word n	—
0x8012_23DC	ASI_MEM_ADDR	Memory Address and R/W Control	—
0x8012_23E0 + 0x004 × n	ASI_MEM_RDATAn (n = 0 to 3)	Memory Read Data Word n	—
0x8012_3404	STATN_STATUS	Statistics Status Register	—
0x8012_3408	STATN_CONFIG	Statistics Configure Register	—
0x8012_340C	STATN_CONTROL	Statistics Control Register	—
0x8012_3410	STATN_CLEARVALUE_LO	Statistics Clear Value Lower Register	—
0x8012_3468 + 0x48 × n	ODISCn (n = 0 to 3)	Port n Discarded Outgoing Frame Count Register	—
0x8012_346C + 0x48 × n	IDISC_VLANn (n = 0 to 3)	Port n Discarded Incoming VLAN Tagged Frame Count Register	—

Table 29.3 ETHSW register map (9 of 12)

Address	Register symbol	Register name	Write protect
0x8012_3470 + 0x48 × n	IDISC_UNTAGGEDn (n = 0 to 3)	Port n Discarded Incoming VLAN Untagged Frame Count Register	—
0x8012_3474 + 0x48 × n	IDISC_BLOCKEDn (n = 0 to 3)	Port n Discarded Incoming Blocked Frame Count Register	—
0x8012_3478 + 0x48 × n	IDISC_ANYn (n = 0 to 3)	Port n Discarded Any Frame Count Register	—
0x8012_347C + 0x48 × n	IDISC_SRCFLTn (n = 0 to 2)	Port n Discarded Address Source Count Register	—
0x8012_3480 + 0x48 × n	TX_HOLD_REQ_CNTn (n = 0 to 2)	Port n TX Hold Request Count Register	—
0x8012_3484 + 0x48 × n	TX_FRAG_CNTn (n = 0 to 2)	Port n TX for Preemption Count Register	—
0x8012_3488 + 0x48 × n	RX_FRAG_CNTn (n = 0 to 2)	Port n RX Continuation Count Register	—
0x8012_348C + 0x48 × n	RX_ASSY_OK_CNTn (n = 0 to 2)	Port n RX Preempted Frame Success Count Register	—
0x8012_3490 + 0x48 × n	RX_ASSY_ERR_CNTn (n = 0 to 2)	Port n RX Preempted Frame Incorrect Count Register	—
0x8012_3494 + 0x48 × n	RX_SMD_ERR_CNTn (n = 0 to 2)	Port n RX SMD Frame Count Register	—
0x8012_3498 + 0x48 × n	TX_VERIFY_OK_CNTn (n = 0 to 2)	Port n TX VERIFY Frame Count Register	—
0x8012_349C + 0x48 × n	TX_RESPONSE_OK_CNTn (n = 0 to 2)	Port n TX RESPONSE Frame Count Register	—
0x8012_34A0 + 0x48 × n	RX_VERIFY_OK_CNTn (n = 0 to 2)	Port n RX VERIFY Frame Count Register	—
0x8012_34A4 + 0x48 × n	RX_RESPONSE_OK_CNTn (n = 0 to 2)	Port n RX RESPONSE Frame Count Register	—
0x8012_34A8 + 0x48 × n	RX_VERIFY_BAD_CNTn (n = 0 to 2)	Port n RX Error VERIFY Frame Count Register	—
0x8012_34AC + 0x48 × n	RX_RESPONSE_BAD_CNTn (n = 0 to 2)	Port n RX Error RESPONSE Frame Count Register	—
0x8012_3B00	MMCTL_OUT_CT	Cut-Through Register	—
0x8012_3B04	MMCTL_CTFL_P0_3_ENA	Cut-Through Frame Length Enable Register	—
0x8012_3B20 + 0x4 × n	MMCTL_YELLOW_BYTE_LENGTH_Pn (n = 0 to 2)	Port n Yellow Period Byte Length Register	—
0x8012_3B40 + 0x4 × n	MMCTL_POOLn_CTR (n = 0 to 1)	Memory Pool Counter	—
0x8012_3B60	MMCTL_POOL_GLOBAL	Memory Pool Configuration Register	—
0x8012_3B64	MMCTL_POOL_STATUS	Memory Pool Status Register	—
0x8012_3B68	MMCTL_POOL_QMAP	Queue MAP Register	—
0x8012_3B6C	MMCTL_QGATE	Queue Gate State Register	—
0x8012_3B70	MMCTL_QTRIG	Queue Trigger Register	—
0x8012_3B74	MMCTL_QFLUSH	Flush Event Select Register	—
0x8012_3B78	MMCTL_QCLOSED_STATUS_P0_3	Queue Closed Status Register	—
0x8012_3B80 + 0x4 × n	MMCTL_1FRAME_MODE_Pn (n = 0 to 2)	Port n 1-Frame Mode Configuration Register	—
0x8012_3BA0	MMCTL_P0_3_QUEUE_STATUS	Queue Status Indicator	—
0x8012_3BA8	MMCTL_P0_3_FLUSH_STATUS	Queue Flush Status Indicator	—

Table 29.3 ETHSW register map (10 of 12)

Address	Register symbol	Register name	Write protect
0x8012_3BB0	MMCTL_DLY_QTRIGGER_CTRL	Delayed Queue Trigger Control Register	—
0x8012_3BB4	MMCTL_PREEMPT_QUEUES	Preemptable Queues Configures Register	—
0x8012_3BB8	MMCTL_HOLD_CONTROL	Request Preemption Register	—
0x8012_3BBC	MMCTL_PREEMPT_STATUS	Preemption State Register	—
0x8012_3BC0 + 0x4 × n	MMCTL_CQF_CTRL_Pn (n = 0 to 3)	Port n Cyclic Queuing and Forwarding Control Register	—
0x8012_3BE0	MMCTL_P0_3_QCLOSED_NONEMPTY	Port Queue Status Register	—
0x8012_3BE8	MMCTL_PREEMPT_EXTRA	Frame Preemption Extra Configuration Register	—
0x8012_3C00	DLR_CONTROL	DLR Control Register	—
0x8012_3C04	DLR_STATUS	DLR Status Register	—
0x8012_3C08	DLR_ETH_TYP	DLR Ethernet Type Register	—
0x8012_3C0C	DLR_IRQ_CONTROL	DLR Interrupt Control Register	—
0x8012_3C10	DLR_IRQ_STAT_ACK	DLR Interrupt Status/ACK Register	—
0x8012_3C14	DLR_LOC_MAClo	DLR Local MAC Address Low Register	—
0x8012_3C18	DLR_LOC_MAChi	DLR Local MAC Address High Register	—
0x8012_3C20	DLR_SUPR_MAClo	DLR Supervisor MAC Address Low Register	—
0x8012_3C24	DLR_SUPR_MAChi	DLR Supervisor MAC Address High Register	—
0x8012_3C28	DLR_STATE_VLAN	DLR Ring Status/VLAN Register	—
0x8012_3C2C	DLR_BEC_TMOUT	DLR Beacon Timeout Register	—
0x8012_3C30	DLR_BEC_INTRVL	DLR Beacon Interval Register	—
0x8012_3C34	DLR_SURR_IPADR	DLR Supervisor IP Address Register	—
0x8012_3C38	DLR_ETH_STYP_VER	DLR Sub Type/Protocol Version Register	—
0x8012_3C3C	DLR_INV_TMOUT	DLR Beacon Timeout Timer Register	—
0x8012_3C40	DLR_SEQ_ID	DLR Sequence ID Register	—
0x8012_3C58	DLR_DSTlo	DLR Beacon Destination Address Low Register	—
0x8012_3C5C	DLR_DSThi	DLR Beacon Destination Address High Register	—
0x8012_3C60	DLR_RX_STAT0	DLR Received Frame Statistic Register 0	—
0x8012_3C64	DLR_RX_ERR_STAT0	DLR Received Frame Error Statistic Register 0	—
0x8012_3C6C	DLR_RX_LF_STAT0	DLR Received Frame Loop Filter Statistic Register 0	—
0x8012_3C70	DLR_RX_STAT1	DLR Received Frame Statistic Register 1	—
0x8012_3C74	DLR_RX_ERR_STAT1	DLR Received Frame Error Statistic Register 1	—
0x8012_3C7C	DLR_RX_LF_STAT1	DLR Received Frame Loop Filter Statistic Register 1	—
0x8012_3D00	PRP_CONFIG	PRP Configuration Register	—
0x8012_3D04	PRP_GROUP	PRP Port Group Register	—
0x8012_3D08	PRP_SUFFIX	PRP RCT Suffix	—
0x8012_3D0C	PRP_LANID	PRP LAN Identifier	—
0x8012_3D10	DUP_W	PRP Max Duplicate Detection Window Size	—
0x8012_3D14	PRP_AGETIME	PRP Aging Time Define Register	—
0x8012_3D18	PRP_IRQ_CONTROL	PRP Interrupt Control Register	—
0x8012_3D1C	PRP_IRQ_STAT_ACK	PRP Interrupt Status/ACK Register	—
0x8012_3D20	RM_ADDR_CTRL	PRP History Memory Transactions Control Register	—

Table 29.3 ETHSW register map (11 of 12)

Address	Register symbol	Register name	Write protect
0x8012_3D24	RM_DATA	PRP Memory Data Register	—
0x8012_3D28	RM_DATA_HI	PRP Memory Data Higher Register	—
0x8012_3D2C	RM_STATUS	PRP Memory Controller Status Indication	—
0x8012_3D30	TxSeqTooLate	PRP Frame Transmission Retrieval of Failed Sequence	—
0x8012_3D34	CntErrWrongLanA	PRP Wrong ID LAN-A Count Register	—
0x8012_3D38	CntErrWrongLanB	PRP Wrong ID LAN-B Count Register	—
0x8012_3D3C	CntDupLanA	PRP Duplicate LAN-A Count Register	—
0x8012_3D40	CntDupLanB	PRP Duplicate LAN-B Count Register	—
0x8012_3D44	CntOutOfSeqLowA	PRP Sequence Error Low LAN-A Count Register	—
0x8012_3D48	CntOutOfSeqLowB	PRP Sequence Error Low LAN-B Count Register	—
0x8012_3D4C	CntOutOfSeqA	PRP Sequence Error LAN-A Count Register	—
0x8012_3D50	CntOutOfSeqB	PRP Sequence Error LAN-B Count Register	—
0x8012_3D54	CntAcceptA	PRP Valid Frame LAN-A Count Register	—
0x8012_3D58	CntAcceptB	PRP Valid Frame LAN-B Count Register	—
0x8012_3D5C	CntMissing	PRP Drop History Adjustment Count	—
0x8012_3E00	HUB_CONFIG	HUB Configuration Register	—
0x8012_3E04	HUB_GROUP	HUB Port Group Register	—
0x8012_3E08	HUB_DEFPORT	HUB Default Port Selection Register	—
0x8012_3E0C	HUB_TRIGGER_IMMEDIATE	HUB Transmission Trigger Immediate Register	—
0x8012_3E10	HUB_TRIGGER_AT	HUB Transmission Trigger At Register	—
0x8012_3E14	HUB_TTIME	HUB Transmission Time Define Register	—
0x8012_3E18	HUB_IRQ_CONTROL	HUB Interrupt Control Register	—
0x8012_3E1C	HUB_IRQ_STAT_ACK	HUB Interrupt Status/ACK Register	—
0x8012_3E20	HUB_STATUS	HUB Status Register	—
0x8012_3E24	HUB_OPORT_STATUS	HUB Output Port Status Register	—
0x8012_3E80	TDMA_CONFIG	TDMA Configuration Register	—
0x8012_3E84	TDMA_ENA_CTRL	TDMA Scheduling Enable Control Register	—
0x8012_3E88	TDMA_START	TDMA Start Time Set Register	—
0x8012_3E8C	TDMA_MODULO	TDMA System Timer Modulo	—
0x8012_3E90	TDMA_CYCLE	TDMA Periodic Cycle Set Register	—
0x8012_3E94	TCV_SEQ_ADDR	TCV Sequence Address Register	—
0x8012_3E98	TCV_SEQ_CTRL	TCV Sequence Table Control Register	—
0x8012_3E9C	TCV_SEQ_LAST	TCV Sequence Last Entry	—
0x8012_3EA0	TCV_D_ADDR	TCV Data Address Register	—
0x8012_3EA4	TCV_D_OFFSET	TCV Data Offset Register	—
0x8012_3EA8	TCV_D_CTRL	TCV Data Control Register	—
0x8012_3EAC	TDMA_CTR0	TDMA Counter 0	—
0x8012_3EB0	TDMA_CTR1	TDMA Counter 1	—
0x8012_3EB4	TDMA_TCV_START	TDMA TCV Sequence Entry Start	—
0x8012_3EB8	TIME_LOAD_NEXT	TDMA Calculated Next Loading Time	—
0x8012_3EBC	TDMA_IRQ_CONTROL	TDMA IRQ Control Register	—

Table 29.3 ETHSW register map (12 of 12)

Address	Register symbol	Register name	Write protect
0x8012_3EC0	TDMA_IRQ_STAT_ACK	TDMA IRQ Status/ACK Register	—
0x8012_3EC4	TDMA_GPIO	TDMA GPIO Register	—
0x8012_3F00 + 0x4 × n	RXMATCH_CONFIGn (n = 0 to 3)	RX Pattern Matcher Configuration for Port n	—
0x8012_3F40 + 0x4 × n	PATTERN_CTRLn (n = 0 to 11)	RX Pattern Matcher Function Control for Pattern n	—
0x8012_3F80	PATTERN_IRQ_CONTROL	RX Pattern Matcher Interrupt Control Register	—
0x8012_3F84	PATTERN_IRQ_STAT_ACK	RX Pattern Matcher Interrupt Status/ACK Register	—
0x8012_3F88	PTRN_VLANID	Custom VLAN ID Register	—
0x8012_3F8C	PATTERN_SEL	RX Pattern Number Selection Register	—
0x8012_3FC0	PTRN_CMP_30	Pattern Compare Value Bytes 3 .. 0	—
0x8012_3FC4	PTRN_CMP_74	Pattern Compare Value Bytes 7 .. 4	—
0x8012_3FC8	PTRN_CMP_118	Pattern Compare Value Bytes 11 .. 8	—
0x8012_3FD0	PTRN_MSK_30	Pattern Mask for Bytes 3 .. 0	—
0x8012_3FD4	PTRN_MSK_74	Pattern Mask for Bytes 7 .. 4	—
0x8012_3FD8	PTRN_MSK_118	Pattern Mask for Bytes 11 .. 8	—
0x8011_0800 + 0x100 × n	SWTMENn (n = 0 to 3)	PTP Timer Pulse Output Enable n Register	—
0x8011_0804 + 0x100 × n	SWTMSTSECn (n = 0 to 3)	PTP Timer Pulse Start Second n Register	—
0x8011_0808 + 0x100 × n	SWTMSTNSn (n = 0 to 3)	PTP Timer Pulse Start Nanosecond n Register	—
0x8011_080C + 0x100 × n	SWTMPSECn (n = 0 to 3)	PTP Timer Pulse Period Second n Register	—
0x8011_0810 + 0x100 × n	SWTMPNSn (n = 0 to 3)	PTP Timer Pulse Period Nanosecond n Register	—
0x8011_0814 + 0x100 × n	SWTMWTHn (n = 0 to 3)	PTP Timer Pulse Width n Register	—
0x8011_0818 + 0x100 × n	SWTMMAXPn (n = 0 to 3)	PTP Timer Pulse Max Second n Register	—
0x8011_081C + 0x100 × n	SWTMLATSECn (n = 0 to 3)	PTP Timer Pulse Latch Second n Register	—
0x8011_0820 + 0x100 × n	SWTMLATNSn (n = 0 to 3)	PTP Timer Pulse Latch Nanosecond n Register	—
0x8011_0404	SWPTPOUTSEL	ETHSW_PTPOUT Select Register	PRCMD

Table 29.4 ETHSW related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
SWTMENn (n = 0 to 3), SWTMSTSECn (n = 0 to 3), SWTMSTNSn (n = 0 to 3), SWTMPSECn (n = 0 to 3), SWTMPNSn (n = 0 to 3), SWTMWTHn (n = 0 to 3), SWTMMAXPn (n = 0 to 3), SWTMLATSECn (n = 0 to 3), SWTMLATNSn (n = 0 to 3), SWPTPOUTSEL	MRCTLE.MRTCLE05	MSTPCRE.MSTPCRE03	SLVACCCTL6.ETHSS_SL
Other than the above	MRCTLE.MRTCLE02	MSTPCRE.MSTPCRE01	SLVACCCTL6.ETHSW_SL

29.3 Register Description

ETHSW has "Ethernet Switch" and "PTP pulse generator" (See previous figure). Registers for setting "Ethernet Switch" and "PTP pulse generator" are described in following sections.

Before setting these registers, setting registers SWCTRL and SWDUPC in Network SS is necessary. See section 27.3.7. SWCTRL : Switch Core Control Register and section 27.3.8. SWDUPC : Switch Core Duplex Mode Register.

The application must not assume that registers containing more than one field result in atomic write operations when modifying multiple fields. One example is the BR_CONTROL_Pn register in which PREEMPT_ENA should be written last when enabling preemption to avoid false start conditions (for instance, setting VERIFY_DIS to 1 together with PREEMPT_ENA can result in a VERIFY frame to be transmitted).

29.3.1 Switch Base Registers

29.3.1.1 REVISION : Switch Core Version Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0000

Bit position:	31																												0
Bit field:	REV[31:0]																												
Value after reset:	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	REV[31:0]	Revision Fixed value 0x42000600	R

29.3.1.2 SCRATCH : Scratch Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x04

Bit position:	31																												0
Bit field:	SCRATCH[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SCRATCH[31:0]	The Scratch Register provides a memory location to test the register access. It returns all written data in inverted form.	R/W

29.3.1.3 PORT_ENA : Port Enable Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x08

Bit position:	31											19				16												3		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	RXENA[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	TXENA[3:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	TXENA[3:0]	Transmit Enable Mask One bit per port. The switch will transmit frames only to ports enabled in this mask. Frames directed to a disabled port are discarded. The exception is all BPDU frames originating from the management port, to which this mask does not apply, allowing to transmit frames to disabled ports (see section 29.3.1.9. MGMT_CONFIG : Management Configuration Register).	R/W
15:4	—	The read values are undefined. The write value should be 0.	R/W
19:16	RXENA[3:0]	Receive Enable Mask One bit per port. The switch will serve only input ports that are enabled in this mask.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The enable bits (both per port bits) reset defaults are controlled by STRAP_SX_ENB bit and STRAP_HUB_ENB bit of SWCTRL register.

STRAP_SX_ENB = 0, STRAP_HUB_ENB = 0: RXENA = 0x0, TXENA = 0x0.

STRAP_SX_ENB = 0, STRAP_HUB_ENB = 1: RXENA = 0x3, TXENA = 0x3.

STRAP_SX_ENB = 1, STRAP_HUB_ENB = 0: RXENA = 0x7, TXENA = 0x7.

STRAP_SX_ENB = 1, STRAP_HUB_ENB = 1: RXENA = 0xF, TXENA = 0xF.

STRAP_SX_ENB and STRAP_HUB_ENB are described in section 27.3.7. SWCTRL : Switch Core Control Register.

29.3.1.4 UCAST_DEFAULT_MASK0 : Unicast Default Mask Register 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x0C

Bit position:	31																												3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UCASTDM[3:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1

Bit	Symbol	Function	R/W
3:0	UCASTDM[3:0]	Default Unicast Resolution 1 bit per port. For unicast frame flooding resolution, the default output port list if the destination address is not found in the MAC Lookup table.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

The UCAST_DEFAULT_MASK0 register is used to mask unicast frames of each port of the ethernet switch. When the destination address is unicast and not found in the address table, the frame is transferred to the ports where the mask is enabled.

29.3.1.5 VLAN_VERIFY : Verify VLAN Domain Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x10

Bit position:	31											19			16													3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	VLANDISC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	VLANVERI[3:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	VLANVERI[3:0]	Verify VLAN Domain 1 bit per port (bit 0 = port 0) When this bit is set to 1 (enabled), a frame is accepted from the port as valid only when the input port is a member of the VLAN domain of the frame. When this bit is set to 0 (disabled), frames are forwarded without VLAN domain checking.	R/W
15:4	—	The read values are undefined. The write value should be 0.	R/W
19:16	VLANDISC[3:0]	Discard Unknown 1 bit per port (bit 16 = port 0). When set and a frame is received with a VLAN ID that is unknown, or has no VLAN tag, the frame is discarded and not forwarded (the default broadcast is ignored).	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.6 BCAST_DEFAULT_MASK0 : Broadcast Default Mask Register 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x14

Bit position:	31																											3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BCASTDM[3:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1

Bit	Symbol	Function	R/W
3:0	BCASTDM[3:0]	Default Broadcast Resolution 1 bit per port. For broadcast/flooding resolution, the default output port list.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

The BCAST_DEFAULT_MASK0 register is used to mask broadcast frames of each port of the ethernet switch. When the destination address is broadcast, the frame is transferred to the ports given in this mask.

29.3.1.7 MCAST_DEFAULT_MASK0 : Multicast Default Mask Register 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x18

Bit position:	31																											3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCASTDM[3:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1

Bit	Symbol	Function	R/W
3:0	MCASTDM[3:0]	Default Multicast Resolution 1 bit per port. Used for multicast/flooding resolution. The default output port list is used instead of the BCAST_DEFAULT_MASK, when the received frame carries a multicast address.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

The MCAST_DEFAULT_MASK0 register is used to mask multicast frames of each ports of the ethernet switch. When the destination address is multicast, the frame is transferred to the ports given in this mask.

29.3.1.8 INPUT_LEARN_BLOCK : Input Learning Block Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	LEARNDIS[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	BLOCKEN[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	BLOCKEN[3:0]	Blocking Enable 1 bit per port, enable blocking (bit 0 = port 0)	R/W
15:4	—	The read values are undefined. The write value should be 0.	R/W
19:16	LEARNDIS[3:0]	Learning Disable 1 bit per port, disable learning (bit 16 = port 0)	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The INPUT_LEARN_BLOCK register is used to set the address learning function and frame blocking function of each port of the ethernet switch.

When blocking is enabled for a port (bit = 1), only Bridge Protocol data units are accepted on that input, all other frames are discarded. When learning is disabled for a port (bit = 1), source address lookup and learning do not occur. Both functions operate independently from each other.

Note: Source addresses from incoming BPDU frames are not learned, independent of these settings (to avoid table updates if for example, loops exist).

29.3.1.9 MGMT_CONFIG : Management Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PORTMASK[3:0]				PRIORITY[2:0]			—	—	—	MGMT_DISC	MGMT_EN	DISCARD	ENABLE	MSG_TRANS	—	PORT[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	x	x	x	x	x	1	0	0	x	0	0	1	1

Bit	Symbol	Function	R/W
3:0	PORT[3:0]	The Port number of the port that should act as a management port. Keep the initial value.	R/W
4	—	The read value is undefined. The write value should be 0.	R/W
5	MSG_TRANS	Set (latched) when a BPDU message is transmitted from the management port to any output port. This bit can be used for handshaking to indicate that the port mask bits are used and can now be changed again by setting it to 0. The bit is not asserted for normal frames and is reset by writing into the register. Will assert regardless of whether the frame was transmitted successfully or not. The control bit MGMT_EN must be set to 1.	R/W
6	ENABLE	If set, all Bridge Protocol Frames (BPDU) are forwarded exclusively to the management port specified in bits [3:0]. If cleared, Bridge Protocol Frames (BPDU) are forwarded as any other frame or discarded if the discard bit is set.	R/W

Bit	Symbol	Function	R/W
7	DISCARD	If set, BPDU frames are discarded always.	R/W
8	MGMT_EN	If set, BPDU frames received at the management port are forwarded to the ports given in the portmask given in this register, bypassing the normal forwarding decisions (except forced forwarding). If cleared, Bridge Protocol Frames (BPDU) are forwarded as any other frame or discarded if the discard bit is set.	R/W
9	MGMT_DISC	This bit is the same as DISCARD (bit 7) but for the management port.	R/W
12:10	—	The read values are undefined. The write value should be 0.	R/W
15:13	PRIORITY[2:0]	Priority to use for transmitted BPDU frames if non-zero. The setting is ignored if it is 0 (normal priority resolution operates). This bit can be used to put a management frame in a high-priority output queue for fast delivery.	R/W
19:16	PORTMASK[3:0]	Portmask for transmission of management frames. When the management port transmits a frame to the switch, it is forwarded to all ports in this portmask (bit 16 = port 0, bit 17 = port 1, …, bit 19 = port 3). When the mask is all 0, the frame is forwarded normally (either from the MAC address lookup or multicast flooding). Forced forwarding takes precedence over this portmask and management frames are forwarded as specified by the forced forwarding portmask if active. When using forced forwarding, this portmask must be written with all 0.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The MGMT_CONFIG register is used to configure the bridge management port of the ethernet switch and to enable the management port that receives bridge protocol frames. It is necessary to set the internal interface port (port 3) in the management port.

29.3.1.10 MODE_CONFIG : Mode Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	STATSRESET	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CUT_THRU_EN[3:0]				—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	—	The read values are undefined. The write value should be 0.	R/W
11:8	CUT_THRU_EN[3:0]	Port Cut through Support Enable One bit per port. Bit 8 = port 0, bit 9 = port 1, bit 10 = port 2, bit 11 = port 3. When a port is enabled for cut through, a frame received is forwarded to the destination before completion of the frame reception. Cut through is possible only if both ports (where frame is received and is forwarded) have the cut through bit set. Otherwise, the frame is forwarded in normal store and forward behavior.	R/W
30:12	—	The read values are undefined. The write value should be 0.	R/W
31	STATSRESET	Reset Statistics Counters Command. When set during a write, all statistics counters are cleared. When this bit is set to 1, the write values of all other bits are ignored. Set this bit to 0 when writing other bits.	R/W

The MODE_CONFIG register is used to reset the statistics counter in the ethernet switch.

29.3.1.11 VLAN_IN_MODE : VLAN Input Manipulation Mode Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	P3VLANINMD[1:0]		P2VLANINMD[1:0]		P1VLANINMD[1:0]		P0VLANINMD[1:0]	
Value after reset:	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	P0VLANINMD[1:0]	Port 0 Define Behavior of VLAN Input Manipulation Function 0 0: Mode 1 0 1: Mode 2 1 0: Mode 3 1 1: Mode 4	R/W
3:2	P1VLANINMD[1:0]	Port 1 Define Behavior of VLAN Input Manipulation Function 0 0: Mode 1 0 1: Mode 2 1 0: Mode 3 1 1: Mode 4	R/W
5:4	P2VLANINMD[1:0]	Port 2 Define Behavior of VLAN Input Manipulation Function 0 0: Mode 1 0 1: Mode 2 1 0: Mode 3 1 1: Mode 4	R/W
7:6	P3VLANINMD[1:0]	Port3 Define Behavior of VLAN Input Manipulation Function 0 0: Mode 1 0 1: Mode 2 1 0: Mode 3 1 1: Mode 4	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.12 VLAN_OUT_MODE : VLAN Output Manipulation Mode Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	P3VLANOUTMD[1:0]		P2VLANOUTMD[1:0]		P1VLANOUTMD[1:0]		P0VLANOUTMD[1:0]	
Value after reset:	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	P0VLANOUTMD[1:0]	Port 0 Define Behavior of VLAN Output Manipulation Function 0 0: No output manipulation 0 1: Mode 1, Strip Mode 1 0: Mode 2, Tag Through 1 1: Mode 3, Transparent	R/W
3:2	P1VLANOUTMD[1:0]	Port 1 Define Behavior of VLAN Output Manipulation Function 0 0: No output manipulation 0 1: Mode 1, Strip Mode 1 0: Mode 2, Tag Through 1 1: Mode 3, Transparent	R/W
5:4	P2VLANOUTMD[1:0]	Port 2 Define Behavior of VLAN Output Manipulation Function 0 0: No output manipulation 0 1: Mode 1, Strip Mode 1 0: Mode 2, Tag Through 1 1: Mode 3, Transparent	R/W
7:6	P3VLANOUTMD[1:0]	Port 3 Define Behavior of VLAN Output Manipulation Function 0 0: No output manipulation 0 1: Mode 1, Strip Mode 1 0: Mode 2, Tag Through 1 1: Mode 3, Transparent	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.13 VLAN_IN_MODE_ENA : VLAN Input Mode Enable Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x30

Bit position:	31																												3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLANINMDEN[3:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	VLANINMDEN[3:0]	Enable the input processing according to the VLAN_IN_MODE for a port (1 bit per port).	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.14 VLAN_TAG_ID : VLAN Tag ID Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x34

Bit position:	31																15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLANTAGID[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	VLANTAGID[15:0]	The VLAN type field (TPID) value to expect to identify a VLAN tagged frame. Default value is 0x8100.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

The VLAN_TAG_ID register is used to set a VLAN tag ID to identify a VLAN tagged frame. In valid IEEE802.1Q, a VLAN tag ID is defined as 0x8100.

29.3.1.15 BCAST_STORM_LIMIT : Broadcast Storm Protection Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x38

Bit position:	31																16	15													0
Bit field:	BCASTLIMIT[15:0]																	TMOUT[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TMOUT[15:0]	Timeout in steps of 65535 switch operating clock cycles.	R/W
31:16	BCASTLIMIT[15:0]	Number of broadcast frames (-1) that can be accepted on a port during a timeout period. If more are received, they are discarded. The counter is implemented per port independently. However, the limit is used for all ports. When 0, no limit is set. The value is set 1 less (for example, set 9 to allow 10 frames).	R/W

29.3.1.16 MCAST_STORM_LIMIT : Multicast Storm Protection Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C

Bit position:	31																16														0
Bit field:	MCASTLIMIT[15:0]																	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	—	The read values are undefined. The write value should be 0.	R/W
31:16	MCASTLIMIT[15:0]	Number of multicast frames (-1) that can be accepted on a port during a timeout period. If more are received, they are discarded. The counter is implemented per port independently. However, the limit is used for all ports. When 0, no limit is set.	R/W

The timeout setting is taken from the value configured in BCAST_STORM_LIMIT[15:0]. Here only the number of acceptable multicast frames is configured.

29.3.1.17 MIRROR_CONTROL : Port Mirroring Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	EG_DA_MATCH	EG_SA_MATCH	ING_DA_MATCH	ING_SA_MATCH	EG_MAP_EN	ING_MAP_EN	MIRROR_EN	—	—	PORT[1:0]	
Value after reset:	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PORT[1:0]	The port number of the port that acts as the mirror port and receives all mirrored frames. Valid setting range is 0 to 3.	R/W

Bit	Symbol	Function	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	MIRROR_EN	Mirroring 0: Disabled 1: Enabled	R/W
5	ING_MAP_EN	If set, the ingress map is enabled (MIRROR_ING_MAP).	R/W
6	EG_MAP_EN	If set, the egress map is enabled (MIRROR_EG_MAP).	R/W
7	ING_SA_MATCH	If set, only frames received on an ingress port with a source address matching the value programmed in MIRROR_ISRC registers are mirrored. Other frames are not mirrored.	R/W
8	ING_DA_MATCH	If set, only frames received on an ingress port with a destination address matching the value programmed in MIRROR_IDST registers are mirrored. Other frames are not mirrored.	R/W
9	EG_SA_MATCH	If set, only frames transmitted on an egress port with a source address matching the value programmed in MIRROR_ESRC registers are mirrored. Other frames are not mirrored.	R/W
10	EG_DA_MATCH	If set, only frames transmitted on an egress port with a destination address matching the value programmed in MIRROR_EDST registers are mirrored. Other frames are not mirrored.	R/W
31:11	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.18 MIRROR_EG_MAP : Port Mirroring Egress Port Definition Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x44

Bit position:	31																												3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EMAP[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	EMAP[3:0]	Port Mirroring Egress Port Definitions 1 bit per port. When this bit is 1 (enabled), frames destined to the port(s) are mirrored to the mirror port.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.19 MIRROR_ING_MAP : Port Mirroring Ingress Port Definition Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x48

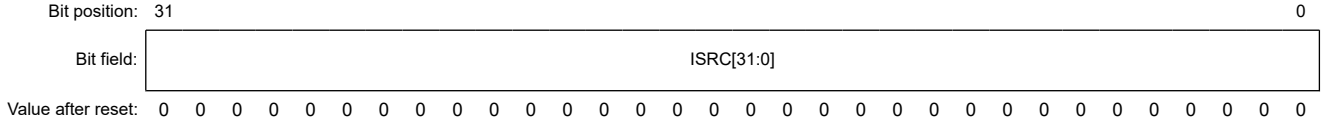
Bit position:	31																												3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IMAP[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IMAP[3:0]	Port Mirroring Ingress Port Definitions 1 bit per port. When this bit is 1 (enabled), frames from the port(s) are mirrored to the mirror port.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.20 MIRROR_ISRC_0 : Ingress Source MAC Address for Mirror Filtering Register 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x4C

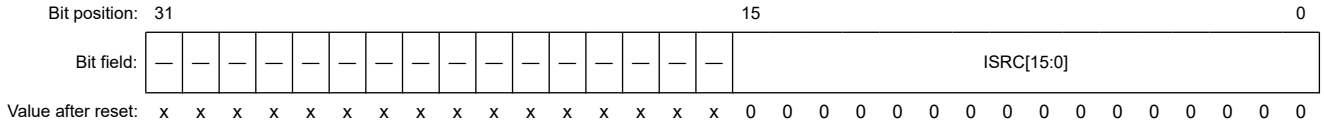


Bit	Symbol	Function	R/W
31:0	ISRC[31:0]	Ingress Source MAC Address for Mirror Filtering Lowest 32-bit of address.	R/W

29.3.1.21 MIRROR_ISRC_1 : Ingress Source MAC Address for Mirror Filtering Register 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x50

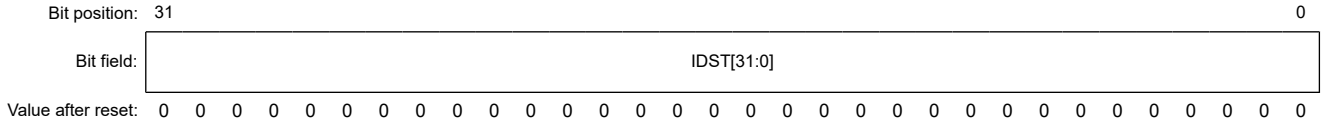


Bit	Symbol	Function	R/W
15:0	ISRC[15:0]	Ingress Source MAC Address for Mirror Filtering Upper 16-bit of address.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.22 MIRROR_IDST_0 : Ingress Destination MAC Address for Mirror Filtering Register 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x54

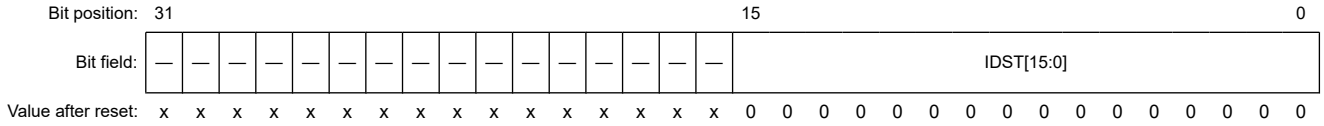


Bit	Symbol	Function	R/W
31:0	IDST[31:0]	Ingress Destination MAC Address for Mirror Filtering Lowest 32-bit of address.	R/W

29.3.1.23 MIRROR_IDST_1 : Ingress Destination MAC Address for Mirror Filtering Register 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x58



Bit	Symbol	Function	R/W
15:0	IDST[15:0]	Ingress Destination MAC Address for Mirror Filtering Upper 16-bit of address.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.24 MIRROR_ESRC_0 : Egress Source MAC Address for Mirror Filtering Register 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x5C

Bit position:	31																												0
Bit field:	ESRC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ESRC[31:0]	Egress Source MAC Address for Mirror Filtering Lowest 32-bit of address.	R/W

29.3.1.25 MIRROR_ESRC_1 : Egress Source MAC Address for Mirror Filtering Register 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x60

Bit position:	31															15													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ESRC[15:0]													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ESRC[15:0]	Egress Source MAC Address for Mirror Filtering Upper 16-bit of address.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.26 MIRROR_EDST_0 : Egress Destination MAC Address for Mirror Filtering Register 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x64

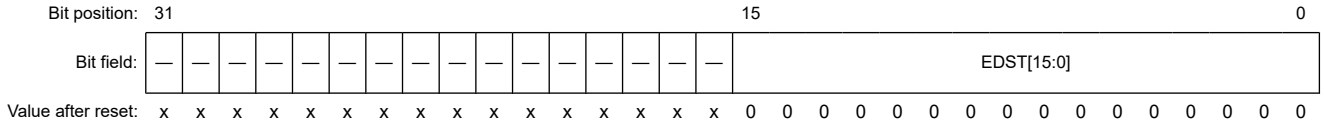
Bit position:	31																												0
Bit field:	EDST[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	EDST[31:0]	Egress Destination MAC Address for Mirror Filtering Lowest 32-bit of address.	R/W

29.3.1.27 MIRROR_EDST_1 : Egress Destination MAC Address for Mirror Filtering Register 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x68

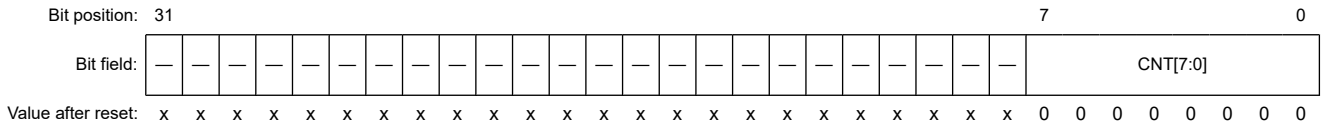


Bit	Symbol	Function	R/W
15:0	EDST[15:0]	Egress Destination MAC Address for Mirror Filtering Upper 16-bit of address.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.28 MIRROR_CNT : Mirror Filtering Count Value Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x6C

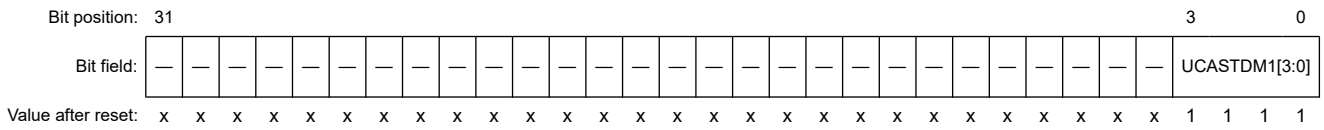


Bit	Symbol	Function	R/W
7:0	CNT[7:0]	Count Value for Mirror Filtering Every n-th frame is forwarded to the mirror port if enabled. A value of 1 or 0 means every frame.	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.29 UCAST_DEFAULT_MASK1 : Unicast Default Mask Register 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x70



Bit	Symbol	Function	R/W
3:0	UCASTDM1[3:0]	Default Unicast Resolution Mask 1 1 bit per port. For unicast frame flooding resolution, the default output port list if the destination address is not found in the MAC lookup table.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.30 BCAST_DEFAULT_MASK1 : Broadcast Default Mask Register 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x74

Bit position:	31																											3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BCASTDM1[3:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1

Bit	Symbol	Function	R/W
3:0	BCASTDM1[3:0]	Default Broadcast Resolution Mask 1 1 bit per port. For broadcast/flooding resolution, the default output port list.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.31 MCAST_DEFAULT_MASK1 : Multicast Default Mask Register 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x78

Bit position:	31																											3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCASTDM1[3:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1

Bit	Symbol	Function	R/W
3:0	MCASTDM1[3:0]	Default Multicast Resolution Mask 1 1 bit per port. Used for broadcast/flooding resolution. The default output port list is used instead of the BCAST_DEFAULT_MASK1, when the received frame carries a multicast address.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.32 PORT_XCAST_MASK_SEL : Port Mask Select Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x7C

Bit position:	31																											3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSEL[3:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	MSEL[3:0]	Mask Select 1 bit per port that selects which UCAST, MCAST, and BCAST Default Mask register set to use for forwarding the respective frames. This facilitates separation of a bridge domain between various switch ports so that broadcast, multicast, or unknown destination addresses flooding do not cross over two networks when VLAN verification is not used. If VLAN verification is used, then it is better to use the VLAN identification function to separate traffic between networks. 0: XCAST_DEFAULT_MASK0 registers are selected 1: XCAST_DEFAULT_MASK1 registers are selected	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.33 QMGR_ST_MINCELLS : Minimum Memory Cell Statistics Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x88

Bit position:	31														10											0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STMINCELLS[10:0]				
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	STMINCELLS[10:0]	Minimum Free Cell Indication Statistic providing the lowest number of free cells reached in memory during operation since this statistic was last cleared. The value is reset to the maximum when a write to the register with any value is performed.	R/W
31:11	—	The read values are undefined. The write value should be 0.	R/W

The QMGR_ST_MINCELLS register is used to indicate the minimum number of free cells in the memory of the ethernet switch.

29.3.1.34 QMGR_RED_MIN4 : RED Minimum Threshold Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x94

Bit position:	31																									0
Bit field:	CFGRED_MINTH4[31:0]																									
Value after reset:	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
31:0	CFGRED_MINTH4[31:0]	Random Early Detection (RED) Minimum Threshold for Queues 0 to 3 An 8-bit value per queue, global for all ports: CFGRED_MINTH4[7:0]: Queue 0 CFGRED_MINTH4[15:8]: Queue 1 CFGRED_MINTH4[23:16]: Queue 2 CFGRED_MINTH4[31:24]: Queue 3	R/W

29.3.1.35 QMGR_RED_MAX4 : RED Maximum Threshold Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x98

Bit position:	31																									0
Bit field:	CFGRED_MAXTH4[31:0]																									
Value after reset:	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	1

Bit	Symbol	Function	R/W
31:0	CFGRED_MAXTH4[31:0]	Random Early Detection (RED) Maximum Threshold for Queues 0 to 3 An 8-bit value per queue, global for all ports: CFGRED_MAXTH4[7:0]: Queue 0 CFGRED_MAXTH4[15:8]: Queue 1 CFGRED_MAXTH4[23:16]: Queue 2 CFGRED_MAXTH4[31:24]: Queue 3	R/W

29.3.1.36 QMGR_RED_CONFIG : RED Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x9C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	GACTIVITY_EN	—	—	—	—	QUEUE_RED_EN[3:0]			
Value after reset:	x	x	x	x	x	x	x	0	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	QUEUE_RED_EN[3:0]	Enable Random Early Detection (RED) (when this bit is 1) or Tail Drop (when this bit is 0) congestion management for a queue. One bit per queue: bit 0 = queue 0, bit 1 = queue 1, and so on. Per queue setting, but global for all queues of all ports.	R/W
7:4	—	The read values are undefined. The write value should be 0.	R/W
8	GACTIVITY_EN	Enable Averaging on Global Switch Activity (when this bit is 1) or on port local activity (when this bit is 0) only.	R/W
31:9	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.37 IMC_STATUS : Input Memory Controller Status Register

Base address: ETHSW = 0x8012_0000

Offset address: 0xA0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	MEM_FULL	DE_INIT	DE_ERR	CF_ERR	CELLS_AVAILABLE[23:0]							
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CELLS_AVAILABLE[23:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	CELLS_AVAILABLE[23:0]	Total number of memory cells (128-byte units) available in the shared memory (real time).	R
24	CF_ERR	Cell Factory Empty Error Indication that memory overflow occurred. There are more memory cells requested than available. This is a fatal error that must not occur during normal operation as it may indicate memory leakage. This bit is cleared by reading it.	R
25	DE_ERR	Deallocation Error Indication that memory cells cannot be deallocated and are lost for further usage. This is a fatal error that must not occur during normal operation as it may indicate memory leakage. This bit is cleared by reading it.	R
26	DE_INIT	Asserts during Memory Initialization (deallocation module) Asserts after reset as long as memory initialization is ongoing and then clears when the memory is ready for normal operation. The switch must not be enabled before the memory is ready for operation.	R

Bit	Symbol	Function	R/W
27	MEM_FULL	Latched Indication that Memory is or was Full Asserts when all cells of the memory have been allocated (memory congestion). As a result, frames are discarded. This is not an error. This bit is cleared by reading it.	R
31:28	—	The read values are undefined.	R

Input Memory Controller (IMC) status/debug information. The IMC controls the shared memory which stores all frames from all ports.

The reset value may not be read as initialization begins immediately after reset (changing this register's value depending on progress). Register reads 0x08000400 when memory initialization is completed.

29.3.1.38 IMC_ERR_FULL : Input Port Memory Full and Truncation Indicator Register

Base address: ETHSW = 0x8012_0000

Offset address: 0xA4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IPC_ERR_TRUNC[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IPC_ERR_FULL[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IPC_ERR_FULL[3:0]	Memory was full at start of a frame reception. A frame has been discarded. Shows congestion occurred on an input to the shared memory. One bit per port (bit 0 = port 0, bit 1 = port 1, and so on). This bit is cleared by reading it.	R
15:4	—	The read values are undefined.	R
19:16	IPC_ERR_TRUNC[3:0]	Memory became full while a frame was received and was partly written into memory. A frame has been truncated and discarded. Shows congestion occurred on an input to the shared memory. One bit per port (bit 0 = port 0, bit 1= port 1, and so on). This bit is cleared by reading it.	R
31:20	—	The read values are undefined.	R

29.3.1.39 IMC_ERR_IFACE : Input Port Memory Error Indicator Register

Base address: ETHSW = 0x8012_0000

Offset address: 0xA8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	WBUF_OVF[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IPC_ERR_IFACE[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IPC_ERR_IFACE[3:0]	Error indication on memory input (receive from MAC) that a frame has been truncated and discarded. This may occur during normal operation if, for example, input rules discard a frame after it is written into memory (destination portmask is all 0, or VLAN domain verification fails, or source address not found discard is enabled and source was not found). It also indicates internal (fatal) errors if the memory access failed. One bit per port (bit 0 = port 0, bit 1 = port 1, and so on). This bit is cleared by reading it.	R
15:4	—	The read values are undefined.	R
19:16	WBUF_OVF[3:0]	Error indicating an overflow in the input write buffer to the memory controller (a small decoupling FIFO at every MAC RX). This must not occur during normal operation. If it occurs, the received frame is corrupted and is marked as error to be discarded eventually by the memory controller. One bit per port (bit 0 = port 0, bit 1 = port 1, and so on). This bit is cleared by reading it.	R
31:20	—	The read values are undefined.	R

29.3.1.40 IMC_ERR_QOFLOW : Output Port Queue Overflow Indicator Register

Base address: ETHSW = 0x8012_0000

Offset address: 0xAC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OP_ERR[3:0]
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	OP_ERR[3:0]	A frame cannot be stored in an output queue of the port as the queue FIFO overflowed (write occurred into full fifo). The frame is ignored but stays stored in memory. This should not occur during normal operation. This is a fatal error as the memory allocated by that frame is not freed, and resulting in memory leakage. It indicates an error in the congestion function which must normally prevent writing a frame into a queue when the queue is not capable of accepting it. One bit per port (bit 0 = port 0, bit 1 = port 1, and so on). This bit is cleared by reading it.	R
31:4	—	The read values are undefined.	R

29.3.1.41 IMC_CONFIG : Input Memory Controller Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0xB0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	CTFL_EMPTY_MD	SPEED_HIPRI_THR[2:0]			RSV_ENA	WFQ_EN
Value after reset:	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WFQ_EN	Enable weighted fair queuing (when this bit is 1) or strict priority (when this bit is 0, default) output queue scheduling. When this bit is set, the output queues are weighted with factors 1, 2, 4, and 8 for queues 0, 1, 2, and 3 respectively (queue 3 is highest priority). A higher weight causes the queue to be served with higher priority. This bit must be set to 0 when TDMA scheduling is used.	R/W
1	RSV_ENA	Enable Memory Reservations to Operate When set to 1, the per-pool memory allocations can be configured. When set to 0, all memory is shared by all ports and queues.	R/W
4:2	SPEED_HIPRI_THR[2:0]	High-Priority Speed Threshold This register defines which speed as a 3-bit value is considered a high speed port. Ports at or above this speed value are serviced faster by the output memory controller than other ports.	R/W
5	CTFL_EMPTY_MD	When this bit is set to 0, a frame received in Cut-Through mode that cannot allocate an entry in the CTFL is forwarded as store and forward. When this bit is set to 1, if an entry in the CTFL is not available, the frame length is reported as 0 bytes and on transmission, the frame is held until Cut-Through is enabled again.	R/W
31:6	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.42 IMC_ERR_ALLOC : Input Port Error Indicator Register

Base address: ETHSW = 0x8012_0000

Offset address: 0xB4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DISC_LATE[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	DISC_FULL[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DISC_FULL[3:0]	Per port discard indication due to memory pool going empty. Per port indication that one of the queues was full and a frame was discarded. This bit is cleared by reading it.	R
15:4	—	The read values are undefined.	R
19:16	DISC_LATE[3:0]	Per port discard indication due to lateness in the priority resolution. The priority resolution can be delayed by the pattern matchers. If it arrives too late (after approximately 100 bytes into the frame), the frame is discarded. This bit is cleared by reading it.	R
31:20	—	The read values are undefined.	R

29.3.1.43 GPARSERn : [n + 1]th Parser of 1st Block (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x00C0 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CMP_MASK_OR	OFFSET_PLUS2	CMP16	IPPROTOCOL	IPDATA	SKIPVLAN	VALID	OFFSET_DA	—	OFFSET[5:0]					
Value after reset:	x	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	COMPARE_VAL[7:0]								MASK_VAL2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MASK_VAL2[7:0]	Mask for single byte compares or 2nd compare value (if bit 30 = 1) or least significant bits of a 16-bit compare value (if bit 28 = 1). When used as a mask (bit 28, 30 = 0, 0), the data from the frame is ANDed with this mask, then compared to the compare value. All bits having a 1 in the mask will be compared with the data in the frame. All bits having a 0 will be 0 for the compare, however this requires the compare value to have those bits also set to 0.	R/W
15:8	COMPARE_VAL[7:0]	The value to compare the frame data with at the given offset.	R/W
21:16	OFFSET[5:0]	An offset in bytes where to find the data for comparison within the frame. The offset value starts at 0 to indicate the very first byte after offset start. The offset start can be either the type/length field of the frame, that is, 0 = first byte of type/length field) or the payload following an IP header (see IPDATA). Valid values range from 0 to 60.	R/W
22	—	The read value is undefined. The write value should be 0.	R/W
23	OFFSET_DA	When set, the offset starts counting from the first byte of the MAC destination address. The SKIPVLAN, IPDATA, and IPPROTOCOL bits must be 0 when this bit is set.	R/W
24	VALID	Indicate that this entry is valid (when this bit is 1) and should be used. When this bit is 0, the parser result always indicates "no match" and none of the other bits are relevant.	R/W
25	SKIPVLAN	When set, any optional VLAN tags found in the frame are skipped and the parser starts operating at the first byte following any VLAN tags. When cleared, the parser starts with the first byte following the source MAC address. If this bit is 0, this means the first byte of the Type/length field is the first data for any compares, allowing to check for ether Types.	R/W
26	IPDATA	When set, the offset starts with the first byte following an IP header if an IP frame is processed. The following fields are skipped: ● any VLAN tags if present (implicitly acts as SKIPVLAN = 1) ● for Ipv4 the header and any header options ● for Ipv6 only the base header. If the bit is set, it implicitly requires an IP frame. If the frame is not an IP frame, the parser will report "no match" and not continue to inspect the frame. When cleared, the offset starts with the first data following the MAC source address (skipping VLAN tags if allowed by "SKIPVLAN"). When this bit is set, the SKIPVLAN bit has no meaning and can be set to any value.	R/W
27	IPPROTOCOL	When set, the compare value is compared with the protocol field found within the IP header for both IPv4 and IPv6 frames. It implicitly acts as SKIPVLAN = 1 skipping any VLAN tags if present. The offset setting has no meaning and is ignored. If the bit is set, but the frame is not an IPv4/v6 frame the parser will report "no match" and not continue to inspect the frame. When cleared, the offset is used normally on all frames. When this bit is set, the bits 25, 26, 28, 29, 30 have no meaning and should be set to 0. MASK_VAL2[7:0] is applied and should be 0xFF to achieve an exact match.	R/W
28	CMP16	When set, MASK_VAL2[7:0] is used as a value to perform a 16-bit compare. COMPARE_VAL[7:0] represent the byte at the given offset and MASK_VAL2[7:0] represent the byte following at offset + 1 which matches the network byte order for 16-bit fields. For example, setting a compare value of 0x0800 and offset 0 matches IP frames. No mask is available in this mode. When cleared, only one byte of data is compared and the mask can be used to mask individual bits.	R/W
29	OFFSET_PLUS2	Repeats the comparison at offset + 2, if the comparison at offset failed. Only usable when CMP16 = 1. This can be used to create a comparison for UDP or TCP port numbers allowing checking if the port number exists in the source or in the destination port number field.	R/W
30	CMP_MASK_OR	Use the MASK_VAL2[7:0] bits as a 2nd compare value. When set, the parser reports a match if the byte at given offset matches COMPARE_VAL[7:0] or MASK_VAL2[7:0]. Only usable when CMP16 = 0.	R/W
31	—	The read value is undefined. The write value should be 0.	R/W

29.3.1.44 GARITHn : Snoop Configuration for Arithmetic [n + 1]th Stage of 1st Block (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x00D0 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	SNP_MD[1:0]		—	—	RESULT_INV	OP
Value after reset:	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SEL_ARITH2	SEL_ARITH1	SEL_ARITH0	SEL_MATCH[3:0]				—	—	—	—	NOT_INP[3:0]			
Value after reset:	x	0	0	0	0	0	0	0	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	NOT_INP[3:0]	Not Input Define, which match result from a parser should be used inverted. One bit per parser. Bit 0 = parser 0, Bit 1 = parser 1, …, Bit 3 = parser 3 When a bit is 1, the match result of the corresponding parser is used inverted in this stage.	R/W
7:4	—	The read values are undefined. The write value should be 0.	R/W
11:8	SEL_MATCH[3:0]	Select Match Define, which parser result is relevant at this stage. One bit per parser. Bit 8 = parser 0, …, Bit 11 = parser 3 When a bit is 1, the match result of the corresponding parser is used in this stage, otherwise it is ignored.	R/W
12	SEL_ARITH0	Select Arithmetic Stage 0 Available only on stages 1 to 3. If set, the result from arithmetic stage 0 is selected in addition to any of the parser results.	R/W
13	SEL_ARITH1	Select Arithmetic Stage 1 Available only on stages 2 and 3. If set, the result from arithmetic stage 1 is selected in addition to any of the parser results.	R/W
14	SEL_ARITH2	Select Arithmetic Stage 2 Available only on stage 3. If set, the result from arithmetic stage 2 is selected in addition to any of the parser results.	R/W
15	—	The read value is undefined. The write value should be 0.	R/W
16	OP	Operation 0: AND all selected inputs 1: OR all selected inputs	R/W
17	RESULT_INV	Result Invert 0: The output is used directly. 1: The output of this stage is inverted.	R/W
19:18	—	The read values are undefined. The write value should be 0.	R/W
21:20	SNP_MD[1:0]	Snoop Mode The last stage of the arithmetic (GARITH3) defines which snooping function should be performed if the result of the arithmetic evaluates to a match result. If both arithmetic blocks (arithmetic 0 and 1) match at the same time and have different mode settings, the lower mode is chosen (for example, if one function is programmed to 01b and the other to 11b, then mode 01b is executed). These bits exist only in the GARITH3 register. The bits are not writable in all others. 0 0: Disabled, no snooping occurs (forward normally) 0 1: Forward to designated management port only 1 0: Forward normally and copy to management port 1 1: Discard the frame	R/W
31:22	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.45 GPARSERn : [n - 3]th Parser of 2nd Block (n = 4 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x00E0 + 0x4 × (n - 4)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CMP_MASK_OR	OFFSET_PLUS2	CMP16	IPPROTOCOL	IPDATA	SKIPVLAN	VALID	OFFSET_DA	—	OFFSET[5:0]					
Value after reset:	x	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	COMPARE_VAL[7:0]								MASK_VAL2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MASK_VAL2[7:0]	Mask for single byte compares or 2nd compare value (if bit 30 = 1) or least significant bits of a 16-bit compare value (if bit 28 = 1). When used as a mask (bit 28, 30 = 0, 0), the data from the frame is ANDed with this mask, then compared to the compare value. All bits having a 1 in the mask are compared with the data in the frame. All bits having a 0 will be 0 for the compare, however this requires the compare value to have those bits also set to 0.	R/W
15:8	COMPARE_VAL[7:0]	The value to compare the frame data with at the given offset.	R/W
21:16	OFFSET[5:0]	An offset in bytes to locate the data for comparison within the frame. The offset value starts at 0 to indicate the very first byte after offset start. The offset start can be either the type or length field of the frame, for example 0 = first byte of type/length field) or the payload following an IP header (see bit 26). Valid values range from 0 to 60.	R/W
22	—	The read value is undefined. The write value should be 0.	R/W
23	OFFSET_DA	When set, the offset starts counting from the first byte of the MAC destination address. The SKIPVLAN, IPDATA, and IPPROTOCOL bits must be 0 when this bit is set.	R/W
24	VALID	Indicates that this entry is valid (when this bit is 1) and should be used. When this bit is 0, the parser result always indicates "no match" and none of the other bits are relevant.	R/W
25	SKIPVLAN	When set, any optional VLAN tags found in the frame are skipped and the parser starts operating at the first byte following any VLAN tags. When cleared, the parser starts with the first byte following the source MAC address. When this bit is 0, this means the first byte of the type/length field is the first data for any compare, allowing to check for either types.	R/W
26	IPDATA	When set, the offset starts with the first byte following an IP header if an IP frame is processed. The following fields are skipped: ● any VLAN tags if present (implicitly acts as SKIPVLAN = 1) ● for Ipv4 the header and any header options ● for Ipv6 only the base header If the bit is set, it implicitly requires an IP frame. If the frame is not an IP frame, the parser reports "no match" and does not continue to inspect the frame. When cleared, the offset starts with the first data following the MAC source address (skipping VLAN tags if allowed by the bit SKIPVLAN. When this bit is set, the SKIPVLAN bit has no meaning and can be set to any value.	R/W
27	IPPROTOCOL	When set, the compare value is compared with the protocol field located within the IP header for both IPv4 and IPv6 frames. It implicitly acts as SKIPVLAN = 1 skipping any VLAN tags if present. The offset setting has no meaning and is ignored. If the bit is set, but the frame is not an IPv4/v6 frame the parser reports a no match and does not continue to inspect the frame. When cleared, the offset is used normally on all frames. When this bit is set, bits 25, 26, 28, 29, and 30 have no meaning and should be set to 0. MASK_VAL2[7:0] is applied therefore should be 0xFF to achieve an exact match.	R/W
28	CMP16	When set, MASK_VAL2[7:0] is used as a value to perform a 16-bit compare. COMPARE_VAL[7:0] represents the byte at the given offset and MASK_VAL2[7:0] represents the byte following at offset + 1 which matches the network byte order for 16-bit fields, (for example setting a compare value of 0x0800 and offset 0 matches IP frames). No mask is available in this mode. When cleared, only one byte of data is compared and the mask can be used to mask individual bits.	R/W

Bit	Symbol	Function	R/W
29	OFFSET_PLUS2	Repeats the comparison at offset + 2, if the comparison at offset failed. Only usable when CMP16 = 1. This can be used to create a comparison for UDP or TCP port numbers allowing for checking if the port number exists in the source or in the destination port number field.	R/W
30	CMP_MASK_OR	Use MASK_VAL2[7:0] as a second compare value. When set, the parser reports a match if the byte at given offset matches COMPARE_VAL[7:0] or MASK_VAL2[7:0]. Only usable when CMP16 = 0.	R/W
31	—	The read value is undefined. The write value should be 0.	R/W

The GPARSERn register is used as a port snooping function parser configuration.

29.3.1.46 GARITHn : Snoop Configuration for Arithmetic [n − 3]th Stage of 2nd Block (n = 4 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x00F0 + 0x4 × (n - 4)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	SNP_MD[1:0]		—	—	RESULT_INV	OP
Value after reset:	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SEL_ARITH2	SEL_ARITH1	SEL_ARITH0	SEL_MATCH[3:0]				—	—	—	—	NOT_INP[3:0]			
Value after reset:	x	0	0	0	0	0	0	0	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	NOT_INP[3:0]	Not Input Define, which match result from a parser should be used inverted. One bit per parser. Bit 0 = parser 4, Bit 1 = parser 5, …, Bit 3 = parser 7 When a bit is 1, the match result of the corresponding parser is used inverted in this stage.	R/W
7:4	—	The read values are undefined. The write value should be 0.	R/W
11:8	SEL_MATCH[3:0]	Select Match Define, which parser result is relevant at this stage. One bit per parser. Bit 8 = parser 4, …, Bit 11 = parser 7 When a bit is 1, the match result of the corresponding parser is used in this stage, otherwise it is ignored.	R/W
12	SEL_ARITH0	Select Arithmetic Stage 0 Available only in stages 1 to 3. If set, the result from arithmetic stage 0 is selected in addition to any of the parser results.	R/W
13	SEL_ARITH1	Select Arithmetic Stage 1 Available only in stages 2 and 3. If set, the result from arithmetic stage 1 is selected in addition to any of the parser results.	R/W
14	SEL_ARITH2	Select Arithmetic Stage 2 Available only in stage 3. If set, the result from arithmetic stage 2 is selected in addition to any of the parser results.	R/W
15	—	The read value is undefined. The write value should be 0.	R/W
16	OP	Operation 0: AND all selected inputs 1: OR all selected inputs	R/W
17	RESULT_INV	Result Invert 0: The output is used directly 1: The output of this stage is inverted	R/W
19:18	—	The read values are undefined. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
21:20	SNP_MD[1:0]	Snoop Mode The last stage of the arithmetic (GARITH7) defines which snooping function should be performed if the result of the arithmetic evaluates to a match result. If both arithmetic blocks (arithmetic 0 and 1) match at the same time and they have different mode settings, the lower mode is chosen. For example, if one function is programmed to 01b and the other to 11b, then mode 01b is executed. These bits exist only in the GARITH7 register. These bits are not writable in other registers. 0 0: Disabled, no snooping occurs (forward normally) 0 1: Forward to designated management port only 1 0: Forward normally and copy to management port 1 1: Discard the frame	R/W
31:22	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.47 VLAN_PRIORITYn : VLAN Priority Register n (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x0100 + 0x4 × n

Bit position:	31								23		21	20		18	17		15	14		12	11		9	8		6	5		3	2		0
Bit field:	—	—	—	—	—	—	—	—	PRIORITY 7[2:0]			PRIORITY 6[2:0]			PRIORITY 5[2:0]			PRIORITY 4[2:0]			PRIORITY 3[2:0]			PRIORITY 2[2:0]			PRIORITY 1[2:0]			PRIORITY 0[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	PRIORITY0[2:0]	Priority 0 Setting Priority for priority 0 of VLAN tag of input frames	R/W
5:3	PRIORITY1[2:0]	Priority 1 Setting Priority for priority 1 of VLAN tag of input frames	R/W
8:6	PRIORITY2[2:0]	Priority 2 Setting Priority for priority 2 of VLAN tag of input frames	R/W
11:9	PRIORITY3[2:0]	Priority 3 Setting Priority for priority 3 of VLAN tag of input frames	R/W
14:12	PRIORITY4[2:0]	Priority 4 Setting Priority for priority 4 of VLAN tag of input frames	R/W
17:15	PRIORITY5[2:0]	Priority 5 Setting Priority for priority 5 of VLAN tag of input frames	R/W
20:18	PRIORITY6[2:0]	Priority 6 Setting Priority for priority 6 of VLAN tag of input frames	R/W
23:21	PRIORITY7[2:0]	Priority 7 Setting Priority for priority 7 of VLAN tag of input frames	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

Note: n = 0: port 0, n = 1: port 1, n = 2: port 2, n = 3: port 3 (management port)

The VLAN_PRIORITYn registers are used to set the priority of the VLAN tag of the input frames.

The ethernet switch has an 8-entry programmable priority lookup table for each port. The priority included in the 3 higher-order bits of the first octet of the VLAN info is used as an index of the lookup table and can be remapped.

29.3.1.48 IP_PRIORITYn : IP Priority Register n (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x0140 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	READ	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	PRIORITY[2:0]			IPV6SELECT	ADDRESS[7:0]							
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ADDRESS[7:0]	COS Table Address Specifying Specify an address for the index of the COS table. The IPv4 priority table is 6 bits (64 entries) and the IPv6 table is 8 bits (256 entries).	R/W
8	IPV6SELECT	IPv6 COS Table Selection Indicates the selection of the IPv6 COS table. 0: The IPv4 table is accessed. The valid range of the ADDRESS[7:0] bits is from 0 to 63. 1: The IPv6 table is accessed. The valid range of the ADDRESS[7:0] bits is from 0 to 255.	R/W
11:9	PRIORITY[2:0]	COS Table Priority When writing, set the priority information to be written to the COS table. When read, the priority information written to the COS table can be read. The address index of the COS table to be read is the address of the last write operation.	R/W
30:12	—	The read values are undefined. The write value should be 0.	R/W
31	READ	COS Table Operation Switching Switches write or read operation to the COS table. Read operation to the register: The priority information of the address of the last write operation is read from the table. Use the following procedure to read the priority information from the COS table: 1. Perform a write (that is, specification of the address to be read) while this bit is 1 and the IPV6SELECT and ADDRESS[7:0] bits are set to the target values. 2. Execute a read operation Write operation to the register: 0: Write the priority information to the COS table 1: Does not write the priority information to the COS table (unchanged)	R/W

Note: n = 0: port 0, n = 1: port 1, n = 2: port 2, n = 3: port 3 (management port)

The IP_PRIORITYn register is used to make and refer to the COS table settings by writing to the COS table or specifying the settings for reference to a priority level. Each port of the ethernet switch has class of services (COS) tables for IPv4 and IPv6. In the COS table entries for IPv4, the 6-bit DiffServ field of the frame is provided as an index to the Lookup table and a 3-bit priority can be set. In the COS table entries for IPv6, the 8-bit COS field of the frame is provided as an index to the Lookup table and a 3-bit priority can be set.

29.3.1.49 PRIORITY_CFGn : Priority Configuration Register n (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x0180 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PCP_REMAP[23:8]															
Value after reset:	1*1	1*1	1*1	1*1	1*1	0	1*1	0	1*1	1*1	0	0	0	1*1	1*1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCP_REMAP[7:0]								PCP_REMAP_DIS	DEFAULTPRI[2:0]			TYPE_EN	MACEN	IPEN	VLANEN
Value after reset:	1*1	0	0	0	1*1	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VLANEN	VLAN Priority Enable Enables VLAN priority for a frame received on port n. 0: VLAN priority is not used. The priority field of the VLAN tag of a frame is ignored. 1: Priority in the switch is determined using the priority field of the VLAN tag of a frame according to the VLAN_PRIORITYn setting for the port on which the frame was received.	R/W
1	IPEN	IP Priority Enable Enables IP priority for a frame received on port n. 0: IP priority is not used and IP DiffServ/COS fields of the frame are ignored 1: Priority in the switch is determined using the IP DiffServ/COS field according to the IP_PRIORITYn setting for the port on which the frame was received	R/W
2	MACEN	MAC Based Priority Enable 0: MAC priority is ignored 1: Priority information found within the MAC address table (for a static entry)	R/W
3	TYPE_EN	TYPE Based Priority Enable 0: Type fields are ignored 1: The type/length field of the frame (following any VLAN tags) is compared to all PRIORITY_TYPE configured values. If any of the values matches, the priority information from that matching type is used.	R/W
6:4	DEFAULTPRI[2:0]	Default Priority Enable Setting Set the default priority of a frame received on port n. When neither of the priorities is used, this priority is applied. The valid range is from 0 to 7.	R/W
7	PCP_REMAP_DIS	Disables PCP remapping when set to 1. See PCP_REMAP[23:0] field.	R/W
31:8	PCP_REMAP[23:0]	PCP Remapping function On ingress frame's VLAN.PCP is remapped according to this function and the frame's VLAN tag is modified accordingly (section 29.4.3.2. VLAN Input Processing Function (5) VLAN PCP remap). Default value keeps the same PCP value. The function is only present for line ports. PCP_REMAP[2:0]: Remap value for PCP 0 PCP_REMAP[5:3]: Remap value for PCP 1 PCP_REMAP[8:6]: Remap value for PCP 2 PCP_REMAP[11:9]: Remap value for PCP 3 PCP_REMAP[14:12]: Remap value for PCP 4 PCP_REMAP[17:15]: Remap value for PCP 5 PCP_REMAP[20:18]: Remap value for PCP 6 PCP_REMAP[23:21]: Remap value for PCP 7	R/W

Note: n = 0: port 0, n = 1: port 1, n = 2: port 2, n = 3: port 3 (management port)

Note 1. The initial value of the PRIORITY_CFG3 register for management port (Port 3) is 0x00000000. Others are 0xFAC68800.

The PRIORITY_CFGn register is used to set each port for how received frames are assigned according to the priority of the queues in the switch using the priority in the frames, and VLAN PCP remapping function.

When several types of the priority fields are enabled, priority reassignment is executed according to the higher priority (IP priority (DiffServ/COS), VLAN priority, and default priority).

29.3.1.50 PRIORITY_TYPE1 : Priority Type Register 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x01B8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PRIORITY[2:0]			VALID
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TYPEVAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TYPEVAL[15:0]	Type Priority A 16-bit value to be compared against the type/length field of the frame at receive. For example, a value of 0x88F7 would match IEEE 1588 frames. All such PRIORITY_TYPE registers are simultaneously searched and a type match occurs if a compare is successful. The value is relevant (and used for comparison) only if the valid bit is set. This value is always compared against the type/length field of the frame following any VLAN tags. Up to two VLAN tags can be present in the frame. When used with the management port special frame tagging (see section 29.3.1.91. MGMT_TAG_CONFIG : Management Tag Configuration Register), no VLAN tags can be supported for the tag insertion selection. That is, the management tag is inserted only if this is the first tag found in the frame. At the same time, for priority definition, VLAN tags are supported.	R/W
16	VALID	If set indicates, this register contains valid data. If cleared (default), the data from this register is ignored. When using with the management port special frame tagging, the valid bit has no relevance. The type is compared if enabled in the section 29.3.1.91. MGMT_TAG_CONFIG : Management Tag Configuration Register. This allows for using the type for the management tag only, or at the same time for priority resolution (useful for 1588 frames).	R/W
19:17	PRIORITY[2:0]	The priority value to use if a match occurs. PRIORITY_CFGn.TYPE_EN bit of a port controls the use of this priority.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The PRIORITY_TYPE1 register is used to compare value for prioritizing frames with a specific type field.

29.3.1.51 PRIORITY_TYPE2 : Priority Type Register 2

Base address: ETHSW = 0x8012_0000

Offset address: 0x01BC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PRIORITY[2:0]			VALID
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TYPEVAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

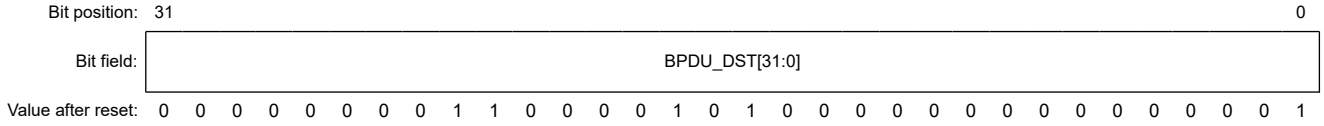
Bit	Symbol	Function	R/W
15:0	TYPEVAL[15:0]	Type Priority A 16-bit value to be compared against the type/length field of the frame at receive. For example, a value of 0x88F7 would match IEEE 1588 frames. All such PRIORITY_TYPE registers are simultaneously searched and a type match occurs if a compare is successful. The value is relevant (and used for comparison) only if the valid bit is set. This value is always compared against the type/length field of the frame following any VLAN tags. Up to two VLAN tags can be present in the frame. When used with the management port special frame tagging (see section 29.3.1.91. MGMT_TAG_CONFIG : Management Tag Configuration Register), no VLAN tags can be supported for the tag insertion selection. That is, the management tag is inserted only if this is the first tag found in the frame. At the same time, for priority definition, VLAN tags are supported.	R/W
16	VALID	If set indicates, this register contains valid data. If cleared (default), the data from this register is ignored. When using with the management port special frame tagging, the valid bit has no relevance. The type is compared if enabled in the section 29.3.1.91. MGMT_TAG_CONFIG : Management Tag Configuration Register. This allows for using the type for the management tag only, or at the same time for priority resolution (useful for 1588 frames).	R/W
19:17	PRIORITY[2:0]	The priority value to use if a match occurs. PRIORITY_CFGn.TYPE_EN bit of a port controls the use of this priority.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The PRIORITY_TYPE2 register is used to compare value for prioritizing frames with a specific type field.

29.3.1.52 MGMT_ADDR0_lo : Lower MAC Address 0 for Bridge Protocol Frame Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01C0

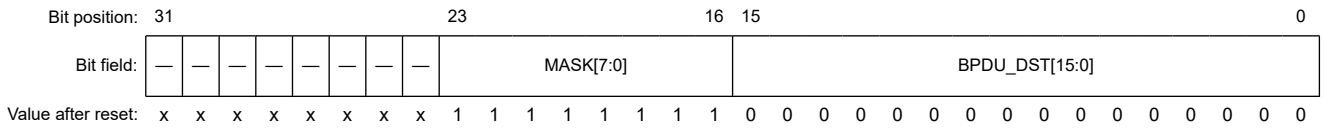


Bit	Symbol	Function	R/W
31:0	BPDU_DST[31:0]	Additional MAC address defining a Bridge Protocol Frame (BPDU) in addition to the commonly-known addresses First 4 bytes of MAC address: Bits [7:0] = 1st, bits [15:8] = 2nd, bits [23:16] = 3rd, bits [31:24] = 4th.	R/W

29.3.1.53 MGMT_ADDR0_hi : Higher MAC Address 0 for Bridge Protocol Frame Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01C4



Bit	Symbol	Function	R/W
15:0	BPDU_DST[15:0]	Bits [7:0] is 5th byte, bits [15:8] is 6th (last) byte	R/W
23:16	MASK[7:0]	8-bit mask for comparing the last byte of the MAC address. Each bit that is 1 in the mask is compared. The mask is ANDed with the last byte of the MAC address (6th byte in frame) before comparison. Setting the mask to 0xFF compares all bits. A mask of 0x00 ignores the last byte of the MAC address and a match occurs on any value. All mask bits with 0 must also have 0 in the corresponding MAC address bit (bits [15:8] in this register).	R/W

Bit	Symbol	Function	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

The MGMT_ADDR0_hi register specifies the last 2 octets of the MAC address and the mask/valid bits.

Note: The compare is always active. To disable it, set a default value. Reset defaults to 01-80-c2-00-00-00 mask 0xFF.

29.3.1.54 MGMT_ADDR1_lo : Lower MAC Address 1 for Bridge Protocol Frame Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01C8

Bit position:	31																												0
Bit field:	BPDU_DST[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
31:0	BPDU_DST[31:0]	Additional MAC address defining a Bridge Protocol Frame (BPDU) in addition to the commonly-known addresses First 4 bytes of MAC address: Bits [7:0] = 1st, bits [15:8] = 2nd, bits [23:16] = 3rd, bits [31:24] = 4th.	R/W

29.3.1.55 MGMT_ADDR1_hi : Higher MAC Address 1 for Bridge Protocol Frame Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01CC

Bit position:	31								23							16	15													0
Bit field:	—	—	—	—	—	—	—	—	MASK[7:0]								BPDU_DST[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	BPDU_DST[15:0]	Bits [7:0] is 5th byte, bits [15:8] is 6th (last) byte.	R/W
23:16	MASK[7:0]	8-bit mask for comparing the last byte of the MAC address. Each bit that is 1 in the mask is compared. The mask is ANDed with the last byte of the MAC address (6th byte in frame) before comparison. Setting the mask to 0xFF compares all bits. A mask of 0x00 ignores the last byte of the MAC address and a match occurs on any value. All mask bits with 0 must also have 0 in the corresponding MAC address bit (bits [15:8] in this register).	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

The MGMT_ADDR1_hi register specifies the last 2 octets of the MAC address and the mask/valid bits.

Note: The compare is always active. To disable it, set a default value. Reset defaults to 01-80-c2-00-00-00 mask 0xFF.

29.3.1.56 MGMT_ADDR2_lo : Lower MAC Address 2 for Bridge Protocol Frame Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01D0

Bit position:	31																												0
Bit field:	BPDU_DST[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
31:0	BPDU_DST[31:0]	Additional MAC address defining a Bridge Protocol Frame (BPDU) in addition to the commonly-known addresses. First 4 bytes of MAC address: Bits [7:0] = 1st, bits [15:8] = 2nd, bits [23:16] = 3rd, bits [31:24] = 4th.	R/W

29.3.1.57 MGMT_ADDR2_hi : Higher MAC Address 2 for Bridge Protocol Frame Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01D4

Bit position:	31								23								16	15													0
Bit field:	—	—	—	—	—	—	—	—	MASK[7:0]								BPDU_DST[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	BPDU_DST[15:0]	Bits [7:0] is 5th byte, bits [15:8] is 6th (last) byte.	R/W
23:16	MASK[7:0]	8-bit mask for comparing the last byte of the MAC address. Each bit that is 1 in the mask is compared. The mask is ANDed with the last byte of the MAC address (6th byte in frame) before comparison. Setting the mask to 0xFF compares all bits. A mask of 0x00 ignores the last byte of the MAC address and a match occurs on any value. All mask bits with 0 must also have 0 in the corresponding MAC address bit (bits [15:8] in this register).	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

The MGMT_ADDR2_hi register specifies the last 2 octets of the MAC address and the mask/valid bits.

Note: The compare is always active. To disable it, set a default value. Reset defaults to 01-80-c2-00-00-00 mask 0xFF.

29.3.1.58 MGMT_ADDR3_lo : Lower MAC Address 3 for Bridge Protocol Frame Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01D8

Bit position:	31																														0
Bit field:	BPDU_DST[31:0]																														
Value after reset:	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
31:0	BPDU_DST[31:0]	Additional MAC address defining a Bridge Protocol Frame (BPDU) in addition to the commonly-known addresses. First 4 bytes of MAC address: Bits [7:0] = 1st, bits [15:8] = 2nd, bits [23:16] = 3rd, bits [31:24] = 4th.	R/W

29.3.1.59 MGMT_ADDR3_hi : Higher MAC Address 3 for Bridge Protocol Frame Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01DC

Bit position:	31								23								16	15													0
Bit field:	—	—	—	—	—	—	—	—	MASK[7:0]								BPDU_DST[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	BPDU_DST[15:0]	Bits [7:0] is 5th byte, bits [15:8] is 6th (last) byte.	R/W
23:16	MASK[7:0]	8-bit mask for comparing the last byte of the MAC address. Each bit that is 1 in the mask is compared. The mask is ANDed with the last byte of the MAC address (6th byte in frame) before comparison. Setting the mask to 0xFF compares all bits. A mask of 0x00 ignores the last byte of the MAC address and a match occurs on any value. All mask bits with 0 must also have 0 in the corresponding MAC address bit (bits [15:8] in this register).	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

The MGMT_ADDR3_hi register specifies the last 2 octets of the MAC address and the mask/valid bits.

Note: The compare is always active. To disable it, set a default value. Reset defaults to 01-80-c2-00-00-00 mask 0xFF.

29.3.1.60 SRCFLT_ENA : MAC Source Address Filtering Enable Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DSTENA[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SRCENA[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	SRCENA[2:0]	Per-Source Port Enable Enables for which ports the MAC source address filtering check is enabled when frames are received by the enabled port.	R/W
15:3	—	The read values are undefined. The write value should be 0.	R/W
19:16	DSTENA[3:0]	Per-Destination Port Enable Enables for which destination ports the source filtering is applied. When a frame is subject to filtering check and the MAC address is different, the ports enabled in DSTENA are filtered out from the final forwarding mask.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The SRCFLT_ENA register is used for the per-port MAC address source filtering enable.

29.3.1.61 SRCFLT_CONTROL : MAC Source Address Filtering Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01E4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	WATCHDOG_TIME[15:0]															
Value after reset:	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WATCHDOG_ENA	MGMT_FWD
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	MGMT_FWD	Management Forward Enable When this bit is set to 1 and a MAC source address filter check results in a port being removed from the destination port mask, the frame is then copied to the management port.	R/W
1	WATCHDOG_ENA	When set to 1, a watchdog is enabled. The watchdog is used to trigger the interrupt SRCFLT_WD_INT when the time in WATCHDOG_TIME expires. The watchdog starts to operate immediately when this bit is 1. It resets every time a frame is received in a port with SRCENA set to 1 and the source address of the frame matches SRCFLT_MACADDR.	R/W
15:2	—	The read values are undefined. The write value should be 0.	R/W
31:16	WATCHDOG_TIME[15:0]	Defines the watchdog expire time in milliseconds. The default is 2000 milliseconds.	R/W

29.3.1.62 SRCFLT_MACADDR_LO : Lower MAC Filtering Address Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01E8

Bit position:	31																												0
Bit field:	SRCFLT_MACADDR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SRCFLT_MACADDR[31:0]	MAC address to use in source filtering The mapping to the first 32 bits of the MAC source address MACSA[31:0] to be matched against.	R/W

29.3.1.63 SRCFLT_MACADDR_HI : Higher MAC Filtering Address Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01EC

Bit position:	31															16	15													0
Bit field:	MASK[15:0]																SRCFLT_MACADDR[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SRCFLT_MACADDR[15:0]	MAC address to use in source filtering The mapping to the last 16 bits of the MAC source address MACSA[47:32] to be matched against.	R/W
31:16	MASK[15:0]	The mask to apply to the last 16 bits of the MAC address When set to 0, the corresponding bit in MACSA[47:32] is ignored. For the mask to work correctly, the corresponding bit in MACSA[47:32] must also be set to 0.	R/W

29.3.1.64 PHY_FILTER_CFG : Debounce Filter Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x01FC

Bit position:	31													18	16								8							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	FLT_EN[2:0]		—	—	—	—	—	—	—	FILTER_DURATION[8:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	0	0	0	0	1	0	1	1	1

Bit	Symbol	Function	R/W
8:0	FILTER_DURATION[8:0]	This is the amount of time to wait after the last phy_link (ETHSW_PHYLINKn: n = port) transition from 0 to 1 to acknowledge the link-up condition. Each unit corresponds to 2^{16} switch clock periods. At 200 MHz each unit is 0.328 ms. The default value sets a debounce time of approximately 7.5 ms.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
18:16	FLT_EN[2:0]	Per-port Enable Mask Set to 1 to enable the link-up debounce filter on each port.	R/W
31:19	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.65 SYSTEM_TAGINFOn : One VLAN ID Field n for VLAN Input Manipulation (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x0200 + 0x4 × n

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SYSVLANINFO[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SYSVLANINFO[15:0]	System VLAN Info (prio/cfi/vid) for Port n	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.66 AUTH_PORTn : Port n Authentication Control and Configuration (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x0240 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	GUEST_MASK[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EAPOL_PNUM[3:0]				ACHG_UNAUTH	—	—	—	—	—	EAPOL_UC_EN	BPDU_EN	GUEST_EN	EAPOL_EN	CTRL_BOTH	AUTH
Value after reset:	0	0	1	1	0	x	x	x	x	x	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	AUTH	Authorized 0: The port is unauthorized restricting forwarding according to the rules configured in this register 1: The port is in the authorized state and operates normally without restrictions. The other bits of this register, except for bits [15:12] and bit [2], are not relevant.	R/W
1	CTRL_BOTH	Controlled Both 0: The port operates in Controlled Directions (mode in). This allows frames from other ports to be forwarded to this port for transmission, but reception is still restricted. 1: The port operates in Controlled Directions (mode both). This means other ports cannot transmit any traffic on this port, except the port specified in the EAPOL destination port.	R/W

Bit	Symbol	Function	R/W
2	EAPOL_EN	EAPOL Enable Enable (when this bit is 1) / Disable (when this bit is 0) Reception of EAPOL Frames. When enabled, EAPOL frames are forwarded to the port given in EAPOL destination port number set in EAPOL_PNUM[3:0]. This is independent from the authorized state. If disabled and the port is: ● Unauthorized — EAPOL frames are discarded ● Authorized — EAPOL frames are treated as BPDU frames and forwarded to the management port This bit is disabled if implementing the HELD state of the authentication state machine where a port is unauthorized and is not allowed to accept EAPOL frames.	R/W
3	GUEST_EN	Guest Enable Enable (when this bit is 1) / Disable (when this bit is 0) Reception of non EAPOL Frames. When enabled and the port is unauthorized, such frames are forwarded only to the given guest port mask set in GUEST_MASK[3:0]. When this bit is enabled, the normal VLAN manipulation function (VLAN insertion) can be used if required.	R/W
4	BPDU_EN	BPDU Enable Enable (when this bit is 1) / Disable (when this bit is 0) Reception of BPDU Frames. When enabled and the port is unauthorized, BPDU frames are forwarded to the management port normally. Otherwise, BPDU frames are discarded. Although EAPOL frames use BPDU multicast addresses, they are not discarded by setting this bit to 0.	R/W
5	EAPOL_UC_EN	EAPOL Unicast Enable Normally EAPOL frames must use the dedicated PAE multicast destination address. However, if this bit is set and a unicast destination address is found, the frame is accepted if it matches the MAC address of the port as configured in the ADDR registers of the MAC. The frame is then forwarded to the EAPOL destination port number given in bits [15:12].	R/W
10:6	—	The read values are undefined. The write value should be 0.	R/W
11	ACHG_UNAUTH	Automatic Port Change to Unauthorized Enables automatic port change to unauthorized. When this bit is set to 1 (enabled) and a frame of unknown source is received at the port, the authorized (AUTH) bit of this register is cleared. The port changes into an unauthorized state and it is up to the application to re-authenticate the port and eventually set the authorized bit again. This bit is valid only if the port has learning disabled (see section 29.3.1.8. INPUT_LEARN_BLOCK : Input Learning Block Register) and is configured to discard frames of unknown source (see section 29.3.1.83. LK_CTRL : Learning/Lookup Function Global Configuration Register). The interrupt (INT_CONFIG.LK_NEW_SRC) can be used to notify the application of this event.	R/W
15:12	EAPOL_PNUM[3:0]	EAPOL Port Number 4 bits port number where to send EAPOL frames. Typically it is the management port (port number = 3), but can be an external port if desired (with limitations). The setting must be identical for all ports. It is not recommended to change this setting. If it is modified and a port changes to authorized without keeping the EAPOL enable bit [2] set, then EAPOL frames are treated as normal BPDU frames. Therefore, an EAPOL log off message does not reach the configured port but is forwarded to the management port instead. The setting is also valid when a port is authorized and matches this number. If this port receives frames, it can always forward to unauthorized ports, even when controlled both is set.	R/W
19:16	GUEST_MASK[3:0]	Destination port mask with all ports that are allowed to receive non-EAPOL frames from this port while it is unauthorized and guest (GUEST_EN) is enabled. Bit 16 = port 0, Bit 17 = port 1, and so on. ● If all 0, frames are discarded even if guest is enabled ● If a destination port in this list is unauthorized and has its CTRL_BOTH bit set, it is removed from this list automatically during forwarding ● If a destination port in this list is authorized, it is not removed from this list. This allows forwarding of a frame from an unauthorized to an authorized port. This can be intentional, for example to implement DHCP with an external server connected at an authorized port. It is the responsibility of the application to ensure the update of all guest masks of all ports when a port changes to authorized, without receiving frames forwarded to it from unauthorized ports.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The AUTH bit reset defaults are controlled by the STRAP_SX_ENB bit and STRAP_HUB_ENB bit of SWCTRL register.

29.3.1.67 VLAN_RES_TABLEn : 32 VLAN Domain Entries (n = 0 to 31)

Base address: ETHSW = 0x8012_0000

Offset address: 0x0280 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	WT_PRTMSK	WT_TAGMSK	RD_TAGMSK	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VLANID[11:0]												PORTMASK[3:0]			
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	PORTMASK[3:0]	When this bit is set to 1, it defines a port as a member of the VLAN. When bit [28] or bit [29] is set, the tagged bit mask is read/written instead of port mask.	R/W
15:4	VLANID[11:0]	The 12-bit VLAN identifier (VLAN ID) of the entry.	R/W
27:16	—	The read values are undefined. The write value should be 0.	R/W
28	RD_TAGMSK	Read TAG Mask Selects contents of PORTMASK[3:0] when reading the register. If this bit is set during a write into the register, all other bits of the write are ignored (for example bit [30], bit [29], bits [15:0]) and bit [28] of the register toggles (from 1 to 0 or 0 to 1). This is only used to allow changing bit [28] without changing any table contents. When the register is read, the bit indicates the contents of bits [3:0]. This bit is cleared when the register is written with bit [28] = 0. 0: The port mask is returned 1: The tagged bit mask is returned	R/W
29	WT_TAGMSK	Write TAG Mask When this bit is set while writing into the register, bits [3:0] are stored in the tagged bit mask for the domain only. The port mask is left unchanged and the VLAN ID must be set accordingly during the write. The tagged bit mask is allowed to specify for each port if the frames leaving the port should be tagged (when this bit is 1) or untagged (when this bit is 0). If a port is untagged, frames that leave that port have the VLAN tag removed. This tagged bit mask is interpreted by the output manipulation function on all ports that operate in output manipulation mode 3. When bits [30:29] are both 0 or both 1 during a write, bits [3:0] are always written into both the port mask and the tagged bit mask.	R/W
30	WT_PRTMSK	Write Port Mask When this bit is set when writing into the register, bits [3:0] are stored in the port mask for the domain only. The tagged bit mask is left unchanged. The VLAN ID must be set accordingly during the write. When bits [30:29] are both 0 or both 1 during a write, bits [3:0] are always written into both the port mask and the tagged bit mask.	R/W
31	—	The read value is undefined. The write value should be 0.	R/W

29.3.1.68 TOTAL_DISC : Discarded Frame Total Number Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0300

Bit position:	31	0
Bit field:	TOTAL_DISC[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	TOTAL_DISC[31:0]	Total number of incoming frames accepted by MAC RX but discarded in the switch	R

29.3.1.69 TOTAL_BYT_DISC : Discarded Frame Total Bytes Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0304

Bit position:	31																												0
Bit field:	TOTAL_BYT_DISC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TOTAL_BYT_DISC[31:0]	Sum of bytes of frames counted in TOTAL_DISC	R

29.3.1.70 TOTAL_FRM : Processed Frame Total Number Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0308

Bit position:	31																												0
Bit field:	TOTAL_FRM[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TOTAL_FRM[31:0]	Total number of incoming frames processed by the switch	R

Note: The number of incoming frames is also incremented when cut through forwarding is used and a frame with invalid CRC was received, as it has been forwarded. The IfOutErrors of the MAC however, will indicate erroneous frames forwarded.

29.3.1.71 TOTAL_BYT_FRM : Processed Frame Total Bytes Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x030C

Bit position:	31																												0
Bit field:	TOTAL_BYT_FRM[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TOTAL_BYT_FRM[31:0]	Sum of bytes of frames counted in TOTAL_FRM	R

29.3.1.72 IALK_CONTROL : IA Lookup Function Enable Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0340

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CT_ENA[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IA_LKUP_ENA[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IA_LKUP_ENA[3:0]	Per-port Enable to the IA Lookup Table When enabled on a port, the L2 MAC destination address is sent to the IA database.	R/W
15:4	—	The read values are undefined. The write value should be 0.	R/W
19:16	CT_ENA[3:0]	Per-port Cut-Through Mode Enable When set to 0 and a lookup to the IA database results in a found result, the frame is not forwarded cut-through. When set to 1, the frame can be forwarded cut-through if all the other conditions are met.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The IALK_CONTROL register is used as the IA Lookup function per-port enable register.

29.3.1.73 IALK_OUI : IA Frames MAC Address OUI Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0344

Bit position:	31							23								0
Bit field:	—	—	—	—	—	—	—	—	IALK_OUI[23:0]							
Value after reset:	x	x	x	x	x	x	x	x	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
23:0	IALK_OUI[23:0]	IA Frames MAC Address OUI Configures the 24-bit OUI used to compare against the OUI of the receive frame.	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.74 IALK_ID_MIN : Minimum Value ID MAC Address Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0348

Bit position:	31							23								0
Bit field:	—	—	—	—	—	—	—	—	IALK_ID_MIN[23:0]							
Value after reset:	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	IALK_ID_MIN[23:0]	Minimum value for the 24-bit ID in the MAC address Frames below this minimum are not looked up in the IA table.	R/W

Bit	Symbol	Function	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.75 IALK_ID_MAX : Maximum Value ID MAC Address Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x034C

Bit position:	31								23																				0
Bit field:	—	—	—	—	—	—	—	—	IALK_ID_MAX[23:0]																				
Value after reset:	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
23:0	IALK_ID_MAX[23:0]	Maximum value for the 24-bit ID in the MAC address Frames above this minimum are not looked up in the IA table.	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.76 IALK_ID_SUB : Offset Value ID MAC Address Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0350

Bit position:	31								23																				0
Bit field:	—	—	—	—	—	—	—	—	IALK_ID_SUB[23:0]																				
Value after reset:	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	IALK_ID_SUB[23:0]	Offset value to subtract from the 24-bit ID in the MAC address	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.77 IALK_ID_CONFIG : Configures Lookup Response Unknown IDs Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0354

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	INVLD_ID_FLOOD_MASK[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	INVLD_ID_PRIO_VLD	INVLD_ID_PRIO[2:0]			—	—	INVLD_ID_LRN_ENA	INVLD_ID_FLOOD
Value after reset:	x	x	x	x	x	x	x	x	0	0	0	0	x	x	0	0

Bit	Symbol	Function	R/W
0	INVLD_ID_FLOOD	Setting this bit to 1 causes the IA table to return a found response for frames whose ID lies outside the ID range defined by [IA_LK_MAX:IA_LK_MIN] using INVLD_ID_FLOOD_MASK[3:0] bits. When set to 0, an unknown ID returns a not-found response and the switch uses the result from the L2 FDB instead.	R/W

Bit	Symbol	Function	R/W
1	INVLD_ID_LRN_ENA	Setting this bit to 1 allows automatic learning into the L2 FDB for frames with unknown IDs. When 0, learning is inhibited. This bit is only valid when INVLD_ID_FLOOD bit is set to 1.	R/W
3:2	—	The read values are undefined. The write value should be 0.	R/W
6:4	INVLD_ID_PRIO[2:0]	Priority to use for found responses of an invalid ID. This bit is only valid when INVLD_ID_FLOOD bit is set to 1.	R/W
7	INVLD_ID_PRIO_VLD	Indicates if the priority in INVLD_ID_PRIO is valid. This bit is valid only when INVLD_ID_FLOOD bit is set to 1.	R/W
15:8	—	The read values are undefined. The write value should be 0.	R/W
19:16	INVLD_ID_FLOOD_MASK[3:0]	Forwarding mask used for frames whose ID is invalid. This bit is only valid when INVLD_ID_FLOOD bit is set to 1. Setting this mask to 0 causes the frame to be dropped.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The IALK_ID_CONFIG register configures lookup response for unknown IDs (IDs that fall beyond the range of IALK_ID_MAX to IALK_ID_MIN).

29.3.1.78 IALK_VLAN_CONFIG : Configure Lookup Response Unknown VLAN Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0358

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	UNKWN_VLAN_FLOOD_MASK[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VLANS_ENABLED[2:0]			UNKWN_VLAN_PRIO_VLD	UNKWN_VLAN_PRIO[2:0]			—	—	UNKWN_VLAN_LRN_ENA	UNKWN_VLAN_FLOOD
Value after reset:	x	x	x	x	x	0	0	0	0	0	0	0	x	x	0	0

Bit	Symbol	Function	R/W
0	UNKWN_VLAN_FLOOD	When this bit is set to 1, a frame matching the OUI and with a valid ID but having a VLAN ID not matching any of the enabled values in IALK_VLANIDn causes the IA table to return a found response using the forwarding mask in UNKWN_VLAN_FLOOD_MASK[3:0]. When this bit is set to 0, an unknown VLAN causes the IA table to return a not-found response and the switch uses the result from the L2 FDB instead. This bit is only valid when VLANS_ENABLED[2:0] is greater than 0.	R/W
1	UNKWN_VLAN_LRN_ENA	Setting this bit to 1 allows automatic learning into the L2 FDB for frames with unknown VLANs. When 0, learning is inhibited. This bit is only valid when UNKWN_VLAN_FLOOD bit is set to 1.	R/W
3:2	—	The read values are undefined. The write value should be 0.	R/W
6:4	UNKWN_VLAN_PRIO[2:0]	Priority to use for found responses for an unknown VLAN. This bit is only valid when UNKWN_VLAN_FLOOD bit is set to 1.	R/W
7	UNKWN_VLAN_PRIO_VLD	Indicates if the priority in UNKWN_VLAN_PRIO[2:0] is valid. This bit is only valid when UNKWN_VLAN_FLOOD bit is set to 1.	R/W
10:8	VLANS_ENABLED[2:0]	Configures the logical geometry of the IA table by specifying the number of distinct VLAN IDs enabled. When set to 0, no VLANs are supported and the VLAN ID for the frames is ignored. Valid values must be powers of 2 such as 0, 1, 2, or 4.	R/W
15:11	—	The read values are undefined. The write value should be 0.	R/W

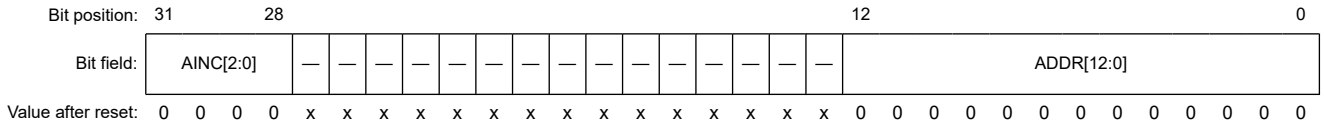
Bit	Symbol	Function	R/W
19:16	UNKWN_VLAN_FLOOD_MASK[3:0]	Forwarding mask used for frames with an unknown VLAN ID. One bit per port (bit 0 = port 0, bit 1 = port 1, and so on). This bit is only valid when UNKWN_VLAN_FLOOD bit is set to 1. Setting this bit to 0 causes the frame to be dropped.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The IALK_VLAN_CONFIG register configures lookup response for unknown VLANs and the logical geometry of the IA lookup database.

29.3.1.79 IALK_TBL_ADDR : IA Lookup Database Address Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x035C



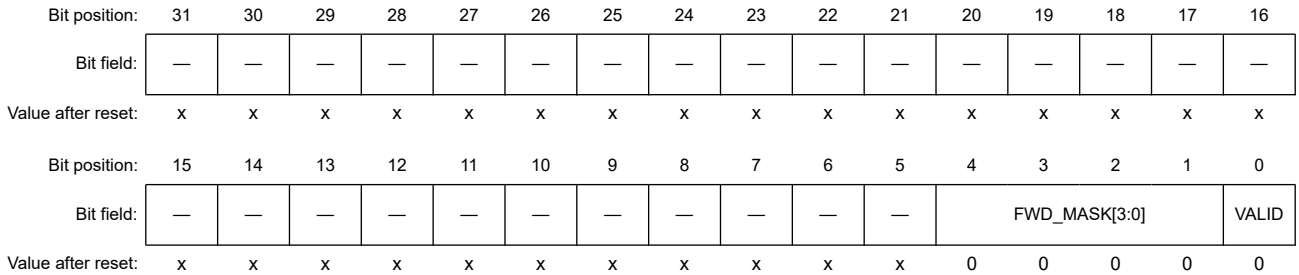
Bit	Symbol	Function	R/W
12:0	ADDR[12:0]	Defines the address to write to or read from the IA Lookup table This is the logical entry address where the LSBits specify the VLAN (based on $\log_2$(VLANS_ENABLED)) and the MSBits the ID. To this address, the value in IALK_ID_SUB is not applied.	R/W
27:13	—	The read values are undefined. The write value should be 0.	R/W
31:28	AINC[2:0]	Auto-Increment Control When set to a value other than 0, the value in ADDR is auto-incremented with each access to IALK_TBL_DATA by this amount. The address rolls over at the maximum address.	R/W

The IALK_TBL_ADDR register configures the address to access and whether to auto-increment with each read or write operation. The data to be written or to be read is accessed using the IALK_TBL_DATA register.

29.3.1.80 IALK_TBL_DATA : IA Lookup Database Data Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0360



Bit	Symbol	Function	R/W
0	VALID	Indicates whether the entry indicated by ADDR is valid or not. When set to 1, the IA Lookup table responds with found and the forwarding mask used is the value in FWD_MASK[3:0]. When set to 0, the response is the flooding mask of VLAN (IALK_VLANIDn.VLANID_FLOOD_MASK[3:0]). If VLAN is not enabled (VLANS_ENABLED is 0) then the response is not-found and the switch uses the response from the L2 FDB.	R/W
4:1	FWD_MASK[3:0]	Forwarding mask used for lookups that hit the entry and when VALID is set to 1. One bit per port (bit 0 = port 0, bit 1 = port 1, and so on).	R/W

Bit	Symbol	Function	R/W
31:5	—	The read values are undefined. The write value should be 0.	R/W

When reading from the IALK_TBL_DATA register, the entry indicated in IALK_TBL_ADDR is returned. When writing to this register, the entry indicated in IALK_TBL_ADDR is written to.

29.3.1.81 IALK_VLANIDn : IA Lookup VLANIDn Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x0380 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	VLANID_PRIO_VLD	VLANID_PRIO[2:0]			—	—	—	—	—	—	—	—	VLANID_FLOOD_MASK[3:0]			
Value after reset:	0	0	0	0	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	VLANID_LRN_ENA	VLANID_ENA	VLANID[11:0]											
Value after reset:	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	VLANID[11:0]	Configure the VLAN ID to be used for VLAN n (n: IALK_VLAN_CONFIG.VLANS_ENABLED). This bit is only valid when VLANID_ENA bit is set to 1. A value of 0 matches any VLAN ID.	R/W
12	VLANID_ENA	Enables this VLAN ID. When set to 1, the VLAN ID of the frame is compared against VLANID[11:0]. If a match is found, the lowest IALK_VLANIDn register selects the VLAN column to look for in the IA table.	R/W
13	VLANID_LRN_ENA	Configures whether automatic learning in the L2 FDB is allowed for frames matching VLAN ID. This also includes frames that match the VLAN ID and that the entry in the IA table is invalid.	R/W
15:14	—	The read values are undefined. The write value should be 0.	R/W
19:16	VLANID_FLOOD_MASK[3:0]	Flooding mask to be used for frames matching this VLAN ID but with an invalid entry in the IA table. One bit per port (bit 0 = port 0, bit 1 = port 1, and so on).	R/W
27:20	—	The read values are undefined. The write value should be 0.	R/W
30:28	VLANID_PRIO[2:0]	Priority to use for found responses.	R/W
31	VLANID_PRIO_VLD	Indicates if the priority in VLANID_PRIO[2:0] is valid.	R/W

The IALK_VLANIDn register is used to configure for each supported and enabled VLAN the ID and the flooding mask.

29.3.1.82 IMC_QLEVEL_Pn : Port n Queued Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x03C0 + 0x4 × n

Bit position:	31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
Bit field:	QUEUE7[3:0]	QUEUE6[3:0]	QUEUE5[3:0]	QUEUE4[3:0]	QUEUE3[3:0]	QUEUE2[3:0]	QUEUE1[3:0]	QUEUE0[3:0]
Value after reset:	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bit	Symbol	Function	R/W
3:0	QUEUE0[3:0]	A 4-bit value per queue indicating the number of frames stored in queue 0 Saturates at 15 if more than 15 frames are stored in the queue.	R
7:4	QUEUE1[3:0]	A 4-bit value per queue indicating the number of frames stored in queue 1 Saturates at 15 if more than 15 frames are stored in the queue.	R
11:8	QUEUE2[3:0]	A 4-bit value per queue indicating the number of frames stored in queue 2 Saturates at 15 if more than 15 frames are stored in the queue.	R
15:12	QUEUE3[3:0]	A 4-bit value per queue indicating the number of frames stored in queue 3 Saturates at 15 if more than 15 frames are stored in the queue.	R
19:16	QUEUE4[3:0]	A 4-bit value per queue indicating the number of frames stored in queue 4 Saturates at 15 if more than 15 frames are stored in the queue.	R
23:20	QUEUE5[3:0]	A 4-bit value per queue indicating the number of frames stored in queue 5 Saturates at 15 if more than 15 frames are stored in the queue.	R
27:24	QUEUE6[3:0]	A 4-bit value per queue indicating the number of frames stored in queue 6 Saturates at 15 if more than 15 frames are stored in the queue.	R
31:28	QUEUE7[3:0]	A 4-bit value per queue indicating the number of frames stored in queue 7 Saturates at 15 if more than 15 frames are stored in the queue.	R

The IMC_QLEVEL_Pn register indicates the port n status (real-time) of the number of frames stored in each of the output queues of the port.

29.3.1.83 LK_CTRL : Learning/Lookup Function Global Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0400

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DISC_UNK_SRC[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IND_VLAN	CLRTBL	—	DISC_UNK_DEST	ALW_MGRT	AGING_EN	LEARN_EN	LKUP_EN
Value after reset:	x	x	x	x	x	x	x	x	0	1	x	0	1	1	1	1

Bit	Symbol	Function	R/W
0	LKUP_EN	Lookup Controller Enable The Lookup controller can be enabled or disabled. When disabled, the switch forwarding engine skips the lookup and floods all incoming frames as if no address was found.	R/W
1	LEARN_EN	Learning Enable When set to 1 (enabled), a frame source address if not found in the lookup memory is added automatically by the hardware. The source lookup is followed by a store operation, storing the MAC address in the next free address of the lookup memory. When set to 0 (disabled), unknown source addresses is not added to the memory automatically and it is up to the software to maintain the table entries.	R/W
2	AGING_EN	Aging Enable When set to 1 (enabled), the aging process continuously scans the table for outdated entries and removes them. When set to 0 (disabled), no aging occurs.	R/W
3	ALW_MGRT	Allow Migration When set to 1 (enabled), existing dynamic entries in the Lookup table are updated if the source port of the entry changed. This requires LEARN_EN bit to be enabled. When set to 0 (disabled), no update occurs on dynamic entries when the incoming port number changes.	R/W

Bit	Symbol	Function	R/W
4	DISC_UNK_DEST	Discard Unknown Destination When set to 1 (enabled) and the destination address is not found, the frame is discarded. This setting does not affect BPDU frames, which are always accepted. When set to 0 (disabled), frames are flooded normally.	R/W
5	—	The read value is undefined. The write value should be 0.	R/W
6	CLRTBL	Clear Table Writes all table entries with 0. When this bit is set to 1, the bit remains 1 until the function has completed. Lookup is disabled during this time and the switch floods all frames. The BUSY bit within register LK_ADDR_CTRL is also set until the function has completed. This bit is set after reset to flush the table. Software might read the bit as 1 if the flushing is not completed and software should not enable the switch until the bit changes to 0.	R/W
7	IND_VLAN	Enable Independent VLAN Learning When enabled, the VLAN ID is stored together with the MAC address in the Lookup table to allow the same MAC address within different VLANs.	R/W
15:8	—	The read values are undefined. The write value should be 0.	R/W
19:16	DISC_UNK_SRC[3:0]	Discard Unknown Source Per port discard if source address not found. Bit 16 = port 0, bit 17 = port 1, and so on. When set to 1 (enabled) and the source address of an incoming frame is not found, the frame is discarded. The source is not learned, even if learning is allowed (INPUT_LEARN_BLOCK). Such discard events are counted in the switch per port input statistics IDISC_BLOCKEDn. When set to 0 (disabled), frames are accepted normally. This setting does not affect BPDU frames, which are always accepted. Independently from this setting, the unknown source interrupt is triggered and the new address is stored internally for retrieval with the LK_ADDR_CTRL.GETLASTNEW command.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

Note: For a normal switch operation, set bits [3:0] to all 1s.

29.3.1.84 LK_STATUS : Status Bits and Table Overflow Counter Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0404

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	LRNEVNT	—	OVERFLOW[13:0]													
Value after reset:	0	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AGEADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
15:0	AGEADDR[15:0]	Address the aging process will inspect when the aging timer expires next time.	R
29:16	OVERFLOW[13:0]	Counts number of table overflows that occurred (a new address was learned but the table had no storage and an older entry was deleted). The counter is cleared by writing into the register and having bit 16 set to 1.	R/W
30	—	The read value is undefined. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31	LRNEVNT	Learn Event Indicates a new source address was detected. The bit is set whenever the lookup task cannot find the source address from a received frame in the MAC address table. This bit is cleared when writing 1 to it or when issuing the GETLASTNEW command in LK_ADDR_CTRL to retrieve the latest entry. The bit asserts for every newly learned source address, independent from automatic learning or any discard options that are configured. If the bit is already set from a previous event, it remains set.	R/W

29.3.1.85 LK_ADDR_CTRL : Address Table Transaction Control and Read/Write Address

Base address: ETHSW = 0x8012_0000

Offset address: 0x0408

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BUSY	DEL_PORT	CLEAR	LOOKUP	WAIT_COMP	READ	WRITE	GETLASTNEW	CLR_STATIC	CLR_DYNAMIC	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	ADDR_MSK[11:0]											
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

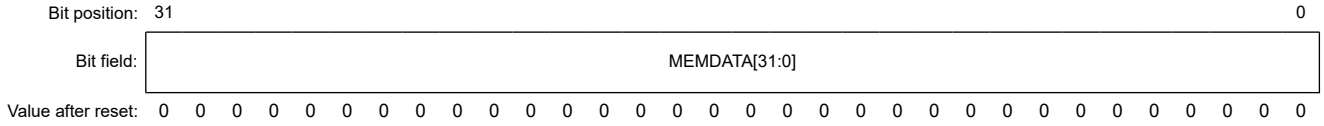
Bit	Symbol	Function	R/W
11:0	ADDR_MSK[11:0]	Memory address for read and write transactions. This is the address of a 69-bit entry. For the DEL_PORT bit, a port mask can be provided in these bits instead of the address. Bit 0 represents port 0, bit 1 port 1, and so on. When a LOOKUP function is executed and the address (LK_DATA_HI[16] = 1) is found, the memory address of the entry is returned. When a LOOKUP function is executed and the address (LK_DATA_HI[16] = 0) is not found, it returns the valid memory address of an empty entry which can be used to store the new data if required (for a 4096-size table, bits [11:3] contain the calculated hash value, bits [2:0] the offset within the 8 entries block).	R/W
21:12	—	The read values are undefined. The write value should be 0.	R/W
22	CLR_DYNAMIC	When set to 1, scans the complete table for valid dynamic entries and deletes them (writes entry with all 0s). This bit is cleared when the function has completed.	R/W
23	CLR_STATIC	When set to 1, scans the complete table for valid static entries and deletes them (writes entry with all 0s). This bit is cleared when the function has completed.	R/W
24	GETLASTNEW	When set to 1, retrieves the last source address that was not found in the table and places it into LK_DATA_LO/HI/HI2. The valid bit of the entry (bit LK_DATA_HI[16]) indicates if the address is new (when valid bit is 1) or not (when valid bit is 0) since the command was last issued. The port mask (LK_DATA_HI[24:21]) indicates at which port the new address was received. Whenever the learning task detects an unknown source address, it stores it into an internal storage that can be retrieved by this command. This occurs for each new source address, no matter if learning is enabled or not. It also stores the address when the no source discard feature is enabled. Only one entry is stored, if another new address is found before the data is read, it overwrites any existing data.	R/W
25	WRITE	When set to 1, perform a single write transaction. LK_DATA_LO/HI/HI2 must be set prior to starting the transactions.	R/W
26	READ	When set to 1, perform a single read transaction. Data is returned in LK_DATA_LO/HI/HI2.	R/W
27	WAIT_COMP	When set to 1, instructs to stall the processor bus until the transaction is completed. This allows performing of consecutive writes into this register with varying commands without the need for polling the BUSY bit.	R/W

Bit	Symbol	Function	R/W
28	LOOKUP	When set to 1, perform a lookup of the MAC address given in LK_DATA_LO/HI/HI2. When the lookup completed, the LK_DATA_HI upper 16-bit are updated with the corresponding bits read from the found entry. The ADDR_MSK field in this register shows the address where the entry is found. If the MAC address was not found, LK_DATA_HI valid bit [16] is 0 and bits [31:17] are arbitrary. The address field of this register is changed to the correct memory address that is empty and can be used to write the new entry. For example, it returns the hash calculated for the MAC address plus an offset to the empty entry within the 8 entries block. When LK_DATA_HI indicates a valid entry, LK_DATA_HI2 is valid too. This bit remains 1 together with the BUSY indication until the function completed.	R
29	CLEAR	When set to 1, writes all 0s to the entry selected by the given address set in ADDR_MSK[11:0]. If this bit is set together with the LOOKUP bit, first a lookup is performed and if the lookup succeeds, the entry is then deleted. The registers LK_DATA_LO/HI/HI2 are also cleared. The memory address in this register is set from the lookup result. If the lookup failed, the clear command is ignored and memory address is arbitrary.	R/W
30	DEL_PORT	When set to 1, scans the complete table for valid dynamic entries that contain the given ports in their destination port mask and deletes the ports or the complete entry. The port mask is provided in the ADDR_MSK[3:0] when writing this register (1 bit per port, bit 0 = port 0, bit 1 = port 1, and so on). The given port mask is ANDed with the port mask of each entry. When the result is non-zero, the entry is processed. Otherwise, it is ignored. When an entry is processed, the bits given in the mask are cleared in the port mask of the entry. If the resulting port mask is all 0s, the entry is deleted from the table (valid bit = 0). The function operates on dynamic entries only and does not affect static entries. This bit remains 1 together with the BUSY indication until the table is completely scanned.	R/W
31	BUSY	Transaction Busy Indication If this bit is 1, controller is busy. As long as the controller is busy, the corresponding command bit is set. When the controller becomes non-busy again, all command bits are cleared.	R

29.3.1.86 LK_DATA_LO : Lower 32-Bit Data of Lookup Memory Entry

Base address: ETHSW = 0x8012_0000

Offset address: 0x040C

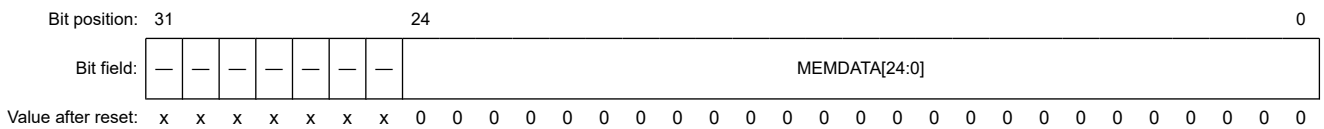


Bit	Symbol	Function	R/W
31:0	MEMDATA[31:0]	Memory Data [31:0] The lower 32-bit data of a memory entry. When writing MAC addresses, the first byte is [7:0] and the 4th byte is [31:24]. When reading, the data returned is the captured data of the last read transaction.	R/W

29.3.1.87 LK_DATA_HI : Higher 25-Bit Data of Lookup Memory Entry

Base address: ETHSW = 0x8012_0000

Offset address: 0x0410

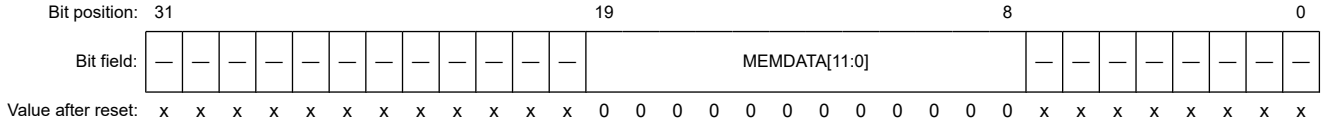


Bit	Symbol	Function	R/W
24:0	MEMDATA[24:0]	Memory Data [56:32] The next 25-bit data of a memory entry. When writing MAC addresses, the 5th byte is [7:0] and the 6th byte is [15:8]. The upper 9-bit are bits [56:48] of the memory.	R/W
31:25	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.88 LK_DATA_HI2 : Higher2 12-Bit Data of Lookup Memory Entry

Base address: ETHSW = 0x8012_0000

Offset address: 0x0414

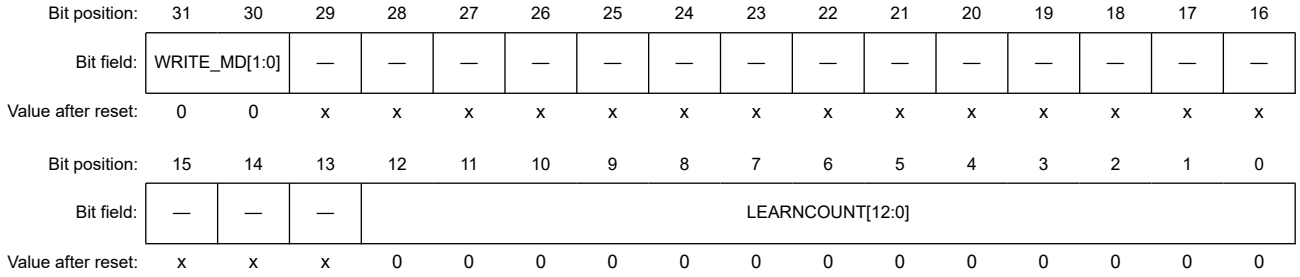


Bit	Symbol	Function	R/W
7:0	—	The read values are undefined. The write value should be 0.	R/W
19:8	MEMDATA[11:0]	Memory Data [68:57] The 12-bit VLAN ID of the entry, if VLAN IDs are stored in the address table.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

29.3.1.89 LK_LEARNCOUNT : Learned Address Count Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0418



Bit	Symbol	Function	R/W
12:0	LEARNCOUNT[12:0]	Number of Learned Addresses	R/W
29:13	—	The read values are undefined. The write value should be 0.	R/W
31:30	WRITE_MD[1:0]	These bits define how the LEARNCOUNT value is modified when writing into the register: 0 0: Sets LEARNCOUNT to given value 0 1: Increments the LEARNCOUNT value by 1 (atomic) 1 0: Decrements the LEARNCOUNT value by 1 (atomic) 1 1: Reserved, write has no effect on LEARNCOUNT	R/W

Software can increment the value LEARNCOUNT when it has added a MAC address into the memory. Software can decrement the value if an entry has been removed.

The value set in the LEARNCOUNT[12:0] is ignored when writing WRITE_MD[1:0] bit to 01b or 10b so that the hardware can change the value in parallel.

29.3.1.90 LK_AGETIME : Period of the Aging Timer

Base address: ETHSW = 0x8012_0000

Offset address: 0x041C

Bit position:	31								23																			0
Bit field:	—	—	—	—	—	—	—	—	AGETIME[23:0]																			
Value after reset:	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	AGETIME[23:0]	24-bit Timer Value	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

The timer is decremented once every 1024 switch operating clock cycles and reloaded with this value when it reaches 0. One table entry is inspected whenever the age-timer expires.

Therefore, a full aging timeout is the amount of time given in this register multiplied by the size of the lookup memory (number of entries).

Initial value: 16384 (for example, at 200 MHz switch operating clock, it is up to 5.7 minutes for 4096 entries table: n × 1024 × period × table-size = 16384 × 1024 × 5 ns × 4096)

29.3.1.91 MGMT_TAG_CONFIG : Management Tag Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x0480

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TAGFIELD[15:0]															
Value after reset:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TYPE2_EN	TYPE1_EN	—	—	AL_FRAMES	ENABLE
Value after reset:	x	x	x	x	x	x	x	x	x	x	0	0	x	x	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Enable Management Tag Insertion Module If set, the function adds tags to outgoing frames and removes tags from incoming frames at the management port. If this bit is 0 (reset default), no frame manipulation occurs.	R/W
1	AL_FRAMES	Enable Tag Insertion for All Frames If set, the function inserts the management tag into all transmitted frames at the management port. This requires that the CPU must be capable of receiving frames of at least 1526 bytes (1518 + 8) for normal frames and 1530 bytes for VLAN tagged frames. If this bit is 0 (reset default), only frames marked as management frames (BPDU frames) are tagged with extra control information. Other frames are not manipulated. The input processing (receive CPU to switch) always operates when ENABLE (bit [0]) = 1, independent from this setting.	R/W
3:2	—	The read values are undefined. The write value should be 0.	R/W
4	TYPE1_EN	When set, frames with a Type field that match the value in PRIORITY_TYPE1.TYPEVAL[15:0] have management tag inserted. This is in addition to BPDU frames which always have tag inserted. The valid bit [16] in PRIORITY_TYPE1 register has no effect for this Type field check. Validity is indicated when this bit is set.	R/W

Bit	Symbol	Function	R/W
5	TYPE2_EN	When set, frames with a Type field that match the value in PRIORITY_TYPE2.TYPEVAL[15:0] have management tag inserted. This is in addition to BPDU frames which always have tag inserted. The valid bit [16] in PRIORITY_TYPE2 register has no effect for this Type field check. Validity is indicated when this bit is set.	R/W
15:6	—	The read values are undefined. The write value should be 0.	R/W
31:16	TAGFIELD[15:0]	The value of the tag that is found in the first Type/Length field of the frame to identify that the control information is present within a frame. For example, [31:24] = first octet, [23:16] = 2nd octet.	R/W

29.3.2 1588 Timestamping Control Module Registers

29.3.2.1 TSM_CONFIG : Timestamping Control Module Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x504

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_TX_EN[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IRQ_ATIME_OVER[1:0]		—	—	IRQ_EVT_PERIOD[1:0]		—	—	IRQ_EVT_OFFSET[1:0]		—	IRQ_TSFIFO_OVR	IRQ_TEST	IRQ_EN
Value after reset:	x	x	0	0	x	x	0	0	x	x	0	0	x	0	0	0

Bit	Symbol	Function	R/W
0	IRQ_EN	Final Interrupt enable Interrupts are possible only when this bit is set to 1.	R/W
1	IRQ_TEST	Software controlled interrupt for testing purposes When set, an interrupt is triggered immediately. Can be used to cause a software controlled interrupt for testing purposes. Self-clearing, always reads back 0.	R/W
2	IRQ_TSFIFO_OVR	Trigger interrupt enable for Transmit Timestamp Capture Overflow event When set to 1, the interrupt is enabled.	R/W
3	—	The read value is undefined. The write value should be 0.	R/W
5:4	IRQ_EVT_OFFSET[1:0]	Per-timer Trigger interrupt enable for the timer offset event When set to 1, the interrupt is enabled.	R/W
7:6	—	The read values are undefined. The write value should be 0.	R/W
9:8	IRQ_EVT_PERIOD[1:0]	Per-timer Trigger interrupt enable for the timer periodical event When set to 1, the interrupt is enabled.	R/W
11:10	—	The read values are undefined. The write value should be 0.	R/W
13:12	IRQ_ATIME_OVER[1:0]	Per-timer Trigger interrupt enable for the timer wrap (reached its maximum) When set to 1, the interrupt is enabled.	R/W
15:14	—	The read values are undefined. The write value should be 0.	R/W
19:16	IRQ_TX_EN[3:0]	Per Port Transmit Timestamp Capture Interrupt Enable One bit for each port: Bit 16 = Port 0 Bit 17 = Port 1 Bit 18 = Port 2 Bit 19 = Port 3 When set, an interrupt is generated when the transmit timestamp register of the port has stored a new timestamp.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The TSM_CONFIG register is used for module configuration and interrupt enable mask.

29.3.2.2 TSM_IRQ_STAT_ACK : Interrupt Status/Acknowledge Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x508

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_TX[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IRQ_ATIME_OVER[1:0]		—	—	IRQ_EVT_PERIOD[1:0]		—	—	IRQ_EVT_OFFSET[1:0]		—	IRQ_TSFIFO_OVR	IRQ_TEST	IRQ_STAT
Value after reset:	x	x	0	0	x	x	0	0	x	x	0	0	x	0	0	0

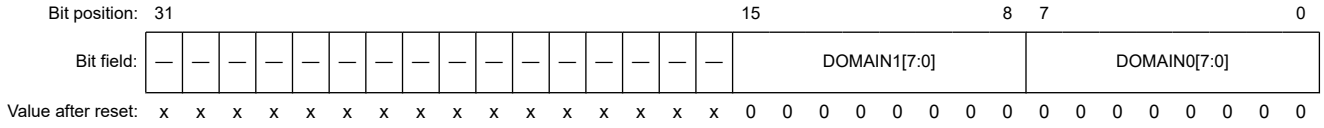
Bit	Symbol	Function	R/W
0	IRQ_STAT	Interrupt Pending Status Asserted as long as the interrupt is active.	R
1	IRQ_TEST	Test Interrupt Pending Status If an event occurs, the corresponding bit is latched high. Writing the bit with 1 clears the interrupt.	R/W
2	IRQ_TSFIFO_OVR	Transmit Timestamp Capture Overflow Interrupt Pending Status If an event occurs, the corresponding bit is latched high. Indicates that an overflow has occurred for a timestamp FIFO. The actual port where the event occurred can be determined by reading the register TS_FIFO_STATUS. The interrupt is cleared in the TS_FIFO_STATUS register by clearing the overflow latched event.	R
3	—	The read value is undefined. The write value should be 0.	R/W
5:4	IRQ_EVT_OFFSET[1:0]	Per-timer Offset Interrupt Pending Status If an event occurs, the corresponding bit is latched high. Writing the bit with 1 clears the interrupt. One bit for a timer.	R/W
7:6	—	The read values are undefined. The write value should be 0.	R/W
9:8	IRQ_EVT_PERIOD[1:0]	Per-timer Periodical Interrupt Pending Status If an event occurs, the corresponding bit is latched high. Writing the bit with 1 clears the interrupt. One bit for a timer.	R/W
11:10	—	The read values are undefined. The write value should be 0.	R/W
13:12	IRQ_ATIME_OVER[1:0]	Per-timer Overflow Interrupt Pending Status If an event occurs, the corresponding bit is latched high. Writing the bit with 1 clears the interrupt. One bit for a timer.	R/W
15:14	—	The read values are undefined. The write value should be 0.	R/W
19:16	IRQ_TX[3:0]	Per Port Transmit Timestamp Capture Interrupt One bit for each port: Bit 16 = Port 0 Bit 17 = Port 1 Bit 18 = Port 2 Bit 19 = Port 3 If the port's transmit timestamp register has stored a new timestamp, the corresponding bit is latched high. Writing a bit with 1 clears the interrupt.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The TSM_IRQ_STAT_ACK register indicates the interrupt status and acknowledge for timestamping control module.

29.3.2.3 PTP_DOMAIN : Domain Number of PTP Frame

Base address: ETHSW = 0x8012_0000

Offset address: 0x50C



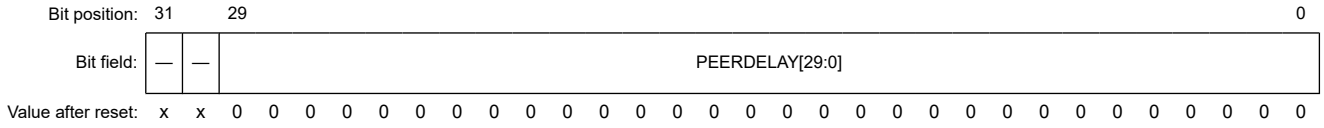
Bit	Symbol	Function	R/W
7:0	DOMAIN0[7:0]	DomainNumber to Match Against for Timer 0 When the PTP domainNumber field matches this value, the hardware timer 0 is used for timestamping the frame on receive and on transmit.	R/W
15:8	DOMAIN1[7:0]	DomainNumber to Match Against for Timer 1 When the PTP domainNumber field matches this value, the hardware timer 1 is used for timestamping the frame on receive and on transmit.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

Sets the default hardware timer to use for timestamping frames when PTPAUTORESPONSE_Pn is enabled based on the domainNumber of the PTP frame.

29.3.2.4 PEERDELAY_Pn_T0 : Port n Peer Delay Value for Timer 0 (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x540 + 0x10 × n

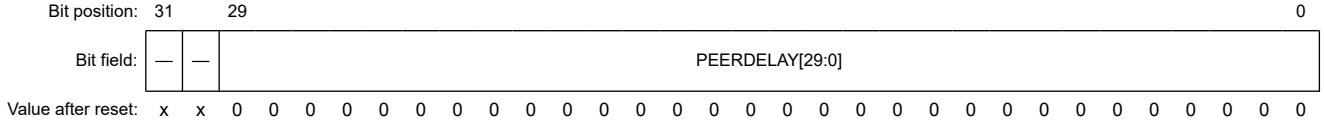


Bit	Symbol	Function	R/W
29:0	PEERDELAY[29:0]	Peer Delay Value Determined at the Port n for Timer 0 The value is added to the correction field of the received frame when a transient time correction is performed. This time value is set by the application after performing peer delay measurement on the port. Required only when the network uses peer-to-peer transparent clocks. A value of 0 is necessary to implement an end-to-end transparent clock. Value is in nanoseconds and must be less than 10^9. It must be set by the application after measuring the latency at the port where it receives SYNC messages.	R/W
31:30	—	The read values are undefined. The write value should be 0.	R/W

29.3.2.5 PEERDELAY_Pn_T1 : Port n Peer Delay Value for Timer 1 (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x544 + 0x10 × n



Bit	Symbol	Function	R/W
29:0	PEERDELAY[29:0]	Peer Delay Value Determined at the Port n for Timer 1 The value is added to the correction field of the received frame when a transient time correction is performed. This time value is set by the application after performing peer delay measurement on the port. Required only when the network uses peer-to-peer transparent clocks. A value of 0 is necessary to implement an end-to-end transparent clock. Value is in nanoseconds and must be less than 10^9. It must be set by the application after measuring the latency at the port where it receives SYNC messages.	R/W
31:30	—	The read values are undefined. The write value should be 0.	R/W

29.3.2.6 TS_FIFO_STATUS : Transmit Timestamp FIFO Status Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x5C0

Bit position:	31											19		16											3		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	FF_OVR[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	FF_VALID[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FF_VALID[3:0]	Per-port indication that a valid timestamp is available in the corresponding FIFO of the port	R
15:4	—	The read values are undefined. The write value should be 0.	R/W
19:16	FF_OVR[3:0]	Per-port indication that a timestamp cannot be written to the FIFO because of the FIFO being full. This indicates that at least one timestamp information was lost. Writing a 1 to the corresponding bit clears the value.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The TS_FIFO_STATUS register indicates the transmit timestamp FIFO status, per-port indication of valid data in the FIFO, and whether an overrun has occurred.

29.3.2.7 TS_FIFO_READ_CTRL : Transmit Timestamp FIFO Read Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x5C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	TS_ID[6:0]							—	TS_SEL	—	TS_VALID	—	—	PORT_NUM[1:0]	
Value after reset:	x	0	0	0	0	0	0	0	x	0	x	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PORT_NUM[1:0]	Port Number to Read from Write to this register to select the port number to read from. Writing to this register invalidates any previously read value (TS_VALID is set to 0).	R/W
3:2	—	This bit is read as 0. The write value should be 0.	R/W
4	TS_VALID	When reading from this register, this bit is 1 if the FIFO indicated by PORT_NUM contained valid data.	R

Bit	Symbol	Function	R/W
5	—	The read value is undefined. The write value should be 0.	R/W
6	TS_SEL	When TS_VALID is 1, TS_SEL indicates the timer used for the read timestamp.	R
7	—	The read value is undefined. The write value should be 0.	R/W
14:8	TS_ID[6:0]	When TS_VALID is 1, TS_ID indicates the ID specified by the application through the management tag control information, if present.	R
31:15	—	The read values are undefined. The write value should be 0.	R/W

The TS_FIFO_READ_CTRL register is used to control transmit timestamp FIFO read control. Selects the FIFO (port) to read from and returns the metadata such as valid indication, timer used for the timestamp, and timestamp ID. On write, the port number is written to select the FIFO (port) to read from. On read, the corresponding FIFO is read if not empty. Every read on this register generates a new read from the FIFO as long as the FIFO has data. Whether the read was successful can be determined by the TS_VALID field.

29.3.2.8 TS_FIFO_READ_TIMESTAMP : 32-bit Timestamp Value Read from FIFO

Base address: ETHSW = 0x8012_0000

Offset address: 0x5C8

Bit position:	31																												0
Bit field:	TIMESTAMP[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TIMESTAMP[31:0]	32-bit timestamp value read from the FIFO When TS_FIFO_READ_CTRL is read and TS_VALID is 1, this register must be read next to read the corresponding 32-bit timestamp value.	R

29.3.3 Interrupt Control Registers

29.3.3.1 INT_CONFIG : Interrupt Enable Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x600

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PATTERN_INT	TDMA_INT	TSM_INT	SRCFLT_WD_INT	EFP_INT	—	—	—	—	—	—	—	—	IRQ_MAC_EEE[2:0]		
Value after reset:	0	0	0	0	0	x	x	x	x	x	x	x	x	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IRQ_LINK[2:0]			HUB_INT	PRP_INT	DLR_INT	IRQ_TEST	LK_NEW_SRC	—	MDIO1	IRQ_EN
Value after reset:	x	x	x	x	x	0	0	0	0	0	0	0	0	x	0	0

Bit	Symbol	Function	R/W
0	IRQ_EN	Interrupt Global Enable When set and if any of the interrupt events occurs and is enabled, the interrupt signal is asserted 1.	R/W
1	MDIO1	Enable Interrupt on Transaction Complete from MDIO Controller When set to 1, the interrupt is enabled.	R/W
2	—	The read value is undefined. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
3	LK_NEW_SRC	Enable Interrupt for New Source Address When set, an interrupt is triggered when the lookup process detects a new (unknown) source address on a port. The interrupt triggers on every unknown source detected, independent of learning (INPUT_LEARN_BLOCK) or whether source discard (LK_CTRL) was enabled or not.	R/W
4	IRQ_TEST	When set, an interrupt is triggered immediately. Can be used to cause a software controlled interrupt for testing purposes.	R/W
5	DLR_INT	Enable Interrupt for DLR When set to 1, the interrupt is enabled. This interrupt event is controlled by the DLR_IRQ_CONTROL register. The interrupt signal (ETHSW_DLR) is not affected by the setting of this bit. See section 29.3.11.4. DLR_IRQ_CONTROL : DLR Interrupt Control Register.	R/W
6	PRP_INT	Enable Interrupt for PRP When set to 1, the interrupt is enabled. This interrupt event is controlled by the PRP_IRQ_CONTROL register. The interrupt signal (ETHSW_PRP) is not affected by the setting of this bit. See section 29.3.12.7. PRP_IRQ_CONTROL : PRP Interrupt Control Register.	R/W
7	HUB_INT	Enable Interrupt for HUB When set to 1, the interrupt is enabled. This interrupt event is controlled by the HUB_IRQ_CONTROL register. The interrupt signal (ETHSW_HUB) is not affected by the setting of this bit. See section 29.3.13.7. HUB_IRQ_CONTROL : HUB Interrupt Control Register.	R/W
10:8	IRQ_LINK[2:0]	Per Line Port Phy Link Change Interrupt Enable When set, an interrupt is generated when the Link status input from PHY of a port changes. One bit for each line port: Bit 8 = Port 0, Bit 9 = Port 1, Bit 10 = Port 2.	R/W
15:11	—	The read values are undefined. The write value should be 0.	R/W
18:16	IRQ_MAC_EEE[2:0]	Per Line Port MAC interrupt When set, an interrupt is generated when the EEE function of the port changed state. Bit 16 = Port 0, Bit 17 = Port 1, Bit 18 = Port 2	R/W
26:19	—	The read values are undefined. The write value should be 0.	R/W
27	EFP_INT	Enable Interrupt for Extended Frame Parser When set to 1, the interrupt is enabled. This interrupt event is controlled by registers in Extended Frame Parser. See section 29.4.19.11. Interrupt Source.	R/W
28	SRCFLT_WD_INT	MAC Address Source Filtering Watchdog Asserts when the source filtering watchdog triggers.	R/W
29	TSM_INT	Enable Interrupt for TSM (Timer, Timestamping) When set to 1, the interrupt is enabled. This interrupt event is controlled by the TSM_CONFIG register. See section 29.3.2.1. TSM_CONFIG : Timestamping Control Module Configuration Register.	R/W
30	TDMA_INT	Enable Interrupt for TDMA scheduler When set to 1, the interrupt is enabled. This interrupt event is controlled by the TDMA_IRQ_CONTROL register. See section 29.3.14.16. TDMA_IRQ_CONTROL : TDMA IRQ Control Register.	R/W
31	PATTERN_INT	Enable Interrupt for RX Pattern Matcher When set to 1, the interrupt is enabled. This interrupt event is controlled by the PATTERN_IRQ_CONTROL register. The interrupt signal (ETHSW_PTRNn, n = 0 to 11) is not affected by the setting of this bit. See section 29.3.15.3. PATTERN_IRQ_CONTROL : RX Pattern Matcher Interrupt Control Register.	R/W

29.3.3.2 INT_STAT_ACK : Interrupt Status/ACK Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x604

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PATTERN_INT	TDMA_INT	TSM_INT	SRCFLT_WD_INT	EFP_INT	—	—	—	—	—	—	—	—	IRQ_MAC_EEE[2:0]		
Value after reset:	0	0	0	0	0	x	x	x	x	x	x	x	x	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IRQ_LINK[2:0]			HUB_INT	PRP_INT	DLR_INT	IRQ_TEST	LK_NEW_SRC	—	MDIO1	IRQ_PEND
Value after reset:	x	x	x	x	x	0	0	0	0	0	0	0	0	x	0	0

Bit	Symbol	Function	R/W
0	IRQ_PEND	Interrupt Pending Status Asserted as long as any pending and enabled interrupt exists. Each interrupt latch bit in this register asserts when the interrupt event occurred, independent of the enable setting in INT_CONFIG. However, the IRQ_PEND asserts only if a latch bit and its corresponding enable bit are both set for any of the events. When set, the signal (ETHSW_INTR) asserts if the global enable bit INT_CONFIG.IRQ_EN is set.	R
1	MDIO1	Latched Interrupt Status for MDIO1 If an event occurs, the corresponding bit is latched high. Writing the bit with 1 clears the latch.	R/W
2	—	The read value is undefined. The write value should be 0.	R/W
3	LK_NEW_SRC	Latched Interrupt Status for LK_NEW_SRC If an event occurs, the corresponding bit is latched high. Writing the bit with 1 clears the latch.	R/W
4	IRQ_TEST	Interrupt Status for IRQ_TEST This bit is set as long as INT_CONFIG.IRQ_TEST is 1. To clear the interrupt, the INT_CONFIG.IRQ_TEST must be cleared.	R
5	DLR_INT	Interrupt Pending Status from DLR Module Asserted as long as the interrupt register of the DLR module has pending interrupts (represents the status of signal ETHSW_DLR). To clear the interrupt, the status/acknowledge register of the DLR module must be written. Writing this bit has no effect. See section 29.3.11.5. DLR_IRQ_STAT_ACK : DLR Interrupt Status/ACK Register.	R
6	PRP_INT	Interrupt Pending Status from PRP Module Asserted as long as the interrupt register of the PRP module has pending interrupts (represents status of the ETHSW_PRP signal). To clear the interrupt, the status/acknowledge register of the PRP module must be written. Writing this bit has no effect. See section 29.3.12.8. PRP_IRQ_STAT_ACK : PRP Interrupt Status/ACK Register.	R
7	HUB_INT	Interrupt Pending Status from Hub Module Asserted as long as the interrupt register of the Hub module has pending interrupts (represents status of the ETHSW_HUB signal). To clear the interrupt, the status/acknowledge register of the Hub module must be written. Writing this bit has no effect. See section 29.3.13.8. HUB_IRQ_STAT_ACK : HUB Interrupt Status/ACK Register.	R
10:8	IRQ_LINK[2:0]	Interrupt Pending per Line Port Phy Link Change Interrupt When set, the corresponding port link state changes. Writing a bit with 1 clears the interrupt.	R/W
15:11	—	The read values are undefined. The write value should be 0.	R/W
18:16	IRQ_MAC_EEE[2:0]	Interrupt Pending Status per Line Port MAC Interrupt When set, an interrupt is generated when the EEE function of the port changed state. Writing a bit with 1 clears the interrupt.	R/W
26:19	—	The read values are undefined. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
27	EFP_INT	Interrupt from Extended Frame Parser If an event occurs, the corresponding bit is latched high. To clear the interrupt the registers in Extended Frame Parser must be written. Writing this bit has no effect. See section 29.4.19.11. Interrupt Source.	R/W
28	SRCFLT_WD_INT	Interrupt Pending Status for MAC Source Filtering Watchdog If an event occurs, the corresponding bit is latched high. Writing a bit with 1 clears the interrupt.	R/W
29	TSM_INT	Interrupt Pending Interrupt Indication from TSM (Timestamping) module If an event occurs, the corresponding bit is latched high. To clear the interrupt the TSM_IRQ_STAT_ACK register must be written. Writing this bit has no effect.	R
30	TDMA_INT	Interrupt Pending Status from TDMA Scheduler Asserted as long as any TDMA interrupt is enabled and is pending. To clear the interrupt, the TDMA_IRQ_STAT_ACK register must be used. Writing this bit has no effect.	R
31	PATTERN_INT	Interrupt Pending Status from RX Pattern Matcher Module Asserted as long as the interrupt register of the module has pending interrupts (represents OR of all the module interrupt output of the RX Pattern Matcher). To clear the interrupt, the status/acknowledge register of the pattern module must be written. Writing this bit has no effect. See section 29.3.15.4. PATTERN_IRQ_STAT_ACK : RX Pattern Matcher Interrupt Status/ACK Register.	R

29.3.4 Hardware Timers Registers

29.3.4.1 ATIME_CTRLn : Timer n Control Register (n = 0, 1)

Base address: ETHSW = 0x8012_0000

Offset address: 0x680 + 0x20 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CAPTURE_ALL	CAPTURE	—	RESTART	—	—	—	EVT_PERIOD_RST	EVT_PERIOD_ENA	—	EVT_OFFSET_ENA	ONE_SHOT	ENABLE
Value after reset:	x	x	x	0	0	x	0	x	x	x	0	0	x	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	When set to 1, the timer starts incrementing. When set to 0, the timer stops at the current value.	R/W
1	ONE_SHOT	Avoid timer wrap around. If set, the timer stops at maximum. An overflow interrupt (TSM_CONFIG.IRQ_ATIME_OVER) occurs (if enabled) when the maximum is reached. When cleared (default), the timer operates continuously.	R/W
2	EVT_OFFSET_ENA	Enable Offset Event When set to 1, an offset-event interrupt can be generated (if enabled) when the offset correction has completed. The bit is cleared when the offset event is reached so no further event is created until the bit is set again. The timer offset value should be set before.	R/W
3	—	The read value is undefined. The write value should be 0.	R/W
4	EVT_PERIOD_ENA	Enable Periodical Event When set to 1, a period event interrupt can be generated (if enabled). The timer period value should be set before.	R/W

Bit	Symbol	Function	R/W
5	EVT_PERIOD_RST	Reset Timer on Periodical Event When set to 1, the timer is reset to 0 when the period setting is reached (causing an periodical event). If cleared, the counter increments continuously unit it wraps around.	R/W
8:6	—	The read values are undefined. The write value should be 0.	R/W
9	RESTART	Resets the Timer to Zero (Command Bit) This has no effect on the counter enable. If the counter is enabled while the command is triggered, the timer is reset to 0 and starts counting from there. When set, all other bits of this register are ignored.	R/W
10	—	The read value is undefined. The write value should be 0.	R/W
11	CAPTURE	Capture Time Value (Command Bit) When set, the current time is captured and can be read from the ATIMEn and ATIME_SECn registers. The bit is cleared when the capture has completed and the time value can be read from the timer register.	R/W
12	CAPTURE_ALL	Capture All Timers Value (Command Bit) This bit is the same as the CAPTURE bit but causes all timers in the system to capture the current time simultaneously. Can be used to validate drifting of the clocks. The bit is cleared when the capture has completed, and the time value can be read from the timer register.	R/W
31:13	—	The read values are undefined. The write value should be 0.	R/W

Note: The command bits can be used to trigger the corresponding events directly. When command bit is set to 1, the write values of all other bits are ignored.

29.3.4.2 ATIMEn : Timer n Count Register (n = 0, 1)

Base address: ETHSW = 0x8012_0000

Offset address: 0x684 + 0x20 × n

Bit position:	31																												0
Bit field:	TIMER_VAL[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TIMER_VAL[31:0]	Timer Value A write sets the timer. A read returns the last capture value. To read the current value, a capture command must be issued in the ATIME_CTRLn register. The value represents true nanoseconds (ns). The seconds value, see section 29.3.4.7. ATIME_SECn : Timer n Seconds Time Register (n = 0, 1), must be programmed before writing to this register.	R/W

29.3.4.3 ATIME_OFFSETn : Timer n Offset Register (n = 0, 1)

Base address: ETHSW = 0x8012_0000

Offset address: 0x688 + 0x20 × n

Bit position:	31																												0
Bit field:	OFFSET[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	OFFSET[31:0]	Value used for performing offset corrections without changing the drift correction The value is used in two different ways: ● If ATIME_OFFS_CORRn is 0, the value is immediately added to the current timer value when this register is written. The value must be given in true nanoseconds. When a negative value is given, the value is effectively subtracted. The actual correction value has an error of 2 timer clock periods, therefore to the desired correction value a correction factor of 2 × ts_cyc (timer clock cycle period) must be added. The rate for ATIME_OFFSETn must be [- (PERIOD - 2 × ATIME_INCn), + (PERIOD - 2 × ATIME_INCn)], where period is the value in ATIME_EVT_PERIODn. This function must not be used when the timer is being used by the TDMA. ● If ATIME_OFFS_CORRn is not 0, the value defines how many offset corrections to be performed. After each correction, the value is decremented by 1 until it reaches 0. When it has reached 0, no more offset corrections are performed. The register is then also reset to 0 and if enabled (EVT_OFFSET_ENA in ATIME_CTRLn register), an offset event interrupt is generated. ATIME_OFFS_CORRn must be written prior to writing this register to define the intended function.	R/W

29.3.4.4 ATIME_EVT_PERIODn : Timer n Periodic Event Register (n = 0, 1)

Base address: ETHSW = 0x8012_0000

Offset address: 0x68C + 0x20 × n

Bit position:	31																												0
Bit field:	PERIOD[31:0]																												
Value after reset:	0	0	1	1	1	0	1	1	1	0	0	1	1	0	1	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PERIOD[31:0]	Value for generating periodic events Each time the nanoseconds timer has reached this time, the period event occurs and the nanoseconds timer restarts. If enabled (EVT_PERIOD_ENA in ATIME_CTRLn register), the period event interrupt is generated. The value represents true nanoseconds (ns). Defaults to 1×10^9 representing one second. Allowable value is 10^9 or 0 only.	R/W

29.3.4.5 ATIME_CORRn : Timer n Correction Period Register (n = 0, 1)

Base address: ETHSW = 0x8012_0000

Offset address: 0x690 + 0x20 × n

Bit position:	31	30																											0
Bit field:	—	CORR_PERIOD[30:0]																											
Value after reset:	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
30:0	CORR_PERIOD[30:0]	Correction Period Defines after how many clock cycles (125 MHz) the timer should be corrected once. Bit 31 is unused and is always 0. The amount of correction is defined in ATIME_INCn.CORR_INC[6:0]. Setting the increment amount to 0 stops the timer for one clock cycle, slowing it down. Larger values can be used to speed up the timer. The correction value is the reciprocal of the ppm deviation between the master and slave oscillators. This value is given in clock cycles, not in nanoseconds as other values.	R/W

Bit	Symbol	Function	R/W
31	—	The read value is undefined. The write value should be 0.	R/W

29.3.4.6 ATIME_INCn : Timer n Increment Register (n = 0, 1)

Base address: ETHSW = 0x8012_0000

Offset address: 0x694 + 0x20 × n

Bit position:	31									22						16	14							8	6						0
Bit field:	—	—	—	—	—	—	—	—	—	OFFS_CORR_INC[6:0]							—	CORR_INC[6:0]							—	CLK_PERIOD[6:0]						
Value after reset:	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	CLK_PERIOD[6:0]	Clock Period of the Timestamping Clock (125 MHz) in nanoseconds The timer increments by this amount with every clock cycle. The value must be set to 0x08 in this LSI. The ATIME_EVT_PERIODn must be an integer multiple of this value.	R/W
7	—	The read value is undefined. The write value should be 0.	R/W
14:8	CORR_INC[6:0]	Correction Increment Value This value is added once every time the correction timer has advanced the number of clock cycles given in ATIME_CORRn. A value smaller than in bits CLK_PERIOD[6:0] slows down the timer. A value larger than in bits CLK_PERIOD[6:0] speeds up the timer.	R/W
15	—	The read value is undefined. The write value should be 0.	R/W
22:16	OFFS_CORR_INC[6:0]	Offset Correction Increment Value This value is added once every time the offset correction counter has advanced the number of clock cycles given in ATIME_OFFS_CORRn.	R/W
31:23	—	The read values are undefined. The write value should be 0.	R/W

29.3.4.7 ATIME_SECn : Timer n Seconds Time Register (n = 0, 1)

Base address: ETHSW = 0x8012_0000

Offset address: 0x698 + 0x20 × n

Bit position:	31																														0
Bit field:	SEC_TIME[31:0]																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SEC_TIME[31:0]	Seconds Time Value The seconds timer is incremented when the nanoseconds timer has reached its period setting. When written, the value is stored in a local register only. It is then loaded into the seconds timer when the ATIMEn register is written. On a read, it returns the last captured value (after read command was issued in ATIME_CTRLn).	R/W

29.3.4.8 ATIME_OFFS_CORRn : Timer n Offset Correction Counter Register (n = 0, 1)

Base address: ETHSW = 0x8012_0000

Offset address: 0x69C + 0x20 × n

Bit position:	31																												0
Bit field:	OFFS_CORR_CNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	OFFS_CORR_CNT[31:0]	Offset Correction Counter Defines, after how many clock cycles (125 MHz) the timer should be incremented once by the amount given in ATIME_INCn.OFFS_CORR_INC[6:0]. These bits are used in combination with the ATIME_OFFSETn value to distribute the time change over a longer time. This avoids jumps in time and causes only low jitter during offset corrections.	R/W

29.3.5 MDIO Master Registers

29.3.5.1 MDIO_CFG_STATUS : MDIO Configuration and Status Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x700

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CLKDIV[8:0]									—	DISPREAM	HOLD[2:0]			READERR	BUSY
Value after reset:	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSY	MDIO Busy If set, a MDIO transaction is currently ongoing. If cleared, the application can access other registers.	R
1	READERR	MDIO Read Error If set, the last read transaction had no response from a PHY and the data read could be invalid. This can happen, if the PHY address does not match any PHY that is available on the MDIO bus.	R
4:2	HOLD[2:0]	MDIO Hold Time Setting The output timing is based on the rising edge of MDC. 0 0 0: 1 PCLKM clock cycle (default) 0 0 1: 3 PCLKM clock cycles 0 1 0: 5 PCLKM clock cycles 0 1 1: 7 PCLKM clock cycles 1 0 0: 9 PCLKM clock cycles 1 0 1: 11 PCLKM clock cycles 1 1 0: 13 PCLKM clock cycles 1 1 1: 15 PCLKM clock cycles	R/W
5	DISPREAM	Disable Preamble 0: Enable Preamble 1: Disable Preamble	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:7	CLKDIV[8:0]	MDIO Clock Divisor A value of 5 to 511 can be set. The frequency is (PCLKM clock freq.) / (2 × divisor + 1). The reset default is 40. Setting the divisor to 0 disables MDC.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.5.2 MDIO_COMMAND : MDIO PHY Command Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x704

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRANINIT	—	—	—	—	—	PHYADDR[4:0]					REGADDR[4:0]				
Value after reset:	0	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	REGADDR[4:0]	Register Address	R/W
9:5	PHYADDR[4:0]	PHY Address	R/W
14:10	—	The read values are undefined. The write value should be 0.	R/W
15	TRANINIT	If set to 1, a read transaction is initiated.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.5.3 MDIO_DATA : MDIO Data Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x708

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MDIO_DATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MDIO_DATA[15:0]	• When written: Initiates a write transaction to the PHY. The MDIO_COMMAND register must be initialized. The busy status bit (MDIO_CFG_STATUS.BUSY) is set immediately and cleared when the write transaction has finished. • When read: Returns the data read from the PHY register after a read transaction has completed (initiated by writing the MDIO_COMMAND register).	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.6 MAC Control/Status Registers

Every MAC provides an individual set of registers are listed in following tables.

29.3.6.1 REV_Pn : Port n MAC Core Revision (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x800 + 0x400 × n

Bit position:	31																												0
Bit field:	REV[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	x	1	x

Bit	Symbol	Function	R/W
31:0	REV[31:0]	MAC Core Revision n = 0 to 2: 0x00010006 n = 3: 0x00010003	R

29.3.6.2 COMMAND_CONFIG_Pn : Port n Command Configuration Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x808 + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	TIMER_SEL	—	—	TX_TRUNCATE	EFPI_SELECT	ENA_10	NO_LGTH_CHK	CNTL_FRM_ENA	—	—	—	—	—	—	—
Value after reset:	x	0	x	x	0	0	0	1	0	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	SW_RESET	—	TX_CRC_APPEND	HD_ENA	TX_ADDR_INS	PAUSE_IGNORE	PAUSE_FWD	—	PAD_EN	PROMIS_EN	ETH_SPEED	TDMA_PREBUF_DIS	RX_ENA	TX_ENA
Value after reset:	x	x	0	0	1	0	0	0	0	0	0	1	0	0	1	1

Bit	Symbol	Function	R/W
0	TX_ENA	Enable/Disable MAC Transmit Path When set to 0, the MAC transmit function is disabled. When set to 1, the MAC transmit function is enabled. Do not change this value to 0. See section 29.4.23.1. Restriction.	R/W
1	RX_ENA	Enable/Disable MAC Receive Path When set to 0, the MAC receive function is disabled. When set to 1, the MAC receive function is enabled.	R/W
2	TDMA_PREBUF_DIS	Disables pre-buffering of frames in the transmit MAC when TDMA is enabled. When set to 1, the MAC does not request a new frame from the IMC until the current frame is completed. This can cause the IPG between frames to be more than the value in TX_IPG_LENGTH. When set to 0, a small buffer in the MAC affects the precision of the TDMA for stopping the port (gating queues) so a bigger time guard band must be used when programming the TCVs.	R/W
3	ETH_SPEED	Operation Mode Definition 0: 10/100 Megabit mode 1: 1 Gigabit mode	R/W
4	PROMIS_EN	Enable/Disable MAC Promiscuous Operation When asserted (set to 1), all frames are received without Unicast address filtering. This bit is always fixed to 1.	R
5	PAD_EN	Enable/Disable Frame Padding Remove on Receive If enabled (set to 1), padding is removed from received frames before they are given to the user application. If disabled (reset to a value 0), no padding is removed on receive by the MAC. This bit is always fixed to 0.	R

Bit	Symbol	Function	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	PAUSE_FWD	Terminate/Forward Pause Frames If enabled (set to 1), pause frames are forwarded to the user application. In normal mode (reset to a value 0), pause frames are terminated and discarded in the MAC. This bit is always fixed to 0.	R
8	PAUSE_IGNORE	Ignore Pause Frame Quanta If enabled (set to 1), received pause frames are ignored by the MAC. When disabled (reset to a value 0), the transmit process is stopped for the amount of time specified in the pause quanta received within the pause frame.	R/W
9	TX_ADDR_INS	Non writable bit, fixed to 0 always.	R
10	HD_ENA	Enable auto full/half-duplex operation (set to 1) or full-duplex only (set to 0). To set half-duplex, set this bit to 1 and PHY_DUPLEX[2:0] bits of Switch Core Duplex Mode Register (SWDUPC) to 0. Otherwise, full-duplex is selected.	R/W
11	TX_CRC_APPEND	Enable CRC Append on Transmit If set to 1, the TX appends a CRC to all outgoing frames. If frame manipulation operations are activated within the switch, all MACs should be appended CRC on transmit. If no frame manipulation is configured within the switch, the CRC may be forwarded together with the frame throughout the switch.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	SW_RESET	Self Clearing Reset Command Bit When set to 1, it clears the statistics registers and disables the MAC transmit and receive data path. Only when this bit is written in the MAC Port 0 register that the statistics of all ports are cleared. Writing this bit on any other register of the port has no effect on the statistics counters. This bit is automatically cleared when the reset is complete.	R/W
22:14	—	The read values are undefined. The write value should be 0.	R/W
23	CNTL_FRM_ENA	MAC Control Frame Enable When set to 0, MAC Control frames (type = 0x8808: Pause frame) with any opcode other than 0x0001 are silently discarded. When set to 1, MAC control frames with any opcode other than 0x0001 are accepted and forwarded to the Client interface.	R/W
24	NO_LGTH_CHK	Payload Length Check Disable When set to 0, the Core checks the payload length of the frame with the Frame Length/Type field. When set to 1, the payload length check is disabled.	R/W
25	ENA_10	This bit has no effect except PHYSPEED bit of STATUS_Pn register.	R/W
26	EFPI_SELECT	Extended Frame Parser (EFP) Enable 0: EFP disable 1: EFP enable	R/W
27	TX_TRUNCATE	Selects whether to truncate the transmitted frames when the frame is received with an error. In cut-through mode, the frame may have commenced transmission before the CRC is checked. Setting this bit to 1 causes the switch to shorten the frame by 8 to 12 bytes when the frame is marked as an errored frame by the switch. When set to 0 the frame length is not modified.	R/W
29:28	—	The read values are undefined. The write value should be 0.	R/W
30	TIMER_SEL	Selects the default timer to use for timestamping operations on transmit and on receive. The value is used when not overridden by the PTP auto-response function, pattern matchers or force forwarding information in a management tag. 0: Use timer 0 1: Use timer 1	R/W
31	—	The read value is undefined. The write value should be 0.	R/W

29.3.6.3 MAC_ADDR_0_Pn : Port n MAC Address Register 0 (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x80C + 0x400 × n

Bit position:	31																												0
Bit field:	MAC_ADDR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MAC_ADDR[31:0]	The first 4 bytes of the MAC address of the port. First byte is bits [7:0]. The MAC address is used on locally generated frames such as pause frames, peer-delay response.	R/W

29.3.6.4 MAC_ADDR_1_Pn : Port n MAC Address Register 1 (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x810 + 0x400 × n

Bit position:	31																15												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MAC_ADDR[15:0]												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MAC_ADDR[15:0]	The last 2 bytes of the MAC address of the port. Bits [7:0] is the 5th byte and bits [15:8] is the 6th byte.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.6.5 FRM_LENGTH_Pn : Port n Maximum Frame Length Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x814 + 0x400 × n

Bit position:	31																	13												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRM_LENGTH[13:0]											
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	1	0	1	1	1	1	1	1	0

Bit	Symbol	Function	R/W
13:0	FRM_LENGTH[13:0]	Maximum Frame Length Defines a 14-bit maximum frame length used by the MAC receive logic to check frames. This is the maximum total length of a frame allowed. Therefore, if VLAN frames must be supported, the value should be adjusted to 1522 or 1526 for single or double tagged VLAN support respectively.	R/W
31:14	—	The read values are undefined. The write value should be 0.	R/W

To allow the management port to accept a frame with the special management tag, the allowed length must be increased by 8 for this port. As typical setting (support double VLAN), 1534 on the management port and 1526 on all other ports.

29.3.6.6 PAUSE_QUANT_Pn : Port n MAC Pause Quanta (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x818 + 0x400 × n

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PAUSE_QUANT[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	PAUSE_QUANT[15:0]	Pause Quanta A 16-bit value and when set, in increment of 512 Ethernet bit times, the pause quanta used in each pause frame is sent to the remote Ethernet device.	R
31:16	—	The read values are undefined.	R

29.3.6.7 MAC_LINK_QTRIG_Pn : Port n Trigger Event Configuration Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x81C + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ENABLE	MODE	DLR_MODE	TRIGGERED	—	—	—	—	QUEUE_MASK[7:0]							
Value after reset:	0	0	0	0	x	x	x	x	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PORT_MASK[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PORT_MASK[3:0]	Per-port Bit Mask When set to 1 for a port, a frame is triggered from the closed queues indicated by QUEUE_MASK.	R/W
15:4	—	The read values are undefined. The write value should be 0.	R/W
23:16	QUEUE_MASK[7:0]	1-bit per queue indicating from which queues a frame is transmitted from the ports indicated by PORT_MASK. A single frame is transmitted per indicated port in PORT_MASK among the queues indicated by QUEUE_MASK.	R/W
27:24	—	The read values are undefined. The write value should be 0.	R/W
28	TRIGGERED	When MODE is set to 1, TRIGGERED indicates whether a frame was transmitted. When MODE is set to 0, TRIGGERED is always 0. This flag clears when the register is written.	R/W
29	DLR_MODE	When set to 0, the DLR state machine is ignored. When set to 1, the Link Queue Trigger occurs only if the DLR state machine is in the NORMAL or FAULT state.	R/W
30	MODE	When set to 0, only a single Link_Status frame is generated. This is to prevent sending multiple frames due to link flapping. When set to 1, a Link_Status frame is generated for every 1 → 0 transition in the link status.	R/W
31	ENABLE	Write to 1 to enable the Link Queue Trigger feature. When the link status (phy_link) transitions from 1 → 0, a trigger event is generated to the memory controller for the ports and queues indicated in PORT_MASK and QUEUE_MASK.	R/W

The MAC_LINK_QTRIG_Pn register enables and configures a trigger event when the link changes from active to lost.

29.3.6.8 PTPCLOCKIDENTITY1_Pn : Port n PTP Clock Identity 1 Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x830 + 0x400 × n

Bit position:	31							24	23							16	15							8	7							0
Bit field:	CLK_IDENTITY3[7:0]								CLK_IDENTITY2[7:0]								CLK_IDENTITY1[7:0]								CLK_IDENTITY0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CLK_IDENTITY0[7:0]	20, portIdentity.ClockIdentity[0]	R/W
15:8	CLK_IDENTITY1[7:0]	21, portIdentity.ClockIdentity[1]	R/W
23:16	CLK_IDENTITY2[7:0]	22, portIdentity.ClockIdentity[2]	R/W
31:24	CLK_IDENTITY3[7:0]	23, portIdentity.ClockIdentity[3]	R/W

The PTPCLOCKIDENTITY1_Pn register sets the first 4 octets of the 8-byte PTP ClockIdentity used for automatic PDELAY_RESP generation.

29.3.6.9 PTPCLOCKIDENTITY2_Pn : Port n PTP Clock Identity 2 Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x834 + 0x400 × n

Bit position:	31							24	23							16	15							8	7							0
Bit field:	CLK_IDENTITY7[7:0]								CLK_IDENTITY6[7:0]								CLK_IDENTITY5[7:0]								CLK_IDENTITY4[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CLK_IDENTITY4[7:0]	24, portIdentity.ClockIdentity[4]	R/W
15:8	CLK_IDENTITY5[7:0]	25, portIdentity.ClockIdentity[5]	R/W
23:16	CLK_IDENTITY6[7:0]	26, portIdentity.ClockIdentity[6]	R/W
31:24	CLK_IDENTITY7[7:0]	27, portIdentity.ClockIdentity[7]	R/W

The PTPCLOCKIDENTITY2_Pn register sets the last 4 octets of the 8-byte PTP ClockIdentity used for automatic PDELAY_RESP generation.

29.3.6.10 PTPAUTORESPONSE_Pn : Port n PTP Auto Response Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x838 + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PORTNUM0[7:0]								PORTNUM1[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	D_TIMER	ARSP_EN
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	ARSP_EN	Auto Response Enable Enable automatic generation of IEEE 1588v2 Layer2 peer delay response (PDELAY_RESP) messages. When this bit is 1, the MAC detects PDELAY_REQ messages causing an automatic reply with PDELAY_RESP. When this bit is 0, the MAC does not cause automatic response generation. The PRIORITY_TYPE registers can be used to define the priority for the response frames.	R/W
1	D_TIMER	Default timer to use for auto-response generation 0: Use timer 0 1: Use timer 1	R/W
15:2	—	The read values are undefined. The write value should be 0.	R/W
23:16	PORTNUM1[7:0]	29, portIdentity.PortNumber[1] (lsb)	R/W
31:24	PORTNUM0[7:0]	28, portIdentity.PortNumber[0] (msb)	R/W

The PTPAUTORESPONSE_Pn register enables automatic PDELAY_RESP message generation and defines the PTP PortNumber.

29.3.6.11 STATUS_Pn : Port n Status Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x840 + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BR_VERIF_ST[2:0]			LK_DST_ERR	TX_UNDFLW	PHYDUPLEX	PHYLINK	PHYSPEED[1:0]	
Value after reset:	x	x	x	x	x	x	x	0	0	0	0	0	1	0	x	x

Bit	Symbol	Function	R/W
1:0	PHYSPEED[1:0]	Currently Active PHY Interface Speed The information is valid only if the link is up (PHYLINK = 1). Otherwise, it can be arbitrary. Fixed to 10b for port 3 (management port) 0 0: 10 Mbps 0 1: 100 Mbps 1 0: 1 Gbps 1 1: Reserved	R

Bit	Symbol	Function	R/W
2	PHYLINK	Link status from PHY interface This is a direct representation of Link status input (ETHSW_PHYLINKn, n: 0 to 2) from PHY. Fixed to 1 for port 3 (management port) 0: link down 1: link up	R
3	PHYDUPLEX	Duplex status from PHY interface The information is valid only if the PHYLINK = 1. Otherwise, it can be arbitrary. This is a direct representation of PHY_DUPLEX[2:0] bits of Switch Core Duplex Mode Register (SWDUPC). Therefore, the actual PHY status may differ from this. Fixed to 1 for port 3 (management port) 0: half-duplex 1: full-duplex	R
4	TX_UNDFLW	Indicates that the transmit MAC underflow. This shall never occur during normal operation. Cleared when writing to 1.	R/W
5	LK_DST_ERR	Indicates that the L2 destination lookup process failed to complete in time before the next frame was received at the port. This should never occur under normal operation. The cause could be from IPG violations in the received frames. Cleared when writing to 1.	R/W
8:6	BR_VERIF_ST[2:0]	Indicates the current status of the verification according to clause 30.14.1.2 of the 802.3br specification Fixed to 000b for port 3 (management port) 0 0 0: Unknown (during reset) 0 0 1: Initial (when preemption is disabled) 0 1 0: Verifying 0 1 1: Succeeded 1 0 0: Failed 1 0 1: Disabled (if preempt verification is disabled)	R
31:9	—	The read values are undefined. The write value should be 0.	R/W

The STATUS_Pn register is a read-only register that contains information of the current speed of the port, link and duplex status as communicated by the PHY interface to the MAC.

29.3.6.12 TX_IPG_LENGTH_Pn : Port n Transmit IPG Length Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x844 + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MINRTC3GAP[4:0]				
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	0	1/0*1	1/0*1	1/0*1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TX_IPG_LENGTH[4:0]				
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	0	1	1	0	0

Bit	Symbol	Function	R/W
4:0	TX_IPG_LENGTH[4:0]	Define transmit interpacket gap in octets. Allowed values are in the range of 8 to 31.	R/W
15:5	—	The read values are undefined. The write value should be 0.	R/W
20:16	MINRTC3GAP[4:0]	Set the MinRTC3Gap parameter which defines the IPG to use when in RED period (as indicated by the TDMA scheduler). This is for use with Profinet IRT. Default is 14 bytes. The management port (Port 3) does not support this field, and bits [20:16] are not writable. Read back always return 0. If a port is not operating in Profinet IRT, bits [20:16] must be written with the same value as bits [4:0].	R/W
31:21	—	The read values are undefined. The write value should be 0.	R/W

Note 1. Initial value of MINRTC3GAP for the management port (Port 3) is 0x00. Others are 0x0E.

29.3.6.13 EEE_CTL_STAT_Pn : Port n MAC EEE Functions Control and Status (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x848 + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	STLH_LPI_IND	—	STLH_TXBUSY	STLH_LPI_TXHOLD	STLH_LPI_REQ
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	0	x	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	ST_LPI_IND	ST_TXAVAIL	ST_TXBUSY	ST_LPI_TXHOLD	ST_LPI_REQ	—	—	—	—	—	LPI_TXHOLD	LPI_REQ	EEE_AUTO
Value after reset:	x	x	x	0	0	0	0	0	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
0	EEE_AUTO	EEE Automatic Mode of Operation When this bit is 1, the MAC begins transmitting LPI sequences as soon as it is in idle for the duration configured in register EEE_IDLE_TIME_Pn. The bit should be set to 1 only following initialization of the timer registers (EEE_IDLE_TIME_Pn, EEE_TWSYS_TIME_Pn). When this bit is 0, the MAC does not perform automatic LPI transmission.	R/W
1	LPI_REQ	Request LPI Transmission when MAC Becomes Idle When this bit is 1, the MAC continues to transmit any available frames in its output queues and then transmit LPI sequences. When it transitioned into LPI, the ST_LPI_REQ status bit asserts. As long as in LPI state, the MAC does not serve new frames arriving in its output queues. Writing LPI_REQ = 1 and LPI_TXHOLD = 0 ensures all frames in the transmit queue are sent before entering LPI. If LPI_TXHOLD is set together with LPI_REQ, the MAC stops immediately after currently ongoing frame transmission completed and does not empty its output queue. Before clearing LPI_REQ, the LPI_TXHOLD bit should be set to prevent the MAC from immediately transmitting frames following LPI removal (for example, not respecting any wake time). A typical application controlled power down sequence is: LPI_REQ = 1 → wait until ST_LPI_REQ = 1 → LPI_TXHOLD = 1 → wait quiet timeout or data available → LPI_REQ = 0 → wait wake timeout → LPI_TXHOLD = 0. This bit can be set at any time (even when EEE_AUTO = 1). Set this bit to 0 for normal operation.	R/W
2	LPI_TXHOLD	MAC Transmission Hold When this bit is 1, the MAC continues to transmit a currently ongoing frame, if any, and then stops reading any more frames from its transmit FIFO/Queue. This bit must be set before LPI_REQ is cleared to ensure the MAC transitions to idle after LPI. Then LPI_TXHOLD is cleared after the (application controlled) wake time expired, allowing the MAC to transmit frames normally again. Using LPI_REQ and LPI_TXHOLD controls requires the application to implement corresponding quiet and wake timers. Set this bit to 0 for normal operation or when EEE_AUTO is used.	R/W
7:3	—	The read values are undefined. The write value should be 0.	R/W
8	ST_LPI_REQ	Status (real time) of Internal LPI_REQ to the MAC Asserts either as a result of writing the LPI_REQ bit, or when the automatic mode is enabled and the idle time has expired.	R
9	ST_LPI_TXHOLD	Status (real time) of Internal LPI_TXHOLD to the MAC Indicates the MAC is stopped and no longer serving frames. It asserts either as a result of writing the LPI_REQ or LPI_TXHOLD bit, or when the automatic mode is enabled and the MAC is not yet allowed to transmit frames.	R
10	ST_TXBUSY	Status (real time) if the MAC is currently transmitting.	R
11	ST_TXAVAIL	Status (real time) if the MAC transmit FIFO has data available for transmission.	R

Bit	Symbol	Function	R/W
12	ST_LPI_IND	Status (real time) of Received LPI Asserts as long as the RS layer of the MAC detects LPI sequences at its receive interface.	R
15:13	—	The read values are undefined. The write value should be 0.	R/W
16	STLH_LPI_REQ	Status (latched high) of Internal LPI_REQ to the MAC Bit clears on register read.	R
17	STLH_LPI_TXHOLD	Status (latched high) of Internal LPI_TXHOLD to the MAC Bit clears on register read.	R
18	STLH_TXBUSY	Status (latched high) if the MAC is/was Transmitting Bit clears on register read.	R
19	—	The read value is undefined. The write value should be 0.	R/W
20	STLH_LPI_IND	Status (latched high) of Received LPI (ST_LPI_IND) Bit clears on register read.	R
31:21	—	The read values are undefined. The write value should be 0.	R/W

The EEE_CTL_STAT_Pn register controls MAC EEE and indicates the status.

29.3.6.14 EEE_IDLE_TIME_Pn : Port n EEE Idle Time Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x84C + 0x400 × n

Bit position:	31																													0
Bit field:	EEE_IDLE_TIME[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	EEE_IDLE_TIME[31:0]	Time (−1) the transmitter must be idle before transmission of LPI begins. A 32-bit value in steps of 32 switch operating clock cycles. A value of 0 disables the timer. The value must be set to 1 less count.	R/W

29.3.6.15 EEE_TWSYS_TIME_Pn : Port n EEE Wake Up Time Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x850 + 0x400 × n

Bit position:	31																													0
Bit field:	EEE_WKUP_TIME[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	EEE_WKUP_TIME[31:0]	Time (−1) after PHY wakeup until the MAC is allowed to begin transmitting the first frame again. A 32-bit value in steps of switch operating clock cycles. A value of 0 disables the timer. The value must be set to 1 less count.	R/W

29.3.6.16 IDLE_SLOPE_Pn : Port n MAC Traffic Shaper Bandwidth Control (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x854 + 0x400 × n

Bit position:	31																			10										0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IDLE_SLOPE[10:0]										
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	IDLE_SLOPE[10:0]	Traffic Shaper Bandwidth Control When a non-zero value is set, the shaper is active and controls the bandwidth. When this bit is 0, the shaper is disabled. The valid value is 2 to 2046.	R/W
31:11	—	The read values are undefined. The write value should be 0.	R/W

29.3.6.17 CT_DELAY_Pn : Port n Cut-Through Delay Indication Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x858 + 0x400 × n

Bit position:	31																					8								0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CT_DELAY[8:0]								
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	CT_DELAY[8:0]	Delay Value in 400 ns / 40 ns / 8 ns increments (frequency of the MII PHY interface)	R/W
31:9	—	The read values are undefined. The write value should be 0.	R/W

The CT_DELAY_Pn register indicates Cut-Through delay. This is a programmable value in MII clock cycles that can delay the assertion of the cut-through flag to the forwarding engine in order to hide the variable latency of the forwarding decision. The default value is 0 unless required for specific protocols like Profinet IRT.

Recommended values are:

- 10 Mbit/s: 0 (400 ns increments)
- 100 Mbit/s: 32 (40 ns increments)
- 1 Gbit/s: 0 (8 ns increments)

29.3.6.18 BR_CONTROL_Pn : Port n 802.3br Frame Configuration Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x85C + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RX_MCRC_INV	TX_MCRC_INV	RX_STRICT_BR_CTL	RX_BR_SMD_DIS	RX_STRICT_PRE
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	TX_VERIFY_TIME[6:0]							—	—	ADDFRAGSIZE[1:0]		—	RESPONSE_DIS	VERIFY_DIS	PREEMPT_ENA
Value after reset:	x	0	0	0	1	0	1	0	x	x	0	0	x	0	0	0

Bit	Symbol	Function	R/W
0	PREEMPT_ENA	When set to 1, enables 802.3br Frame Preemption. This bit must be set to 1 to allow 802.3br operation.	R/W
1	VERIFY_DIS	When set to 1, disables the verify process required for preemption operation. When PREEMPT_ENA is set to 1, preemption is not operational until the verify process validates that the peer port is also capable of performing frame preemption. Setting VERIFY_DIS disables this process and allows frame preemption without running the verify process.	R/W
2	RESPONSE_DIS	When set to 1 prevents the MAC from responding to "verify" frames. "verify" frames are always replied to unless PREEMPT_ENA is set to 0 and RESPONSE_DIS is set to 1. This bit must be kept at 0 to be compliant with 802.3br.	R/W
3	—	The read value is undefined. The write value should be 0.	R/W
5:4	ADDFRAGSIZE[1:0]	Minimum fragment size in increments of 64 bytes. Sets the minimum mPacket size when preempting a Preemptable frame. The default value of 0 corresponds to 64 bytes, a value of 1 corresponds to 128 bytes, and so on.	R/W
7:6	—	The read values are undefined. The write value should be 0.	R/W
14:8	TX_VERIFY_TIME[6:0]	Preemption verification timeout in milliseconds. When preemption is enabled and verification is running, this is the timeout to wait for a RESPONSE frame after sending a VERIFY frames +/− 1 ms. VERIFY frames are sent after the timeout up to two more times. Default value is 10 ms. When the value is set to less than 5 ms, it does not meet the 20% precision required by the 802.3br, so it is recommended to use larger times.	R/W
15	—	The read value is undefined. The write value should be 0.	R/W
16	RX_STRICT_PRE	When set to 1, the preamble is checked so all bytes except the SFD are 0x55. When set to 0, only the last 2 bytes of the preamble are checked (SFD/SMD and FRAG_COUNT). It is recommended to set this bit to 1 to comply with the 802.3br specification. This bit must be set to 0 if only non-802.3br traffic is expected (for example, normal Ethernet traffic) and if custom preamble is used.	R/W
17	RX_BR_SMD_DIS	When set to 1, the receiver does not decode the 802.3br SMDs and assumes all frames are express frames. This bit must be set to 0 for correct operation with 802.3br, and can be set to 1 when 802.3br is not enabled to avoid false detection of SMDs. Setting this bit to 1 also causes any preempted frame to be aborted, bringing the re-assembly state machine to IDLE.	R/W
18	RX_STRICT_BR_CTL	When set to 1, strict checking of VERIFY and RESPONSE frames is enabled. When set to 1, the frame contents and frame length checks are also performed on these frames. The mCRC is always checked regardless of the value of this register. This bit must be set to 0 to be compliant with the functionality described in IEEE 802.3br.	R/W
19	TX_MCRC_INV	When set to 1, the 32-bit XOR mask used to calculate the mCRC for transmitted frames is inverted. This bit must always be written to 0 and only used for debugging.	R/W
20	RX_MCRC_INV	When set to 1, the 32-bit XOR mask used to calculate the mCRC for received frames is inverted. This bit must always be written to 0 and only used for debugging.	R/W
31:21	—	The read values are undefined. The write value should be 0.	R/W

The BR_CONTROL_Pn register is a configuration register for the 802.3br frame preemption.

29.3.7 MAC Statistics Registers

Every MAC provides an individual set of statistics as listed in following tables.

The counters are 64-bit wide. To read a 64-bit value, first, the counter as given in table below must be read. The returned value is the lower 32 bits of the counter. At the same time, the upper 32 bits are latched into register STATS_HIWORD. The STATS_HIWORD is a single global register that stores only the higher counter bits for the most recently read statistics counter. If it is not read before another counter (in any MAC page) is accessed the value is overridden with upper data from the newly read counter. Global means although it can be read from within a MAC page, it exists only once in hardware and the very same register is read independently of which MAC page is used.

The counters can be cleared by two possible commands:

1. Issuing a soft-reset (SW_RESET) to the Port 0 MAC (COMMAND_CONFIG_P0). This also clears the test registers STATS_CLEAR_VALUELO/HI and all counters will be initialized to 0.

2. Writing register STATS_CTRL with bit 0 = 1 (CLRALL). Sets all counters to the value given in registers STATS_CLEAR_VALUELO/HI. Setting a non-zero value can be used for testing purposes. Typically an all zero value should be set for normal operation.

As the counter clearing requires many clock cycles (~ 30 × 4 ports) to sequentially update all counter memory locations a read to any statistics counter should be avoided until the clear is completed. This is indicated by register bit STATS_CTRL.CLRBUSY.

An individual per port clearing of the statistics is not available.

29.3.7.1 AFRAMESTRANSMITTEDOK_Pn : Port n MAC Transmitted Valid Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x868 + 0x400 × n

Bit position:	31																												0
Bit field:	TXVALIDCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXVALIDCOUNT[31:0]	PORT n, this field indicates the number of MAC Valid Transmitted, including pause.	R

29.3.7.2 AFRAMESRECEIVEDOK_Pn : Port n MAC Received Valid Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x86C + 0x400 × n

Bit position:	31																												0
Bit field:	RXVALIDCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXVALIDCOUNT[31:0]	PORT n, this field indicates the number of MAC Valid Received, including pause.	R

29.3.7.3 AFRAMECHECKSEQUENCEERRORS_Pn : Port n MAC FCS Error Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x870 + 0x400 × n

Bit position:	31																												0
Bit field:	FCSERRCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FCSERRCOUNT[31:0]	PORT n, this field indicates the number of MAC Valid Length but CRC error.	R

29.3.7.4 AALIGNMENTERRORS_Pn : Port n MAC Alignment Error Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x874 + 0x400 × n

Bit position:	31																												0
Bit field:	ALGNERRCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ALGNERRCOUNT[31:0]	PORT n, this field indicates the number of MAC Odd Number of Nibbles (MII) Received.	R

29.3.7.5 AOCTETSTRANSMITTEDOK_Pn : Port n MAC Transmitted Valid Frame Octets Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x878 + 0x400 × n

Bit position:	31																												0
Bit field:	TXVALIDOCTETS[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXVALIDOCTETS[31:0]	PORT n, this field indicates the octets (the payload only) of MAC Valid Transmitted.	R

29.3.7.6 AOCTETSRECEIVEDOK_Pn : Port n MAC Received Valid Frame Octets Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x87C + 0x400 × n

Bit position:	31																												0
Bit field:	RXVALIDOCTETS[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXVALIDOCTETS[31:0]	PORT n, this field indicates the octets (the payload only) of MAC Valid Received.	R

29.3.7.7 ATXPAUSEMACCTRLFRAMES_Pn : Port n MAC Transmitted Pause Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x880 + 0x400 × n

Bit position:	31																												0
Bit field:	TXPAUSECOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXPAUSECOUNT[31:0]	PORT n, this field indicates the number of MAC Valid Pause Transmitted.	R

29.3.7.8 ARXPAUSEMACCTRLFRAMES_Pn : Port n MAC Received Pause Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x884 + 0x400 × n

Bit position:	31																												0
Bit field:	RXPAUSECOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXPAUSECOUNT[31:0]	PORT n, this field indicates the number of MAC Valid Pause Received.	R

29.3.7.9 IFINERRORS_Pn : Port n MAC Input Error Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x888 + 0x400 × n

Bit position:	31																												0
Bit field:	INERRCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	INERRCOUNT[31:0]	PORT n, this field indicates the number of MAC Any Error During Reception such as CRC, Length, PHY Error, RX FIFO Overflow.	R

29.3.7.10 IFOUTERRORS_Pn : Port n MAC Output Error Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x88C + 0x400 × n

Bit position:	31																												0
Bit field:	OUTERRCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	OUTERRCOUNT[31:0]	PORT n, this field indicates the number of MAC Frame Transmitted with PHY error.	R

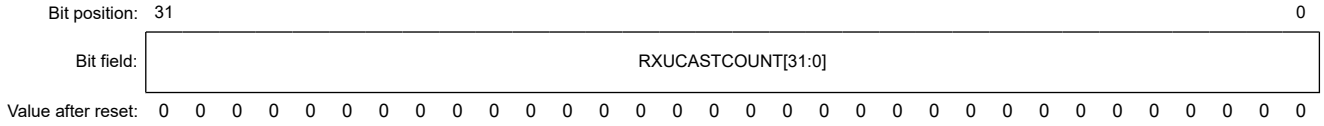
Increments on internal errors (for example, TX FIFO underflow) or when cut-through forwarding was used and a frame with error was received.

This register does not count aborted frames (half-duplex collisions).

29.3.7.11 IFINUCASTPKTS_Pn : Port n MAC Received Unicast Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x890 + 0x400 × n

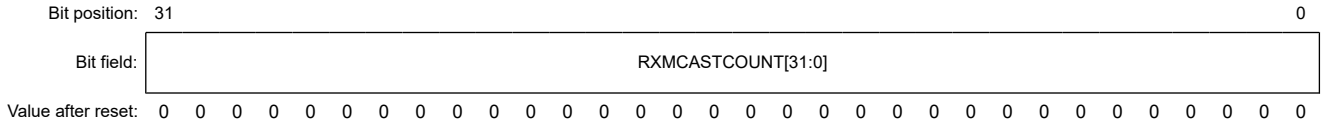


Bit	Symbol	Function	R/W
31:0	RXUCASTCOUNT[31:0]	PORT n, this field indicates the number of MAC Unicast Frame Valid Received.	R

29.3.7.12 IFINMULTICASTPKTS_Pn : Port n MAC Received Multicast Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x894 + 0x400 × n

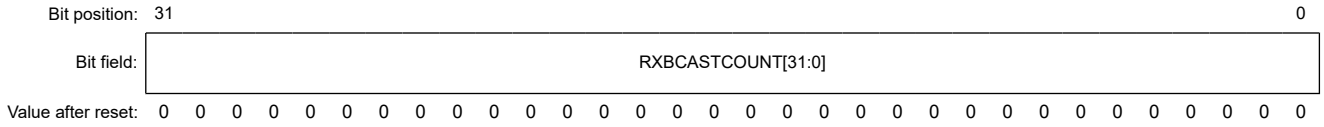


Bit	Symbol	Function	R/W
31:0	RXMCASTCOUNT[31:0]	PORT n, this field indicates the number of MAC Multicast Frame Valid Received.	R

29.3.7.13 IFINBROADCASTPKTS_Pn : Port n MAC Received Broadcast Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x898 + 0x400 × n

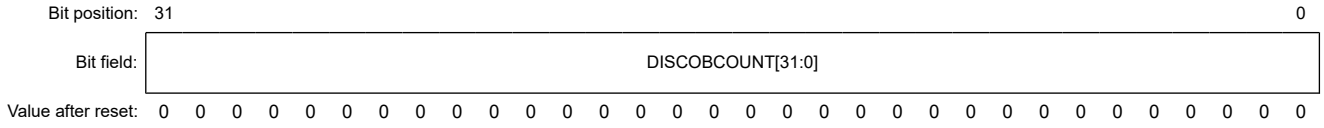


Bit	Symbol	Function	R/W
31:0	RXBCASTCOUNT[31:0]	PORT n, this field indicates the number of MAC Broadcast Frame Valid Received.	R

29.3.7.14 IFOUTDISCARDS_Pn : Port n MAC Discarded Outbound Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x89C + 0x400 × n

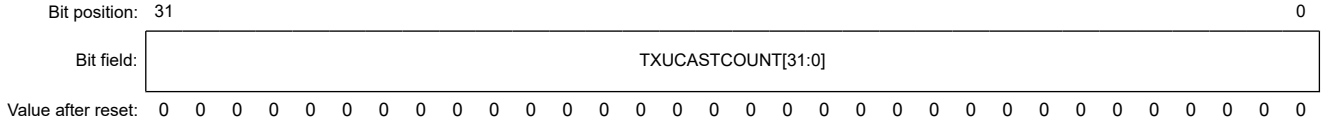


Bit	Symbol	Function	R/W
31:0	DISCOBCOUNT[31:0]	Not Applicable	R

29.3.7.15 IFOUTUCASTPKTS_Pn : Port n MAC Transmitted Unicast Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8A0 + 0x400 × n

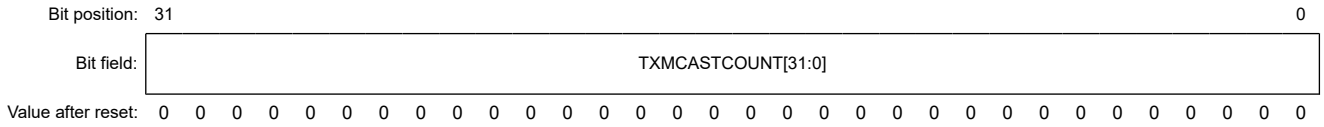


Bit	Symbol	Function	R/W
31:0	TXUCASTCOUNT[31:0]	PORT n, this field indicates the number of MAC Unicast Frame Valid Transmitted.	R

29.3.7.16 IFOUTMULTICASTPKTS_Pn : Port n MAC Transmitted Multicast Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8A4 + 0x400 × n

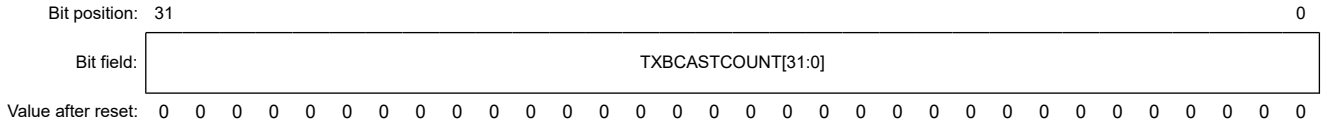


Bit	Symbol	Function	R/W
31:0	TXMCASTCOUNT[31:0]	PORT n, this field indicates the number of MAC Multicast Frame Valid Transmitted.	R

29.3.7.17 IFOUTBROADCASTPKTS_Pn : Port n MAC Transmitted Broadcast Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8A8 + 0x400 × n

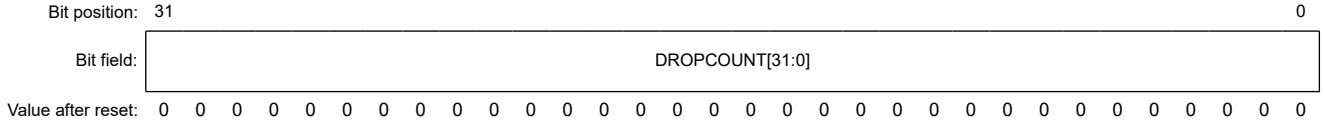


Bit	Symbol	Function	R/W
31:0	TXBCASTCOUNT[31:0]	PORT n, this field indicates the number of MAC Broadcast Frame Valid Transmitted.	R

29.3.7.18 ETHERSTATSDROPEVENTS_Pn : Port n MAC Dropped Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8AC + 0x400 × n

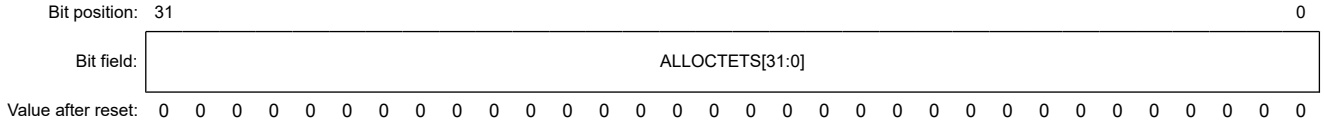


Bit	Symbol	Function	R/W
31:0	DROPCOUNT[31:0]	PORT n, this field indicates the number of MAC RX FIFO Full at frame start.	R

29.3.7.19 ETHERSTATSOCTETS_Pn : Port n MAC All Frame Octets Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8B0 + 0x400 × n

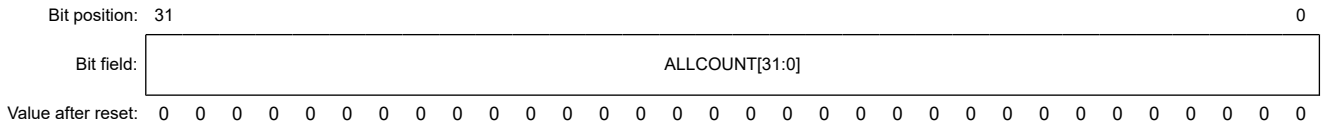


Bit	Symbol	Function	R/W
31:0	ALLOCTETS[31:0]	PORT n, this field indicates the octets of MAC All Frames, Good and Bad, including too long/short.	R

29.3.7.20 ETHERSTATSPKTS_Pn : Port n MAC All Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8B4 + 0x400 × n



Bit	Symbol	Function	R/W
31:0	ALLCOUNT[31:0]	PORT n, this field indicates the number of MAC All Frames, Good and Bad, including too long/short.	R

29.3.7.21 ETHERSTATSUNDERSIZEPKTS_Pn : Port n MAC Too Short Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8B8 + 0x400 × n

Bit position:	31																												0
Bit field:	TOOSHRTCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TOOSHRTCOUNT[31:0]	PORT n, this field indicates the number of MAC too Short, Good CRC.	R

29.3.7.22 ETHERSTATSOVERSIZEPKTS_Pn : Port n MAC Too Long Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8BC + 0x400 × n

Bit position:	31																												0
Bit field:	TOOLONGCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TOOLONGCOUNT[31:0]	PORT n, this field indicates the number of MAC too Long, Good CRC.	R

29.3.7.23 ETHERSTATSPKTS64OCTETS_Pn : Port n MAC 64 Octets Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8C0 + 0x400 × n

Bit position:	31																												0
Bit field:	OCTCNT64[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	OCTCNT64[31:0]	PORT n, this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 64 bytes).	R

29.3.7.24 ETHERSTATSPKTS65TO127OCTETS_Pn : Port n MAC 65 to 127 Octets Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8C4 + 0x400 × n

Bit position:	31																												0
Bit field:	OCTCNT65T127[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	OCTCNT65T127[31:0]	PORT n, this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 65 to 127 bytes).	R

29.3.7.25 ETHERSTATSPKTS128TO255OCTETS_Pn : Port n MAC 128 to 255 Octets Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8C8 + 0x400 × n

Bit position:	31																												0
Bit field:	OCTCNT128T255[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	OCTCNT128T255[31:0]	PORT n, this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 128 to 255 bytes).	R

29.3.7.26 ETHERSTATSPKTS256TO511OCTETS_Pn : Port n MAC 256 to 511 Octets Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8CC + 0x400 × n

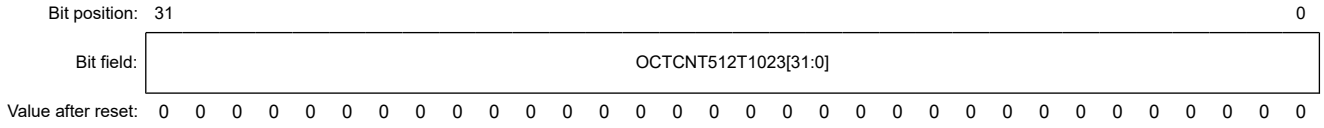
Bit position:	31																												0
Bit field:	OCTCNT256T511[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	OCTCNT256T511[31:0]	PORT n, this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 256 to 511 bytes).	R

29.3.7.27 ETHERSTATSPKTS512TO1023OCTETS_Pn : Port n MAC 512 to 1023 Octets Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8D0 + 0x400 × n

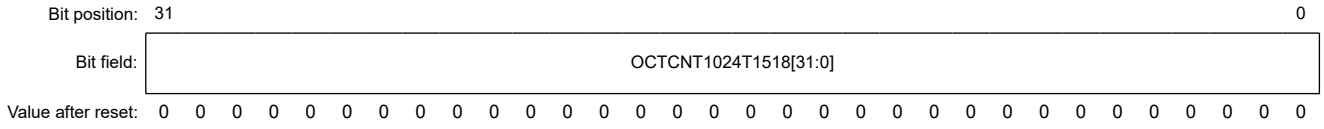


Bit	Symbol	Function	R/W
31:0	OCTCNT512T1023[31:0]	PORT n, this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 512 to 1023 bytes).	R

29.3.7.28 ETHERSTATSPKTS1024TO1518OCTETS_Pn : Port n MAC 1024 to 1518 Octets Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8D4 + 0x400 × n

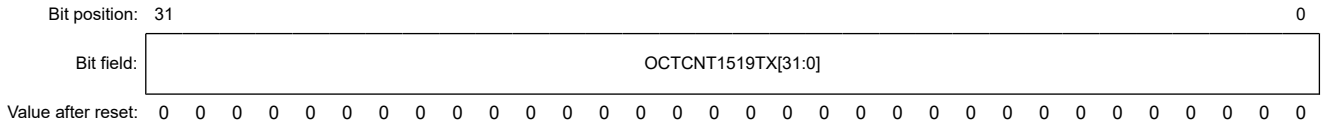


Bit	Symbol	Function	R/W
31:0	OCTCNT1024T1518[31:0]	PORT n, this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 1024 to 1518 bytes).	R

29.3.7.29 ETHERSTATSPKTS1519TOXOCTETS_Pn : Port n MAC Over 1519 Octets Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8D8 + 0x400 × n

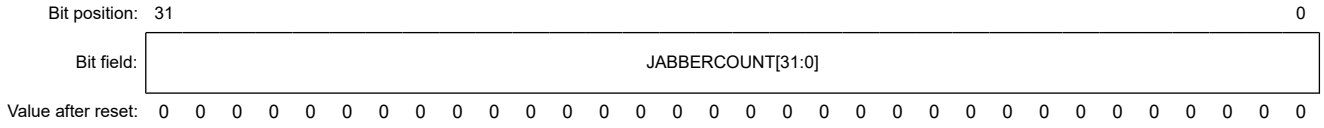


Bit	Symbol	Function	R/W
31:0	OCTCNT1519TX[31:0]	PORT n, this field indicates the number of MAC all Frames, Good and Bad (Packet Size: over 1519 bytes).	R

29.3.7.30 ETHERSTATSJABBERS_Pn : Port n MAC Jabbers Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8DC + 0x400 × n

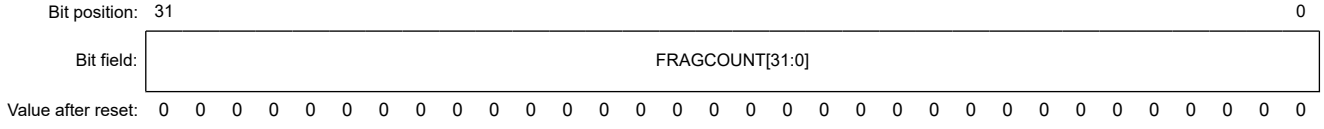


Bit	Symbol	Function	R/W
31:0	JABBERCOUNT[31:0]	PORT n, this field indicates the number of MAC too Long, Bad CRC.	R

29.3.7.31 ETHERSTATSFRAGMENTS_Pn : Port n MAC Fragment Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8E0 + 0x400 × n

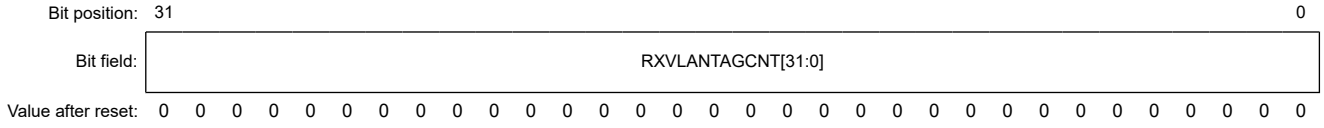


Bit	Symbol	Function	R/W
31:0	FRAGCOUNT[31:0]	PORT n, this field indicates the number of MAC too Short, Bad CRC.	R

29.3.7.32 VLANRECEIVEDOK_Pn : Port n MAC Received VLAN Tagged Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8E8 + 0x400 × n

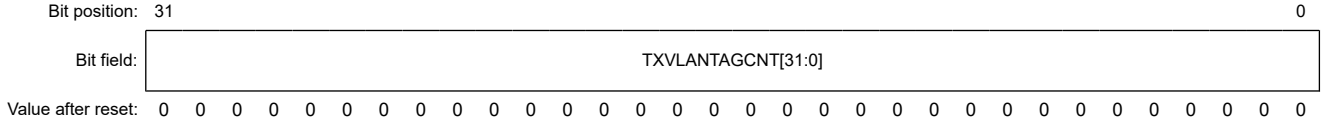


Bit	Symbol	Function	R/W
31:0	RXVLANTAGCNT[31:0]	PORT n, this field indicates the number of Good Frames with VLAN Tag Received.	R

29.3.7.33 VLANTRANSMITTEDOK_Pn : Port n MAC Transmitted VLAN Tagged Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8F4 + 0x400 × n



Bit	Symbol	Function	R/W
31:0	TXVLANTAGCNT[31:0]	PORT n, this field indicates the number of Good Frames with VLAN Tag Transmitted.	R

29.3.7.34 FRAMESRETRANSMITTED_Pn : Port n MAC Retransmitted Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x8F8 + 0x400 × n

Bit position:	31																												0
Bit field:	RETXCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RETXCOUNT[31:0]	PORT n, this field indicates the number of Transmitted Frames that experienced a collision and were retransmitted. Increments once per frame, independent of number of retransmit attempts (increment even if a frame was discarded after 16 attempts). The counter does not increment for late collisions.	R

If the Hub mode retransmit is disabled (see section 29.3.13.1. HUB_CONFIG : HUB Configuration Register), the counter increments when a collision occurred that normally would cause a retransmit, although the frame was discarded.

The counter increments with the next frame transmitted, in case the frame was discarded due to too many retransmits or hub retransmit disabled.

29.3.7.35 STATS_HIWORD_Pn : Port n MAC Statistics Counter High Word Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x900 + 0x400 × n

Bit position:	31																												0
Bit field:	STATS_HIWORD[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	STATS_HIWORD[31:0]	The latched upper 32-bit of the 64 bits MAC Statistics Counter Last Read Returns latched bits [63:32] of the last accessed counter of any MAC page.	R

This register exists once only and is global. Accessing it in any of the MAC pages reads from this same global register. It also means that it is latching for every statistics counter read in any of the MAC pages.

29.3.7.36 STATS_CTRL_Pn : Port n MAC Statistics Control Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x904 + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRBUSY	CLRALL
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	CLRALL	Self Clearing Counter Initialize Command The bit resets itself immediately, therefore always reads back as 0. When written with 1, all statistics counters of all MACs are initialized to the value given in STATS_CLEAR_VALUELO/HI. For test purposes, the clear value is programmable to perform counter wrap around or other memory tests for the counters. STATS_CLEAR_VALUELO/HI should be set to all 0s during normal operation. A clear takes many switch operating clock cycles (up to 30 × 4 ports).	R/W
1	CLRBUSY	Clear in Progress Indication The bit is set when a clear command is triggered (either by CLRALL control bit write, or soft reset (SW_RESET bit) write to Port 0 COMMAND_CONFIG_P0 register. The bit clears when all counters are initialized.	R
31:2	—	The read values are undefined. The write value should be 0.	R/W

This register exists once only and is global.

29.3.7.37 STATS_CLEAR_VALUELO_Pn : Port n MAC Statistics Clear Value Lower Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x908 + 0x400 × n

Bit position:	31	0
Bit field:	STATS_CLEAR_VALUELO[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	STATS_CLEAR_VALUELO[31:0]	PORT n, lower 32-bit of 64 bits value loaded into all counters when clearing all counters with STATS_CTRL_Pn.CLRALL command for test purposes. These bits should be set to 0 normally.	R/W

This register exists once only and is global. Clear value cannot be set for each MAC page.

29.3.7.38 STATS_CLEAR_VALUEHI_Pn : Port n MAC Statistics Clear Value Higher Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x90C + 0x400 × n

Bit position:	31																												0
Bit field:	STATS_CLEAR_VALUEHI[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	STATS_CLEAR_VALUEHI[31:0]	PORT n, upper 32-bit of 64 bits value loaded into all counters when clearing all counters with STATS_CTRL_Pn.CLRALL command for test purposes. These bits should be set to 0 normally.	R/W

This register exists once only and is global. Clear value cannot be set for each MAC page.

29.3.7.39 ADEFERRED_Pn : Port n MAC Deferred Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x910 + 0x400 × n

Bit position:	31																												0
Bit field:	DEFERCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DEFERCOUNT[31:0]	PORT n, this field indicates the number of Frame Transmitted without collision but was deferred at begin.	R

29.3.7.40 AMULTIPLECOLLISIONS_Pn : Port n MAC Multiple Collision Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x914 + 0x400 × n

Bit position:	31																												0
Bit field:	COUNTAFTMLTCOLL[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	COUNTAFTMLTCOLL[31:0]	PORT n, this field indicates the number of Good Frame Transmit after multiple collisions.	R

29.3.7.41 ASINGLECOLLISIONS_Pn : Port n MAC Single Collision Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x918 + 0x400 × n

Bit position:	31																												0
Bit field:	COUNTAFTSNGLCOLL[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	COUNTAFTSNGLCOLL[31:0]	PORT n, this field indicates the number of Good Frame Transmit after single collision.	R

29.3.7.42 ALATECOLLISIONS_Pn : Port n MAC Late Collision Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x91C + 0x400 × n

Bit position:	31																												0
Bit field:	LATECOLLCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	LATECOLLCOUNT[31:0]	PORT n, this field indicates the number of too Late Collision. Frame was aborted and not retransmitted. When this counter increments, the ifOutErrors counter also increments.	R

29.3.7.43 AEXCESSIVECOLLISIONS_Pn : Port n MAC Excessive Collision Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x920 + 0x400 × n

Bit position:	31																												0
Bit field:	EXCCOLLCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	EXCCOLLCOUNT[31:0]	PORT n, this field indicates the number of Frames Discarded due to 16 consecutive collisions. When this counter increments, the ifOutErrors counter also increments.	R

29.3.7.44 ACARRIERSENSEERRORS_Pn : Port n MAC Carrier Sense Error Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x924 + 0x400 × n

Bit position:	31																												0
Bit field:	CSERRCOUNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CSERRCOUNT[31:0]	PORT n, increments during Transmission without Collisions the PHY Carrier Sense Signal (RX_CRS) dropped or never asserted.	R

29.3.8 Extended Frame Parser Registers

29.3.8.1 Pn_QSTMACUs : Qci Stream Filter Table MAC Address Upper Part (n = 0 to 2, s = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2000 + 0x400 × n + 0x028 × s

Bit position:	31															16	15												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DASA	MACA[15:0]												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MACA[15:0]	Qci stream filter table MAC address[47:32]	R/W
16	DASA	MAC address (MACA) select 0: MACA is a source address 1: MACA is a destination address	R/W
31:17	—	The read values are undefined. The write value should be 0.	R/W

For details of the write access procedure, see Figure 29.61.

29.3.8.2 Pn_QSTMACDs : Qci Stream Filter Table MAC Address Downer Part (n = 0 to 2, s = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2004 + 0x400 × n + 0x028 × s

Bit position:	31																												0
Bit field:	MACA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MACA[31:0]	Qci stream filter table MAC address[31:0]	R/W

For details of the write access procedure, see Figure 29.61.

29.3.8.3 Pn_QSTMAMUs : Qci Stream Filter Table MAC Address Mask Upper Part (n = 0 to 2, s = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2008 + 0x400 × n + 0x028 × s

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MACAM[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MACAM[15:0]	Qci stream filter table MAC address bit mask[47:32] 0: No bit mask 1: Bit mask (the bit value of the MAC address is ignored, any value matches)	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

For details of the write access procedure, see Figure 29.61.

29.3.8.4 Pn_QSTMAMDs : Qci Stream Filter Table MAC Address Mask Downer Part (n = 0 to 2, s = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x200C + 0x400 × n + 0x028 × s

Bit position:	31																														0
Bit field:	MACAM[31:0]																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MACAM[31:0]	Qci stream filter table MAC address bit mask[31:0] 0: No bit mask 1: Bit mask (the bit value of the MAC address is ignored, any value matches)	R/W

For details of the write access procedure, see Figure 29.61.

29.3.8.5 Pn_QSFTVLs : Qci Stream Filter Table VLAN (n = 0 to 2, s = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2010 + 0x400 × n + 0x028 × s

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TAGMD[1:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCP[2:0]			DEI	VLANID[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	VLANID[11:0]	Qci stream filter table VLAN ID[11:0]	R/W
12	DEI	Qci stream filter table VLAN DEI	R/W

Bit	Symbol	Function	R/W
15:13	PCP[2:0]	Qci stream filter table VLAN PCP[2:0]	R/W
17:16	TAGMD[1:0]	Qci stream filter table VLAN Tag Mode 0 0: Un Tagged (the frame must be untagged to match to this filter) 0 1: C-Tagged (the frame must be C-Tagged to match to this filter) 1 0: Reserved 1 1: Do not check VLAN Tag (VLAN Tag check is skipped. All frames either untagged or tagged with any VLAN/PCP/DEI may match this filter.)	R/W
31:18	—	The read values are undefined. The write value should be 0.	R/W

For details of the write access procedure, see Figure 29.61.

29.3.8.6 Pn_QSFTVLMs : Qci Stream Filter Table VLAN Mask (n = 0 to 2, s = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2014 + 0x400 × n + 0x028 × s

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCPM[2:0]			DEIM	VLANIDM[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	VLANIDM[11:0]	Qci stream filter table VLAN ID[11:0] bit mask 0: No bit mask 1: Bit mask (the bit value of the VID is ignored, any value matches)	R/W
12	DEIM	Qci stream filter table VLAN DEI bit mask 0: No bit mask 1: Bit mask (the bit value of the DEI is ignored, any value matches)	R/W
15:13	PCPM[2:0]	Qci stream filter table VLAN PCP[2:0] bit mask 0: No bit mask 1: Bit mask (the bit value of the PCP is ignored, any value matches)	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

For details of the write access procedure, see Figure 29.61.

29.3.8.7 Pn_QSFTBLs : Qci Stream Filter Table SDU/Gate/Meter ID (n = 0 to 2, s = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2018 + 0x400 × n + 0x028 × s

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	QSMSM	MSDUE	MSDU[10:0]										
Value after reset:	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MEIDV	—	MEID[2:0]			GAIDV	GAID[2:0]			—	—	—	QSTE
Value after reset:	x	x	x	0	x	0	0	0	0	0	0	0	x	x	x	0

Bit	Symbol	Function	R/W
0	QSTE	Qci stream filter table Enable 0: Stream filter Table disable (All frame unmatch) 1: Stream filter Table enable	R/W
3:1	—	The read values are undefined. The write value should be 0.	R/W
6:4	GAID[2:0]	Qci filter table gate ID	R/W
7	GAIDV	Qci stream filter Gating check enable 0: Gating Check disable 1: Gating Check enable	R/W
10:8	MEID[2:0]	Qci stream filter table Meter ID	R/W
11	—	The read value is undefined. The write value should be 0.	R/W
12	MEIDV	Qci stream filter table Meter ID Valid 0: Meter Check disable 1: Meter Check enable	R/W
15:13	—	The read values are undefined. The write value should be 0.	R/W
26:16	MSDU[10:0]	Qci stream filter table max SDU size Byte unit, Include FCS	R/W
27	MSDUE	Qci stream filter table max SDU size Check enable 0: Check disable 1: Check enable	R/W
28	QSMSM	Qci stream MSDU mode 0: No effect 1: When a frame is discard by MSDU over, all frames are dropped until software clears Pn_QSEIS.QSMOIS[s].	R/W
31:29	—	The read values are undefined. The write value should be 0.	R/W

For details of the write access procedure, see Figure 29.61.

29.3.8.8 Pn_QSMFCs : Qci Stream Match Packet Count (n = 0 to 2, s = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x201C + 0x400 × n + 0x028 × s

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QSMFC[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	QSMFC[15:0]	Qci stream match packet count These bits are cleared by reading it.	R
31:16	—	The read values are undefined.	R

29.3.8.9 Pn_QMSPPCs : Qci MSDU Passed Packet Count (n = 0 to 2, s = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2020 + 0x400 × n + 0x028 × s

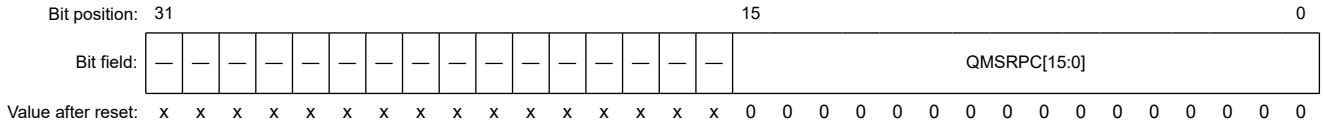
Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QMSPPC[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	QMSPPC[15:0]	Qci msdu passed packet count These bits are cleared by reading it.	R
31:16	—	The read values are undefined.	R

29.3.8.10 Pn_QMSRPCs : Qci MSDU Reject Packet Count (n = 0 to 2, s = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2024 + 0x400 × n + 0x028 × s

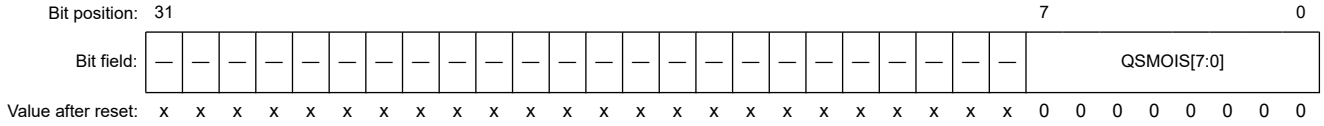


Bit	Symbol	Function	R/W
15:0	QMSRPC[15:0]	Qci MSDU rejected packet count These bits are cleared by reading it.	R
31:16	—	The read values are undefined.	R

29.3.8.11 Pn_QSEIS : Qci Stream Filter Error Interrupt Status (SDU Oversize) (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x21E8 + 0x400 × n

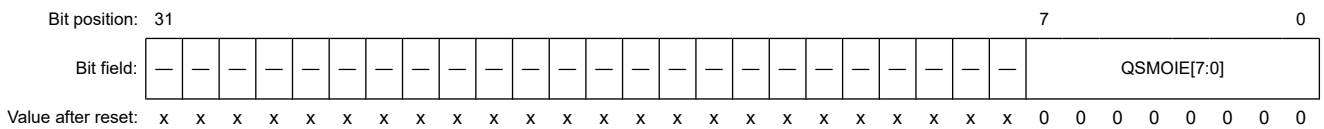


Bit	Symbol	Function	R/W
7:0	QSMOIS[7:0]	MSDU oversize frames Interrupt status[s] s: Stream ID (0 to 7) Writing 1 to this bit will clear it. Writing 0 to this bit has not effect.	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.12 Pn_QSEIE : Qci Stream Filter Error Interrupt Enable (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x21EC + 0x400 × n



Bit	Symbol	Function	R/W
7:0	QSMOIE[7:0]	MSDU oversize frames Interrupt Enable[s] s: Stream ID (0 to 7) Write 0 has no effect. 0: Interrupt which is occurred by Pn_QSEIS.QSMOIS is not appeared the interrupt output pin 1: Interrupt which is occurred by Pn_QSEIS.QSMOIS is appeared the interrupt output pin	R/W

Bit	Symbol	Function	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.13 Pn_QSEID : Qci Stream Filter Error Interrupt Disable (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x21F0 + 0x400 × n

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QSMOID[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	QSMOID[7:0]	MSDU oversize frames Interrupt Disable[s] s: Stream ID (0 to 7) Write 1 will disable Pn_QSEIE.QSMOIE, write 0 has no effect. 0: No change of Pn_QSEIE.QSMOIE 1: Set Pn_QSEIE.QSMOIE to 0	W
31:8	—	The write value should be 0.	W

Read value of this register is 0x00000000.

29.3.8.14 Pn_QGMOD : Qci Gate Mode Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2200 + 0x400 × n

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QGMOD[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	QGMOD[7:0]	Flow gate mode[g] QGMOD is used to select mode. g: Gate ID (0 to 7) 0: No effect 1: When the gate discards a frame due to a closed gate, all frames are dropped until software clears Pn_QGEIS.QGMOIS[g].	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.15 Pn_QGPPC : Qci Gate (All) Passed Packet Count Port n (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2204 + 0x400 × n

Bit position:	31																15													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QGPPC[15:0]													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

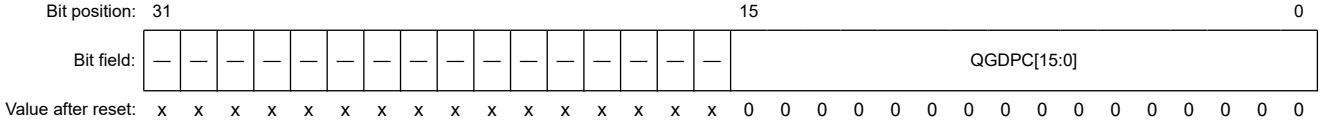
Bit	Symbol	Function	R/W
15:0	QGPPC[15:0]	Qci gate passed packet count These bits are cleared by reading it.	R

Bit	Symbol	Function	R/W
31:16	—	The read values are undefined.	R

29.3.8.16 Pn_QGDPCg : Qci Gate g Dropped Packet Count Port n (n = 0 to 2, g = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2208 + 0x400 × n + 0x008 × g

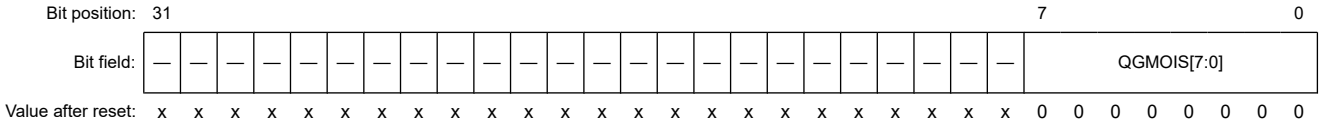


Bit	Symbol	Function	R/W
15:0	QGDPC[15:0]	Qci gate dropped packet count These bits are cleared by reading it.	R
31:16	—	The read values are undefined.	R

29.3.8.17 Pn_QGEIS : Qci Gate Error Interrupt Status (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2244 + 0x400 × n

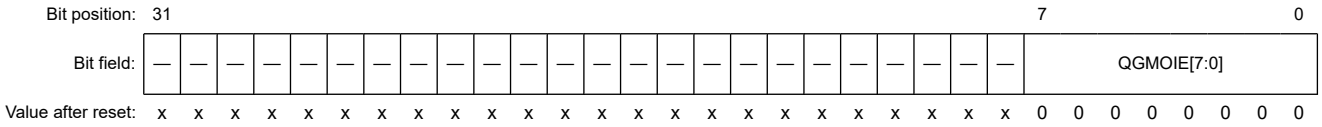


Bit	Symbol	Function	R/W
7:0	QGMOIS[7:0]	Gating error Interrupt status[g] g: Gate ID (0 to 7) Writing 1 to this bit will clear it. Writing 0 to this bit has not effect.	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.18 Pn_QGEIE : Qci Gate Error Interrupt Enable (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2248 + 0x400 × n

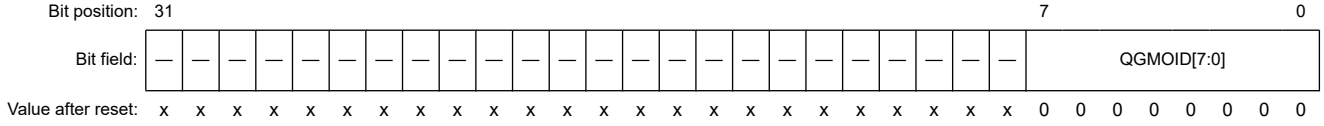


Bit	Symbol	Function	R/W
7:0	QGMOIE[7:0]	Gating error Interrupt Enable[g] g: Gate ID (0 to 7) Write 0 has no effect. 0: Interrupt which is occurred by Pn_QGEIS.QGMOIS is not appeared the interrupt output 1: Interrupt which is occurred by Pn_QGEIS.QGMOIS is appeared the interrupt output	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.19 Pn_QGEID : Qci Gate Error Interrupt Disable (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x224C + 0x400 × n



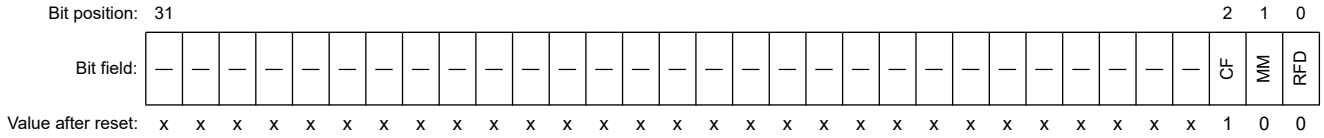
Bit	Symbol	Function	R/W
7:0	QGMOID[7:0]	Gating error Interrupt Disable[g] g: Gate ID (0 to 7) Write 1 will disable Pn_QGEIE.QGMOIE, write 0 has no effect. 0: No change of Pn_QGEIE.QGMOIE 1: Set Pn_QGEIE.QGMOIE to 0	W
31:8	—	The write value should be 0.	W

Read value of this register is 0x00000000.

29.3.8.20 Pn_QMDESCm : Qci Port n Flow Meter m Descriptor Register (n = 0 to 2, m = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2250 + 0x400 × n + 0x014 × m



Bit	Symbol	Function	R/W
0	RFD	Red frame drop RFD is used to drop red frame for meter 0: A red frame is not dropped. 1: A red frame is dropped.	R/W
1	MM	Flow meter mode MM is used to select mode for meter 0: No effect 1: When discard frame by Flow Meter, all frames are dropped until software clears Pn_QMEIS.QRFIS[m].	R/W
2	CF	Coupling flag Coupling flag for meter When coupling flag is 0, if the tokens overflow, these tokens will be added to the next bucket in the chain. When coupling flag is 1, overflow tokens are not added to the next meter.	R/W
31:3	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.21 Pn_QMCBSCm : Qci Meter CBS Configuration Port n, Meter m (n = 0 to 2, m = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2254 + 0x400 × n + 0x014 × m

Bit position:	31														17															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CBS[17:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
17:0	CBS[17:0]	CBS (Committed Burst Size) CBS value for counter Token maximum value	R/W
31:18	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.22 Pn_QMCIRCm : Qci Meter CIR Configuration n m (n = 0 to 2, m = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2258 + 0x400 × n + 0x014 × m

Bit position:	31															16															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CIR[16:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
16:0	CIR[16:0]	CIR (Committed Information Rate) CIR value for counter The token will be incremented depending on this value. CIR[16] is the token increment in byte per clock. CIR[15:0] is the token increment in sub-byte per clock.	R/W
31:17	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.23 Pn_QMGPCm : Qci Meter Green Packet Count (n = 0 to 2, m = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x225C + 0x400 × n + 0x014 × m

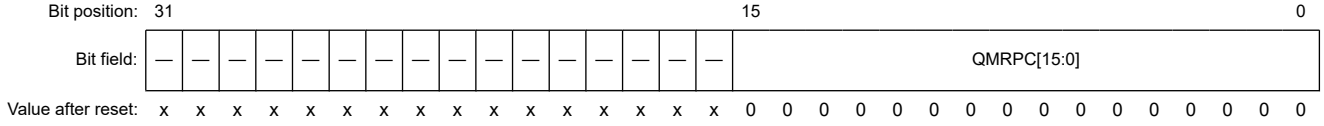
Bit position:	31																15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QMGPC[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	QMGPC[15:0]	Qci meter green packet count These bits are cleared by reading it.	R
31:16	—	The read values are undefined.	R

29.3.8.24 Pn_QMRPCm : Qci Meter Red Packet Count (n = 0 to 2, m = 0 to 7)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2260 + 0x400 × n + 0x014 × m

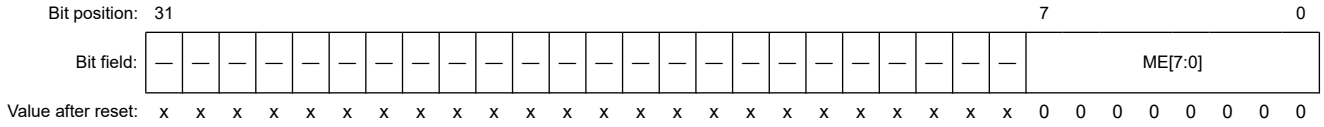


Bit	Symbol	Function	R/W
15:0	QMRPC[15:0]	Qci meter red packet count These bits are cleared by reading it.	R
31:16	—	The read values are undefined.	R

29.3.8.25 Pn_QMEC : Qci Meter Enable Configuration (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x22F0 + 0x400 × n

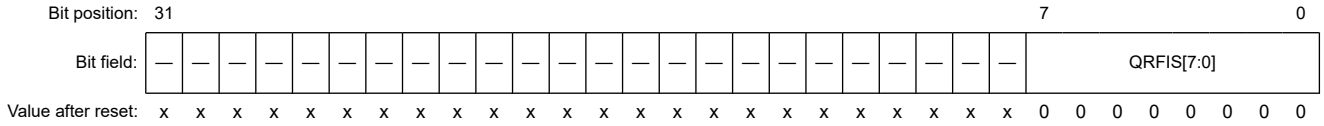


Bit	Symbol	Function	R/W
7:0	ME[7:0]	Enable meter[m] m: Meter ID (0 to 7) If Pn_QSFTBLs.MEIDV = 1 and Pn_QMEC.ME[m] = 0, those settings are invalid and the received packet is discard as Red-Frame.	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.26 Pn_QMEIS : Qci Meter Error Interrupt Status (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x22F4 + 0x400 × n



Bit	Symbol	Function	R/W
7:0	QRFIS[7:0]	Red frames Interrupt status[m] Indicate the status of frame discard by Flow Meter m: Meter ID (0 to 7) Writing 1 to this bit will clear it. Writing 0 to this bit has not effect.	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.27 Pn_QMEIE : Qci Meter Error Interrupt Enable (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x22F8 + 0x400 × n

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QRFIE[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	QRFIE[7:0]	Red frames Interrupt Enable[m] m: Meter ID (0 to 7) Write 0 has no effect. 0: Interrupt which is occurred by Pn_QMEIS.QRFIS is not appeared the interrupt output pin 1: Interrupt which is occurred by Pn_QMEIS.QRFIS is appeared the interrupt output pin	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.28 Pn_QMEID : Qci Meter Error Interrupt Disable (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x22FC + 0x400 × n

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QRFID[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	QRFID[7:0]	Red frames Interrupt Disable[m] m: Meter ID (0 to 7) Write 1 will disable Pn_QMEIE.QRFIE, write 0 has no effect. 0: No change of Pn_QMEIE.QRFIE 1: Set Pn_QMEIE.QRFIE to 0	W
31:8	—	The write value should be 0.	W

Read value of this register is 0x00000000.

29.3.8.29 Pn_PCP_REMAP : Port n VLAN Priority Code Point (PCP) Remap (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2300 + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	PCP_REMAP7[2:0]			PCP_REMAP6[2:0]			PCP_REMAP5[2:0]	
Value after reset:	x	x	x	x	x	x	x	x	1	1	1	1	1	0	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCP_REMAP5[2:0]	PCP_REMAP4[2:0]			PCP_REMAP3[2:0]			PCP_REMAP2[2:0]			PCP_REMAP1[2:0]			PCP_REMAP0[2:0]		
Value after reset:	1	1	0	0	0	1	1	0	1	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	PCP_REMAP0[2:0]	Source PCP = 0 or untagged will be remapped to this value.	R/W
5:3	PCP_REMAP1[2:0]	Source PCP = 1 will be remapped to this value.	R/W
8:6	PCP_REMAP2[2:0]	Source PCP = 2 will be remapped to this value.	R/W
11:9	PCP_REMAP3[2:0]	Source PCP = 3 will be remapped to this value.	R/W
14:12	PCP_REMAP4[2:0]	Source PCP = 4 will be remapped to this value.	R/W
17:15	PCP_REMAP5[2:0]	Source PCP = 5 will be remapped to this value.	R/W
20:18	PCP_REMAP6[2:0]	Source PCP = 6 will be remapped to this value.	R/W
23:21	PCP_REMAP7[2:0]	Source PCP = 7 will be remapped to this value.	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

Remaps TCI.PCP source to TCI.PCP destination.

29.3.8.30 Pn_VLAN_TAG : Port n VLAN TAG Information for Priority Regeneration (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2304 + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TPID[15:0]															
Value after reset:	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCP[2:0]			DEI	VID[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	VID[11:0]	Port Domain VID	R/W
12	DEI	Port Drop Eligible Indicator	R/W
15:13	PCP[2:0]	Port Priority Code Point	R/W
31:16	TPID[15:0]	Port Tag Protocol Identifier	R/W

Untagged or Priority tagged frames will be tagged with the domain VID.

29.3.8.31 Pn_VLAN_MODE : Port n VLAN Mode (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2308 + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	VICM[1:0]		VITM[1:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

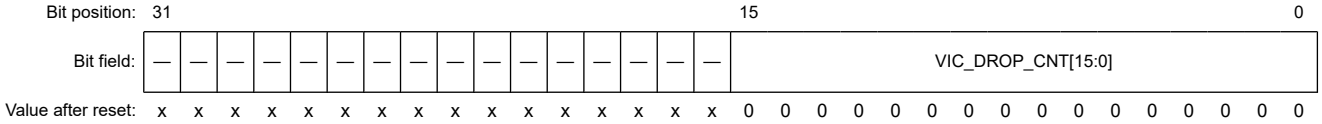
Bit	Symbol	Function	R/W
1:0	VITM[1:0]	VLAN input tagging mode See Table 29.42. 0 0: Disabled, VLAN tag is neither added nor overwritten 0 1: Single Tagging with Passthrough/ VLAN ID (VID) Overwrite 1 0: Single Tagging with Replace 1 1: Tag always	R/W
3:2	VICM[1:0]	VLAN input verification mode See Table 29.41. 0 0: Mode 0 0 1: Mode 1 1 0: Mode 2 1 1: Mode 3	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

Port n VLAN mode for ingress check and tagging.

29.3.8.32 Pn_VIC_DROP_CNT : Port n VLAN Ingress Check Drop Frame Counter (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x230C + 0x400 × n

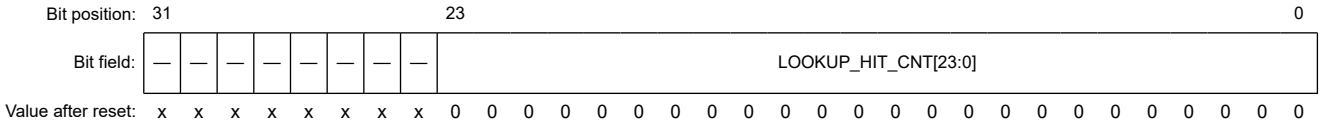


Bit	Symbol	Function	R/W
15:0	VIC_DROP_CNT[15:0]	Port n VLAN ingress check drop frame count Source counter will be reset on read.	R
31:16	—	The read values are undefined.	R

29.3.8.33 Pn_LOOKUP_HIT_CNT : Port n DST Address Lookup Hit Counter (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2328 + 0x400 × n



Bit	Symbol	Function	R/W
23:0	LOOKUP_HIT_CNT[23:0]	Port n Lookup hit count Source counter will be reset on read.	R
31:24	—	The read values are undefined.	R

29.3.8.34 Pn_ERROR_STATUS : Port n Frame Parser Runtime Error Status (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x232C + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	EOVRSZ	EUNDSZ	POVRSZ	PUNDSZ	SOPERR
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SOPERR	SOP error detected in frame parser If second (extra) SOP is received before receiving EOP, this error is asserted. Cleared by writing 1.	R/W
1	PUNDSZ	Preemptable frame under size error detected in frame parser If active frame size of preemptable frame < 64 bytes, this error is asserted. Cleared by writing 1.	R/W
2	POVRSZ	Preemptable frame over size error detected in frame parser If active frame size of preemptable frame > frm_max_length, this error is asserted. Cleared by writing 1.	R/W
3	EUNDSZ	Express frame under size error detected in frame parser If active frame size of express frame < 64 bytes, this error is asserted. Cleared by writing 1.	R/W
4	EOVRSZ	Express frame over size error detected in frame parser If active frame size of express frame > frm_max_length, this error is asserted. Cleared by writing 1.	R/W
31:5	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.35 Pn_ERROR_MASK : Port n Frame Parser Runtime Error Mask (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x2330 + 0x400 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MEOVRSZ	MEUNDSZ	MPOVRSZ	MPUNDSZ	MSOPERR
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MSOPERR	Error mask of SOPERR (SOP error) Active 1	R/W
1	MPUNDSZ	Error mask of PUNDSZ (Preemptable frame under size error) Active 1	R/W
2	MPOVRSZ	Error mask of POVRSZ (Preemptable frame over size error) Active 1	R/W

Bit	Symbol	Function	R/W
3	MEUNDSZ	Error mask of EUNDSZ (Express frame under size error) Active 1	R/W
4	MEOVRSZ	Error mask of EOVRSZ (Express frame over size error) Active 1	R/W
31:5	—	The read values are undefined. The write value should be 0.	R/W

29.3.8.36 CHANNEL_STATE : Enable/Disable State of Ingress Channels

Base address: ETHSW = 0x8012_0000

Offset address: 0x23C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CH2ACT	CH1ACT	CH0ACT
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
0	CH0ACT	State of Ingress channel port 0 0: Channel port 0 is in reset. No frames are forwarded. 1: Channel port 0 is enabled. Frames are forwarded and manipulated as configured.	R
1	CH1ACT	State of Ingress channel port 1 0: Channel port 1 is in reset. No frames are forwarded. 1: Channel port 1 is enabled. Frames are forwarded and manipulated as configured.	R
2	CH2ACT	State of Ingress channel port 2 0: Channel port 2 is in reset. No frames are forwarded. 1: Channel port 2 is enabled. Frames are forwarded and manipulated as configured.	R
31:3	—	The read values are undefined.	R

While deactivated (CHnACT = 0), all state machines of the respective channel n are in Reset state. The values of Extended Frame Parser registers are unchanged. Read/Write operation to Extended Frame Parser registers and to the Active Stream Identification table in memory are still possible.

29.3.8.37 CHANNEL_ENABLE : Enable Operation of Channel

Base address: ETHSW = 0x8012_0000

Offset address: 0x23C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CH2ENA	CH1ENA	CH0ENA
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
0	CH0ENA	Enable of Ingress channel port 0 0: Ignored 1: Enable Channel port 0	W
1	CH1ENA	Enable of Ingress channel port 1 0: Ignored 1: Enable Channel port 1	W
2	CH2ENA	Enable of Ingress channel port 2 0: Ignored 1: Enable Channel port 2	W
31:3	—	The write value should be 0.	W

29.3.8.38 CHANNEL_DISABLE : Disable and Reset Operation of Channel

Base address: ETHSW = 0x8012_0000

Offset address: 0x23C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CH2DIS	CH1DIS	CH0DIS
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
0	CH0DIS	Disable of Ingress channel port 0 0: Ignored 1: Disable and reset Channel port 0	W
1	CH1DIS	Disable of Ingress channel port 1 0: Ignored 1: Disable and reset Channel port 1	W
2	CH2DIS	Disable of Ingress channel port 2 0: Ignored 1: Disable and reset Channel port 2	W
31:3	—	The write value should be 0.	W

29.3.8.39 ASI_MEM_WDATAn : Memory Write Data Word n (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x23CC + 0x004 × n

Bit position:	31	0
Bit field:	WDATA[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	WDATA[31:0]	Destination MAC address regeneration write data Bit allocation of register ASI_MEM_WDATAn: ASI_MEM_WDATA0: WDATA[31:0] ASI_MEM_WDATA1: WDATA[63:32] ASI_MEM_WDATA2: WDATA[95:64] ASI_MEM_WDATA3: WDATA[127:96]	R/W

The Destination MAC address lookup table is accessed by an indirect addressing scheme. The data and control bits are spread over ASI_MEM_WDATAn (n = 0 to 3) registers. There is 1 lookup table per port with 128 entries each. See Table 29.43 and Table 29.44.

29.3.8.40 ASI_MEM_ADDR : Memory Address and R/W Control

Base address: ETHSW = 0x8012_0000

Offset address: 0x23DC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MEM_REQ[2:0]			MEM_WEN	ADDR[6:0]						
Value after reset:	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	ADDR[6:0]	Memory access address The Address is a hash of lookup {DA, VID}: CRC-7 Polynomial (IEC 60870-5) 0x65 = $x^7 + x^6 + x^5 + x^2 + 1$	R/W
7	MEM_WEN	Read/Write enable 0: Read 1: Write	R/W
10:8	MEM_REQ[2:0]	Memory access request Write 1 to request access. Self clear after request complete. Port 0: MEM_REQ[0] Port 1: MEM_REQ[1] Port 2: MEM_REQ[2]	R/W
31:11	—	The read values are undefined. The write value should be 0.	R/W

Lookup table entry address: 0 to 127 entries. Each of the 128 entries corresponds to one table row of the Active Stream Identification table. See section 29.4.19.4. Active Stream Identification.

The request bits MEM_REQ will be self-cleared after a valid transaction. Only 1 request at a time is allowed. A request could be a write operation to one or multiple target channels or a read operation from one channel. All 128 bits of an entry are read or written in one access cycle. Access in byte or word granularity is not possible.

Setting multiple MEM_REQ bits for write operation (MEM_WEN = 1) will write the same memory content to the respective memories of selected channels (0 to 2).

Setting multiple MEM_REQ bits for read operation (MEM_WEN = 0) is prohibited. The result of such read operation is undefined.

29.3.8.41 ASI_MEM_RDATAn : Memory Read Data Word n (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x23E0 + 0x004 × n

Bit position:	31	0
Bit field:	RDATA[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	RDATA[31:0]	Destination MAC address regeneration read data Bit allocation of register ASI_MEM_RDATAn: ASI_MEM_RDATA0: RDATA[31:0] ASI_MEM_RDATA1: RDATA[63:32] ASI_MEM_RDATA2: RDATA[95:64] ASI_MEM_RDATA3: RDATA[127:96]	R

29.3.9 Global Discard and 802.3br Statistics Registers

This address space contains a per-port set of counters for discarded frames and 802.3br statistics. The counters are organized in pages, one page per port. The pages are the same size for all ports except for the management port which does not support 802.3br.

The counters operate in rollover where the counters will roll-over their maximum value. Also, counters can be cleared when read, configurable using the STATN_CONFIG register. Counters can also be cleared by writing to them.

29.3.9.1 STATN_STATUS : Statistics Status Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3404

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSY
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	BUSY	Statistics module is busy Informal busy indicators providing a real-time indication that there is ongoing activity in a module. When this bit is 0, no counting activities occur in that module. This bit may not necessarily assert for ongoing commands and should not be used to check for command completion.	R
31:1	—	The read values are undefined.	R

29.3.9.2 STATN_CONFIG : Statistics Configure Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3408

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RESET	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLEAR_ON_READ	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x

Bit	Symbol	Function	R/W
0	—	The read value is undefined. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
1	CLEAR_ON_READ	When set to 1, a read to a counter resets it to 0. When set to 0 (default), counters are not affected by read.	R/W
30:2	—	The read values are undefined. The write value should be 0.	R/W
31	RESET	Perform a soft-reset on the module When set to 1, all internal functions are aborted and return to a stable state (flushes prescalers). It also triggers a clear of all counter memory (all ports are cleared) by setting STATN_CONTROL.CMD_CLEAR with all mask bits. Capture memory is not reset. Self-clearing bit, always reads 0.	R/W

The STATN_CONFIG register configures the statistics module functions with statistics counters wrap around.

29.3.9.3 STATN_CONTROL : Statistics Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x340C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CMD_CLEAR	—	CLEAR_PRE	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CHANMASK[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

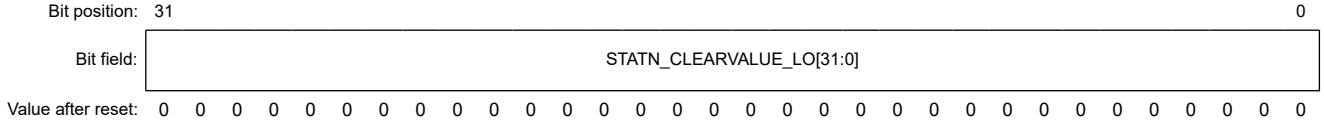
Bit	Symbol	Function	R/W
3:0	CHANMASK[3:0]	One bit per port. Bit 0 = port 0, bit 1 = port 1, and so on. Pages for ports set to 1 are cleared with the clear command. Setting multiple bits means all those ports are cleared. For self-clearing, when the command for a port begins (command is executed, but not necessarily completed yet), the corresponding bit clears itself to 0. A soft reset (STATN_CONFIG.RESET) sets all bits to 1 together with the clear command, causing all memories to be cleared.	R/W
28:4	—	The read values are undefined. The write value should be 0.	R/W
29	CLEAR_PRE	Clear the internal pre-scaler counters of ports when a clear occurs. This bit can be used together with the CMD_CLEAR command to clear the internal pre-scaler counters of the ports. When this bit is 1 and when any of the CMD_CLEAR commands is given, the memory and the prescaler values are cleared (flushes all storage). This is recommended for example, at the start of a port to ensure no pending false increments exist. When this bit is 0, only the memory stored counter values are cleared but the prescaler pending information is not modified and therefore, does not lose counts. This bit should be set to 0 for snapshot generation to avoid losing counts within the pre-scalers.	R/W
30	—	The read value is undefined. The write value should be 0.	R/W
31	CMD_CLEAR	Clear Channel Counters Command When set to 1, all counters of the channels as specified in the CHANMASK bits are cleared to 0. This can take a longer time because internal scheduling arbitration can take several 10s of cycles depending on the counters and the number of channels available and busy. For self resetting, this bit reads 0 when all channels are cleared. The soft reset from STATN_CONFIG also triggers the counter clear function.	R/W

The STATN_CONTROL register controls commands to the module for clearing and latching statistics.

29.3.9.4 STATN_CLEARVALUE_LO : Statistics Clear Value Lower Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3410

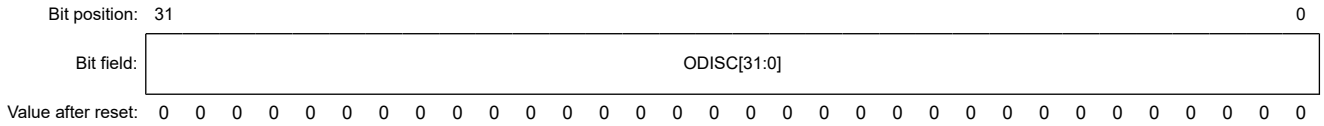


Bit	Symbol	Function	R/W
31:0	STATN_CLEARVALUE_LO[31:0]	32-bit value written into statistics memory when a clear command (STATN_CONTROL.CMD_CLEAR) is triggered (see section 29.3.9.2. STATN_CONFIG : Statistics Configure Register), or when a clear-after-read is used. Set these bits to 0 for a normal operation (intended for testing purposes to initialize the counters). The register clears to 0 when STATN_CONFIG.RESET is asserted.	R/W

29.3.9.5 ODISCn : Port n Discarded Outgoing Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3468 + 0x48 × n

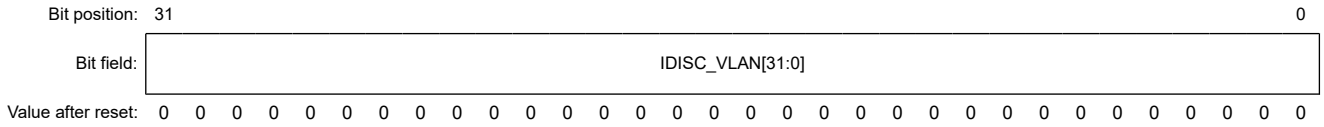


Bit	Symbol	Function	R/W
31:0	ODISC[31:0]	Port n outgoing frames discarded due to output queue congestion.	R

29.3.9.6 IDISC_VLANn : Port n Discarded Incoming VLAN Tagged Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x346C + 0x48 × n



Bit	Symbol	Function	R/W
31:0	IDISC_VLAN[31:0]	Port n incoming frames discarded due to mismatching or missing VLAN ID while VLAN verification was enabled.	R

29.3.9.7 IDISC_UNTAGGEDn : Port n Discarded Incoming VLAN Untagged Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3470 + 0x48 × n

Bit position:	31																												0
Bit field:	IDISC_UNTAGGED[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	IDISC_UNTAGGED[31:0]	Port n incoming frames discarded due to missing VLAN tag.	R

29.3.9.8 IDISC_BLOCKEDn : Port n Discarded Incoming Blocked Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3474 + 0x48 × n

Bit position:	31																												0
Bit field:	IDISC_BLOCKED[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	IDISC_BLOCKED[31:0]	Port n incoming frames discarded (after learning) as the port is configured in blocking mode.	R

29.3.9.9 IDISC_ANYn : Port n Discarded Any Frame Count Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3478 + 0x48 × n

Bit position:	31																												0
Bit field:	IDISC_ANY[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	IDISC_ANY[31:0]	Port n total incoming frames discarded. This includes IDISC_VLAN, IDSIC_UNTAGGED, IDISC_SRCFLT, and IDISC_BLOCKED. For port 3, this includes IDISC_VLAN, IDSIC_UNTAGGED, and IDISC_BLOCKED.	R

29.3.9.10 IDISC_SRCFLTn : Port n Discarded Address Source Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x347C + 0x48 × n

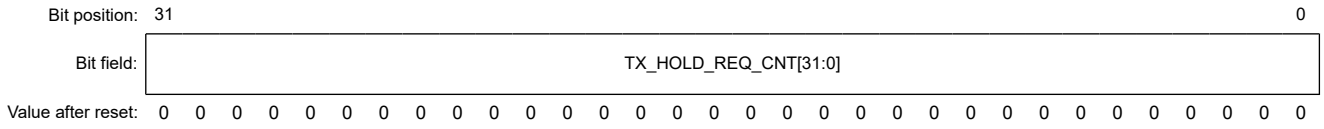
Bit position:	31																												0
Bit field:	IDISC_SRCFLT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	IDISC_SRCFLT[31:0]	Port n counts the number of incoming frames discarded due to the MAC address source filter.	R

29.3.9.11 TX_HOLD_REQ_CNTn : Port n TX Hold Request Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3480 + 0x48 × n

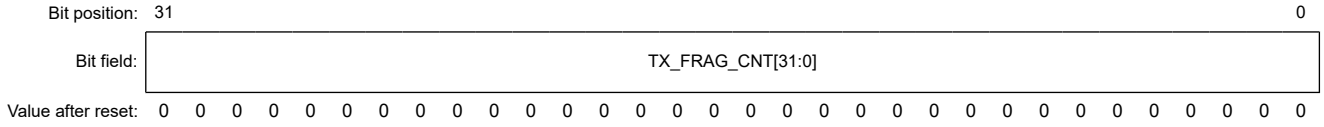


Bit	Symbol	Function	R/W
31:0	TX_HOLD_REQ_CNT[31:0]	Port n increments when MM_CTL.request (hold_req) transitions from FALSE to TRUE.	R

29.3.9.12 TX_FRAG_CNTn : Port n TX for Preemption Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3484 + 0x48 × n

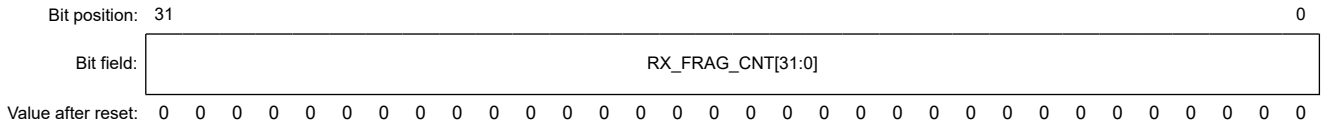


Bit	Symbol	Function	R/W
31:0	TX_FRAG_CNT[31:0]	Port n increments when an additional mPacket is transmitted due to preemption.	R

29.3.9.13 RX_FRAG_CNTn : Port n RX Continuation Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3488 + 0x48 × n



Bit	Symbol	Function	R/W
31:0	RX_FRAG_CNT[31:0]	Port n increments for every continuation mPacket received.	R

29.3.9.14 RX_ASSY_OK_CNTn : Port n RX Preempted Frame Success Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x348C + 0x48 × n

Bit position:	31																												0
Bit field:	RX_ASSY_OK_CNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RX_ASSY_OK_CNT[31:0]	Port n increments when a preempted frame is successfully assembled.	R

29.3.9.15 RX_ASSY_ERR_CNTn : Port n RX Preempted Frame Incorrect Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3490 + 0x48 × n

Bit position:	31																15												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RX_ASSY_ERR_CNT[15:0]												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RX_ASSY_ERR_CNT[15:0]	Port n increments when a preempted frame is incorrectly assembled.	R
31:16	—	The read values are undefined.	R

29.3.9.16 RX_SMD_ERR_CNTn : Port n RX SMD Frame Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3494 + 0x48 × n

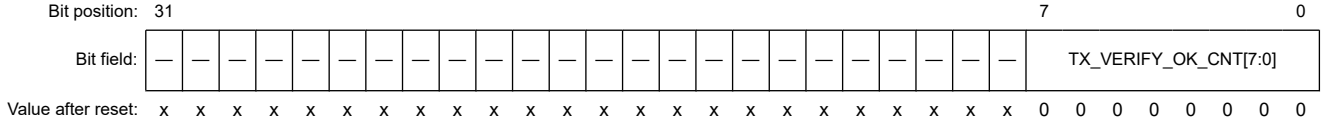
Bit position:	31																15												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RX_SMD_ERR_CNT[15:0]												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RX_SMD_ERR_CNT[15:0]	Port n increments when a frame with an SMD-Cx is received and no assembly is in progress.	R
31:16	—	The read values are undefined.	R

29.3.9.17 TX_VERIFY_OK_CNTn : Port n TX VERIFY Frame Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3498 + 0x48 × n

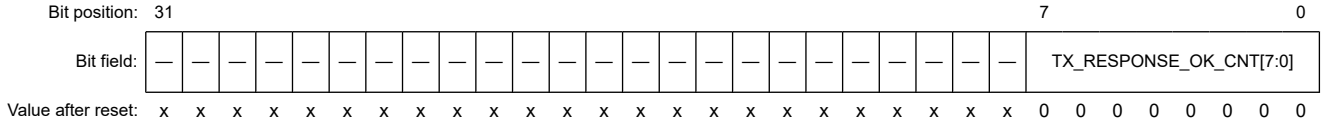


Bit	Symbol	Function	R/W
7:0	TX_VERIFY_OK_CNT[7:0]	Port n increments for every VERIFY frame transmitted.	R
31:8	—	The read values are undefined.	R

29.3.9.18 TX_RESPONSE_OK_CNTn : Port n TX RESPONSE Frame Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x349C + 0x48 × n

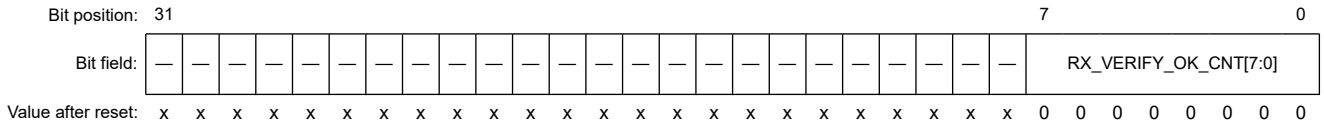


Bit	Symbol	Function	R/W
7:0	TX_RESPONSE_OK_CNT[7:0]	Port n increments for every RESPONSE frame transmitted.	R
31:8	—	The read values are undefined.	R

29.3.9.19 RX_VERIFY_OK_CNTn : Port n RX VERIFY Frame Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x34A0 + 0x48 × n

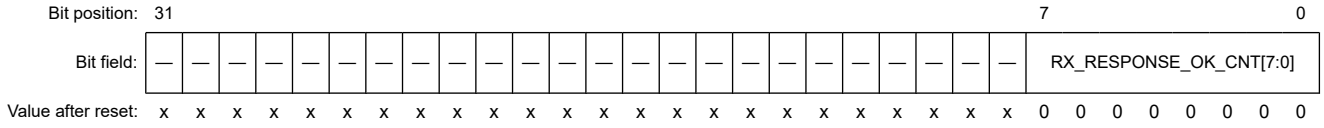


Bit	Symbol	Function	R/W
7:0	RX_VERIFY_OK_CNT[7:0]	Port n increments for every valid VERIFY frame received.	R
31:8	—	The read values are undefined.	R

29.3.9.20 RX_RESPONSE_OK_CNTn : Port n RX RESPONSE Frame Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x34A4 + 0x48 × n

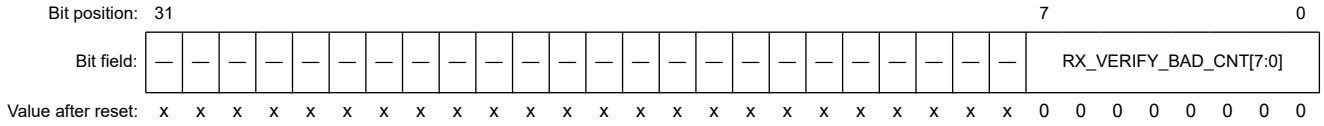


Bit	Symbol	Function	R/W
7:0	RX_RESPONSE_OK_CNT[7:0]	Port n increments for every valid RESPONSE frame received.	R
31:8	—	The read values are undefined.	R

29.3.9.21 RX_VERIFY_BAD_CNTn : Port n RX Error VERIFY Frame Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x34A8 + 0x48 × n

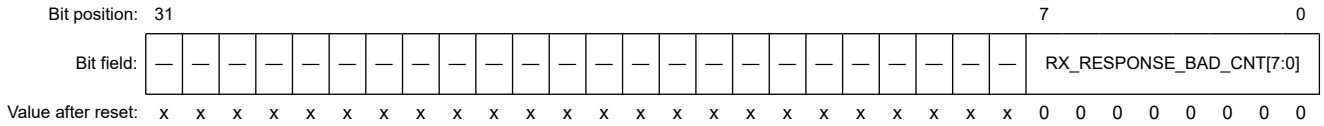


Bit	Symbol	Function	R/W
7:0	RX_VERIFY_BAD_CNT[7:0]	Port n increments for every errored VERIFY frame received.	R
31:8	—	The read values are undefined.	R

29.3.9.22 RX_RESPONSE_BAD_CNTn : Port n RX Error RESPONSE Frame Count Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x34AC + 0x48 × n



Bit	Symbol	Function	R/W
7:0	RX_RESPONSE_BAD_CNT[7:0]	Port n increments for every errored RESPONSE frame received.	R
31:8	—	The read values are undefined.	R

29.3.10 Memory Controller Extended Registers

29.3.10.1 MMCTL_OUT_CT : Cut-Through Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B00

Bit position:	31													18	16													2	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CT_OVR[2:0]		—	—	—	—	—	—	—	—	—	—	—	—	CT_OVR_ENA[2:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	CT_OVR_ENA[2:0]	Per-port bit mask to enable overriding the Cut-Through (CT) behavior of the output ports with CT_OVR. When set to 0, the frames are transmitted CT if the CT flag of the frame context is set.	R/W
15:3	—	The read values are undefined. The write value should be 0.	R/W
18:16	CT_OVR[2:0]	1 bit per-port value to set the Cut Through behavior of the output ports. When set to 0, all frames are sent as Store & Forward (SF) frames. When set to 1, frames with the CT flag set in the frame context are started as soon as the frame context information is available.	R/W
31:19	—	The read values are undefined. The write value should be 0.	R/W

The MMCTL_OUT_CT register allows overriding the output memory controller Cut-Through behavior. Only applicable to Ethernet ports.

29.3.10.2 MMCTL_CTFL_P0_3_ENA : Cut-Through Frame Length Enable Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B04

Bit position:	31								23							16	15							8	7							0
Bit field:	—	—	—	—	—	—	—	—	CTFL_P2_ENA[7:0]								CTFL_P1_ENA[7:0]								CTFL_P0_ENA[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	CTFL_P0_ENA[7:0]	Port 0 bit mask of n bits, where n is the number of queues per port indicating whether the CTFL is used for Cut-Through (CT) frames. When set to 1, a CT frame requires a CTFL entry to be written as a CT frame in the output memory.	R/W
15:8	CTFL_P1_ENA[7:0]	Port 1 bit mask of n bits, where n is the number of queues per port indicating whether the CTFL is used for Cut-Through (CT) frames. When set to 1, a CT frame requires a CTFL entry to be written as a CT frame in the output memory.	R/W
23:16	CTFL_P2_ENA[7:0]	Port 2 bit mask of n bits, where n is the number of queues per port indicating whether the CTFL is used for Cut-Through (CT) frames. When set to 1, a CT frame requires a CTFL entry to be written as a CT frame in the output memory.	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

Per-port/queue enable of the Cut-Through Frame Length table usage for ports 0 through 2. Only applicable to Ethernet ports.

29.3.10.3 MMCTL_YELLOW_BYTE_LENGTH_Pn : Port n Yellow Period Byte Length Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B20 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	YLEN_EN
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	YELLOW_LEN[13:0]														—	—
Value after reset:	0	0	0	0	0	1	0	1	1	1	1	0	1	1	x	x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	R/W
15:2	YELLOW_LEN[13:0]	Length in bytes of the YELLOW period for port n. Determines whether a frame can be transmitted before the YELLOW period expires. The value is programmed in increments of 4 bytes excluding the MAC overhead (IPG, Preamble and FCS if appended) of the frame. Default value assumes a YELLOW period duration of 125 µs, which allows for 1562 bytes. Excluding the MAC maximum overhead of IPG (12 bytes), CRC (4 bytes), Preamble + SFD (8 bytes), and a safety margin (20 bytes). When OUT_CT_ENA transitions from 1 to 0, an internal counter is loaded with YELLOW_LEN and decrements at the port transmission rate. The length of every frame is compared against the value of this counter to determine if it can be transmitted without exceeding the remaining bytes.	R/W
16	YLEN_EN	When set to 1, enables transmission when OUT_CT_ENA is low only if the frame length is less than YELLOW_LEN. If cleared, YELLOW_LEN is ignored and frames are always transmitted in SF mode when OUT_CT_ENA is 0.	R/W
31:17	—	The read values are undefined. The write value should be 0.	R/W

Port n length in bytes of the YELLOW period, one register per port. Used for Profinet IRT when the output cut-through is switched from enabled to disabled. Only relevant for Ethernet ports.

29.3.10.4 MMCTL_POOLn_CTR : Memory Pool Counter (n = 0 to 1)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B40 + 0x4 × n

Bit position:	31						25									16						9									0
Bit field:	—	—	—	—	—	—	USED[9:0]									—	—	—	—	—	—	CELLS[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	CELLS[9:0]	Memory pool configuration for pool n. Configures, in cells, the size of each memory pool. The sum of all cells assigned among all pools including the global pool must be at most MEM_BLKS − IPORTS, where IPORTS is the number of input ports including the management port (when 802.3br is enabled in a port, two blocks must be reserved for it instead of one). Reading back this register returns the current pool utilization. The pools must be configured when there is no traffic and the memory is empty.	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W
25:16	USED[9:0]	Reports the current available number of used cells for this memory pool. The used number of free cells can be calculated as CELLS − USED.	R

Bit	Symbol	Function	R/W
31:26	—	The read values are undefined. The write value should be 0.	R/W

Pool n configuration, one register per counter. Write to define the number of memory cells assigned to pool n.

29.3.10.5 MMCTL_POOL_GLOBAL : Memory Pool Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B60

Bit position:	31						25									16						9								0
Bit field:	—	—	—	—	—	—	USED[9:0]									—	—	—	—	—	—	CELLS[9:0]								
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	1	1	1	1	1	1	1	0	0	1

Bit	Symbol	Function	R/W
9:0	CELLS[9:0]	Memory pool configuration for the global pool. Configures, in cells, the size of the global shared pool. The sum of all blocks assigned to all pools including the global pool must be at most MEM_BLKS − IPORTS, where IPORTS is the number of input ports including the management port (when 802.3br is enabled in a port, two blocks must be reserved for it instead of one). On initialization, all memory cells (MEM_BLKS) are assigned to the global pool minus one cell per input port (IPORTS). The minimum valid value for this field is 3.	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W
25:16	USED[9:0]	Reports the current number of used cells for the global shared pool. The used number of free cells can be calculated as CELLS − USED.	R
31:26	—	The read values are undefined. The write value should be 0.	R/W

The MMCTL_POOL_GLOBAL register is used for global pool configuration. Write to define the number of memory cells assigned to the global pool.

29.3.10.6 MMCTL_POOL_STATUS : Memory Pool Status Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B64

Bit position:	31																							7						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QUEUE_FULL[7:0]						
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	QUEUE_FULL[7:0]	Per-queue pool full indication. Indicates for each queue whether all the blocks in the corresponding pool and global pool are allocated. The value is reset to 0 after it is read.	R
31:8	—	The read values are undefined.	R

The MMCTL_POOL_STATUS register indicates pool full status and whether frames were discarded due to the pool filling up.

29.3.10.7 MMCTL_POOL_QMAP : Queue MAP Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	Q7_ENA	—	—	Q7_MAP	Q6_ENA	—	—	Q6_MAP	Q5_ENA	—	—	Q5_MAP	Q4_ENA	—	—	Q4_MAP
Value after reset:	0	x	x	0	0	x	x	0	0	x	x	0	0	x	x	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	Q3_ENA	—	—	Q3_MAP	Q2_ENA	—	—	Q2_MAP	Q1_ENA	—	—	Q1_MAP	Q0_ENA	—	—	Q0_MAP
Value after reset:	0	x	x	0	0	x	x	0	0	x	x	0	0	x	x	0

Bit	Symbol	Function	R/W
0	Q0_MAP	Queue 0 Memory Pool Configures which memory pool is used for Queue 0 in all ports.	R/W
2:1	—	The read values are undefined. The write value should be 0.	R/W
3	Q0_ENA	Queue 0 Memory Pool Enabled 0: The queue only uses cells from the global pool 1: The queue uses the pool programmed in Q0_MAP	R/W
4	Q1_MAP	Queue 1 Memory Pool Configures which memory pool is used for Queue 1 in all ports.	R/W
6:5	—	The read values are undefined. The write value should be 0.	R/W
7	Q1_ENA	Queue 1 Memory Pool Enabled 0: The queue only uses cells from the global pool 1: The queue uses the pool programmed in Q1_MAP	R/W
8	Q2_MAP	Queue 2 Memory Pool Configures which memory pool is used for Queue 2 in all ports.	R/W
10:9	—	The read values are undefined. The write value should be 0.	R/W
11	Q2_ENA	Queue 2 Memory Pool Enabled 0: The queue only uses cells from the global pool 1: The queue uses the pool programmed in Q2_MAP	R/W
12	Q3_MAP	Queue 3 Memory Pool Configures which memory pool is used for Queue 3 in all ports.	R/W
14:13	—	The read values are undefined. The write value should be 0.	R/W
15	Q3_ENA	Queue 3 Memory Pool Enabled 0: The queue only uses cells from the global pool 1: The queue uses the pool programmed in Q3_MAP	R/W
16	Q4_MAP	Queue 4 Memory Pool Configures which memory pool is used for Queue 4 in all ports.	R/W
18:17	—	The read values are undefined. The write value should be 0.	R/W
19	Q4_ENA	Queue 4 Memory Pool Enabled 0: The queue only uses cells from the global pool 1: The queue uses the pool programmed in Q4_MAP	R/W
20	Q5_MAP	Queue 5 Memory Pool Configures which memory pool is used for Queue 5 in all ports.	R/W
22:21	—	The read values are undefined. The write value should be 0.	R/W
23	Q5_ENA	Queue 5 Memory Pool Enabled 0: The queue only uses cells from the global pool 1: The queue uses the pool programmed in Q5_MAP	R/W
24	Q6_MAP	Queue 6 Memory Pool Configures which memory pool is used for Queue 6 in all ports.	R/W

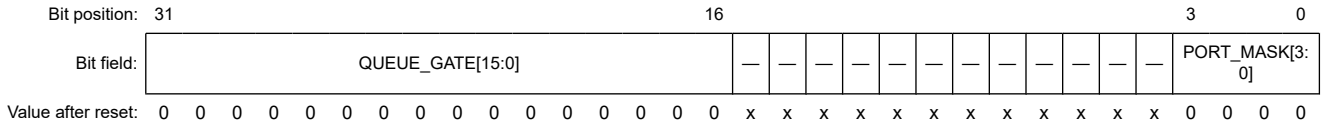
Bit	Symbol	Function	R/W
26:25	—	The read values are undefined. The write value should be 0.	R/W
27	Q6_ENA	Queue 6 Memory Pool Enabled 0: The queue only uses cells from the global pool 1: The queue uses the pool programmed in Q6_MAP	R/W
28	Q7_MAP	Queue 7 Memory Pool Configures which memory pool is used for Queue 7 in all ports.	R/W
30:29	—	The read values are undefined. The write value should be 0.	R/W
31	Q7_ENA	Queue 7 Memory Pool Enabled 0: The queue only uses cells from the global pool 1: The queue uses the pool programmed in Q7_MAP	R/W

The MMCTL_POOL_QMAP register maps each queue to a memory pool. By default, all queues are assigned to pool 0. All queues in the LSI use the same pool regardless of the port.

29.3.10.8 MMCTL_QGATE : Queue Gate State Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B6C



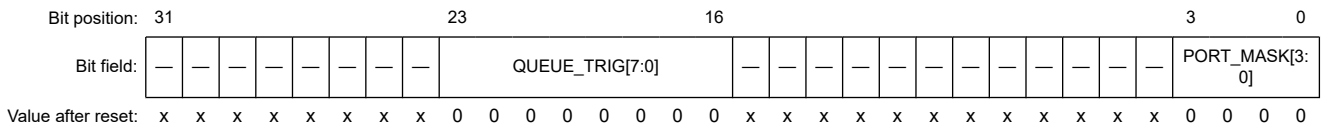
Bit	Symbol	Function	R/W
3:0	PORT_MASK[3:0]	Per-port bit mask. When set to 1 for a port, the queue gate state is changed for that port as indicated by QUEUE_GATE.	W
15:4	—	The write value should be 0.	W
31:16	QUEUE_GATE[15:0]	2-bit per queue indicating the action to be performed on each queue of the ports indicated by PORT_MASK. bits [1:0] = queue 0, bits [3:2] = queue 1, and so on. 0 0: Hold state, does not change the queue gate state 0 1: Toggle state, toggles the gate state 1 0: Gate queue, closes the queue 1 1: Open queue, enables the queue	W

The MMCTL_QGATE register is a write-only register and is used to change the gate state of queues. Writing to this register generates an action for the ports and queues indicating the changes to the gate state of the queues. Only relevant for Ethernet ports.

29.3.10.9 MMCTL_QTRIG : Queue Trigger Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B70



Bit	Symbol	Function	R/W
3:0	PORT_MASK[3:0]	Per-port bit mask. When set to 1 for a port, a frame is triggered from the closed queues indicated by QUEUE_TRIG.	W
15:4	—	The write value should be 0.	W

Bit	Symbol	Function	R/W
23:16	QUEUE_TRIG[7:0]	1-bit per queue indicating from which queues a frame is to be transmitted from the ports indicated by PORT_MASK. When set to 1, a single frame is transmitted per indicated port in PORT_MASK among the queues indicated by QUEUE_TRIG.	W
31:24	—	The write value should be 0.	W

The MMCTL_QTRIG register is a write-only register and is used to trigger sending a frame from a set of queues in selected ports. The action is performed on a register write. The highest priority queue is selected from the enabled queues and a single frame is sent. The queue must be in a closed state. Only relevant for Ethernet ports.

29.3.10.10 MMCTL_QFLUSH : Flush Event Select Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B74

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ACTION[1:0]		QUEUE_MASK[7:0]							
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PORT_MASK[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PORT_MASK[3:0]	Per-port bit mask. When set to 1 for a port, the queue flush status is changed for that port for the queues indicated in QUEUE_MASK.	W
15:4	—	The write value should be 0.	W
23:16	QUEUE_MASK[7:0]	1 bit per queue indicating for which queues of the ports indicated by PORT_MASK the flush state is changed as indicated in ACTION.	W
25:24	ACTION[1:0]	Selects the flush state for the queues indicated by QUEUE_MASK in the ports indicated by PORT_MASK. Possible actions are: 0 0: Disable flush 0 1: Flush when queue is closed, discarding any new frames 1 0: Flush on queue close until empty, then stops flushing 1 1: TRIGGER flush until empty, then return to current flush mode	W
31:26	—	The write value should be 0.	W

The MMCTL_QFLUSH register is a write-only register and is used to generate or stop flush events from selected ports and queues. The action is performed on a register write. Only relevant for Ethernet ports.

29.3.10.11 MMCTL_QCLOSED_STATUS_P0_3 : Queue Closed Status Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B78

Bit position:	31								23								16	15							8	7							0
Bit field:	—	—	—	—	—	—	—	—	P2_STATUS[7:0]									P1_STATUS[7:0]								P0_STATUS[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	P0_STATUS[7:0]	Per-queue closed status of Port 0 (1-bit per queue). A 0 indicates that the queue is open (enabled), and a 1 indicates that the queue is closed (disabled).	R

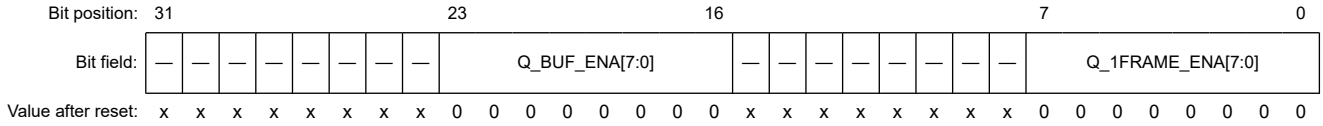
Bit	Symbol	Function	R/W
15:8	P1_STATUS[7:0]	Per-queue closed status of Port 1 (1-bit per queue). A 0 indicates that the queue is open (enabled), and a 1 indicates that the queue is closed (disabled).	R
23:16	P2_STATUS[7:0]	Per-queue closed status of Port 2 (1-bit per queue). A 0 indicates that the queue is open (enabled), and a 1 indicates that the queue is closed (disabled).	R
31:24	—	The read values are undefined.	R

The MMCTL_QCLOSED_STATUS_P0_3 register indicates the queue closed status. Reports whether each queue is open (enabled, 0) or closed (disabled, 1). Only relevant for Ethernet ports.

29.3.10.12 MMCTL_1FRAME_MODE_Pn : Port n 1-Frame Mode Configuration Register (n = 0 to 2)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3B80 + 0x4 × n



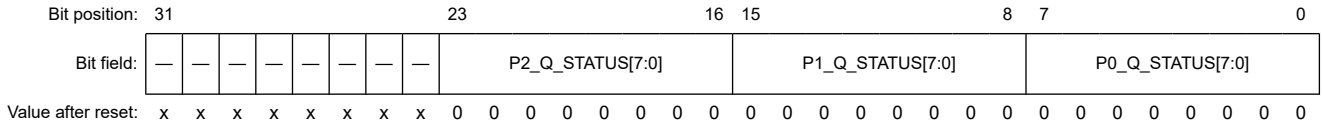
Bit	Symbol	Function	R/W
7:0	Q_1FRAME_ENA[7:0]	1 bit per queue. Setting a bit to 1 enables the 1-frame mode for that queue for port n. In this mode, only one frame is allowed in the queue. If a new frame is received, the old frame is discarded. This mode requires that the queue is in closed state, and frames are transmitted by a triggering action from the register, the pattern matchers, or the TDMA scheduler.	R/W
15:8	—	The read values are undefined. The write value should be 0.	R/W
23:16	Q_BUF_ENA[7:0]	1 bit per queue. Setting a bit to 1 enables the buffer mode behavior for that queue for port n. This mode requires also that Q_1FRAME_ENA is set to 1. When Q_BUF_ENA is set, a frame is always kept in the buffer after it is transmitted. If another frame is received in the queue, the old frame is discarded. This mode requires that the queue is in closed state, and frames are transmitted by a triggering action from the register, the pattern matchers, or the TDMA scheduler.	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

MMCTL_1FRAME_MODE_Pn register, one register per port. Per-queue control to enable 1-frame mode, where up to 1 frame is stored in the indicated queues, and whether the frame is discarded or kept after transmission. Only relevant for Ethernet ports.

29.3.10.13 MMCTL_P0_3_QUEUE_STATUS : Queue Status Indicator

Base address: ETHSW = 0x8012_0000

Offset address: 0x3BA0



Bit	Symbol	Function	R/W
7:0	P0_Q_STATUS[7:0]	Port 0 Per-Queue Bit Indication 0: Queue is empty 1: Queued frames	R

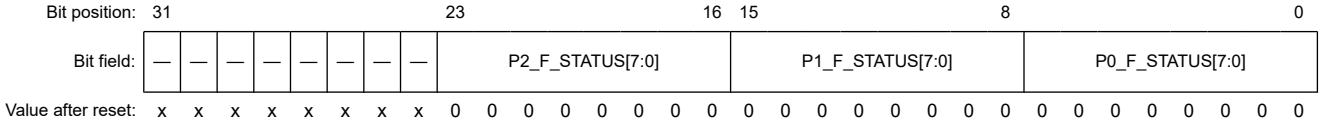
Bit	Symbol	Function	R/W
15:8	P1_Q_STATUS[7:0]	Port 1 Per-Queue Bit Indication 0: Queue is empty 1: Queued frames	R
23:16	P2_Q_STATUS[7:0]	Port 2 Per-Queue Bit Indication 0: Queue is empty 1: Queued frames	R
31:24	—	The read values are undefined.	R

The MMCTL_P0_3_QUEUE_STATUS register indicates per-port/queue status, whether the queue is empty (read 0) or not (read 1). Only relevant for Ethernet ports.

29.3.10.14 MMCTL_P0_3_FLUSH_STATUS : Queue Flush Status Indicator

Base address: ETHSW = 0x8012_0000

Offset address: 0x3BA8



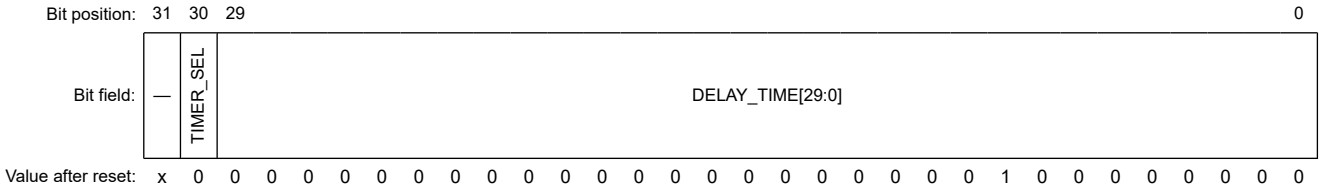
Bit	Symbol	Function	R/W
7:0	P0_F_STATUS[7:0]	Port 0 per-Queue Bit Indication on whether the queue is flushing frames (read 1) or not (read 0).	R
15:8	P1_F_STATUS[7:0]	Port 1 per-Queue Bit Indication on whether the queue is flushing frames (read 1) or not (read 0).	R
23:16	P2_F_STATUS[7:0]	Port 2 per-Queue Bit Indication on whether the queue is flushing frames (read 1) or not (read 0).	R
31:24	—	The read values are undefined.	R

The MMCTL_P0_3_FLUSH_STATUS register indicates per-port/queue status, whether the queue is flushing frames (read 1) or not (read 0). Only relevant for Ethernet ports.

29.3.10.15 MMCTL_DLY_QTRIGGER_CTRL : Delayed Queue Trigger Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3BB0



Bit	Symbol	Function	R/W
29:0	DELAY_TIME[29:0]	30-bit time in nanoseconds indicates the time after the trigger request from the pattern matchers to generate the event.	R/W
30	TIMER_SEL	Select the source timer to use for calculating the time. 0: Timer 0 1: Timer 1	R/W
31	—	The read value is undefined. The write value should be 0.	R/W

The MMCTL_DLY_QTRIGGER_CTRL register controls time for delayed queue triggering. Contains a timer select and a 30-bit time in nanoseconds for delayed queue triggering. When a pattern matcher generates a delay trigger event, this time is added to the current time in the selected timer to calculate the time for the trigger event to occur.

29.3.10.16 MMCTL_PREEMPT_QUEUES : Preemptable Queues Configures Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3BB4

Bit position:	31																15								8	7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PREEMPT_ON_QCLOSE[7:0]									PREEMPT_ENA[7:0]								
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	

Bit	Symbol	Function	R/W
7:0	PREEMPT_ENA[7:0]	Per-queue enable bit to configure which queues are used for preemptable traffic. Set to 1 the corresponding bit to configure a queue to be preemptable. The express queues always have priority over preemptable queues.	R/W
15:8	PREEMPT_ON_QCLOSE[7:0]	Per-queue configuration bit to enable preempting a frame when the queue goes from OPEN to CLOSED. When the corresponding bit is set to 1 and the queue is configured as preemptable in PREEMPT_ENA, a queue close event causes the current frame to be preempted, if preemption is operational.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

The MMCTL_PREEMPT_QUEUES register configures which queues are preemptable queues when preemption is enabled and operational. Only relevant for Ethernet ports with 802.3br support.

29.3.10.17 MMCTL_HOLD_CONTROL : Request Preemption Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3BB8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	Q_HOLD_REQ_RELEASE[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	Q_HOLD_REQ_FORCE[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	Q_HOLD_REQ_FORCE[2:0]	A per-port bit that forces a preempt request using MM_CTL.request (hold_req). When this bit is set to 1, it overrides other sources of hold request, including the TDMA controller.	R/W
15:3	—	The read values are undefined. The write value should be 0.	R/W
18:16	Q_HOLD_REQ_RELEASE[2:0]	A per-port bit that forces a release of preemption request using MM_CTL.request (hold_req). When this bit is set to 1, it overrides other sources of hold request, including the TDMA controller and Q_HOLD_REQ_FORCE[2:0].	R/W
31:19	—	The read values are undefined. The write value should be 0.	R/W

Allows for the firmware to request a preemption when preemption is enabled and operational (MM_CTL.request (hold_req)). Only relevant for Ethernet ports with 802.3br support.

29.3.10.18 MMCTL_PREEMPT_STATUS : Preemption State Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3BBC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	HOLD_REQ_STATE[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	PREEMPT_STATE[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	PREEMPT_STATE[2:0]	A per-port bit that indicates if a port is in a preempted state. This is a real-time indication meant for debugging.	R
15:3	—	The read values are undefined.	R
18:16	HOLD_REQ_STATE[2:0]	A per-port bit that indicates if a port is preempted using MM_CTL.request (hold_req). This is a real-time indication meant for debugging.	R
31:19	—	The read values are undefined.	R

Per-port preemption state. Indicates if a port is preempting a frame and the state of MM_CTL.request (hold_req). Only relevant for Ethernet ports with 802.3br support.

29.3.10.19 MMCTL_CQF_CTRL_Pn : Port n Cyclic Queuing and Forwarding Control Register (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3BC0 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	REF_SEL0	USE_SOP0	GATE_SEL0[2:0]			QUEUE_SEL0[2:0]			PRIO_ENABLE0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

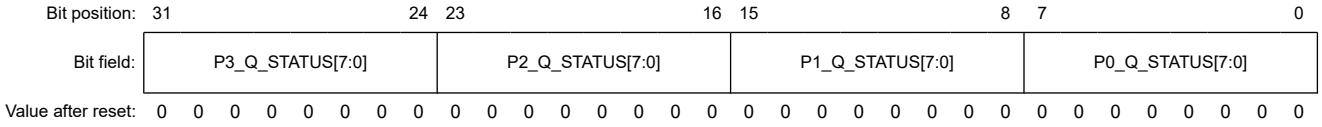
Bit	Symbol	Function	R/W
7:0	PRIO_ENABLE0[7:0]	A per-queue enable to select which ingress priorities are queued in the two CQF queues.	R/W
10:8	QUEUE_SEL0[2:0]	Select which two physical queues are used for CQF. The queues used are QUEUE_SEL0 and QUEUE_SEL0 + 1. Frames are written into QUEUE_SEL0 when the gate control selected with GATE_SEL0 is 0, and into QUEUE_SEL0 + 1 when the gate control is 1.	R/W
13:11	GATE_SEL0[2:0]	Select which gate control signal is used for selecting the output queue (these signals are the same as the ETHSW_TDMAOUT pins).	R/W
14	USE_SOP0	When set to 1, the CFQ queue is determined when the SOP is received at the frame writer in the memory controller. When set to 0, the queue is determined when the EOP is received at the frame writer.	R/W
15	REF_SEL0	Select whether the gate control signal used for the CQF group is based on the egress port when set to 0, or the ingress port when set to 1.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

Controls for configuring Cyclic Queuing and Forwarding (CQF) for port n.

29.3.10.20 MMCTL_P0_3_QCLOSED_NONEMPTY : Port Queue Status Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3BE0



Bit	Symbol	Function	R/W
7:0	P0_Q_STATUS[7:0]	Port 0 per-queue bit indication on whether the queue transitioned from open to closed state while frames were still queued.	R
15:8	P1_Q_STATUS[7:0]	Port 1 per-queue bit indication on whether the queue transitioned from open to closed state while frames were still queued.	R
23:16	P2_Q_STATUS[7:0]	Port 2 per-queue bit indication on whether the queue transitioned from open to closed state while frames were still queued.	R
31:24	P3_Q_STATUS[7:0]	Port 3 per-queue bit indication on whether the queue transitioned from open to closed state while frames were still queued.	R

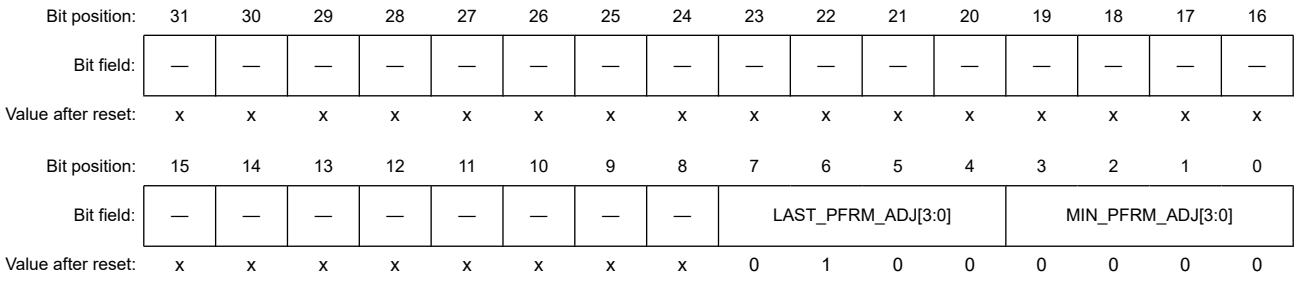
Per-port/queue status indications for ports 0 through 3. Latches when a queue is closed and there are frames still queued. Only relevant for Ethernet ports.

The value is reset to 0 after the bit is read.

29.3.10.21 MMCTL_PREEMPT_EXTRA : Frame Preemption Extra Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3BE8



Bit	Symbol	Function	R/W
3:0	MIN_PFRM_ADJ[3:0]	Adjust the minimum mPacket length, in increments of 4 bytes.	R/W
7:4	LAST_PFRM_ADJ[3:0]	Adjust the preemptable threshold when reaching the end of the frame, in increments of 4 bytes. Incrementing this value increments the length of the last mPacket.	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

Must not be changed.

29.3.11 DLR Module Registers

29.3.11.1 DLR_CONTROL : DLR Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	US_TIME[11:8]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	US_TIME[7:0]								—	—	—	IGNORE_INVTM	—	LOOP_FILTER_ENA	AUTOFLUSH	ENABLE
Value after reset:	0	1	1	0	0	1	0	0	x	x	x	0	x	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Enable DLR extension module. When set, the DLR module becomes active. When DLR is enabled, the LOOP_FILTER_ENA must also be enabled for proper DLR operation. When cleared, no DLR frame detection occurs and all DLR module statistics registers are cleared.	R/W
1	AUTOFLUSH	Enable automatic flushing of unicast entries in address table if ring reconfiguration occurs (see also DLR interrupt IRQ_flush_macaddr_ena in DLR_IRQ_CONTROL).	R/W
2	LOOP_FILTER_ENA	Enable the loop filter function. When set to 1, the ingress loop filter is enabled. This can be enabled regardless of the DLR ENABLE state, allowing the loop filter function to operate when DLR is not used.	R/W
3	—	The read value is undefined. The write value should be 0.	R/W
4	IGNORE_INVTM	Enable ignore beacon frames with invalid timeout timer. When enabled (set to 1) frames with timeout timer value not within a range of 200 microseconds to 500 milliseconds are ignored and parameters are not locally stored or considered for state transitions. The invalid timeout timer value is always stored within the DLR_INV_TMOUT register irrespective of the value of this bit. Ignored frames are forwarded normally.	R/W
7:5	—	The read values are undefined. The write value should be 0.	R/W
19:8	US_TIME[11:0]	Number of clock cycles required for 1 microsecond for the switch operating clock. This LSI operates at 200 MHz, therefore this register must be set to 0xC8. The value after reset must be changed.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The DLR_CONTROL register is used to select the DLR operation.

29.3.11.2 DLR_STATUS : DLR Status Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TOPOLOGY[7:0]								—	—	—	—	—	—	LINK_STATUS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NODE_STATE[7:0]								—	—	—	—	—	—	LastBcnRcvPort[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
1:0	LastBcnRcvPort[1:0]	Last Beacon Receive Port Bit 0: Set if beacon frame from active supervisor received on port 0 Bit 1: Set if beacon frame from active supervisor received on port 1	R
7:2	—	The read values are undefined.	R
15:8	NODE_STATE[7:0]	Local Node Current State 0x00: IDLE_STATE 0x01: NORMAL_STATE 0x02: FAULT_STATE Others: Reserved	R
17:16	LINK_STATUS[1:0]	Link Status This is a direct representation of the top level input pins (ETHSW_PHYLINK0, ETHSW_PHYLINK1). Bit 16: Set if port 0 link is up Bit 17: Set if port 1 link is up	R
23:18	—	The read values are undefined.	R
31:24	TOPOLOGY[7:0]	Current Network Topology 0x00: Linear topology when local node is in IDLE state 0x01: Ring topology when local node is not in IDLE state Others: Reserved	R

29.3.11.3 DLR_ETH_TYP : DLR Ethernet Type Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C08

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DLR_ETH_TYP[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1

Bit	Symbol	Function	R/W
15:0	DLR_ETH_TYP[15:0]	Ethernet type to compare for DLR frame detection (initial value is 0x80E1)	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.11.4 DLR_IRQ_CONTROL : DLR Interrupt Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	atomic_AND	atomic_OR	low_int_en	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IRQ_frm_dscrd1	IRQ_frm_dscrd0	IRQ_bec_rcv1_ena	IRQ_bec_rcv0_ena	IRQ_invalid_tmr_ena	IRQ_ip_addr_chng_ena	IRQ_sup_ignord_ena	IRQ_link_chng1_ena	IRQ_link_chng0_ena	IRQ_supr_chng_ena	IRQ_bec_tmr1_exp_ena	IRQ_bec_tmr0_exp_ena	IRQ_stop_nbchk1_ena	IRQ_stop_nbchk0_ena	IRQ_flush_macaddr_ena	IRQ_state_chng_ena
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IRQ_state_chng_ena	Enable Interrupt for State Change When enabled, an interrupt is generated when state change occurs for the local beacon based DLR ring node. The interrupt service routine must re-read the beacon frame parameters and clear the bit afterwards.	R/W
1	IRQ_flush_macaddr_ena	Enable Flush Local MAC Address Table Interrupt. When enabled, an interrupt is generated when unicast MAC address learning table should be flushed.	R/W
2	IRQ_stop_nbchk0_ena	Enable Stop Request Neighbor Check Timeout Timer Interrupt for Port 0.	R/W
3	IRQ_stop_nbchk1_ena	Enable Stop Request Neighbor Check Timeout Timer Interrupt for Port 1.	R/W
4	IRQ_bec_tmr0_exp_ena	Enable Interrupt on Beacon Timeout Timer Expire for Port 0.	R/W
5	IRQ_bec_tmr1_exp_ena	Enable Interrupt on Beacon Timeout Timer Expire for Port 1.	R/W
6	IRQ_supr_chng_ena	Enable Interrupt on Ring Supervisor Change.	R/W
7	IRQ_link_chng0_ena	Enable Link Status Change Interrupt Event for Port 0.	R/W
8	IRQ_link_chng1_ena	Enable Link Status Change Interrupt Event for Port 1.	R/W
9	IRQ_sup_ignord_ena	Enable interrupt on beacon frame detection from a supervisor with lower precedence than the current ring supervisor or lower numeric value for MAC address when precedence is same.	R/W
10	IRQ_ip_addr_chng_ena	Enable interrupt on IP address change detection within beacon frame from ring supervisor.	R/W
11	IRQ_invalid_tmr_ena	Enable interrupt on invalid range for beacon timeout timer value detection.	R/W
12	IRQ_bec_rcv0_ena	Enable interrupt on beacon frame detection on port 0.	R/W
13	IRQ_bec_rcv1_ena	Enable interrupt on beacon frame detection on port 1.	R/W
14	IRQ_frm_dscrd0	Enable interrupt on frame discard due to source address match with the local address on port 0.	R/W
15	IRQ_frm_dscrd1	Enable Interrupt on Frame discard due to source address match with the local address on port 1.	R/W
28:16	—	The read values are undefined. The write value should be 0.	R/W
29	low_int_en	Enable active-low interrupt. Asserted to use active-low interrupt signal instead of active-high interrupt signal.	R/W
30	atomic_OR	When set during a register-write, the enable bits are ORed into the current setting of the register. By writing this bit at the same time, only the target bit can be set to 1.	W
31	atomic_AND	When set during a register-write, the enable bits are ANDed with the current setting of the register. By writing this bit at the same time, only the target bit can be set to 0.	W

If any of the interrupt enable bit is set and a corresponding event occurs, the DLR interrupt (DLR_INT bit) within the global INT_STAT_ACK interrupt status register asserts. In addition, ETHSW_DLR interrupt asserts. To clear any of the interrupts, the DLR_IRQ_STAT_ACK register must be written.

29.3.11.5 DLR_IRQ_STAT_ACK : DLR Interrupt Status/ACK Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	frm_dscrd1_IRQ_pending	frm_dscrd0_IRQ_pending	bec_rcv1_IRQ_pending	bec_rcv0_IRQ_pending	invalid_tmr_IRQ_pending	ip_chng_IRQ_pending	sup_ignord_IRQ_pending	Link1_IRQ_pending	Link0_IRQ_pending	suprchng_IRQ_pending	bec_tmr1_IRQ_pending	bec_tmr0_IRQ_pending	nbchk1_IRQ_pending	nbchk0_IRQ_pending	flush_IRQ_pending	state_chng_IRQ_pending
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	state_chng_IRQ_pending	Latched State Change Event Asserts whenever local beacon based DLR node state changes. To clear the latch, the bit must be written with 1.	R/W
1	flush_IRQ_pending	Latched Flush Event for MAC Address Learning Table When asserted, MAC address learning table should be flushed. To clear the latch, the bit must be written with 1.	R/W
2	nbchk0_IRQ_pending	Stop Request Event for Neighbor Check Timeout Timer for Port 0 When asserted, neighbor check timeout timer should be stopped. To clear the latch, the bit must be written with 1.	R/W
3	nbchk1_IRQ_pending	Stop Request Event for Neighbor Check Timeout Timer for Port 1 When asserted, neighbor check timeout timer should be stopped. To clear the latch, the bit must be written with 1.	R/W
4	bec_tmr0_IRQ_pending	Beacon Timeout Timer Expire Event for Port 0 Asserted when beacon timeout timer expired for port 0. To clear the latch, the bit must be written with 1.	R/W
5	bec_tmr1_IRQ_pending	Beacon Timeout Timer Expire Event for Port 1 Asserted when beacon timeout timer expired for port 1. To clear the latch, the bit must be written with 1.	R/W
6	supr_chng_IRQ_pending	Latched Supervisor Change Event Asserted when ring supervisor changes for DLR ring. To clear the latch, the bit must be written with 1.	R/W
7	Link0_IRQ_pending	Latched Link Status Change Event for Port 0 Asserts whenever link status change is detected on Port 0. To clear the latch, the bit must be written with 1.	R/W
8	Link1_IRQ_pending	Latched Link Status Change Event for Port 1 Asserts whenever link status change is detected on Port 1. To clear the latch, the bit must be written with 1.	R/W
9	sup_ignord_IRQ_pending	Latched Event for Beacon Frame Detection from Ignored Supervisor Asserted when beacon frame detected from a supervisor with lower precedence or lower numeric value for the MAC address when precedence is the same. To clear the latch, the bit must be written with 1.	R/W
10	ip_chng_IRQ_pending	Latched IP Address Change Event Asserted when IP address change detection within beacon frame from ring supervisor. To clear the latch, the bit must be written with 1.	R/W
11	invalid_tmr_IRQ_pending	Latched Event on Invalid Beacon Timeout Timer Value Detection Within Beacon Frame on Port 0 or Port 1 To clear the latch, the bit must be written with 1.	R/W
12	bec_rcv0_IRQ_pending	Latched Event on Beacon Frame Detection on Port 0 To clear the latch, the bit must be written with 1.	R/W

Bit	Symbol	Function	R/W
13	bec_rcv1_IRQ_pending	Latched Event on Beacon Frame Detection on Port 1 To clear the latch, the bit must be written with 1.	R/W
14	frm_dscrd0_IRQ_pending	Latched Event on Frame Discard Due to Source Address Match with the Local Address on Port 0 (Loop Filter) To clear the latch, the bit must be written with 1.	R/W
15	frm_dscrd1_IRQ_pending	Latched Event on Frame Discard Due to Source Address Match with the Local Address on Port 1 (Loop Filter) To clear the latch, the bit must be written with 1.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.11.6 DLR_LOC_MAClo : DLR Local MAC Address Low Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C14

Bit position:	31																												0
Bit field:	LOC_MAC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	LOC_MAC[31:0]	First 4 octets of the Local MAC address for loop filter Bits [7:0] is first byte, …, bits [31:24] is 4th byte of address.	R/W

29.3.11.7 DLR_LOC_MAChi : DLR Local MAC Address High Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C18

Bit position:	31																15												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LOC_MAC[15:0]												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	LOC_MAC[15:0]	Last 2 octets of local MAC address for loop filter Bits [7:0] is 5th byte of the local MAC address. Bits [15:8] is 6th (last) byte of the local MAC address.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.11.8 DLR_SUPR_MAClo : DLR Supervisor MAC Address Low Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C20

Bit position:	31																												0
Bit field:	SUPR_MAC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SUPR_MAC[31:0]	First 4 octets of the active ring supervisor of the MAC address extracted from the Source Address field of the beacon frame. Bits [7:0] is first byte, …, bits [31:24] is 4th byte of address.	R

Type1 parameter (see (6) Beacon Frame Parameter Extraction.)

29.3.11.9 DLR_SUPR_MAChi : DLR Supervisor MAC Address High Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C24

Bit position:	31								23							16	15													0
Bit field:	—	—	—	—	—	—	—	—	PRECE[7:0]								SUPR_MAC[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SUPR_MAC[15:0]	Last 2 octets of the active ring supervisor of the MAC address extracted from the Source Address field of the beacon frame. Bits [7:0] is 5th byte, bits [15:8] is 6th byte.	R
23:16	PRECE[7:0]	Precedence value of the ring supervisor extracted from the Supervisor precedence field of the beacon frame.	R
31:24	—	The read values are undefined.	R

Type1 parameter (see (6) Beacon Frame Parameter Extraction.)

29.3.11.10 DLR_STATE_VLAN : DLR Ring Status/VLAN Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	VLANINFO[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	VLAN VALID	RINGSTAT[7:0]							
Value after reset:	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RINGSTAT[7:0]	DLR ring state extracted from the Ring State field of the beacon frame. 0x01: RING_NORMAL_STATE 0x02: RING_FAULT_STATE Others: Unused	R
8	VLANVALID	VLAN Valid When asserted, the current VLAN control field bits [31:16] contain valid VLAN ID.	R
15:9	—	The read values are undefined.	R
31:16	VLANINFO[15:0]	IEEE 802.1Q VLAN Tag control field extracted from the VLAN info field of the beacon frame.	R

Type1 parameter (see (6) Beacon Frame Parameter Extraction.)

29.3.11.11 DLR_BEC_TMOUT : DLR Beacon Timeout Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C2C

Bit position:	31																												0
Bit field:	BEC_TMOUT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	BEC_TMOUT[31:0]	Beacon timeout timer value extracted from the Beacon Timeout in microseconds field of the beacon frame. Invalid timeout timer values are not written to this register and are not considered for state transition when ignore enabled with bit 4 of the DLR_CONTROL register. Expected value is in the range between minimum 200 microseconds and maximum 500 milliseconds. Typical value is 1960 microseconds.	R

Type1 parameter (see (6) Beacon Frame Parameter Extraction.)

29.3.11.12 DLR_BEC_INTRVL : DLR Beacon Interval Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C30

Bit position:	31																												0
Bit field:	BEC_INTRVL[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	BEC_INTRVL[31:0]	Beacon interval extracted from the Beacon Interval field of the beacon frame Expected value is in a range between minimum 100 microseconds and maximum 100 milliseconds. Typical value is 400 microseconds.	R

Type1 parameter (see (6) Beacon Frame Parameter Extraction.)

29.3.11.13 DLR_SURR_IPADR : DLR Supervisor IP Address Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C34

Bit position:	31																												0
Bit field:	SUPR_IPADR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SUPR_IPADR[31:0]	IP address of the ring supervisor extracted from the Source IP address field of the beacon frame. A value of 0x0 can be received when supervisor has no IP address.	R

Type2 parameter (see (6) Beacon Frame Parameter Extraction.)

29.3.11.14 DLR_ETH_STYP_VER : DLR Sub Type/Protocol Version Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C38

Bit position:	31								23								16	15							8	7							0
Bit field:	—	—	—	—	—	—	—	—	SPORT[7:0]									PROTVER[7:0]								SUBTYPE[7:0]								
Value after reset:	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SUBTYPE[7:0]	DLR Ring Ether Sub Type extracted from the Ring Sub Type field of the beacon frame. Value should be 0x02.	R
15:8	PROTVER[7:0]	DLR Ring Protocol Version extracted from the Ring Protocol Version field of the beacon frame.	R
23:16	SPORT[7:0]	Source port extracted from the Source Port field of the beacon frame.	R
31:24	—	The read values are undefined.	R

Type1 parameter (see (6) Beacon Frame Parameter Extraction.)

29.3.11.15 DLR_INV_TMOUT : DLR Beacon Timeout Timer Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C3C

Bit position:	31																															0
Bit field:	INV_TMOUT[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	INV_TMOUT[31:0]	Last out of range Beacon timeout timer value extracted from beacon frame on any of the port. Valid range is between 200 microseconds and 500 milliseconds. This register is always written whenever a new invalid value is received. Contains a valid value when bit 11 of the DLR_IRQ_STAT_ACK register is asserted.	R

Type3 parameter (see (6) Beacon Frame Parameter Extraction.)

29.3.11.16 DLR_SEQ_ID : DLR Sequence ID Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C40

Bit position:	31																															0
Bit field:	SEQ_ID[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SEQ_ID[31:0]	Sequence ID of the last beacon frame extracted from the Sequence ID field of the beacon frame on port 0 or port 1. Sequence ID of the ignored frames is not stored.	R

Type2 parameter (see (6) Beacon Frame Parameter Extraction.)

29.3.11.17 DLR_DSTlo : DLR Beacon Destination Address Low Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C58

Bit position:	31																												0
Bit field:	DLR_DST[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
31:0	DLR_DST[31:0]	First 4 octets of the beacon frame destination multicast address (01-21-6C-00-00-01). Bits [7:0] is first byte, …, bits [31:24] is 4th byte of address.	R/W

29.3.11.18 DLR_DSThi : DLR Beacon Destination Address High Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C5C

Bit position:	31															15													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DLR_DST[15:0]													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DLR_DST[15:0]	Last 2 octets of the beacon frame destination multicast address (01-21-6C-00-00-01). Bits [7:0] is 5th byte, bits [15:8] is 6th byte of address.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.11.19 DLR_RX_STAT0 : DLR Received Frame Statistic Register 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C60

Bit position:	31																												0
Bit field:	RX_STAT0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RX_STAT0[31:0]	Number of Beacon Frames Received on Port 0 Counter increments with the beacon frames which matches the destination MAC address, Ether type, DLR frame type, and CRC. Ignored frames due to type mismatch or CRC error are not counted. The counters are cleared if the DLR module is disabled.	R

29.3.11.20 DLR_RX_ERR_STAT0 : DLR Received Frame Error Statistic Register 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C64

Bit position:	31																												0
Bit field:	RX_ERR_STAT0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RX_ERR_STAT0[31:0]	Number of Beacon Frames Received with CRC Error on Port 0 Counter increments with the beacon frames which match the destination MAC address, Ether type, DLR frame type but with CRC error. The counters are cleared if the DLR module is disabled.	R

29.3.11.21 DLR_RX_LF_STAT0 : DLR Received Frame Loop Filter Statistic Register 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C6C

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RX_LF_STAT0[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RX_LF_STAT0[7:0]	Number of discarded frames in port 0 due to loop filtering when LOOP_FILTER_ENA is set to 1. Saturates at 255. Clear when written. Also cleared when LOOP_FILTER_ENA is set to 0.	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.11.22 DLR_RX_STAT1 : DLR Received Frame Statistic Register 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C70

Bit position:	31																													0
Bit field:	RX_STAT1[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RX_STAT1[31:0]	Number of Beacon Frames Received on Port 1 Counter increments with the beacon frames which match the destination MAC address, Ether type, DLR frame type, and CRC. Ignored frames due to type mismatch or CRC error are not counted. The counters are cleared if the DLR module is disabled.	R

29.3.11.23 DLR_RX_ERR_STAT1 : DLR Received Frame Error Statistic Register 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C74

Bit position:	31																													0
Bit field:	RX_ERR_STAT1[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RX_ERR_STAT1[31:0]	Number of Beacon Frames Received with CRC Error on Port 1 Counter increments with the beacon frames which match the destination MAC address, Ether type, DLR frame type but with CRC error. The counters are cleared if the DLR module is disabled.	R

29.3.11.24 DLR_RX_LF_STAT1 : DLR Received Frame Loop Filter Statistic Register 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x3C7C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RX_LF_STAT1[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RX_LF_STAT1[7:0]	Number of discarded frames in port 1 due to loop filtering when LOOP_FILTER_ENA is set to 1. Saturates at 255. Clear when written. Also cleared when LOOP_FILTER_ENA is set to 0.	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.12 PRP Module Registers

29.3.12.1 PRP_CONFIG : PRP Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRP_AGE_ENA
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RCT_LEN_CHK_DIS	TX_RCT_1588	TX_RCT_UNKNOWN	TX_RCT_MULTICAST	TX_RCT_BROADCAST	TX_RCT_MODE[1:0]		RX_REMOVE_RCT	RX_DUP_ACCEPT	PRP_ENA
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRP_ENA	Enable PRP Operation All other configuration registers must have been configured correctly before setting this bit. When cleared, the statistics counters are cleared also. Before re-enabling, the history memory should be cleared by executing the RM_ADDR_CTRL.CLEAR_MEMORY command.	R/W
1	RX_DUP_ACCEPT	Enable Duplicate Accept Mode of Operation at Receive When this bit is 1, the receivers do not perform duplicate detection and forward all received frames unmodified. When this bit is 0, the receivers try to detect and discard duplicates. Set this bit to 0 for normal operation. Even when this bit is 0, BPDU and IEEE 1588 frames are always accepted without a duplicate check. Using receive duplicate accept can be used to perform duplicate detection at the application and not having the hardware to perform it. Aside from this use, duplicate accept mode is a test setting (IEC: 4.2.6) not intended to be used during normal operation.	R/W

Bit	Symbol	Function	R/W
2	RX_REMOVE_RCT	Allow PRP Port RX to Remove the RCT When this bit is 1, the RCT is removed from frames received at the PRP ports before forwarding. This has no effect on other ports. When this bit is 0, the RCT is not removed. Duplicate detection occurs normally, independent from this setting. BPDU and IEEE 1588 Layer 2 frames are not affected by this setting and are received unmodified always. The RCT is removed only if the source node is known to be a Double attached nodes (DAN) as indicated by having a port mask containing both ports within the address table and the LAN ID matches the port where it is received.	R/W
4:3	TX_RCT_MODE[1:0]	Control Appending the RCT to Transmitted Frames on the Redundant Ports Normally the setting should be 00b or 01b. This allows Single attached nodes (SAN) and Double attached nodes (DAN) within the PRP LAN segments ensuring the RCT is only appended to frames sent to DANs. BPDU frames do not get an RCT appended (IEC: 4.2.7.5.1), independent from this setting. Frames forwarded to a port that is not within the PRP_GROUP do not get an RCT appended (even if at the same time the frame is duplicated to the PRP_GROUP where it can get the RCT appended). The following modes are available. 0 0: Append RCT if group only: Append RCT, if frame is duplicated to exactly the ports listed in the PRP_GROUP register. If the frame is sent to only one of the ports or is for example, flooded also to other ports, no RCT is appended. 0 1: Append RCT to group always: Append the RCT always if the frame is duplicated to at least the two ports listed in the PRP_GROUP. This includes flooded frames even if they are additionally sent on other ports outside the PRP_GROUP. If the frame is sent to only one of the PRP_GROUP ports, no RCT is appended. 1 0: Disable RCT append: The transmitters do not append the RCT even if frames are duplicated to the PRP_GROUP ports. This represents the IEC: 4.2.6 duplicate accept mode of operation. 1 1: Append RCT always forced: The transmitters append the RCT to all frames transmitted on any of the PRP_GROUP ports, no matter if they are duplicated or not or if they are flooded.	R/W
5	TX_RCT_BROADCAST	Should be 1 normally. 0: Do not append RCT to broadcast frames. This further restricts modes 00b, 01b, 11b. 1: Allow appending RCT to broadcast frames.	R/W
6	TX_RCT_MULTICAST	Should be 1 normally. 0: Do not append RCT to multicast frames. This further restricts modes 00b, 01b, 11b. 1: Allow appending RCT to multicast frames.	R/W
7	TX_RCT_UNKNOWN	Should be 1 normally. 0: Do not append RCT to frames flooded because the destination is unknown. This further restricts modes 00b, 01b, 11b. 1: Allow appending RCT to frames that have unknown destination and are hence flooded.	R/W
8	TX_RCT_1588	Setting this bit affects IEEE 1588 frames that are forwarded through the switch (for example, when used as RedBox) to both PRP_GROUP ports. Locally generated IEEE 1588 frames (peer-delay request/response) are not affected by this setting. Normally, this bit should be set to 0 as IEEE 1588 frames should not be considered for duplicate detection (IEC 62439-3: A.4.4). 0: Do not append RCT to IEEE 1588 frames (frame type 0x88F7) frames even if they are duplicated. 1: Allow appending RCT to IEEE 1588 frames. Defines to treat those frames as any other frame according to the configured TX_RCT_xxx settings above.	R/W
9	RCT_LEN_CHK_DIS	When set to 1, disables the RCT length field checking against the actual frame length.	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W
16	PRP_AGE_ENA	Enable History Memory Aging Timer The timer value is configured with the PRP_AGETIME register.	R/W
31:17	—	The read values are undefined. The write value should be 0.	R/W

29.3.12.2 PRP_GROUP : PRP Port Group Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	LANB_MASK[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	PRP_GROUP[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	PRP_GROUP[2:0]	Defines which two ports should be treated as redundant ports for PRP. Exactly 2 bits must be set in the mask. Only line ports are supported. Bit 0 = Port 0, Bit 1 = Port 1, Bit 2 = Port 2 Setting that is not exactly 2 bits leads to unpredictable behavior.	R/W
15:3	—	The read values are undefined. The write value should be 0.	R/W
18:16	LANB_MASK[2:0]	Defines which of the ports is considered the LAN B port. This port has the RCT containing the LAN B identifier. The other port uses LAN A identifier for the RCT. Exactly 1 bit must be set to match one of the 2 bits in the PRP_GROUP mask. Bit 16 = Port 0, Bit 17 = Port 1, Bit 18 = Port 2 The setting can be changed at any time during operation. For example, if management detects the ports were connected to the wrong networks, it is sufficient to swap the declaration of LAN A/B in this register and there is no need to physically change connections.	R/W
31:19	—	The read values are undefined. The write value should be 0.	R/W

The PRP_GROUP register defines two ports that must be used for PRP and their LAN ID association.

29.3.12.3 PRP_SUFFIX : PRP RCT Suffix

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D08

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRP_SUFFIX[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	1	0	0	0	1	1	1	1	1	0	1	1

Bit	Symbol	Function	R/W
15:0	PRP_SUFFIX[15:0]	The Redundancy Control Trailer (RCT) suffix (initial value is 0x88FB)	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.12.4 PRP_LANID : PRP LAN Identifier

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D0C

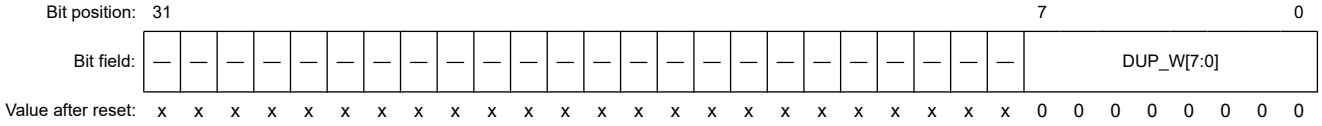
Bit position:	31																							7		4	3		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LANBID[3:0]			LANAID[3:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	1	1	1	0	1	0

Bit	Symbol	Function	R/W
3:0	LANAID[3:0]	LAN A Identifier	R/W
7:4	LANBID[3:0]	LAN B Identifier	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.12.5 DUP_W : PRP Max Duplicate Detection Window Size

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D10

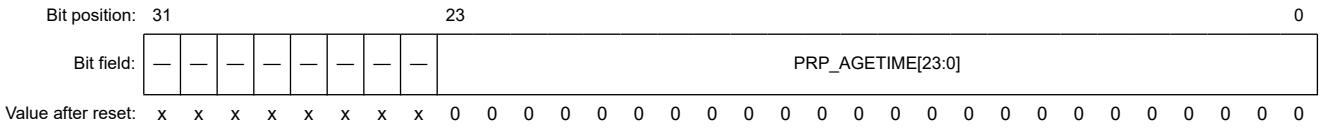


Bit	Symbol	Function	R/W
7:0	DUP_W[7:0]	Maximum Duplicate Detect Window Size If the current sequence number of the frame is in the range of expected, it is treated as duplicate (it is an old sequence number). If a sequence number with a value beyond this window is found, the frame is accepted and an increment of CntOutOfSeqLow occurs. The valid value is 2 to 255. Recommended value is > 50. Setting value of 0 or 1 causes all frames with a sequence number less than the minimum received to be accepted with an out-of-sequence error indication.	R/W
31:8	—	The read values are undefined. The write value should be 0.	R/W

29.3.12.6 PRP_AGETIME : PRP Aging Time Define Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D14



Bit	Symbol	Function	R/W
23:0	PRP_AGETIME[23:0]	Timeout in steps of 32 switch operating clock cycles to control aging of duplicate history data. Every age-time one entry of the history table is inspected for invalidating. Because only a single age bit is used, this requires at maximum two table rounds until an entry is removed. The timeout must be chosen to guarantee the full table is covered in EntryForgetTime / 2 (200 ms). A 24-bit value is available. A value of 0 disables the aging process. For an example setting of 200 ms with a 4096-entries table and the switch clock running at 200 MHz (5 ns): Age-time = 200 ms / (4096 × 5 ns) / 32 = 305.	R/W
31:24	—	The read values are undefined. The write value should be 0.	R/W

29.3.12.7 PRP_IRQ_CONTROL : PRP Interrupt Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SEQMISSING	OUTOFSEQ	WRONGLAN	MEMTOOLATE
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
0	MEMTOOLATE	Enable Interrupt for Memory Error Indications. When enabled, an interrupt is generated when the memory transactions for a frame cannot be completed in time. For transmit, it means a frame was sent without appending an RCT. For receive, it means an RCT cannot be checked and the frame was accepted.	R/W
1	WRONGLAN	Enable interrupt for frames received at a redundant port with an invalid LAN identifier in its redundancy trailer. This event also caused the incrementing of the CntErrWrongLan statistics.	R/W
2	OUTOFSEQ	Enable interrupt for frames received and accepted but have an unexpected sequence number. This event also caused the incrementing of the CntOutOfSeq statistics.	R/W
3	SEQMISSING	Enable interrupt for frames received and accepted that caused the history to skip a sequence number that was never received (for example, a missing sequence number is being ignored and is now treated as a candidate for dropping). This event also caused the incrementing of the CntMissing statistics.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

If any of the interrupt enable bits is set and a corresponding event occurs, the PRP interrupt (PRP_INT bit) within the global INT_STAT_ACK interrupt status register and the ETHSW_PRP interrupt assert. To clear any of the interrupts, write to the PRP_IRQ_STAT_ACK register.

29.3.12.8 PRP_IRQ_STAT_ACK : PRP Interrupt Status/ACK Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SEQMISSING	OUTOFSEQ	WRONGLAN	MEMTOOLATE
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
0	MEMTOOLATE	Interrupt Pending Indication Bits corresponding to the bits defined in PRP_IRQ_CONTROL. If an event occurs, the corresponding bit is latched high. To clear the bit, write 1 to it. The latches operate independently of the PRP_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W

Bit	Symbol	Function	R/W
1	WRONGLAN	This bit functions the same as MEMTOOLATE bit.	R/W
2	OUTOFSEQ	This bit functions the same as MEMTOOLATE bit.	R/W
3	SEQMISSING	This bit functions the same as MEMTOOLATE bit.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

The PRP_IRQ_STAT_ACK register is used to clear any of the interrupts.

29.3.12.9 RM_ADDR_CTRL : PRP History Memory Transactions Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BUSY	—	CLEAR	—	—	READ	WRITE	—	CLEAR_MEMORY	CLEAR_DYNAMIC	—	—	—	—	—	—
Value after reset:	0	x	0	x	x	0	0	x	0	0	x	x	x	x	x	x

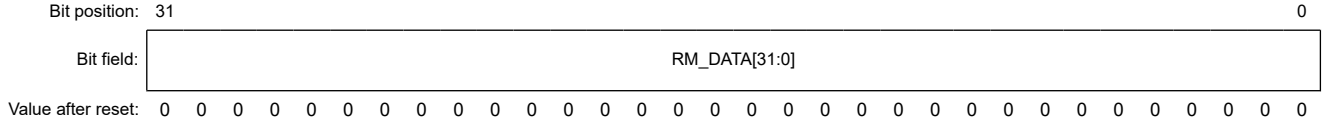
Bit position:	15	14	13	12	11											0
Bit field:	—	—	—	—	address[11:0]											
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	address[11:0]	Memory Address for Read and Write Transactions This is the address of an entry. The address is the same as the address of an entry in the MAC address lookup table. Therefore, to find data for a specific MAC address, the MAC address table must first be searched (lookup) to retrieve the address of an entry.	R/W
21:12	—	The read values are undefined. The write value should be 0.	R/W
22	CLEAR_DYNAMIC	When set to 1, scan the complete table for valid dynamic history entries and deletes them (writes entry with all 0s). The stored sequence numbers (for transmit RCT generation) of source addresses associated with non-redundant local ports are not modified. This bit is cleared when the function has completed.	R/W
23	CLEAR_MEMORY	When set to 1, write all memory locations with 0. This bit is cleared when the function has completed. The bit is set on reset to clear the memory and deasserts when completed.	R/W
24	—	The read value is undefined. The write value should be 0.	R/W
25	WRITE	When set to 1, perform a Single Write Transaction. The RM_DATA/RM_DATA_HI registers must be set prior to starting the transaction. This bit is cleared when the function has completed.	R/W
26	READ	When set to 1, perform Single Read Transaction. Data is returned in RM_DATA/RM_DATA_HI registers. This bit is cleared when the function has completed.	R/W
28:27	—	The read values are undefined. The write value should be 0.	R/W
29	CLEAR	When set to 1, write all 0s to the entry selected by the given address. Avoids the need to set the data registers to a 0 value. This bit is cleared when the function has completed.	R/W
30	—	The read value is undefined. The write value should be 0.	R/W
31	BUSY	Transaction Busy Indication Controller is busy when this bit is 1. As long as the controller is busy, the corresponding command bit is set. When it becomes non-busy again, all command bits are cleared. No other commands are allowed when BUSY = 1. A write to this register must be avoided as long as it is busy.	R

29.3.12.10 RM_DATA : PRP Memory Data Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D24

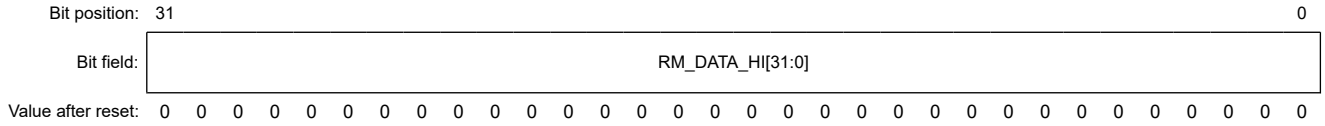


Bit	Symbol	Function	R/W
31:0	RM_DATA[31:0]	Memory data register for read/write transactions controlled by RM_ADDR_CTRL. It contains bits [31:0] of a memory entry.	R/W

29.3.12.11 RM_DATA_HI : PRP Memory Data Higher Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D28

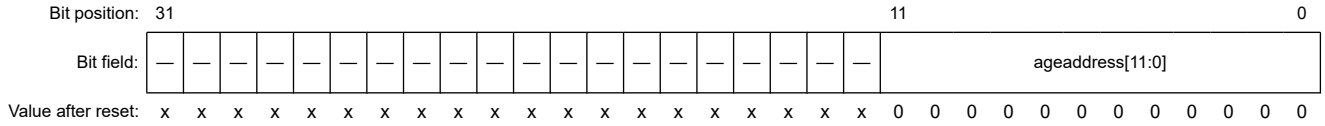


Bit	Symbol	Function	R/W
31:0	RM_DATA_HI[31:0]	A Second Data Register It contains bits [63:32] of a memory entry.	R/W

29.3.12.12 RM_STATUS : PRP Memory Controller Status Indication

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D2C

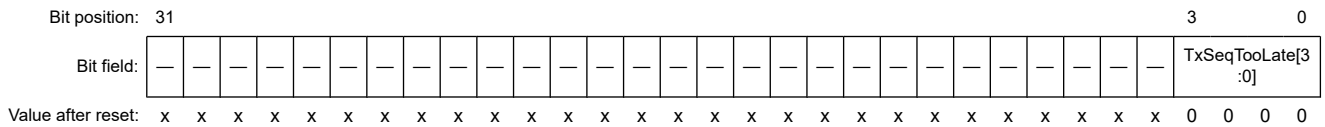


Bit	Symbol	Function	R/W
11:0	ageaddress[11:0]	Address of an entry which the aging process inspects when the aging timer expires next time.	R
31:12	—	The read values are undefined.	R

29.3.12.13 TxSeqTooLate : PRP Frame Transmission Retrieval of Failed Sequence

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D30

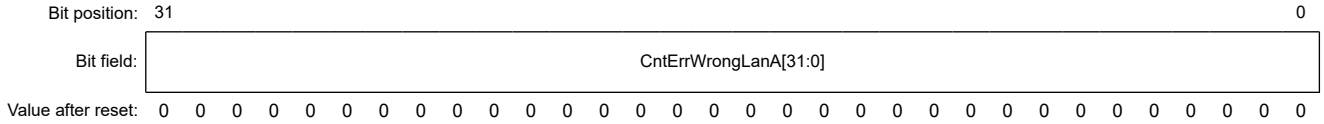


Bit	Symbol	Function	R/W
3:0	TxSeqTooLate[3:0]	Retrieval of a Sequence Number Failed A frame was transmitted and duplicated without appending the RCT. One bit per port, latched high when such an event occurred. Writing 1 to the register in the corresponding bit position clears the latch. This can happen on ports that transmit to the redundant ports therefore, the RedBox function requires to retrieve the sequence number for the frame (based on its source address). If that source address lookup took too long and the frame got forwarded before the lookup can complete, the frame is transmitted without appending an RCT (it is still duplicated). Latches are also cleared when PRP_CONFIG.PRP_ENA = 0.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

29.3.12.14 CntErrWrongLanA : PRP Wrong ID LAN-A Count Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D34

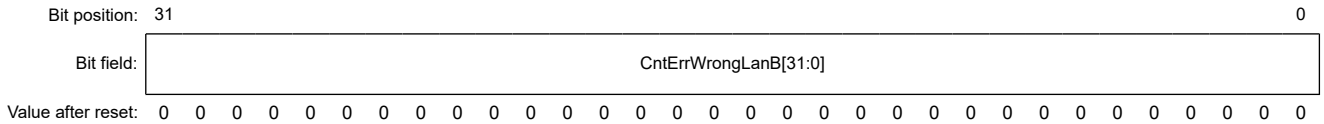


Bit	Symbol	Function	R/W
31:0	CntErrWrongLanA[31:0]	Valid frames received on LAN A which have an RCT (valid length + suffix) but LAN ID is not matching LAN A.	R

29.3.12.15 CntErrWrongLanB : PRP Wrong ID LAN-B Count Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D38

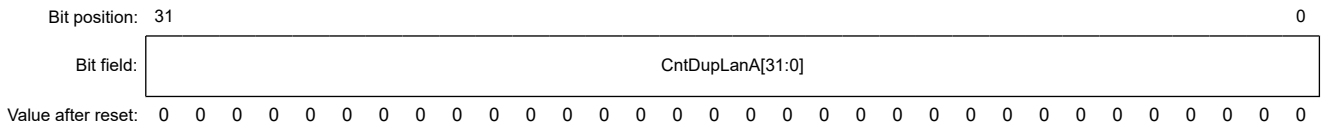


Bit	Symbol	Function	R/W
31:0	CntErrWrongLanB[31:0]	Valid frames received on LAN B which have an RCT (valid length + suffix) but LAN ID is not matching LAN B.	R

29.3.12.16 CntDupLanA : PRP Duplicate LAN-A Count Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D3C

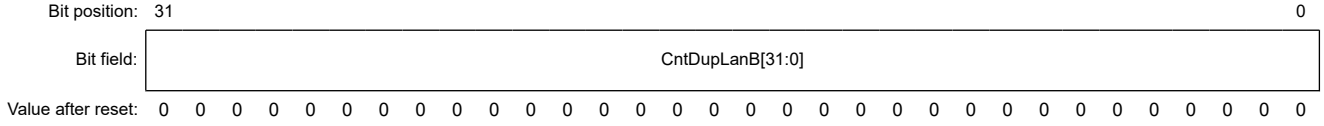


Bit	Symbol	Function	R/W
31:0	CntDupLanA[31:0]	Valid frames received on LAN A that were dropped by duplicate detection.	R

29.3.12.17 CntDupLanB : PRP Duplicate LAN-B Count Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D40

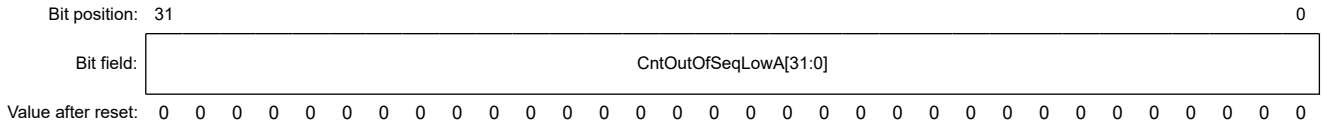


Bit	Symbol	Function	R/W
31:0	CntDupLanB[31:0]	Valid frames received on LAN B that were dropped by duplicate detection.	R

29.3.12.18 CntOutOfSeqLowA : PRP Sequence Error Low LAN-A Count Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D44

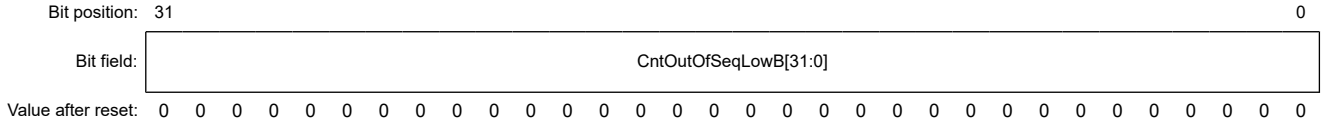


Bit	Symbol	Function	R/W
31:0	CntOutOfSeqLowA[31:0]	Valid and accepted frames received on LAN A with a sequence number less than last window (DUP_W).	R

29.3.12.19 CntOutOfSeqLowB : PRP Sequence Error Low LAN-B Count Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D48

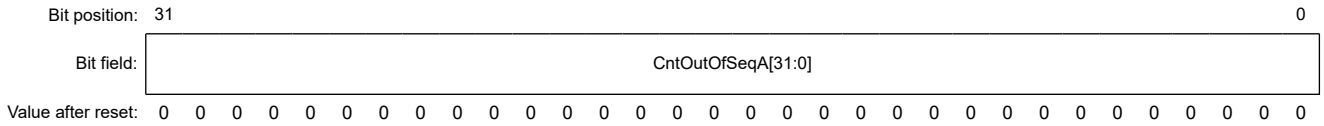


Bit	Symbol	Function	R/W
31:0	CntOutOfSeqLowB[31:0]	Valid and accepted frames received on LAN B with a sequence number less than last window (DUP_W).	R

29.3.12.20 CntOutOfSeqA : PRP Sequence Error LAN-A Count Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D4C

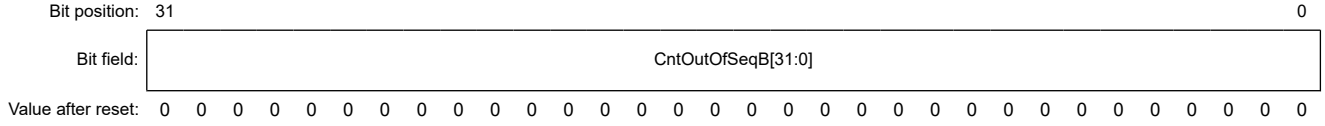


Bit	Symbol	Function	R/W
31:0	CntOutOfSeqA[31:0]	Valid and accepted frames received on LAN A with an unexpected sequence number. Increments also with CntOutOfSeqLowA.	R

29.3.12.21 CntOutOfSeqB : PRP Sequence Error LAN-B Count Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D50

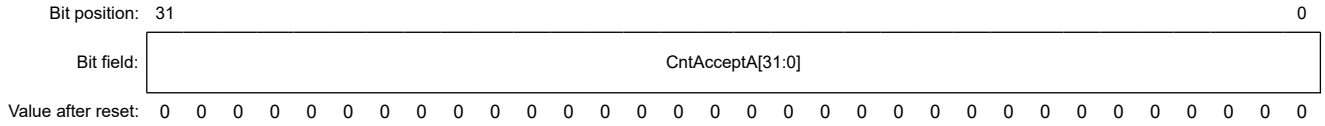


Bit	Symbol	Function	R/W
31:0	CntOutOfSeqB[31:0]	Valid and accepted frames received on LAN B with an unexpected sequence number. Increments also with CntOutOfSeqLowB.	R

29.3.12.22 CntAcceptA : PRP Valid Frame LAN-A Count Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D54

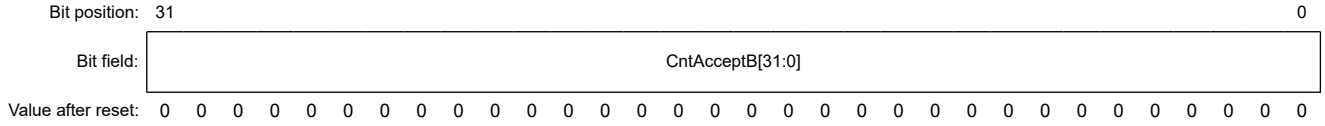


Bit	Symbol	Function	R/W
31:0	CntAcceptA[31:0]	Valid frames received on LAN A which had a valid sequence number in the expected range.	R

29.3.12.23 CntAcceptB : PRP Valid Frame LAN-B Count Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D58

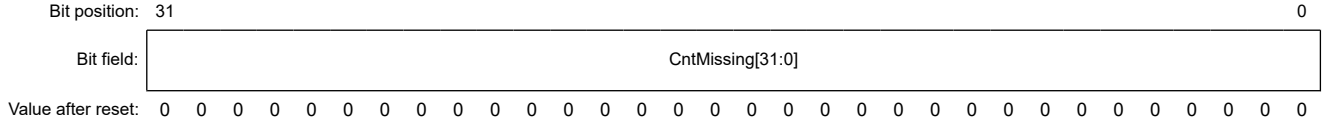


Bit	Symbol	Function	R/W
31:0	CntAcceptB[31:0]	Valid frames received on LAN B which had a valid sequence number in the expected range.	R

29.3.12.24 CntMissing : PRP Drop History Adjustment Count

Base address: ETHSW = 0x8012_0000

Offset address: 0x3D5C



Bit	Symbol	Function	R/W
31:0	CntMissing[31:0]	Indicates adjustment of the drop history as a frame was received with a sequence number of expected + history + 1. This occurs if the same frame was dropped in both LAN segments (one sequence number is missing) and the history is now extended beyond that sequence number (causing it to be treated as drop allowed).	R

29.3.13 Integrated Hub Module Registers

29.3.13.1 HUB_CONFIG : HUB Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	JAM_WAIT_IDLE	PRMB_GEN_DIS	CRS_GEN	IPG_WAIT[2:0]			—	TIMER_SEL	HUB_ISOLATE	TRIGGER_MODE	RETRANSMIT_ENA	HUB_ENA
Value after reset:	x	x	x	x	0	0	0	1	1	0	x	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HUB_ENA	Enable Integrated HUB Operation All other configuration registers must be configured correctly before setting this bit. When cleared, the HUB statistics counters are also cleared, if any.	R/W
1	RETRANSMIT_ENA	Enable Hub Retransmit Capability When this bit is 1 and a collision occurs while a MAC of the Hub Group is transmitting, the MAC is allowed to retransmit if possible (following IEEE 802.3 half-duplex backoff rules). When this bit is 0 and a collision occurs, the MAC does not retransmit the frame and the frame is discarded. This bit must be set to 0 when TRIGGER_MODE = 1.	R/W
2	TRIGGER_MODE	Enable Single Frame Trigger Mode When this bit is 1, the HUB sends a single frame (single MAC transmit) only when allowed by a trigger event. Trigger events are generated by the settings in HUB_TRIGGER_IMMEDIATE and HUB_TRIGGER_AT registers, individually per port. The RETRANSMIT_ENA bit must be set to 0 when trigger mode is active.	R/W
3	HUB_ISOLATE	Isolate all hub ports from the other ports of the switch and allow communication with management port only. It is then up to the application of the management port to implement some bridging functionality to other ports as required. When this bit is 1, frames received at any of the hub ports are always forwarded to the management port. No lookup or flooding occurs. In addition, all frames from ports outside the Hub Group that need forwarding to the Hub Group are always forwarded to the management port. The setting has no effect on forwarding in-between non-hub ports and forwarding to/from the management port. When this bit is 0, normal forwarding is performed and the HUB_DEFPORT mask is used when forwarding traffic to the hub-group from any other port. This setting only controls forwarding within the switch. The Hub PHY copy function is not affected by this setting.	R/W
4	TIMER_SEL	Select the timer to use for timed triggers 0: Timer 0 1: Timer 1	R/W
5	—	The read value is undefined. The write value should be 0.	R/W
8:6	IPG_WAIT[2:0]	Configures the number of bytes to wait before allowing a new frame through the HUB. Do not change.	R/W

Bit	Symbol	Function	R/W
9	CRS_GEN	When set to 1, the integrate HUB generates a local CRS signal whenever a frame is being copied. This prevents the local ports from transmitting frames until the frame is completed. Default is disabled (relies on the CRS from the PHY). Do not change.	R/W
10	PRMB_GEN_DIS	When set to 1, the preamble regeneration when transmitting frames is disabled. When 0, the HUB ensures that the frame contains 7 bytes of preamble plus the SFD. Do not change.	R/W
11	JAM_WAIT_IDLE	When set to 1, the HUB will wait until all ports are IDLE after a collision event to resume forwarding traffic between the ports. When 0, the HUB will resume forwarding traffic when at most 1 port is non-IDLE. Do not change.	R/W
31:12	—	The read values are undefined. The write value should be 0.	R/W

The HUB_CONFIG register enables the integrated hub operation and set mode.

29.3.13.2 HUB_GROUP : HUB Port Group Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	HUB_GROUP[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	HUB_GROUP[2:0]	Define all ports that should be combined to a Hub Group. One bit per port (bit 0 = port 0, bit 1 = port 1, and bit 2 = port 2). A portmask with at least 1 bit set (typically at least 2 are set). The MACs of the Hub group must be configured to identical speed and half-duplex operation. The group should not collide with any other group settings of the switch that may be active such as DLR as this can lead to unpredictable behavior. Only setting a single port allows the port to use the trigger capabilities.	R/W
31:3	—	The read values are undefined. The write value should be 0.	R/W

29.3.13.3 HUB_DEFPORT : HUB Default Port Selection Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	HUB_DEFPORT[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	HUB_DEFPORT[2:0]	The default port within the Hub Group where all traffic from a port outside the group is forwarded to port (bit 0 = port 0, bit 1 = port 1, and bit 2 = port 2). If a frame should be forwarded to any of the hub ports, the frame is sent to this port only. The copy function of the hub copies it to all PHY interfaces of the group eventually. Portmask with exactly 1 bit from the hub group set. Forced forwarding of the management port can be used to direct frames to other ports within the group.	R/W
31:3	—	The read values are undefined. The write value should be 0.	R/W

29.3.13.4 HUB_TRIGGER_IMMEDIATE : HUB Transmission Trigger Immediate Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	HUB_TRIGGER_IMMEDIATE[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	HUB_TRIGGER_IMMEDIATE[2:0]	Trigger immediate transmission of a single frame from given port within the hub group (bit 0 = port 0, bit 1 = port 1, and bit 2 = port 2). Portmask with at maximum having a single bit set. Setting a port bit that is not within the hub group has no effect (register write is ignored). When set, the specified port is allowed to send a single frame immediately. The register is cleared when transmission started and an interrupt can be generated. If no frame is waiting at the port at the time of writing this register, the trigger is kept active until a frame arrives. It is then transmitted and the register is cleared.	R/W
31:3	—	The read values are undefined. The write value should be 0.	R/W

29.3.13.5 HUB_TRIGGER_AT : HUB Transmission Trigger At Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	HUB_TRIGGER_AT[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	HUB_TRIGGER_AT[2:0]	Trigger Transmission of a Single Frame at a Specific Time (bit 0 = port 0, bit 1 = port 1, and bit 2 = port 2). Portmask with at maximum having a single bit set. Setting a port bit that is not within the hub group has no effect (register write is ignored). The HUB_TTIME register should be written first to avoid unexpected triggering on an older/invalid timer value. The register is cleared when transmission has started and an interrupt can be generated. The register is also cleared if the port has no frame to transmit at the time of the trigger. The trigger interrupt occurs and the register clears but no frame is transmitted. A frame arriving after the trigger time is not transmitted (this is in contrast to the trigger immediate function).	R/W
31:3	—	The read values are undefined. The write value should be 0.	R/W

29.3.13.6 HUB_TTIME : HUB Transmission Time Define Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E14

Bit position:	31																												0
Bit field:	HUB_TTIME[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	HUB_TTIME[31:0]	Define the Time Value when a Trigger Should Occur A 32-bit value compared with the timer. The value is compared with the 32-bit timer selected by HUB_CONFIG.TIMER_SEL and if the timer reaches (or crosses) the value, the port given in HUB_TRIGGER_AT is allowed to transmit one frame.	R/W

29.3.13.7 HUB_IRQ_CONTROL : HUB Interrupt Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TRIGGER_TIMER	TRIGGER_IMMEDIATE	CHANGE_DET	RX_TRIGGER[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RX_TRIGGER[2:0]	Enable Interrupt on Receive Pattern Match Trigger Function One bit per port. Bit 0 = Port 0, Bit 1 = Port 1, Bit 2 = Port 2. When enabled, an interrupt is generated when any receive pattern matcher caused the indicated hub port to transmit one frame (trigger mode) and transmission has begun. This interrupt is in addition to the match interrupt capability associated with the pattern that can be enabled in the PATTERN_CTRL register. This trigger interrupt typically occurs shortly after the pattern match interrupt occurred (about 1 µs later in a 100 Mbps port which is the time between frame receive and transmit start following the IPG delay). Therefore, it can cause two separate interrupts for the same event to the system if both are enabled, which may cause unwanted processing overhead.	R/W
3	CHANGE_DET	Enable interrupt for hub TX state machine port state change request detection Assert when the HUB_STATUS register bit TX_Change_Pending asserts.	R/W

Bit	Symbol	Function	R/W
4	TRIGGER_IMMEDIATE	Enable interrupt when hub transmit started after writing the HUB_TRIGGER_IMMEDIATE register The event is caused when the transmission has started and the HUB_TRIGGER_IMMEDIATE register is cleared and can accept a new command.	R/W
5	TRIGGER_TIMER	Enable interrupt when hub transmit started after writing the HUB_TRIGGER_TIME register and the timeout value is reached (register HUB_TTIME). The event is caused when the transmission has started and the HUB_TRIGGER_TIME register is cleared.	R/W
31:6	—	The read values are undefined. The write value should be 0.	R/W

If any of the interrupt enable bits is set and a corresponding event occurs, the Hub interrupt (HUB_INT) within the global INT_STAT_ACK Interrupt Status register asserts. To clear any of the interrupts, write to the HUB_IRQ_STAT_ACK register.

29.3.13.8 HUB_IRQ_STAT_ACK : HUB Interrupt Status/ACK Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TRIGGER_TIMER	TRIGGER_IMMEDIATE	CHANGE_DET	RX_TRIGGER[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RX_TRIGGER[2:0]	Interrupt Pending Indication Bits corresponding to the bits defined in HUB_IRQ_CONTROL. If an event occurs, the corresponding bit is latched high. To clear the bit, write 1 to it. The latches operate independently of the HUB_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W
3	CHANGE_DET	This bit functions the same as RX_TRIGGER bit.	R/W
4	TRIGGER_IMMEDIATE	This bit functions the same as RX_TRIGGER bit.	R/W
5	TRIGGER_TIMER	This bit functions the same as RX_TRIGGER bit.	R/W
31:6	—	The read values are undefined. The write value should be 0.	R/W

To clear any of the interrupts, write to the HUB_IRQ_STAT_ACK register.

29.3.13.9 HUB_STATUS : HUB Status Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TX_Change_Pending	Speed_OK	TX_BUSY	TX_ACTIVE	—	—	—	—	—	—	PORTS_ACTIVE[2:0]		
Value after reset:	x	x	x	0	0	0	0	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	PORTS_ACTIVE[2:0]	When this bit is 1, it shows the currently active ports of the Hub group which are allowed for transmit. It can differ from the HUB_GROUP setting if a port is disabled or has no link for example. One bit per port. Bit 0 = Port 0, Bit 1 = Port 1, Bit 2 = Port 2.	R
8:3	—	The read values are undefined.	R
9	TX_ACTIVE	When this bit is 1, the hub global transmit state machine has successfully entered Hub mode and is now controlling the hub group. The bit reacts with some delay following a write into the HUB_CONFIG/HUB_GROUP registers until it successfully enters or leaves the hub operational mode. The state machine may not enter hub mode even when enabled through the HUB_CONFIG register, as long as the ports are still transferring data or a misconfiguration exists (speed differences, no group defined).	R
10	TX_BUSY	When this bit is 1, the local device currently transmits data to all ports within the hub group. The bit is informal and toggles during normal operation.	R
11	Speed_OK	When this bit is 1, it indicates that the port speed of all group ports match. This bit must be set to 1 to allow the TX state machine to enter active state and may indicate a misconfiguration or link failure that changed the speed of the attached PHY of a port. It is important that all PHYs at all time when they are part of the Hub group operate with the same MAC interface speed even if the link is disconnected (the PHY must not change its MAC interface speed from 100 Mbps to 10 Mbps after the link dropped). The hub is inoperable (not transmitting) and cannot enter the active state as long as this bit is 0. The bit is valid as soon as the hub group is defined (HUB_GROUP) and enabled (HUB_CONFIG). It may give an indication why the TX_ACTIVE is not getting set.	R
12	TX_Change_Pending	Indicate a pending change request in the hub transmitter that is unsolved and cause the hub to stop operation (no longer performing any transmissions). A state change is requested when a port of the hub group changed its operational mode unexpectedly (link fail, MAC speed change). It also asserts if, for example, the hub is disabled in the middle of a transmission until the transmission has ended normally, which will then allow the hub to become disabled (and eventually deassert TX_ACTIVE). The bit deasserts if the misconfiguration has been resolved and the hub is allowed to perform transmissions. The hub is inoperable if this bit is set and the misconfiguration may need the intervention of the application to be resolved (speed mismatch persists).	R
31:13	—	The read values are undefined.	R

The HUB_STATUS register shows informal real-time status information of the hub.

29.3.13.10 HUB_OPORT_STATUS : HUB Output Port Status Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	HUB_OPORT_STATUS[2:0]		
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	HUB_OPORT_STATUS[2:0]	Per Output Port Data Available Status One bit per port. Bit 0 = Port 0, …, Bit 2 = Port 2. When this bit is set to 1, it indicates the output port has data available for transmission (any queue of the port has data). This is a real-time indication of all line ports of the switch, not only the ones of the hub group.	R
31:3	—	The read values are undefined.	R

29.3.14 TDMA Scheduler Registers

29.3.14.1 TDMA_CONFIG : TDMA Configuration Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	OUT_CT_ENA[3:0]				—	—	—	—	IN_CT_ENA[3:0]			
Value after reset:	x	x	x	x	1	1	1	1	x	x	x	x	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TIMER_SEL_ACTIVE	—	—	HOLD_REQ_CLR	OUT_CT_WREN	IN_CT_WREN	RED_OVRD	RED_OVRD_ENA	RED_PERIOD	—	TIMER_SEL	WAIT_START	TDMA_ENA
Value after reset:	x	x	x	0	x	x	0	0	0	0	0	0	x	0	0	0

Bit	Symbol	Function	R/W
0	TDMA_ENA	Enable TDMA Scheduler The bit should be set only after all other TDMA scheduler configuration registers have been correctly configured. When set, the scheduler begins with its first cycle at the time given in TDMA_START. Before that time is reached, the queue state is not changed, but can be preset with the control register MMCTL_QGATE. Clearing the bit is allowed any time to disable the TDMA scheduler allowing all ports to transmit normally again. The scheduler must be enabled before the timer has reached the value in TDMA_START. Enabling it too late causes the scheduler to start after the timer has wrapped around and then reached the start value.	R/W
1	WAIT_START	Status bit which is set as long as the scheduler is enabled but has not yet reached the time given in register TDMA_START. The bit clears when the start time has reached and the scheduler has begun with the very first cycle.	R

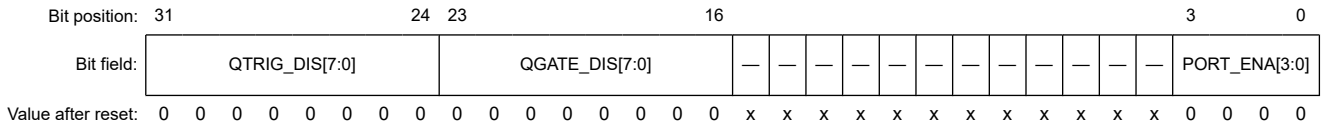
Bit	Symbol	Function	R/W
2	TIMER_SEL	Select which timer to use as the time source for the scheduler The selected timer can only be changed when TDMA_ENA is 0. Read TIMER_SEL_ACTIVE to verify that the written value is being used. 0: Timer 0 1: Timer 1	R/W
3	—	The read value is undefined. The write value should be 0.	R/W
4	RED_PERIOD	Read only bit indicating the current period for Profinet This bit is set to 1 if the period is RED, set by the TCV. It may differ from the current period if RED_OVRD_ENA is set to 1.	R
5	RED_OVRD_ENA	Enables overriding the RED period status, regardless of the indication by the TCV. The value is set in RED_OVRD.	R/W
6	RED_OVRD	Override Value for the RED Period When set to 1, the period is overridden to be RED, otherwise set to GREEN. Valid only when RED_OVRD_ENA is set to 1.	R/W
7	IN_CT_WREN	Enable writing the IN_CT_ENA control to the ingress ports.	W
8	OUT_CT_WREN	Enable writing the OUT_CT_ENA control to the egress ports.	W
9	HOLD_REQ_CLR	Writing 1 to this register clears the state of TDMA hold request.	W
11:10	—	The read values are undefined. The write value should be 0.	R/W
12	TIMER_SEL_ACTIVE	Return the current timer being used for the TDMA Scheduler 0: Timer 0 1: Timer 1	R
15:13	—	The read values are undefined. The write value should be 0.	R/W
19:16	IN_CT_ENA[3:0]	On read, return the current status of the ingress Cut-Through enable indicated by the TDMA scheduler. On write, override the ingress Cut-Through enable if IN_CT_WREN is also 1.	R/W
23:20	—	The read values are undefined. The write value should be 0.	R/W
27:24	OUT_CT_ENA[3:0]	On read, return the current status of the egress Cut-Through enable indicated by the TDMA scheduler. On write, override the egress Cut-Through enable if OUT_CT_WREN is also 1.	R/W
31:28	—	The read values are undefined. The write value should be 0.	R/W

The TDMA_CONFIG register configures and enables the TDMA scheduler. All other TDMA-related registers must be configured before enabling the scheduler.

29.3.14.2 TDMA_ENA_CTRL : TDMA Scheduling Enable Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E84



Bit	Symbol	Function	R/W
3:0	PORT_ENA[3:0]	Set to 1 to indicate that a port is operating in TDMA mode. When set to 1 for a port, the port does not prefetch another frame until the current frame in progress is done and if TDMA_PREBUF_DIS in COMMAND_CONFIG is set to 1. This helps adding precision to the queue gating operations indicated by the TDMA at the expense of loss of line rate.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
23:16	QGATE_DIS[7:0]	One bit per output queue. When a bit is set to 1, the TDMA scheduler gating commands do not affect the queue even if the queue mask in the TCV control data is set to 1.	R/W
31:24	QTRIG_DIS[7:0]	One bit per output queue. When a bit is set to 1, the TDMA scheduler triggering commands do not affect the queue even if the queue mask in the TCV control data is set to 1.	R/W

The TDMA_ENA_CTRL register controls whether the TDMA scheduler acts over specific ports and queues.

29.3.14.3 TDMA_START : TDMA Start Time Set Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E88

Bit position:	31																												0
Bit field:	TDMA_START[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TDMA_START[31:0]	Set the start time for the very first cycle after system initialization has completed. The value is compared with the system time (selected in TDMA_CONFIG.TIMER_SEL) and when it is reached (crossed), the scheduler begins with its first cycle. The 2nd cycle is then at TDMA_START + TDMA_CYCLE.	R/W

29.3.14.4 TDMA_MODULO : TDMA System Timer Modulo

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E8C

Bit position:	31																												0
Bit field:	TDMA_MODULO[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TDMA_MODULO[31:0]	The System Timer Modulo This setting gives the maximum time value + 1 the 32-bit timer (selected TDMA_CONFIG.TIMER_SEL) will reach before it wraps around. For example, if the timer represents nanoseconds and wraps once per second, then the modulo given in this register must be 1,000,000,000. This means the timer presents values from 0 to 999,999,999 at the 32-bit timer input. It is not expected that the values at the 32-bit timer increment continuously. Relevant for this setting is only the possible range of values. Setting a value of 0 indicates a wrap-around at 2^{32}.	R/W

29.3.14.5 TDMA_CYCLE : TDMA Periodic Cycle Set Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E90

Bit position:	31																												0
Bit field:	TDMA_CYCLE[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TDMA_CYCLE[31:0]	The periodic cycle time for the scheduler given in system timer time. For example, if the 32-bit timer (selected TDMA_CONFIG.TIMER_SEL) represents nanoseconds and the cycle time is 800 µs, a value of 800,000 must be set. The cycle time can be manipulated any time during TDMA operation to possibly compensate for a drifting system timer (usually the system timer should be adjusted and locked to a master clock).	R/W

29.3.14.6 TCV_SEQ_ADDR : TCV Sequence Address Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E94

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADDR_AINC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	TCV_S_ADDR[11:0]											
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	TCV_S_ADDR[11:0]	Address to write to or read from in the TCV sequence table.	R/W
30:12	—	The read values are undefined. The write value should be 0.	R/W
31	ADDR_AINC	When set to 1, read and write operations performed using TCV_SEQ_CTRL causes the address in TCV_S_ADDR to auto-increment after the operation.	R/W

The TCV_SEQ_ADDR register programs the address (entry) to read or write in the TCV sequence table. This register also enables address auto-increment.

29.3.14.7 TCV_SEQ_CTRL : TCV Sequence Table Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E98

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	READ_MODE	—	GPIO[7:0]								—	—	—	—	—	—
Value after reset:	0	x	0	0	0	0	0	0	0	0	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TCV_D_IDX[8:0]									INT	START
Value after reset:	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	START	Indicate this TCV must be executed after the next cycle start When set, identifies this TCV to be a cycle start vector. The OFFSET in the TCV is taken as the offset after the next cycle starts. The offset value can be 0. This bit should be always set to 1 for entry 0.	R/W
1	INT	Indicates this TCV generates an interrupt to the CPU when activated When set to 1, an interrupt is generated to the CPU when this TCV is executed (for example, after the indicated OFFSET relative to the cycle start).	R/W
10:2	TCV_D_IDX[8:0]	Index to the TCV Data Entry Index in the TCV Data Table. Select which entry in the TCV data to use for the time offset and control information.	R/W
21:11	—	The read values are undefined. The write value should be 0.	R/W
29:22	GPIO[7:0]	Generic bits that control the output pins ETHSW_TDMAOUTn (n = 0 to 7) The value of these bits is propagated to ETHSW_TDMAOUTn when the vector is executed. The output pin can be configured for level, toggle, or strobe operation using the TDMA_GPIO register.	R/W
30	—	The read value is undefined. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31	READ_MODE	When set to 1, a read operation is performed instead of writing to the TCV sequence table. The read data (START, INT, TCV_D_IDX[8:0], and GPIO) can be obtained by reading this register afterwards. On read, this field always returns 0.	R/W

Control data to be written or read from the TCV sequence table. When written, the value of the field READ_MODE indicates whether the TCV sequence table is read or written from or to the address programmed in TCV_SEQ_ADDR.

29.3.14.8 TCV_SEQ_LAST : TCV Sequence Last Entry

Base address: ETHSW = 0x8012_0000

Offset address: 0x3E9C

Bit position:	31				27												16				11											0
Bit field:	—	—	—	—	ACTIVE[11:0]												—	—	—	—	LAST[11:0]											
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	LAST[11:0]	Defines the last entry to read from the TCV sequence table when the TDMA scheduler is operating. After reading from this LAST entry, the next entry read is entry 0.	R/W
15:12	—	The read values are undefined. The write value should be 0.	R/W
27:16	ACTIVE[11:0]	Return the active TCV sequence entry.	R
31:28	—	The read values are undefined. The write value should be 0.	R/W

The TCV_SEQ_LAST register defines which is the last entry in the TCV_SEQ to be read, afterwards, it rolls back to entry TDMA_TCV_START. TCV entries are read sequentially starting from TDMA_TCV_START up to (TDMA_TCV_START + TCV_SEQ_LAST).

29.3.14.9 TCV_D_ADDR : TCV Data Address Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3EA0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AINC_WR_ENA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADDR[8:0]								
Value after reset:	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	ADDR[8:0]	Address to read from/write to in the TCV data table Must be set before accessing TCV_D_CTRL and TCV_D_OFFSET.	R/W
30:9	—	The read values are undefined. The write value should be 0.	R/W
31	AINC_WR_ENA	Auto-Increment Enable When set to 1, auto-increments after writes to TCV_D_CTRL.	R/W

The TCV_D_ADDR register is the address register for read and write operations to the TCV data table.

29.3.14.10 TCV_D_OFFSET : TCV Data Offset Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3EA4

Bit position:	31																												0
Bit field:	TCV_D_OFFSET[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TCV_D_OFFSET[31:0]	32-bit time offset for the TCV data entry indicated by TCV_D_ADDR. When accessing the table, TCV_D_OFFSET must be read or written before TCV_D_CTRL.	R/W

29.3.14.11 TCV_D_CTRL : TCV Data Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3EA8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PMASK[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	QGATE[7:0]								HOLD_REQ	GATE_MODE	TRIGGER_MODE	IN_CT_ENA	OUT_CT_ENA	RED_PERIOD	INC_CTR1	INC_CTR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INC_CTR0	Increment Control for Counter 0 When set to 1, the counter is enabled. This counter can be used to keep track of the number of cycles since the start of the TDMA operation. For Profinet, it tracks the SendClock number so it is possible for software to recover the current state after a lockup or software reboot. This may not be required as the current cycle number can be determined by reading the current absolute time.	R/W
1	INC_CTR1	Increment Control for Counter 1 When set to 1, the counter is enabled. This is a smaller counter that can trigger periodic interrupts. For Profinet, it can be used to keep track of the phase when the reduction ratio is greater than 1.	R/W
2	RED_PERIOD	Period Color Control (for Profinet IRT) When set to 1, indicates to the ingress RED_GUARD and MAC bridge functions that the RED period is active. This control is passed to the pattern matchers. Set this bit to 0 if period color identification is not required.	R/W
3	OUT_CT_ENA	Output Cut-Through Enable Enable Cut-Through (CT) operation in the output memory read controller. This is used to disable CT in the YELLOW period and initiate the YELLOW port byte counter. When GATE_MODE and TRIGGER_MODE are both 0 and OUT_CT_ENA is set to 1, the cut-through state for each port is set to the value in PMASK where 1 enables cut-through, and 0 disables it. If GATE_MODE is set to 1 then the cut-through state is changed for the ports selected in PMASK(P) to the value in OUT_CT_ENA. If GATE_MODE is set to 0 and TRIGGER_MODE is set to 1, the cut-through state is not changed. Set this bit to 1 if control on the Cut-Through mode of the switch is not required.	R/W

Bit	Symbol	Function	R/W
4	IN_CT_ENA	Input Cut-Through Enable When set to 1, allows changing the ingress cut-through state in the input frame receiver. The IN_CT_ENA is effective when GATE_MODE and TRIGGER_MODE are both 0. When GATE_MODE and TRIGGER_MODE are both 0, and IN_CT_ENA is set to 1, the ingress cut-through state is changed for all ports to the values in PMASK(P), where 1 enables cut-through, and 0 disables it. Set this bit to 1 if control on the Cut-Through mode of the switch is not required.	R/W
5	TRIGGER_MODE	Trigger mode enable when set to 1. GATE_MODE must be 0, otherwise, GATE_MODE has precedence. When set to 1, QGATE causes a single-frame transmission between the closed queues selected QGATE on the ports selected by PMASK. GATE_MODE must be set to 0 for trigger operation.	R/W
6	GATE_MODE	Gate mode enable when set to 1. When set to 1, changes the state of the queues indicated in QGATE for the ports selected by PMASK.	R/W
7	HOLD_REQ	Preemption hold request. Generates a hold request to ports enabled in PMASK. When set to 1, a hold request operation (see section 29.4.18.7. MM_CTL.request (hold_req)) can be generated for the ports selected in PMASK. TRIGGER_MODE must be set to 0 for HOLD_REQ to be executed.	R/W
15:8	QGATE[7:0]	Bits mask, one per output queue Bit mask that controls which queues are gated or triggered in the current time slot. A value of 0 closes (disables) the queue when GATE_MODE is 1. A value of 1 opens (enables) the queue if GATE_MODE is set to 1, or triggers a frame if TRIGGER_MODE is 1 and GATE_MODE is 0. PMASK selects which ports are affected.	R/W
19:16	PMASK[3:0]	Bits mask, one per output port Bit mask (one per output port) that controls which queues of the ports are gated, triggered, hold request generation, and which ports change their Cut-Through mode setting. A value of 1 enables the port.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

The TCV_D_CTRL register control bits for the TCV entry indicated by TCV_D_ADDR. When accessing the table, TCV_D_OFFSET must be read or written before TCV_D_CTRL.

29.3.14.12 TDMA_CTR0 : TDMA Counter 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x3EAC

Bit position:	31																													0
Bit field:	TDMA_CTR0[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TDMA_CTR0[31:0]	32-bit counter that is incremented when the TCV field INC_CTR0 is set to 1. The counter can be initialized with software by writing a value. The counter rolls over from 0xFFFFFFFF to 0x00000000.	R/W

29.3.14.13 TDMA_CTR1 : TDMA Counter 1

Base address: ETHSW = 0x8012_0000

Offset address: 0x3EB0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	INT_VALUE[7:0]								MAX[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	WRITE_ENA	VALUE[7:0]							
Value after reset:	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VALUE[7:0]	Current Counter Value Can be written if bit WRITE_ENA is set to 1. On read, returns the current counter value.	R/W
8	WRITE_ENA	Write Enable for VALUE Must be set to 1 to override the counter value. This bit is a self-clearing bit.	W
15:9	—	The read values are undefined. The write value should be 0.	R/W
23:16	MAX[7:0]	Counter Maximum Value The counter increments up to MAX and then rolls over back to 0.	R/W
31:24	INT_VALUE[7:0]	Interrupt Value Set the value at which an interrupt to the CPU is generated.	R/W

TDMA_CTR1 is an 8-bit counter that is incremented when the TCV field INC_CTR1 is set to 1. The counter can be initialized with software by writing a value.

29.3.14.14 TDMA_TCV_START : TDMA TCV Sequence Entry Start

Base address: ETHSW = 0x8012_0000

Offset address: 0x3EB4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TDMA_TCV_START[11:0]											
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	TDMA_TCV_START[11:0]	Define the TCV_SEQ entry to start from. When the TDMA is first enabled or the last TCV sequence is executed, the cycle is restarted from this TCV sequence entry.	R/W
31:12	—	The read values are undefined. The write value should be 0.	R/W

29.3.14.15 TIME_LOAD_NEXT : TDMA Calculated Next Loading Time

Base address: ETHSW = 0x8012_0000

Offset address: 0x3EB8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TIME_LOAD_NEXT[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TIME_LOAD_NEXT[31:0]	Status giving the calculated time the scheduler loads into its internal compare register after the current running slot end is reached (not the end of the current slot).	R

29.3.14.16 TDMA_IRQ_CONTROL : TDMA IRQ Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3EBC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CTR1_INT_EN	—	—	—	—	—	—	—	—	—	—	—	—	TCV_INT_EN
Value after reset:	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	TCV_INT_EN	Enable Interrupts Generated by the TCV	R/W
12:1	—	The read values are undefined. The write value should be 0.	R/W
13	CTR1_INT_EN	Enable Interrupts Generated from Counter 1	R/W
31:14	—	The read values are undefined. The write value should be 0.	R/W

The TDMA_IRQ_CONTROL register is the interrupt enable for TDMA scheduler interrupts.

If any of the interrupt enable bit is set and a corresponding event occurs, the TDMA interrupt (TDMA_INT) within the global INT_STAT_ACK interrupt status register asserts. To clear any of the interrupts, write to the TDMA_IRQ_STAT_ACK register.

29.3.14.17 TDMA_IRQ_STAT_ACK : TDMA IRQ Status/ACK Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3EC0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CTR1_ACK	—	—	—	—	—	—	—	—	—	—	—	—	TCV_ACK
Value after reset:	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	TCV_ACK	TCV Execution Event If an event occurs, the bit is latched high. To clear the bit, write 1 to it. The latches operate independently of the TDMA_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected. To determine the index of the interrupt, the ISR should read the active TCV index from TCV_SEQ_LAST.ACTIVE. However, this is the current value and if the ISR takes too long to service, then ACTIVE may be already in a subsequent index.	R/W
12:1	—	The read values are undefined. The write value should be 0.	R/W

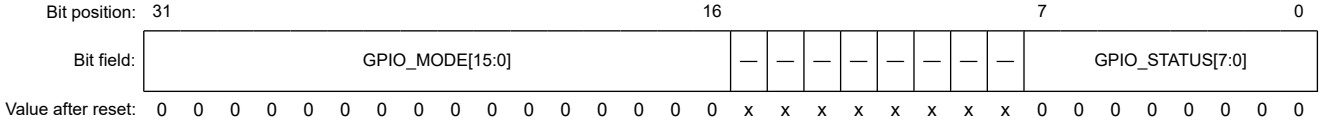
Bit	Symbol	Function	R/W
13	CTR1_ACK	Counter 1 Event If an event occurs, the bit is latched high. To clear the bit, write 1 to it. The latches operate independently of the TDMA_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W
31:14	—	The read values are undefined. The write value should be 0.	R/W

To clear any of the TDMA scheduler interrupts, write to the TDMA_IRQ_STAT_ACK register.

29.3.14.18 TDMA_GPIO : TDMA GPIO Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3EC4



Bit	Symbol	Function	R/W
7:0	GPIO_STATUS[7:0]	Status of the GPIO Output Pins The GPIO pins are driven directly from the TCV sequence table. This is the real-time status of the pins. Setting a pin in strobe mode may cause the state to be missed.	R
15:8	—	The read values are undefined. The write value should be 0.	R/W
31:16	GPIO_MODE[15:0]	2 bits per GPIO pin to configure its operating mode Bit 0/1 = GPIO0, …, Bit 6/7 = GPIO3, and so on. 0: Level mode 1: Strobe mode. The strobe duration is 16 cycles of the switch operating clock. 2: Toggle mode. A setting of 1 in the TCV sequence table for the corresponding pin causes a toggle in the GPIO. 3: Reserved	R/W

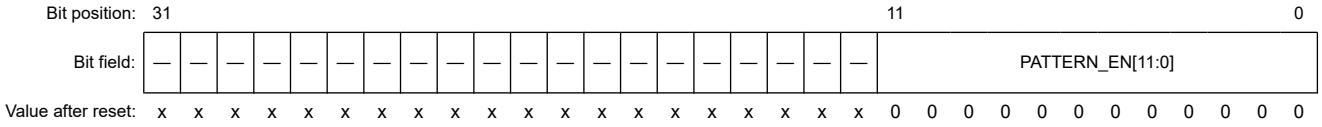
The TDMA_GPIO register indicates the GPIO configuration and status.

29.3.15 RX Pattern Matcher Registers

29.3.15.1 RXMATCH_CONFIGn : RX Pattern Matcher Configuration for Port n (n = 0 to 3)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3F00 + 0x4 × n



Bit	Symbol	Function	R/W
11:0	PATTERN_EN[11:0]	Enable Patterns on the Port (RX) One bit per pattern. When a bit is set, the corresponding pattern is searched within frames received at the port. The match function of the pattern executed is defined in its PATTERN_CTRLn (n: pattern). Multiple patterns can be enabled at the same time (allowing any match). If multiple patterns are enabled, they should be exclusive to each other. If multiple patterns match simultaneously, the executing function is unpredictable if the controls differ.	R/W
31:12	—	The read values are undefined. The write value should be 0.	R/W

The RXMATCH_CONFIGn register configures/enables pattern matchers for port n. The register defines the active patterns that are searched individually for every port. One register per port exists.

29.3.15.2 PATTERN_CTRLn : RX Pattern Matcher Function Control for Pattern n (n = 0 to 11)

Base address: ETHSW = 0x8012_0000

Offset address: 0x3F40 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	QUEUESEL[3:0]				—	TIMER_SEL	MATCH_LT	SWAP_BYTES	IMC_TRIGGER_DLY	IMC_TRIGGER	—	—	PORTMASK[3:0]			
Value after reset:	0	0	0	0	x	0	0	0	0	0	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LEARNING_DIS	PRIORITY[2:0]			VLAN_SKIP	MATCH_NOT_RED	MATCH_RED	HUBTRIGGER	FORCE_FORWARD	TIMER_SEL_OVR	MODE[1:0]		SET_PRIO	DISCARD	MGMTFWD	MATCH_NOT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MATCH_NOT	When set, a match is reported and the functions of this control are executed if the pattern does not match. When cleared (default), the following control executes when the pattern matches an incoming frame. This bit is available only in PATTERN_CTRL0. It is ignored and must be 0 in all other PATTERN_CTRLn registers. When set, the PATTERN_CTRL0 is used if no patterns (including PATTERN_CTRL0) report a match and the PATTERN_CTRL0 has been enabled for a port.	R/W
1	MGMTFWD	When set, the frame is forwarded to the management port only (suppressing destination address lookup). If the pattern is applied to the management port, this setting has no effect.	R/W
2	DISCARD	When set, the frame is discarded.	R/W
3	SET_PRIO	Set frame priority, overriding normal classification. When a match occurs, the priority of the frame is set as given in the PRIORITY bits. This priority takes precedence over (ignores) settings in the PRIORITY_CFGn register of the port where the frame is received.	R/W
5:4	MODE[1:0]	Selects the operating mode 0 0: Mode 1: Fixed 12-byte match 0 1: Mode 2: 2-byte table lookup 1 0: Mode 3: 2-byte range match at set offset 1 1: Mode 4: 2-byte range not-match at set offset	R/W
6	TIMER_SEL_OVR	Overrides the default timer to use by timestamp operations when set to 1, using instead the value in TIMER_SEL.	R/W
7	FORCE_FORWARD	When set, the frame is forwarded to the ports indicated in PORTMASK, ignoring the result from L2 lookups.	R/W
8	HUBTRIGGER	When set, the port defined in the PORTMASK setting is allowed for transmitting one frame. Usable for Hub mode only and if the hub is configured to operate in Trigger mode. The port mask should contain only a single port from within the hub group. A hub trigger is a global event and can occur only once until the enabled port transmits the frame. Therefore, if multiple patterns match simultaneously, only one trigger (of which, one is unknown) is executed. All other triggers that may follow are ignored as long as a pending trigger has not been processed (that is, frame transmit has not started yet).	R/W
9	MATCH_RED	Enable the pattern matcher only when the TDMA indicates that this is the RED period.	R/W
10	MATCH_NOT_RED	Enable the pattern matcher only when the TDMA indicates that this is not the RED period.	R/W

Bit	Symbol	Function	R/W
11	VLAN_SKIP	When set to 1, for operating modes 1, 2, and 3. The first Length/Type after the MAC source address is compared against 0x8100. If it matches, a VLAN tag is assumed and 4 bytes are skipped.	R/W
14:12	PRIORITY[2:0]	Priority of the frame used when SET_PRIO is set. The priority is used to forward the frame into the corresponding output queue of a port. A higher value defines a higher priority.	R/W
15	LEARNING_DIS	When set to 1, the hardware learning function is not executed.	R/W
19:16	PORTMASK[3:0]	A port mask used depending on the control bits (for example, HUBTRIGGER). One bit per port: Bit 16 = Port 0, Bit 17 = Port 1, Bit 18 = Port 2.	R/W
21:20	—	The read values are undefined. The write value should be 0.	R/W
22	IMC_TRIGGER	When set, the ports defined in the PORTMASK setting are allowed for transmitting one frame from the queues indicated by QUEUESEL. The trigger request is sent to the integrated memory controller.	R/W
23	IMC_TRIGGER_DLY	When set, the ports defined in the PORTMASK setting are allowed for transmitting one frame from the queues indicated by QUEUESEL. The trigger request is sent to the integrated memory controller and the event is delayed by the value programmed in MMCTL_DLY_QTRIGGER_CTRL.	R/W
24	SWAP_BYTES	Applicable only for operating modes 1, 2, and 3. When set to 1, the byte order is swapped from the order received by the frame. When set to 0, the first byte received by the frame is set into position 0 for comparison. When set to 1, the first byte received is set into position 3 (for mode 1) or position 2 (for mode 2 and 3) for comparison.	R/W
25	MATCH_LT	For operating modes 1, 2, and 3. When set to 1, the Length/Type field in the frame after the MAC source address is compared against the value in length_type in the compare register. If VLAN_SKIP is set and the frame has a VLAN tag with Length/Type of 0x8100 then the comparison is performed in the Length/Type following the VLAN tag.	R/W
26	TIMER_SEL	Override value to use when TIMER_SEL_OVR is set to 1 for selecting the timer for this frame. 0: Timer 0 1: Timer 1	R/W
27	—	The read value is undefined. The write value should be 0.	R/W
31:28	QUEUESEL[3:0]	A queue selector for the HUBTRIGGER function. Selects the queue to trigger a frame, or sets from 0x8 to 0xF to select one among all queues.	R/W

The PATTERN_CTRLn register controls the function that is executed on pattern matcher n.

For every pattern data set, one PATTERN_CTRLn register defines the function that is executed when the pattern matches onto a received frame. If a pattern is applied to received traffic of a port controlled by the register of the port RXMATCH_CONFIGn (n = 0 to 3).

A control bit acts independently of each other. Therefore, if multiple bits are set, those functions will all execute and/or affect the frame simultaneously.

29.3.15.3 PATTERN_IRQ_CONTROL : RX Pattern Matcher Interrupt Control Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3F80

Bit position:	31												19		16					11											0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ERROR_INT[3:0]				—	—	—	—	MATCHINT[11:0]											
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	MATCHINT[11:0]	Enable Interrupt on Receive Pattern Match One bit per pattern. Bit 0 = pattern 0, …, Bit 7 = pattern 7, … If set and the corresponding pattern match occurs when processing received frames (on any port), an interrupt occurs. The corresponding ETHSW_PTRNn (n = 0 to 11) interrupt also asserts as long as the interrupt is not cleared. The interrupt occurs following the CRC check of the frame by the MAC RX. Even if the frame is eventually dropped by the switch due to other rules, the match interrupt can still occur (it does not occur if the frame is dropped by the MAC).	R/W
15:12	—	The read values are undefined. The write value should be 0.	R/W
19:16	ERROR_INT[3:0]	Enable Interrupt on Internal Pattern Matcher Error One bit per port. Bit 16 = port 0, …, Bit 19 = port 4 When set to 1, an interrupt occurs. The error occurs because the pattern matcher response was not received before a new frame was started at the port.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

Interrupt enable control.

If any of the patterns match (globally) and the interrupt enable bit of the pattern is set, the PATTERN interrupt (PATTERN_INT) within the global INT_STAT_ACK interrupt status register asserts. To clear any of the interrupts, write to the PATTERN_IRQ_STAT_ACK register.

29.3.15.4 PATTERN_IRQ_STAT_ACK : RX Pattern Matcher Interrupt Status/ACK Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3F84

Bit position:	31												19				16				11											0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ERROR_INT[3:0]				—	—	—	—	MATCHINT[11:0]											
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	MATCHINT[11:0]	Interrupt pending indication for the corresponding pattern match events (see section 29.3.15.3. PATTERN_IRQ_CONTROL : RX Pattern Matcher Interrupt Control Register). If an event occurs, the corresponding bit is latched high. To clear the bit, write 1 to it. The latches operate independently of the PATTERN_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W
15:12	—	The read values are undefined. The write value should be 0.	R/W
19:16	ERROR_INT[3:0]	Interrupt pending indication for a pattern matcher error, per port. If an event occurs, the corresponding bit is latched high. To clear the bit, write 1 to it.	R/W
31:20	—	The read values are undefined. The write value should be 0.	R/W

Interrupt status and acknowledge register.

To clear any of the pattern matcher interrupts, write to the PATTERN_IRQ_STAT_ACK register.

29.3.15.5 PTRN_VLANID : Custom VLAN ID Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3F88

Bit position:	31																15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PTRN_VLANID[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PTRN_VLANID[15:0]	Custom VLAN ID to use. The default VLAN ID 0x8100 is always considered by the hardware. This value can be changed to detect other VLANs like 0x8808.	R/W
31:16	—	The read values are undefined. The write value should be 0.	R/W

29.3.15.6 PATTERN_SEL : RX Pattern Number Selection Register

Base address: ETHSW = 0x8012_0000

Offset address: 0x3F8C

Bit position:	31																											3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PATTERN_SEL[3:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PATTERN_SEL[3:0]	Define the pattern number which is selected for read/write through the PTRN_CMP_* and PTRN_MSK_* registers. A read or write operation into any of the PTRN_CMP and PTRN_MSK registers affects the pattern that is selected by this register. To configure a specific pattern dataset, this selection register must first be set to the pattern number of interest.	R/W
31:4	—	The read values are undefined. The write value should be 0.	R/W

29.3.15.7 PTRN_CMP_30 : Pattern Compare Value Bytes 3 .. 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x3FC0

Bit position:	31																												0
Bit field:	PTRN_CMP_30[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PTRN_CMP_30[31:0]	Pattern Compare Value Bytes 3 .. 0 First byte (byte 0) is bits [7:0]. 4th byte (byte 3) is bits [31:24].	R/W

29.3.15.8 PTRN_CMP_74 : Pattern Compare Value Bytes 7 .. 4

Base address: ETHSW = 0x8012_0000

Offset address: 0x3FC4

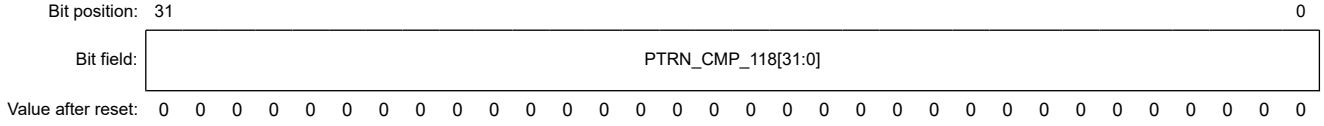
Bit position:	31																												0
Bit field:	PTRN_CMP_74[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PTRN_CMP_74[31:0]	Pattern Compare Value Bytes 7 .. 4 First byte (byte 4) is bits [7:0]. 4th byte (byte 7) is bits [31:24].	R/W

29.3.15.9 PTRN_CMP_118 : Pattern Compare Value Bytes 11 .. 8

Base address: ETHSW = 0x8012_0000

Offset address: 0x3FC8

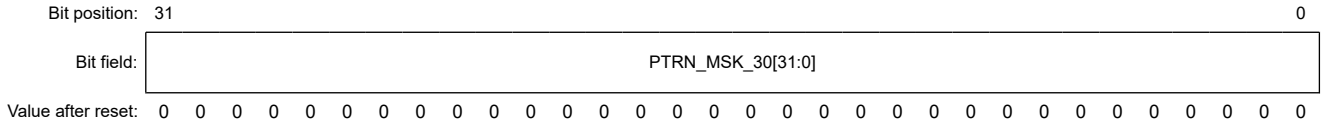


Bit	Symbol	Function	R/W
31:0	PTRN_CMP_118[31:0]	Pattern Compare Value Bytes 11 .. 8 First byte (byte 8) is bits [7:0]. 4th byte (byte 11) is bits [31:24].	R/W

29.3.15.10 PTRN_MSK_30 : Pattern Mask for Bytes 3 .. 0

Base address: ETHSW = 0x8012_0000

Offset address: 0x3FD0

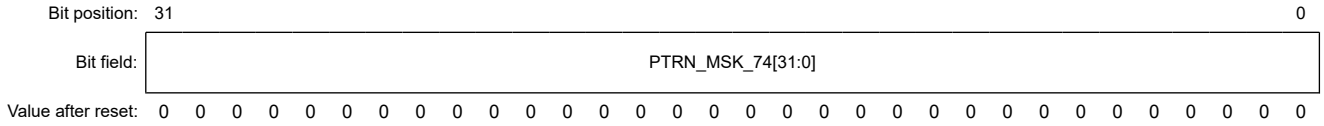


Bit	Symbol	Function	R/W
31:0	PTRN_MSK_30[31:0]	Pattern Mask for Bytes 3 .. 0	R/W

29.3.15.11 PTRN_MSK_74 : Pattern Mask for Bytes 7 .. 4

Base address: ETHSW = 0x8012_0000

Offset address: 0x3FD4

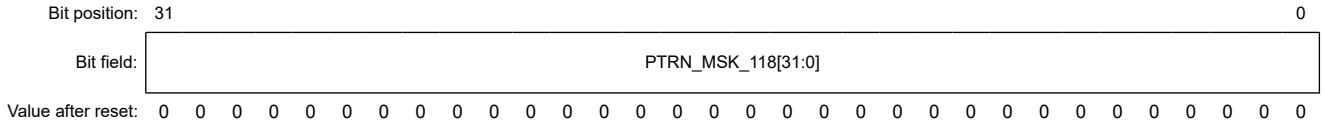


Bit	Symbol	Function	R/W
31:0	PTRN_MSK_74[31:0]	Pattern Mask for Bytes 7 .. 4	R/W

29.3.15.12 PTRN_MSK_118 : Pattern Mask for Bytes 11 .. 8

Base address: ETHSW = 0x8012_0000

Offset address: 0x3FD8



Bit	Symbol	Function	R/W
31:0	PTRN_MSK_118[31:0]	Pattern Mask for Bytes 11 .. 8	R/W

29.3.16 PTP Timer Pulse Control Registers

The registers control the PTP pulse generator. The PTP pulse generator operates based on the current time of the PTP timer and the register values and generates a pulse signal of any cycle from any start time. The pulse width can be set arbitrarily.

The register controls the pulse signal of ETHSW_PTPOUTn (n = 0 to 3) for interrupt signals or external output pins.

29.3.16.1 SWTMENn : PTP Timer Pulse Output Enable n Register (n = 0 to 3)

Base address: ETHSW_PTP = 0x8011_0400

Offset address: 0x400 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OUTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OUTEN	Enable ETHSW_PTPOUTn Signal Output 0: Disabled 1: Enabled	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set after setting completion of SWTMSTSEC, SWTMSTNS, SWTMPSEC, SWTMPNS, SWTMWTH, and SWTMMAXP registers. When this register is set to 1, changing SWTMSTSEC, SWTMSTNS, SWTMPSEC, SWTMPNS, SWTMWTH, and SWTMMAXP has no effect.

29.3.16.2 SWTMSTSECn : PTP Timer Pulse Start Second n Register (n = 0 to 3)

Base address: ETHSW_PTP = 0x8011_0400

Offset address: 0x404 + 0x100 × n

Bit position:	31	0
Bit field:	STSEC[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	STSEC[31:0]	Start Time by Second	R/W

29.3.16.3 SWTMSTNSn : PTP Timer Pulse Start Nanosecond n Register (n = 0 to 3)

Base address: ETHSW_PTP = 0x8011_0400

Offset address: 0x408 + 0x100 × n

Bit position:	31	0
Bit field:	STNS[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

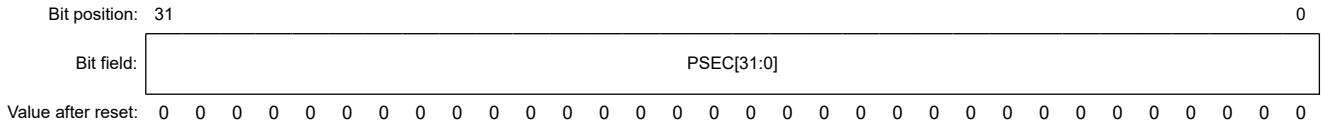
Bit	Symbol	Function	R/W
31:0	STNS[31:0]	Start Time by Nanosecond	R/W

Note: When SWTMMAXP is not 0, the value set in this register must be less than the value set in SWTMMAXP (SWTMSTNS < SWTMMAXP). When SWTMMAXP is 0, this register can be set to any value.

29.3.16.4 SWTMPSECn : PTP Timer Pulse Period Second n Register (n = 0 to 3)

Base address: ETHSW_PTP = 0x8011_0400

Offset address: 0x40C + 0x100 × n

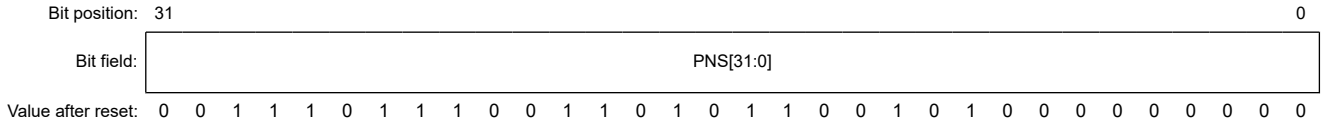


Bit	Symbol	Function	R/W
31:0	PSEC[31:0]	Period by Second	R/W

29.3.16.5 SWTMPNSn : PTP Timer Pulse Period Nanosecond n Register (n = 0 to 3)

Base address: ETHSW_PTP = 0x8011_0400

Offset address: 0x410 + 0x100 × n



Bit	Symbol	Function	R/W
31:0	PNS[31:0]	Period by Nanosecond	R/W

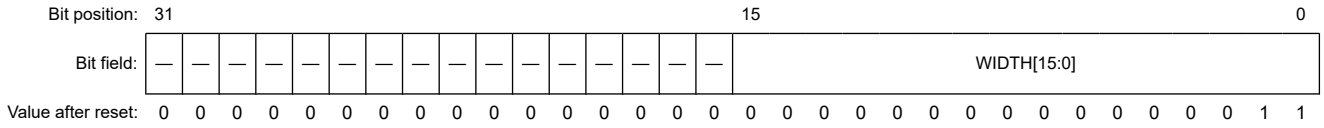
Note: The value set in this register must be larger than 16 × ATIME_INC[6:0] value (128 ns).

Note: When SWTMMAXP is not 0, the value set in this register must be less than or equal to the value set in SWTMMAXP (SWTMPNS ≤ SWTMMAXP). When SWTMMAXP is 0, this register can be set to any value.

29.3.16.6 SWTMWTHn : PTP Timer Pulse Width n Register (n = 0 to 3)

Base address: ETHSW_PTP = 0x8011_0400

Offset address: 0x414 + 0x100 × n



Bit	Symbol	Function	R/W
15:0	WIDTH[15:0]	Set the Pulse Width of ETHSW_PTPOUTn in the cycle number of ts_clk (8 ns). Output pulse width is the set value × 8 (ns). The ETHSW_PTPOUTn is set to 0 when this register is 0x0000.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: If ETHSW_PTPOUT is used as pulse event, do not change the value of this register (use the initial value). When it is used as external signal, appropriate width must be specified.

29.3.16.7 SWTMMAXPn : PTP Timer Pulse Max Nanosecond n Register (n = 0 to 3)

Base address: ETHSW_PTP = 0x8011_0400

Offset address: 0x418 + 0x100 × n

Bit position:	31																												0
Bit field:	MAXP[31:0]																												
Value after reset:	0	0	1	1	1	0	1	1	1	0	0	1	1	0	1	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MAXP[31:0]	Sets the boundary value in nanoseconds to carry from the nanosecond field to the second field. The same value as ATIME_EVT_PERIOD register must be set. Allowable value is 10^9 or 0 only.	R/W

Note: This register is set to the same value as of ATIME_EVT_PERIOD register.

Note: This register sets the boundary value in nanoseconds to carry from the nanosecond field to the second field. When the nanosecond field reaches the set value of this register, the second field is carried up. This register is normally used with the initial value of 0x3B9ACA00 (= 10^9). In this case, the nanosecond field counts from 0x0 to 0x3B9AC9FF (= 10^9 - 1) and then carries to the second field. If 0 is set in this register, the boundary value for carrying to the second field is interpreted as 0x100000000 (= 2^{32}). In this case, the nanosecond field is fully counted from 0x0 to 0xFFFFFFFF (= 2^{32} - 1) and then carries to the second field.

29.3.16.8 SWTMLATSECn : PTP Timer Pulse Latch Second n Register (n = 0 to 3)

Base address: ETHSW_PTP = 0x8011_0400

Offset address: 0x41C + 0x100 × n

Bit position:	31																												0
Bit field:	LATSEC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	LATSEC[31:0]	Latch the time of second count at the ETHSW_PTPOUTn rising edge.	R/W

Note: This register is updated when the PTPOUT signal rises, and the updated value is expressed in the formula t_{START} + t_{PERIOD} × n (n = 0, 1, 2, …) in seconds of the rise target time. t_{START} = {SWTMSTSEC, SWTMSTNS}, t_{PERIOD} = {SWTMPSEC, SWTMPNS}

Note: Since SWTMLATSEC is a 32-bit register, it will be 0x00000000 when incremented in the state of 0xFFFFFFFF.

29.3.16.9 SWTMLATNSn : PTP Timer Pulse Latch Nanosecond n Register (n = 0 to 3)

Base address: ETHSW_PTP = 0x8011_0400

Offset address: 0x420 + 0x100 × n

Bit position:	31																												0
Bit field:	LATNS[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	LATNS[31:0]	Latch the time of nanosecond count at the ETHSW_PTPOUTn rising edge.	R/W

Note: This register is updated when the PTPOUT signal rises, and the updated value is expressed in the formula t_{START} + t_{PERIOD} × n (n = 0, 1, 2, …) in nanoseconds of the rise target time. t_{START} = {SWTMSTSEC, SWTMSTNS}, t_{PERIOD} = {SWTMPSEC, SWTMPNS}

29.3.16.10 SWPTPOUTSEL : ETHSW_PTPOUT Select Register

Base address: ETHSW_PTP = 0x8011_0400

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	EVTSEL3	EVTSEL2	EVTSEL1	EVTSEL0	IOSEL3	IOSEL2	IOSEL1	IOSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IOSEL0	Select the source of the ETHSW_PTPOUT0 output signal 0: Pulse generator 0 output 1: TDMA_GPIO[4] output	R/W
1	IOSEL1	Select the source of the ETHSW_PTPOUT1 output signal 0: Pulse generator 1 output 1: TDMA_GPIO[5] output	R/W
2	IOSEL2	Select the source of the ETHSW_PTPOUT2 output signal 0: Pulse generator 2 output 1: TDMA_GPIO[6] output	R/W
3	IOSEL3	Select the source of the ETHSW_PTPOUT3 output signal 0: Pulse generator 3 output 1: TDMA_GPIO[7] output	R/W
4	EVTSEL0	Select the source of the ETHSW_PTPOUT0 event for GIC, DMAC, and ELC 0: Pulse generator 0 output 1: TDMA_GPIO[4] output	R/W
5	EVTSEL1	Select the source of the ETHSW_PTPOUT1 event for GIC, DMAC, and ELC 0: Pulse generator 1 output 1: TDMA_GPIO[5] output	R/W
6	EVTSEL2	Select the source of the ETHSW_PTPOUT2 event for GIC, DMAC, and ELC 0: Pulse generator 2 output 1: TDMA_GPIO[6] output	R/W
7	EVTSEL3	Select the source of the ETHSW_PTPOUT3 event for GIC, DMAC, and ELC 0: Pulse generator 3 output 1: TDMA_GPIO[7] output	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The SWPTPOUTSEL register is used to specify the source of ETHSW_PTPOUTn event and output signal (n = 0 to 3).

This register can only be written when protection is unlocked by the specific sequence using the section 27.3.1. PRCMD : Ethernet Protect Register. For the protection unlock procedure, see section 27.3.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register.

29.4 Operation

29.4.1 Ethernet Frame Format Overview

29.4.1.1 Overview

The IEEE 802.3 Standard defines the Ethernet Frame format as follows:

- An Ethernet frame has a minimum length of 64 bytes and a maximum length of 1518 bytes, excluding the preamble and the SFD bytes.
- An Ethernet frame consists of the following fields:
  - Seven bytes preamble
  - Start frame delimiter (SFD)
  - Two address fields
  - Length or type field
  - Data field
  - Frame check sequence (CRC value)
  - An EXTENSION field is defined only for half-duplex implementations and is not supported.

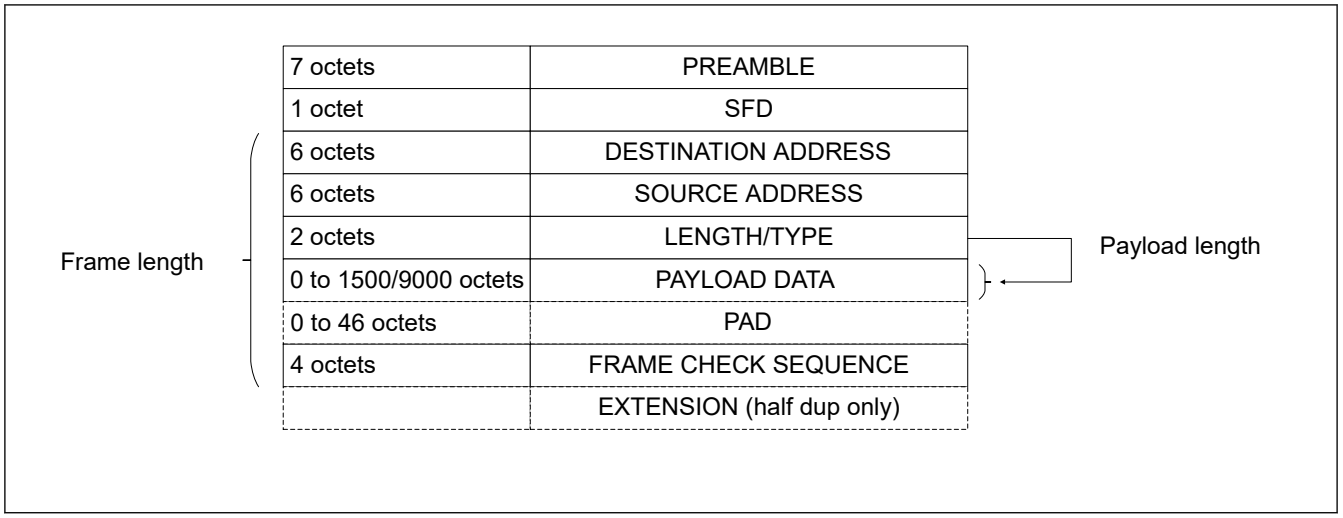


Figure 29.3 MAC frame format overview

Optionally MAC frames can be VLAN tagged with an additional 4-byte field (TPID and VLAN Info) inserted between the MAC Source Address and the Type/Length Field. VLAN tagging is defined by the IEEE P802.1Q specification. VLAN tagged frames have a maximum length of 1522 bytes (tagged standard frames) or 9022 bytes (tagged jumbo frames), excluding the preamble and the SFD bytes.

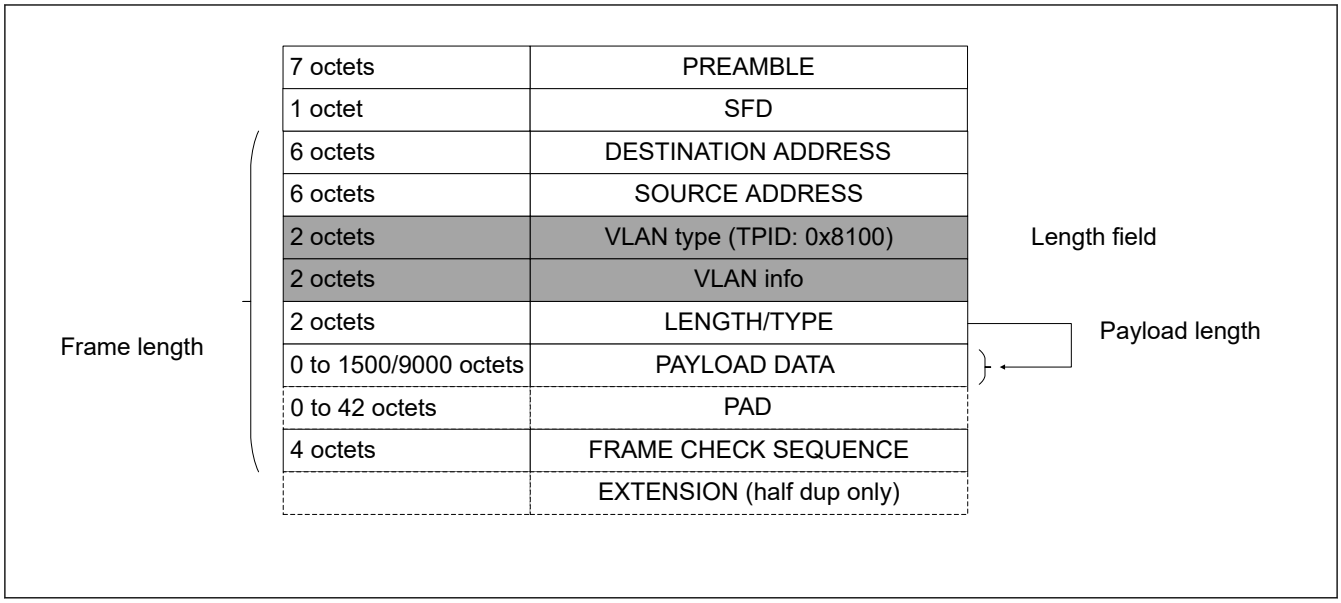


Figure 29.4 VLAN tagged MAC frame format overview

Table 29.5 MAC frame definition

Term	Description
Frame length	The length, in octets, defines the length of the complete frame without preamble and SFD. A frame has a valid length if it contains at least 64 octets and does not exceed the programmed maximum length (typical 1518).
Payload Length	The Length/Type field indicates the length of the payload section of the frame or identifies the type of the frame. The most significant byte is sent/received first. ● If the Length/Type field is set to a value less than 1536 (0x600), it is interpreted as a Length field indicating the number of following payload octets. A Length/Type field less than 46 indicates the payload is padded so that the minimum frame length requirement (64 bytes) is met. For VLAN tagged frames, a value less than 42 indicates a padded frame. ● If the Length/Type field is set to a value larger or equal to 1536 (0x600), it is interpreted as a Type field.
Destination and Source Address	48-Bit MAC addresses The least significant byte is sent/received first and the first two bits (two least significant bits) of the MAC address are used to distinguish MAC frames.

In typical Ethernet switching application, the MAC, on receive, should be programmed to accept every frame (Promiscuous mode), to check but not to change the frame FCS and forward the frame with the FCS field to the Switch. On transmit, the MAC should be programmed not to overwrite the source MAC address received from the switch and to transmit the frame with the FCS received from the switch.

However, since frame manipulation functions within the switch are used, CRC is stripped by the MAC receivers and is appended by the MAC transmitters.

29.4.1.2 MAC Address Overview

The destination address bit 0 is used to differentiate multicast and unicast addresses:

- If bit 0 is set 0, the MAC address is an individual (unicast) address.
- If bit 0 is set 1, the MAC address defines a group address (multicast address).
- If all 48 bits of the MAC address are set to 1, it indicates a broadcast address.

In addition, the first 24 bits of the MAC address define a vendor ID.

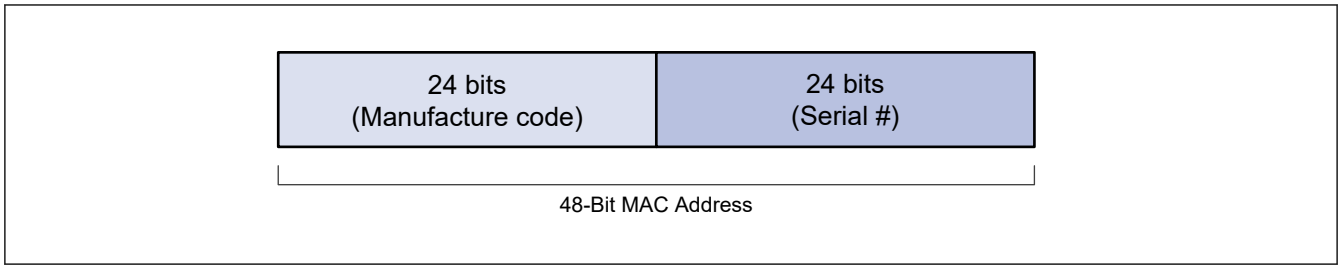


Figure 29.5 MAC address overview

29.4.1.3 VLAN Tag Overview

A VLAN tagged frame provides a 16-bit VLAN information field. For VLAN priority resolution (see (2) VLAN Priority Look Up), the switch uses the upper 4 bits of the first word (octet) of the VLAN Info field, that is the 3 bits priority field and the 1 bit CFI field can be used for priority classification.

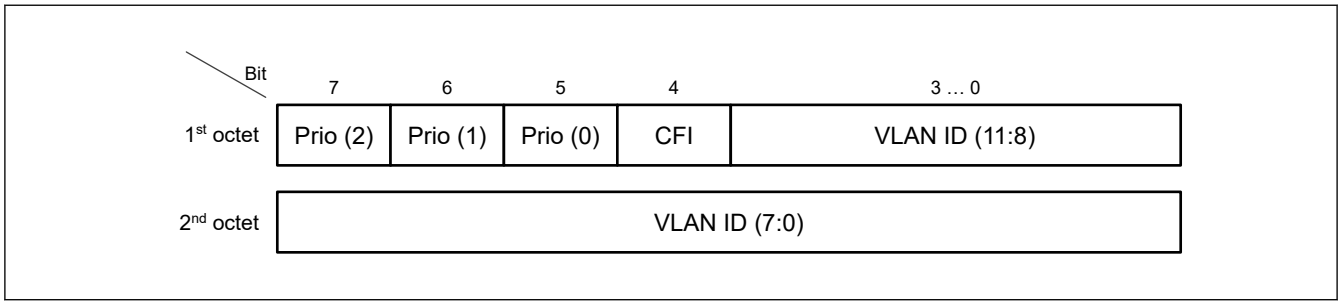


Figure 29.6 VLAN info fields

29.4.1.4 Pause Frames

The receiving device to indicate congestion to the emitting device, which should then stop sending data, generates a pause frame.

The Length/Type set to 0x8808 indicates a pause frame. The first two bytes of a pause frame following the type defines a 16-bit Opcode field set to 0x0001 always. A 16-bit pause quanta is defined in the frame payload byte 2 (byte P1) and byte 3 (byte P2) as defined in Table 29.6. The pause quanta byte P1 is the most significant.

Table 29.6 Pause frame format (values in hex)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
55	55	55	55	55	55	55	D5	01	80	C2	00	00	01
Preamble							SFD	Multicast Destination Address					

15	16	17	18	19	20	21	22	23	24	25	26	27-68
00	00	00	00	00	00	88	08	00	01	hi	lo	00
Source Address						Type		Opcode		P1	P2	Pad (42)

69	70	71	72
xx	xx	xx	xx
CRC-32			

There is no Payload Length field found within a pause frame and a pause frame is always padded with 42 bytes (0x00). If a pause frame with a pause value greater than 0 (XOFF Condition) is received, the MAC stops transmitting data as soon as the current frame transfer is completed. The MAC stops transmitting data for the value defined in pause quanta. One pause quanta fraction refers to 512-bit times.

Pause frames are terminated in the MAC layer and are not forwarded by the switch.

29.4.1.5 Interspersed Traffic Frame Formats

When traffic preemption is enabled, frames are interspersed as express frames are transmitted. The IEEE 802.3br standard defines an mPacket as a segment of a preempted MAC frame, which retains the format of an Ethernet frame and has control information to allow reconstructing the original MAC frame.

The IEEE 802.3br specification describes the two frame formats for traffic interspersing as shown in Figure 29.7. The first format is used for express traffic, and the initial mPacket of a preemptable frame (which can be the whole frame if it does not get preempted). The difference with a regular Ethernet frame is that the SFD is replaced by the SMD (Start mPacket Delimiter). The second format is for mPackets that are a continuation of a preemptable frame (a preemptable frame that was preempted). The SFD and one byte of preamble are replaced by the SMD and a fragment count.

The SMD encodings are defined in the IEEE 802.3br standard as shown in Table 29.7. Express packets use SMD-E, which is identical as the SFD for standard Ethernet frames. Preemptable frames use the SMD-Sx encoding, of which there are four different values. The SMD-Sx is changed for every packet so the receiver can detect a new mPacket start in the case of the loss of the last mPacket of the previous frame. Continuation mPackets use the encodings of SMD-Cx of which there are also four different values to detect the loss of up to three mPackets along with FRAG_COUNT to detect the loss of fragments (see Table 29.8). Finally, SMD-R and SMD-V are used for response and verification frames (see section 29.4.1.6. Preemption Verify and Response Frames).

The CRC of an mPacket is referred as the mCRC and is calculated from the first octet of the full preemptable frame, up to the last octet of the current mPacket. See clause 99.3.6 of the IEE802.3br specification for more details.

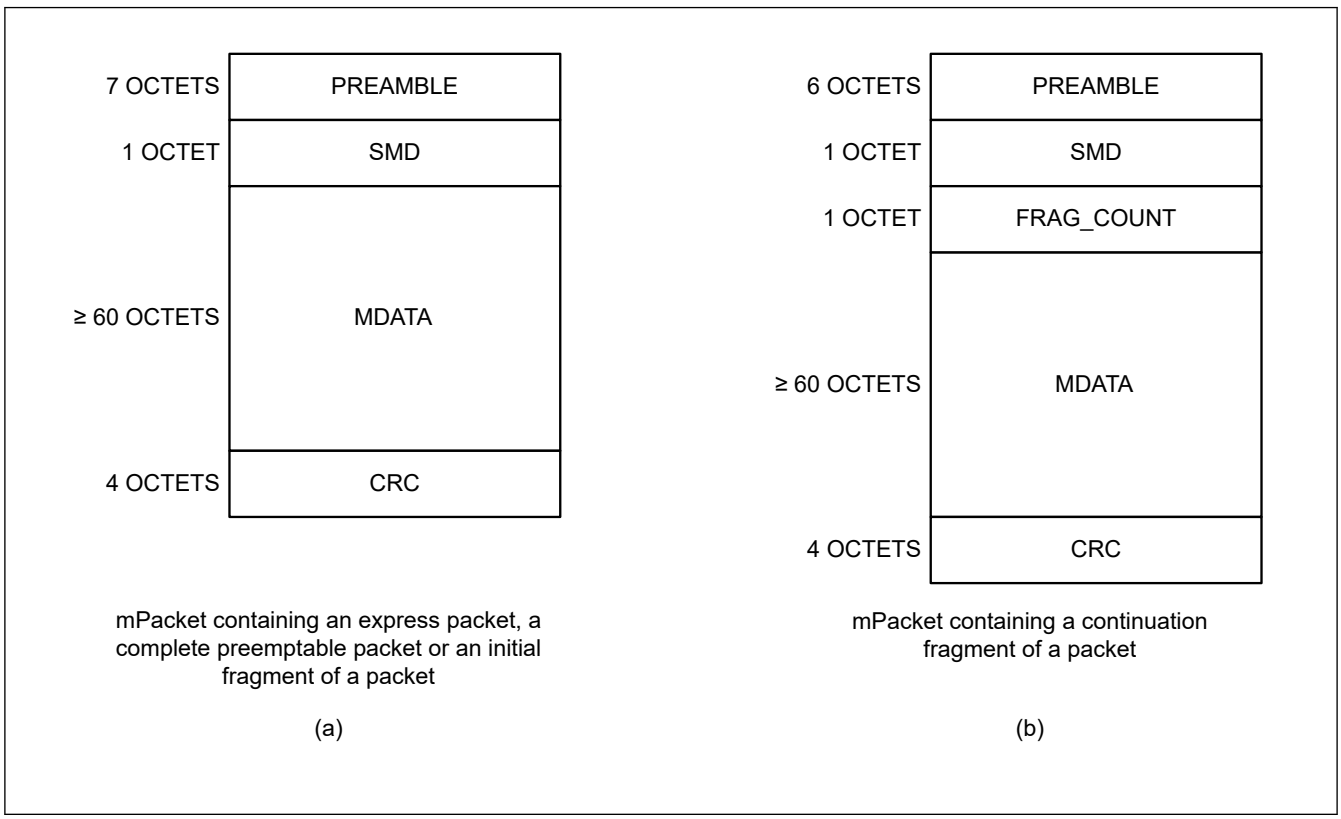


Figure 29.7 Interspersed traffic frame formats

Table 29.7 SMD encoding (1 of 2)

mPacket type	Notation	Frame count	Value
Verify packet	SMD-V	—	0x07
Respond packet	SMD-R	—	0x19
Express packet	SMD-E	—	0xD5

Table 29.7 SMD encoding (2 of 2)

mPacket type	Notation	Frame count	Value
Preemptable packet start	SMD-S0	0	0xE6
	SMD-S1	1	0x4C
	SMD-S2	2	0x7F
	SMD-S3	3	0xB3
Continuation fragment	SMD-C0	0	0x61
	SMD-C1	1	0x52
	SMD-C2	2	0x9E
	SMD-C3	3	0x2A

Table 29.8 FRAG_COUNT values

Frag_count	Value
0	0xE6
1	0x4C
2	0x7F
3	0xB3

29.4.1.6 Preemption Verify and Response Frames

When traffic preemption is enabled, the IEEE 802.3br specification defines to additional frames that are used to verify that both link partners support traffic preemption.

The verify frame is generated when one preemption is enabled by software. This frame is a minimum length Ethernet frame with the SFD replaced by SMD-V and the Data field (mData) consists of 60 bytes of 0x00. The FCS is an mCRC (same FCS but with a different inversion mask).

A response frame is generated in response to a verify frame and its format is the same as a verify frame but using SMD-R instead.

Table 29.9 Verify and response frames

Frame	Preamble (7)	SMD (1)	mData (60)	FCS (4)
Verify	7 × 0x55	SMD-V (0x07)	60 × 0x00	mCRC
Response	7 × 0x55	SMD-R (0x19)	60 × 0x00	mCRC

29.4.1.7 Industrial Automation — TSN Frames

The IEC/IEEE 60802 specification is a joint effort defining the use of TSN for industrial applications. Of interest is the definition of the IA-TSN profile that lists requirements for devices to be able to forward time-sensitive traffic over TSN. The time-sensitive frames are identified using the destination MAC address and VLAN ID. The OUI is a multicast reserved value and the Extension Identifier is used as an index that, together with a VLAN ID, defines a single stream.

Table 29.10 shows the definition of the IA-TSN profile MAC destination address. The OUI value shown is the selected value at the time of writing but can be changed with the control register IALK_OUI. The ID field identifies the stream and it is selected for each stream as the streams are created in the network.

See section 29.4.3.9. Industrial Automation Profile for TSN (TSN-IA-Profile) Forwarding Table for details on the IA-TSN-profile support.

Table 29.10 IA-TSN profile MAC destination address

	OUI			Extension Identifier		
Octet	1	2	3	4	5	6
Value	0x01	0x0E	0xCF	ID[23:16]	ID[15:8]	ID[7:0]

29.4.2 IP Frame Format

29.4.2.1 Definitions

The following sections use the term datagram to describe the protocol specific data unit, which is found within the payload section of its container entity.

For example, an IP datagram specifies the payload section of an Ethernet frame. A TCP datagram specifies the payload section within an IP datagram.

29.4.2.2 Ethernet Types

IP datagrams are carried in the payload section of an Ethernet frame. The Ethernet Type/Length field is used to differentiate several datagram types. Table 29.11 lists the Ethernet types.

Table 29.11 Ethernet type value examples

Type	Description
0x8100	VLAN tagged frame. The actual type is found 4 octets later in the frame
0x0800	IPv4
0x86DD	IPv6
0x8808	MAC control frames (slow protocols, flow control)

29.4.2.3 IPv4 Datagram Format

Figure 29.8 shows the IP Version 4 (IPv4) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words. The most significant bit is bit 31 in the Figure 29.8. The first sent/received byte is the leftmost byte of the first word (for example, Version/IHL field).

The IP Header can contain additional options, which are always padded if necessary to guarantee the payload following the header is aligned to a 32-bit boundary.

The IP header is followed by the payload immediately, which can contain further protocol headers such as TCP or UDP as indicated by the protocol field value. The complete IP datagram is transported in the payload section of an Ethernet frame.

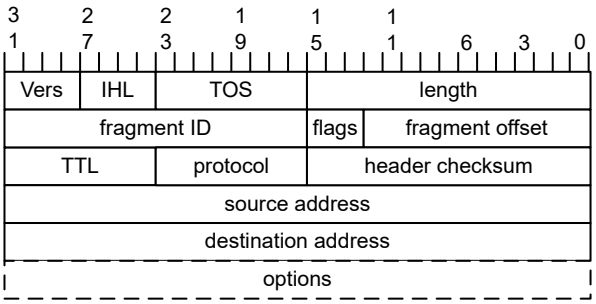


Figure 29.8 IPv4 header format

Table 29.12 IPv4 header fields (1 of 2)

Field Name	Description
Vers	4-bit IP version information. It is 4 for IPv4 frames.
IHL	4-bit IP header length information. Determines number of 32-bit words found within the IP header. The default value, if no options are present is 5.
TOS	Type of Service/DiffServ field
Length	Total length of the datagram in bytes. It includes all octets of header and payload.

Table 29.12 IPv4 header fields (2 of 2)

Field Name	Description
Fragment ID, flags, fragment offset	Fields used for IP fragmentation
TTL	Time-to-live. If 0, datagram must be discarded.
Protocol	Protocol Identifier of protocol that follows in the datagram
Header checksum	Checksum over all IP header fields
Source address	Source IP address
Destination address	Destination IP address

29.4.2.4 IPv6 Datagram Format

Figure 29.9 shows the IP Version 6 (IPv6) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words and has a fixed length of 10 words (40 bytes).

The next header field identifies the type of the header to follow the IPv6 header. It is defined identical to the protocol identifier within IPv4 with new definitions for identifying so-called extension headers, which can be inserted between the IPv6 header and the protocol header, shifting the protocol header accordingly.

The most significant bit is bit 31 in Figure 29.9. The first sent/received byte is the leftmost byte of the first word (for example, Version/Traffic class fields).

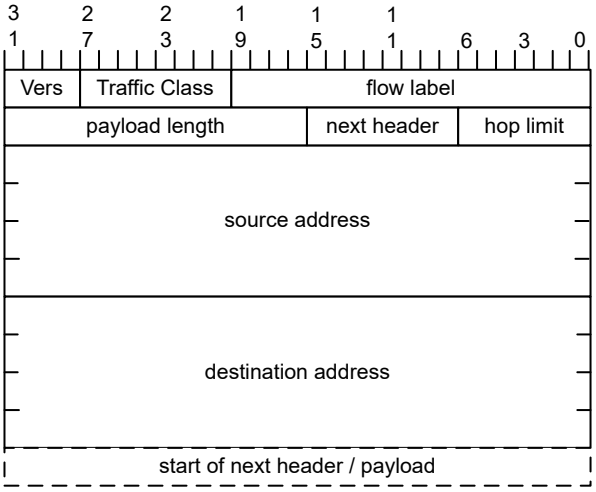


Figure 29.9 IPv6 header format

Table 29.13 IPv6 header fields (1 of 2)

Field Name	Description
Vers	4-bit IP version information. It is 6 for all IPv6 frames.
Traffic Class	8-bit field defining the traffic class.
Flow Label	20-bit flow label identifying frames of the same flow.
Payload Length	16-bit length of the datagram payload in bytes. It includes all octets following the IPv6 header.
Next Header	Identifies the header that follows the IPv6 header. This can be the protocol header or any IPv6 defined extension header.
Hop limit	Hop counter, decremented by 1 by each station that forwards the frame. If hop limit is 0, the frame must be discarded.
Source Address	128-bit IPv6 source address

Table 29.13 IPv6 header fields (2 of 2)

Field Name	Description
Destination Address	128-bit IPv6 destination address

29.4.3 Switch Functional Description

29.4.3.1 Overview

The switch implements the following main functions:

- Input/output VLAN processing
- Input frame parsing and priority extraction
- Input port selection
- Output port resolution
- Frame queuing
- Output queue scheduling

29.4.3.2 VLAN Input Processing Function

(1) Overview

The VLAN input processing function is used on each switch input port to inspect and manipulate the VLAN tag of frames entering the switch. It performs the following functions:

- Input frame parsing
- VLAN tag insertion or manipulation

Based on the information of the input processing function, the frame can be switched to the corresponding output port or is discarded.

VLAN input processing is not performed for BPDU frames.

(2) Terms and Definitions

- VLAN information: The 16-bit field following the VLAN type field within a frame.
- VLAN ID: The lower 12 bits of the VLAN information field.
- VLAN priority: The upper 3 bits of the VLAN information field. Used to prioritize incoming frames. A value of 0 represents lowest priority, a value of 7 represents highest priority.

(3) Configuration Information

The switch management provides the following information to configure and control the operation of the function:

- SYSTEM_TAGINFOn: 16-bit value. The VLAN information field (VLAN ID and priority) used for tag insertion operations.
- Mode of operation: There are different modes of operation, which define how incoming frames must be processed for a port. The function can be enabled and configured individually per port: See registers VLAN_IN_MODE*1 and VLAN_IN_MODE_ENA*2.

Note 1. See section 29.3.1.11. VLAN_IN_MODE : VLAN Input Manipulation Mode Register.
Note 2. See section 29.3.1.13. VLAN_IN_MODE_ENA : VLAN Input Mode Enable Register.

Note: If the VLAN input processing function is not enabled (the port bit of VLAN_IN_MODE_ENA register = 0), the mode setting has no effect.

(4) Modes of Operation

The VLAN input processing function modifies the frames before they enter the switching engine. This means, if a VLAN tag is inserted, the switch only acts on the inserted VLAN tag (priority) and any original tag that was found in the frame before the modification, if any, has no effect within the switch.

In addition, if VLAN verification is enabled for a port (see section 29.3.1.5. VLAN_VERIFY : Verify VLAN Domain Register), the VLAN ID used for insertion (SYSTEM_TAGINFOn) must also be configured in the global VLAN resolution table (see section 29.3.1.67. VLAN_RES_TABLEn : 32 VLAN Domain Entries (n = 0 to 31)), to ensure the switch accepts frames, which contain the inserted tag.

When, in any of the modes, a tag is inserted, it is always inserted as first tag (outer) and its information field is set as programmed in the SYSTEM_TAGINFOn register for the port n where the frame is received.

Input manipulation can be enabled per port with register VLAN_IN_MODE_ENA and its port individual mode is configured in register VLAN_IN_MODE.

Table 29.14 Input manipulation modes

Mode	VLAN_IN_MODE bits [1:0]	Description
1	00b	Single Tagging with Passthrough/VLAN ID (VID) Overwrite. Insert Tag if untagged frame. Leave frame unmodified if tagged and VID > 0. If tagged with VID = 0 (priority tagged), then the VID is overwritten with the VID from SYSTEM_TAGINFO and priority is kept.
2	01b	Single Tagging with Replace If untagged, add the tag, if single tagged, overwrite the tag.
3	10b	Tag Always Insert a tag always. This results in a single tagged frame when an untagged is received, and a double tagged frame, when a single tagged frame is received (or triple tagged if double-tagged received).
4	11b	Reserved Mode is not implemented and should not be configured.

(5) VLAN PCP remap

The VLAN.PCP field can be remapped by the switch on ingress, regardless of whether VLAN input manipulation being enabled or the VLAN mode. This function is only available for the line ports (i.e. not for the management ports). The remapping also applies to the VLAN using the EtherType 0x8100 (0x8808 is not supported), and only for the outmost VLAN matching this EtherType.

The PCP is remapped before priority resolution is performed which means that the VLAN Priority Lookup function, section 29.4.3.5. Frame Classification and Priority Resolution (2) VLAN Priority Look Up, uses the remapped PCP value.

The PCP remapping is configurable per port via the control register PRIORITY_CFGn.PCP_REMAP. The configuration is a simple remap where each PCP value from 0 to 7 can be mapped to a new value between 0 to 7. The default configuration is to not do any remapping.

29.4.3.3 VLAN Output Processing Function

(1) Overview

The VLAN output processing function is used on a switch output port to manipulate the VLAN tag of the outgoing frames that leave the switch. Frames are processed based on the output processing mode, and the number of tags the frame contains.

VLAN output processing is not performed on BPDU frames.

(2) Configuration Information

The switch management provides the information on operating mode to configure and control the operation of the function. There are three different modes of operation, which define how the outgoing frames should be processed.

(3) Modes of Operation

The VLAN output processing function is configured to operate in one of the following modes, which define the way outgoing frames should be treated.

(a) Mode 0: Disabled

No frame manipulation occurs.

(b) Mode 1: Strip Mode

In Strip mode, all the tags (single or double) are removed from frame.

(c) Mode 2: Tag Through Mode

Always removes first tag from frame only. In Tag Through mode, the inner tag is passed through while the outer tag is removed for a double tagged frame. The following rules apply:

- When a single tagged frame is received, strip the tag from the frame
- When a double tagged frame is received, strip the outer tag from the frame

(d) Mode 3: VLAN Domain Mode / Transparent Mode

The following function is implemented: VLAN Domain Mode: The first tag of a frame is removed (same as Mode 2) when the VLAN is defined as untagged for the port. The following rules apply:

- If VLAN ID of the frame is found in the VLAN table (see (b) VLAN Domain Resolution / VLAN Table) and the port is defined as tagged for the VLAN, the frame is not modified.
- If VLAN ID of the frame is found in the VLAN table and the port is defined as untagged for the VLAN, the first VLAN tag is removed from the frame.
- If VLAN ID of the frame is not found in the VLAN table, the frame is not modified.

Note: The VLAN table is extended by a 2nd port mask to store the tagged bit for each entry of every port (in addition to the member port mask). This tagged bit mask is accessed using the control bits [30:28] when writing the VLAN_RES_TABLEn registers.

29.4.3.4 Frame Snooping

(1) Overview

To allow inspecting specific frames of different protocols for management purposes, a programmable generic frame parsing module is implemented that allows marking a frame for snooping. Snooping allows several functions such as a copy is sent to the management port for inspection while the frame is forwarded normally.

The generic approach allows searching for multiple arbitrary length patterns within a frame to identify (match) a frame for snooping. The following is a non-exhaustive list of examples how the generic parsing can be used:

- Mark frames with specific Ethernet type
- Mark frames with specific payload contents
- Mark IP frames with any IP protocol and sub protocol (extension headers) values
- Mark TCP/IP or UDP/IP frames with specific port numbers

When a frame is marked for snooping, it can be forwarded to the management port exclusively or as a copy, or become discarded.

If none of the given rules match, the frame is processed normally. If a frame is coming from the management port itself, the snooping rules are ignored (to avoid the frame be routed back to the management port again). However, the management port can use forced forwarding mechanisms to direct any frame to specific output ports if needed.

The function is configured using registers GPARSERn and GARITHn, and the management port is defined by register MGMT_CONFIG.

(2) Snooping Dataflow Description

Figure 29.10 shows the functions involved to mark a frame for snooping. Every frame is inspected, in parallel, with up to 8 individually programmable parsers. These parsers allow comparing a specific pattern from 1 to 16 bits at a certain position within the frame and report a match if the pattern is found.

The match result of each individual parser is then provided to an arithmetic block, which can perform Boolean operations on the result from up to 4 parsers to form a final match decision. Two of such arithmetic modules are available and independently programmable.

When any of the arithmetic blocks reports a match, the final snooping decision for the frame is provided to the switch forwarding.

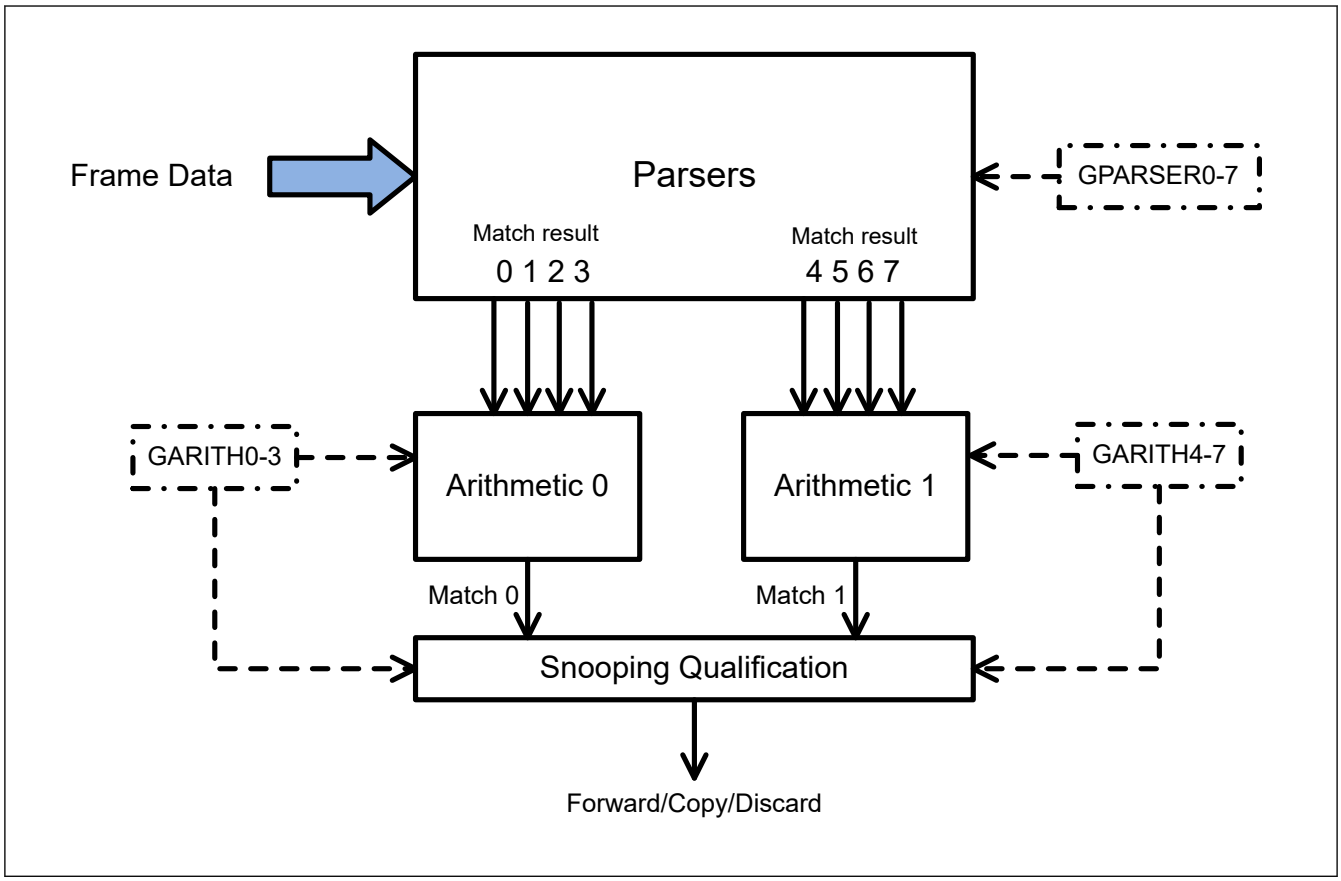


Figure 29.10 Snooping function overview

(3) Generic Parser

Each individual parser can inspect a frame as programmed by register GPARSERn (n = 0 to 7). Figure 29.11 shows the interpretation of the individual settings.

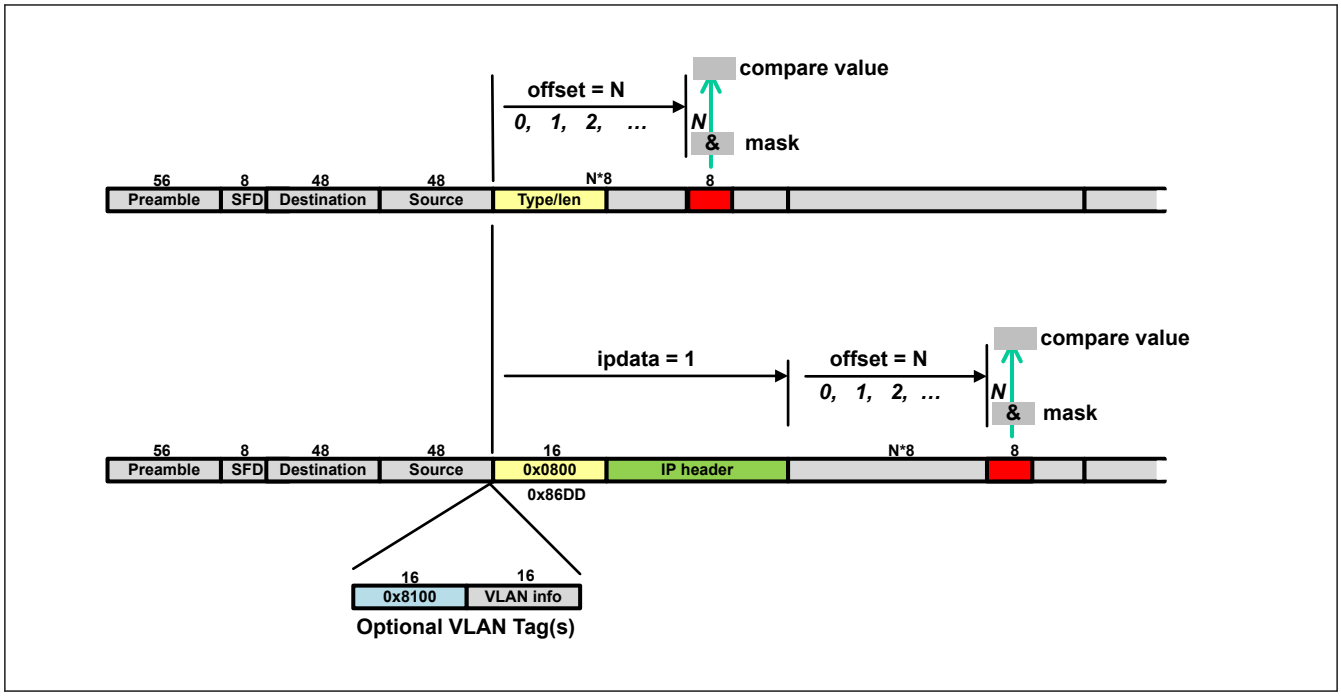


Figure 29.11 Generic parser configuration parameters

The parser acts as follows:

1. It extracts the bytes at the given offset of the frame. The offset can either be specified starting after the MAC source address, starting after any optional VLAN tags, or starting after an IP header.
2. It then applies a bitmask (AND) to the byte to mask out bits that should be ignored.
3. It then compares the result with the given pattern (compare value). If the pattern matches, a match indication is produced.

Instead of the mask & compare function, additional compare options allow to compare against a 16-bit value (for type comparisons) or alternatively (OR) two different byte values instead.

(4) Parser Limitations

The parser when inspecting frames has limitations. If a limit is reached, the parser is ignored and the frame is forwarded normally as if no parser was active (treated as a no match).

The limitations are as follows:

- The bytes that can be inspected can be at maximum at position 67 after the first byte of the destination address of the frame. If the offset together with, for example, VLAN tags falls beyond this limit, the snooping may not have an effect on the frame.
- The bytes within the last 8 bytes of a frame cannot be inspected. That is, if the last byte of the CRC field is 8 bytes or less after the byte to compare.

(5) Arithmetic Function

The match result of four parsers is connected to an arithmetic module combining the results of the parsers to a final match decision.

The arithmetic function implements 4 stages (arith 0, 1, 2, 3 as shown in Figure 29.12) and Boolean operations at every stage allow to implement complex combinations of the parser results. At each stage, the following can be configured:

- Select parser match result (match) or its inverted (no match)
- Select a result of a previous stage (only for the last 3 stages, arith 1, 2, 3)
- Define if the selected inputs should be ORed or ANDed and if the result should be inverted

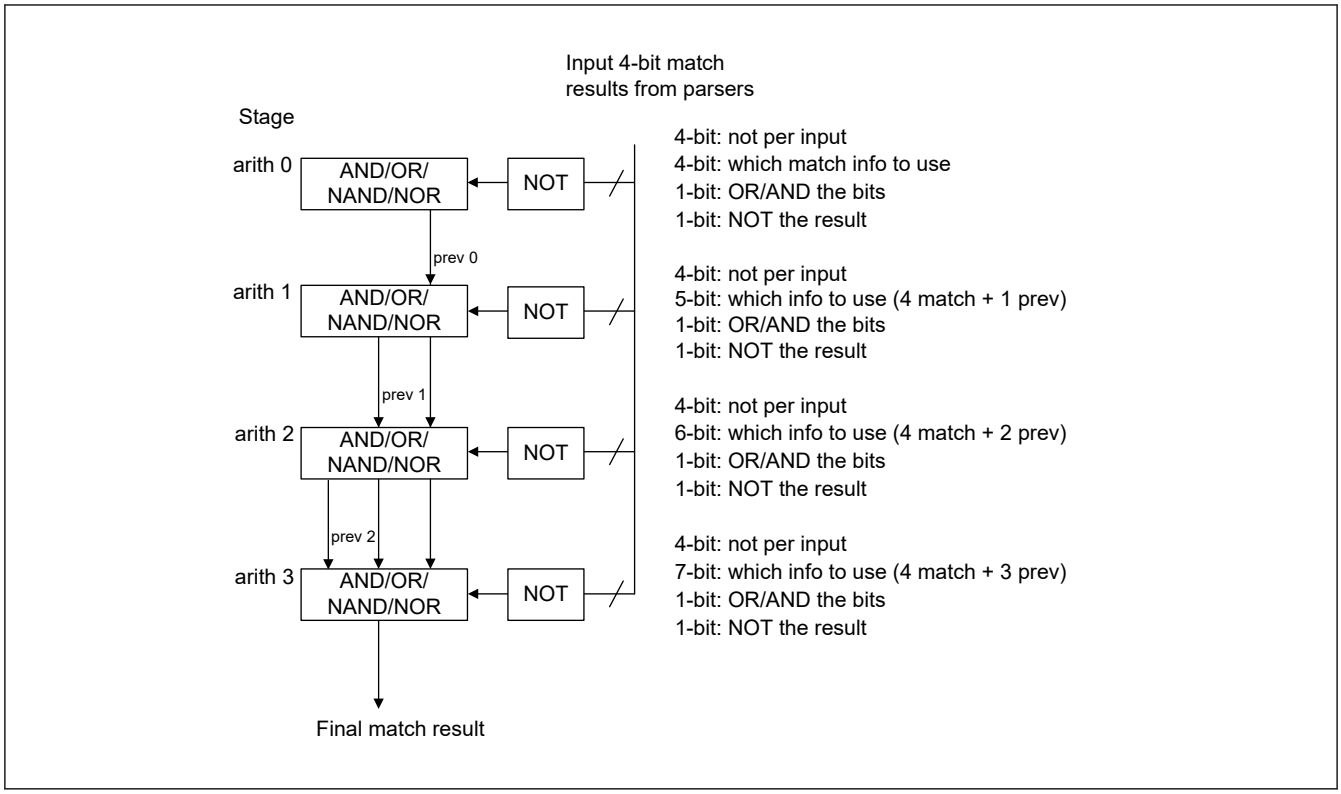


Figure 29.12 Snooping arithmetic function

The arithmetic function for the first 4 parsers (0 to 3) is configured in registers GARITH0-3. The arithmetic function of the next 4 parsers (4 to 7) is configured in registers GARITH4-7.

The configuration of the last stage (arith 3, register GARITH3 or GARITH7) also includes the configuration bits for the snooping function to define if the frame should be forwarded to management or copied or discarded eventually.

(6) Snooping Qualification

The last stage following the arithmetic modules qualifies the results from both arithmetic functions to provide the final decision to the switch how it should process the frame:

1. Forward it to the management port only.
2. Forward it normally and also copy it to the management port.
3. Discard it.

If both arithmetic functions match for the same frame but have different snooping actions configured, the action with the lower number is executed (forward to management port only before copy or before discard).

29.4.3.5 Frame Classification and Priority Resolution

(1) Overview

When a frame is received on an input port, several information are extracted from the frame as the Ethernet MAC Address, VLAN tag, and IP headers to determine the frame type and perform the relevant classification actions.

In addition, the MAC address table can provide a priority indication for the destination MAC address if the switch management software has programmed the address table accordingly (static entries).

The frame is classified as a high priority or as a low priority frame and is eventually queued in a corresponding priority queue at the output ports. A higher queue number is given a higher priority.

(2) VLAN Priority Look Up

On each port, an 8-entry programmable priority table is implemented. The registers VLAN_PRIORITYn contain the priority mapping for a port n. The switch uses 3 bits from the VLAN tag information (3 bit VLAN priority, see section

29.4.1.3. VLAN Tag Overview) to extract the corresponding bits from the table, that indicates which priority the frame should be finally classified.

The index in the mapping table is the 3 bits of the first octet of the VLAN info with bit 5 (Prio (0)) being the LSB and bit 7 (Prio (2)) being the MSB (see section 29.4.1.3. VLAN Tag Overview for VLAN tag details).

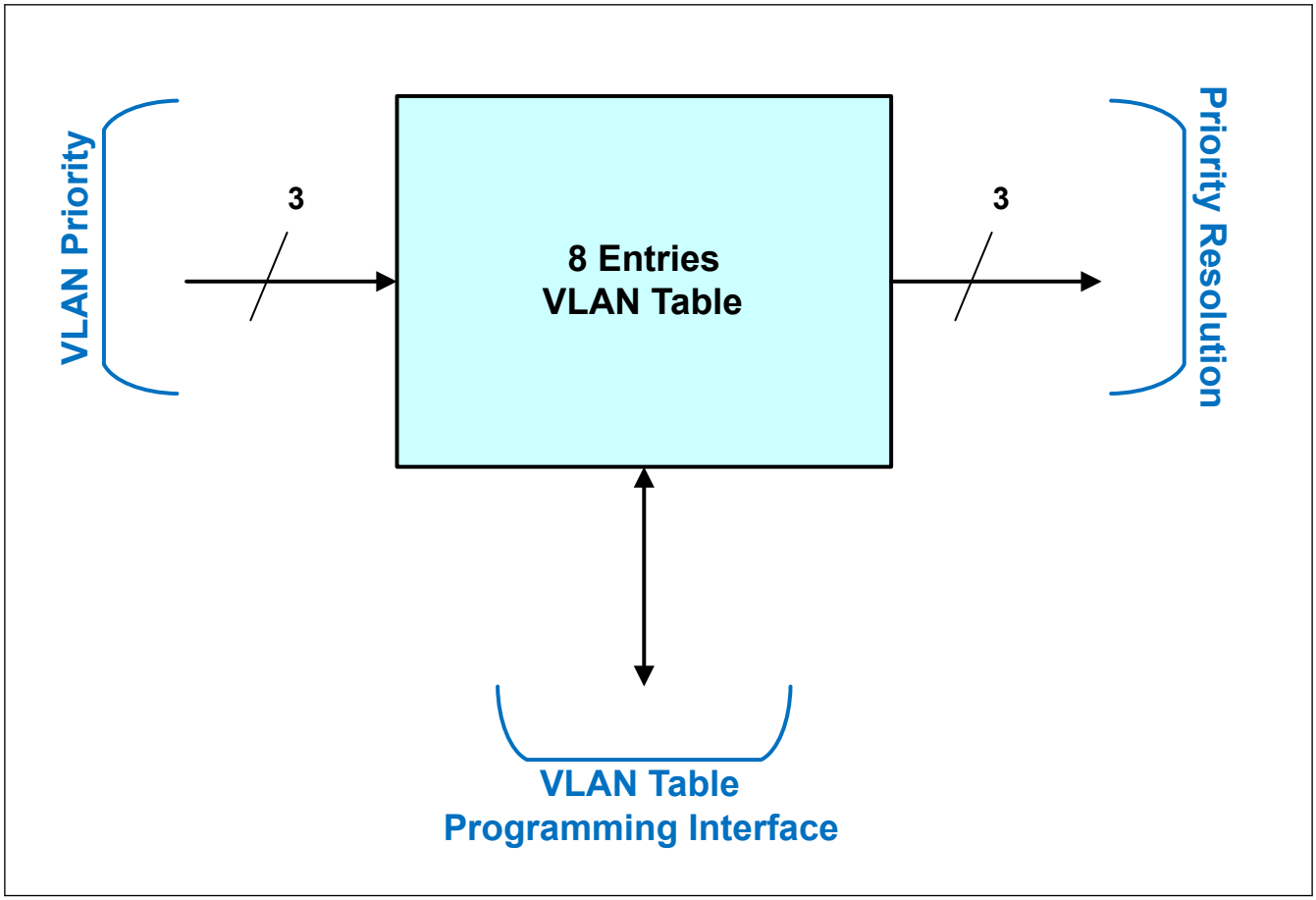


Figure 29.13 VLAN table overview

(3) IPv4 and IPv6 Priority Look Up

The switch can classify both IPv4 and IPv6 frames: A 64-entry table is implemented per port to classify the IPv4 Frames and a 256-entry table is implemented per port to classify IPv6 Frames (the IP classify tables) (see section 29.3.1.48. IP_PRIORITYn : IP Priority Register n (n = 0 to 3)).

On the IPv4 classify table entry, the 6-bit DiffServ field of the frame is provided and the table returns the 3-bit priority information.

On the IPv6 classify table entry, the 8-bit Traffic Class field is provided and the table returns the 3-bit priority information.

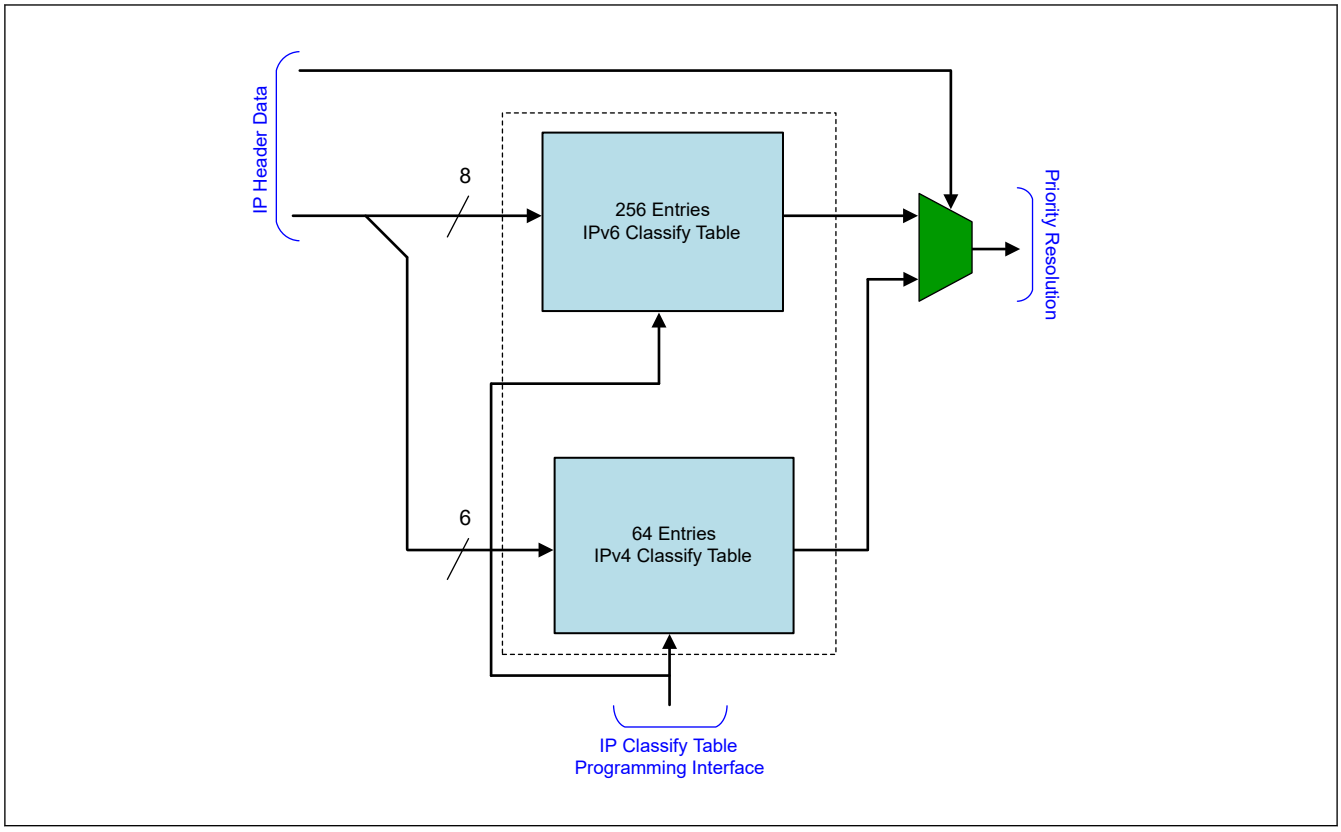


Figure 29.14 IP classify tables overview

(4) Priority Resolution

The priority resolution function, for a Port n, is, on each port independently, programmable with the registers PRIORITY_CFGn to enable or disable VLAN, IP, MAC address, or type based classification.

In addition, the optional Receive Pattern Matcher (see section 29.4.14. Receive Pattern Matcher) can be used to override the classification.

The priority resolution implements the following ruleset (and precedence) depending on which classifications are enabled (PRIORITY_CFGn) and which fields are found within the frame:

- If the Pattern Match classification is enabled (PATTERN_CTRL.SET_PRIO) and a match occurs, the defined priority from the PATTERN_CTRL register is used, ignoring the settings in PRIORITY_CFGn
- Otherwise, if IP classification is enabled and IP header found → map priority according to IP_PRIORITY table
- Otherwise, if VLAN classification is enabled and VLAN tag found → map priority according to VLAN_PRIORITY table
- Otherwise, if MAC classification is enabled and MAC address found → take priority from address table
- Otherwise, if TYPE classification is enabled and the frame type (following any VLAN tags) matches an entry of the PRIORITY_TYPE defined values, use the priority from that matching PRIORITY_TYPE
- Otherwise, if it is a BPDU frame transmitted by the management port, and the priority setting in the register MGMT_CONFIG is non-zero, this priority is used
- Otherwise, use default priority as specified in PRIORITY_CFG

(5) Bridge Control Protocol Identification

To allow for implementation of bridge control protocols like the Spanning tree protocol, an input module is implemented that marks all management frames (Bridge Protocol Data Units, BPDU) when they enter the switch. The mark then can be used by the input port blocking function (see (b) Input Port Blocking) to drop the frame after the address learning.

In addition, the function can be configured to pass all frames or to pass only management frames (covering spanning tree port states such as blocking, listening, learning) and discard all other frames.

29.4.3.6 Input Port Selection

The port selection constantly checks (polling) all input ports for available data and if any data is available, a port is selected and frame data is read from the input. All inputs are served simultaneously using a multiplexed access for storing the frame data into the shared memory. No receive FIFO exists allowing for very low forwarding delays as no copying from a receive FIFO into an output queue is needed.

29.4.3.7 Layer 2 Look Up Engine

(1) Overview

The switch is using the lookup memory to perform address lookups and determine to which port a frame should be forwarded to. By using a port mask, it is possible to forward a frame to one or many destination ports (for example, multicast). For instance, software can add (static) multicast addresses and define that traffic is forwarded only to the ports that have subscribed to the multicast domain.

In addition to the destination lookup, the source address of each incoming frame is searched, and if found, the corresponding aging bit of an entry (dynamic record) becomes set. The aging bit is used by the aging process to determine if an entry should be removed from the memory. The aging process periodically clears the aging bit of each entry and verifies if the bit is set again after some time. If the bit is not set, the entry is considered to be outdated and can be removed from the lookup memory.

(2) Hash Based Lookup

A hash code is calculated using the frame destination MAC address. It is used as an entry (address) to a table, which contains, for each hash value MAC addresses with destination port mask and validity information.

As one hash code value can represent more than one MAC address, the memory implements for each hash value, up to eight MAC address entries, which are searched linearly.

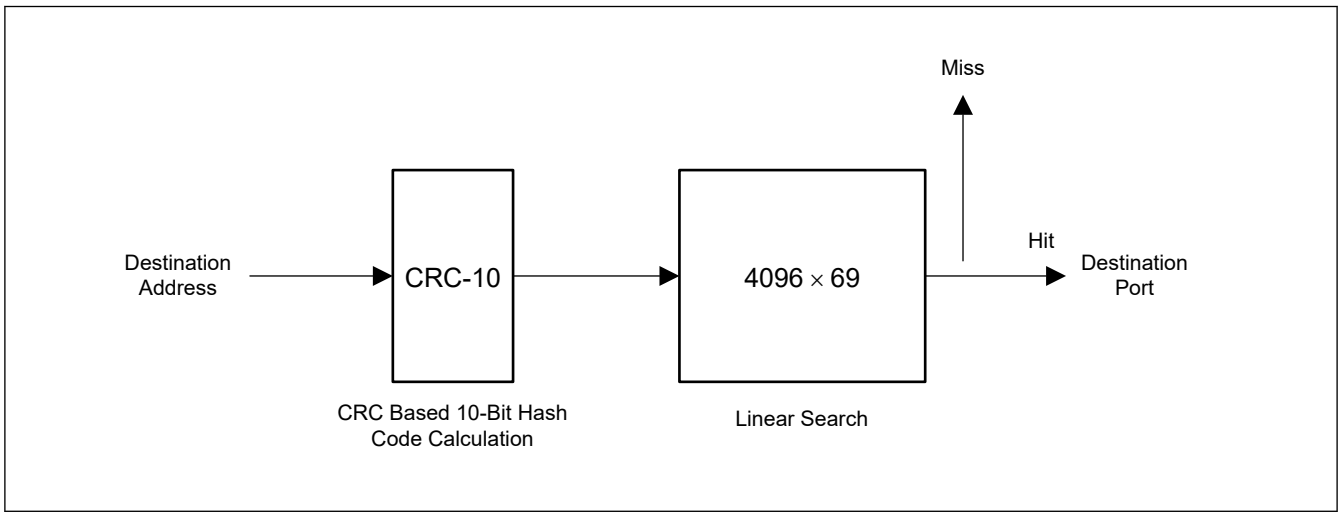


Figure 29.15 Port look-up overview

(3) Hash Code

For MAC address table 4096 entries, a 10-bit hash value is calculated from the 48-bit MAC address. The hash code is using a CRC-10:

$x^{10} + x^9 + x^5 + x^4 + x + 1$ (0x233)

(4) Address Memory

The address memory is divided into blocks. Each block contains eight records, which contain up to 69-bit of information each. Each record contains the 48-bit MAC address (first byte in bits [7:0]), VLAN number if IVL is supported, and provides the necessary forwarding information as well as priority or aging information. Figure 29.16 and Figure 29.17 show the physical format of each record. See section 29.3.1.83. LK_CTRL : Learning/Lookup Function Global Configuration Register for details on how to manually program individual records.

Two types of records are defined:

- Dynamic Record:
  The dynamic entry provides the MAC address together with destination port number and aging information. These entries are created by the learning function based on received frames. Dynamic entries are deleted by the aging function if not updated.
- Static Multiport/Priority Record:
  Switch Management can also write static entries in the table, which can include priority information as well as multiple destination ports for forwarding. The MAC address can be unicast or multicast. These records can be used to specify the ports to participate in a specific multicast domain or to assign a MAC address based priority to a frame. The aging and learning functions ignore static records.

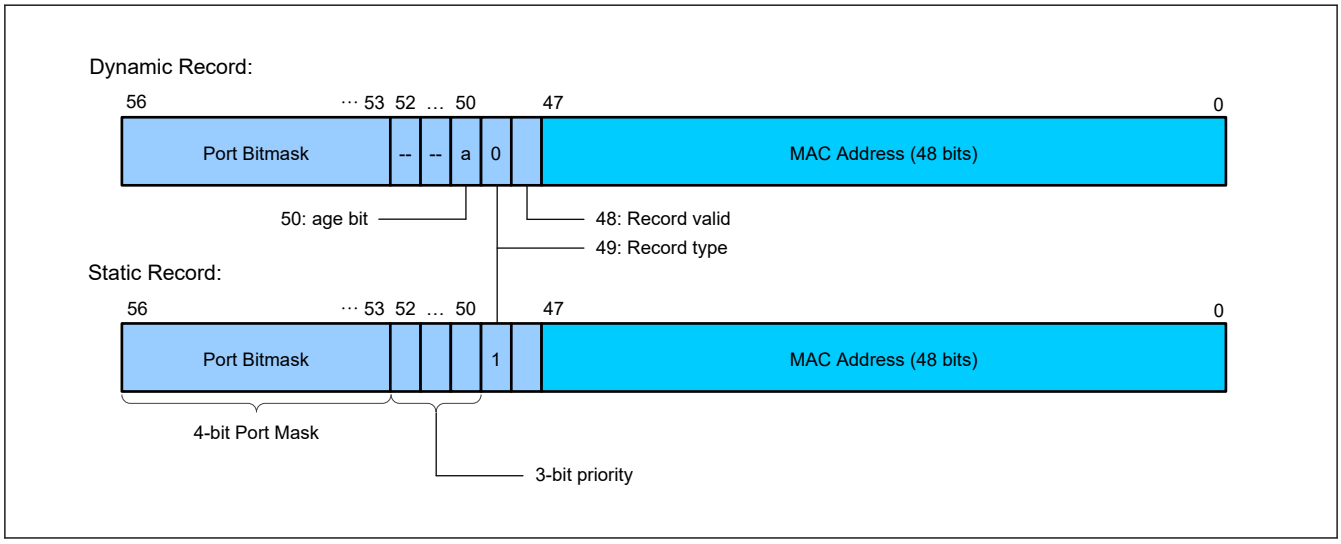


Figure 29.16 Address memory record types (IVL support disabled)

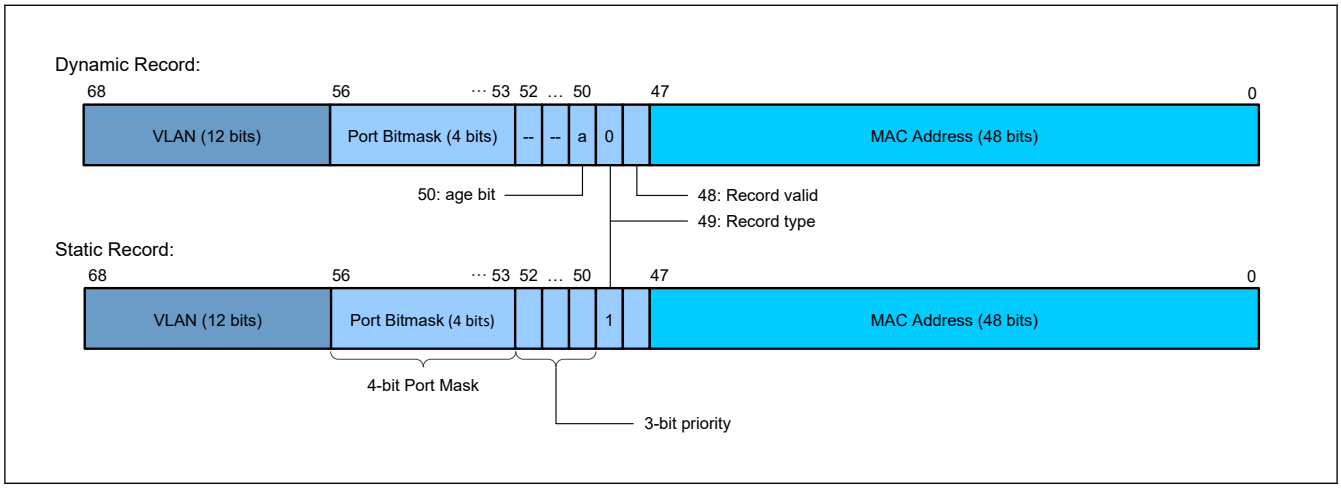


Figure 29.17 Address memory record types (IVL support enabled)

Bit 49 of the record determines which type of record is found in the table:

- If the bit is 0, a dynamic entry is available and bit 50 holds the aging bit. Bits [56:53] indicate at which port the address has been received.
- If the bit is 1, a static entry is available with a 3-bit priority field followed by the port bit mask.

Bit 53 of the record represents port 0, bit 54 port 1, and so on. The frame is forwarded to all ports that have a value of 1 in the port bitmask. The source port is removed dynamically from the bitmask during forwarding (a frame is not forwarded to the port where it came from).

29.4.3.8 Layer 2 Look Up Engine Operational Description

(1) General

The host CPU can be used to configure the lookup options. The switch performs self initialization of the address table after reset for normal operation. Lookup and Learning are performed fully autonomously by the hardware without processor intervention.

It should be noted, that all address table memory transactions from the processor consume time that is lost for the hardware to perform lookups required for frame forwarding. It influences the overall switch performance and therefore should be reduced to a minimum during operation.

(2) Memory Write

Writing to memory is performed in two steps.

1. The LK_DATA_LO/HI/HI2 registers are set with the data to write.
2. The LK_ADDR_CTRL register is written to start the transaction.

When the transaction is started (by writing into LK_ADDR_CTRL), the interface controller requires some time until the transaction is completed. Software can either poll the BUSY bit within the LK_ADDR_CTRL register or set the WAIT_COMP bit when writing to the register. Using the WAIT_COMP bit ensures that the CPU cannot access the registers until the transaction is completed (the processor bus is held in wait, freezing processor activity).

(3) Memory Read

Reading from the memory is performed in two steps similar to writing. However, first, the transaction is executed and the data is delivered to the LK_DATA_LO/HI/HI2 registers after completion.

1. The register LK_ADDR_CTRL is written to start a READ transaction.
2. On completion of the transaction, data is available in LK_DATA_LO/HI/HI2.

(4) Power on Initialization

After power on reset, the switch performs self initialization of the lookup table functions and enables automatic learning. The host CPU may perform additional configuration or add static table entries as needed. The learning functionality and other options are configured with the Learning/Lookup Function Global Configuration Register (LK_CTRL).

(5) Destination MAC Address Lookup

The 48-bit destination MAC address of each frame is extracted. If the frame is a unicast or multicast frame (see section 29.4.1.2. MAC Address Overview for definition) the look up function can provide three results with three different associated actions performed:

1. The address is found in the table:
   The switch forwards the frame to the ports provided by the port mask of the entry.
2. The address is in the table but is associated with the port on which it was received:
   The switch discards the frame and does not forward it to any port.
3. The address is not found in the table:
   The switch engine sends the received frame depending on the type of destination address to all ports (flooding) listed in:
   - UCAST_DEFAULT_MASK0/UCAST_DEFAULT_MASK1 for unicast frames
   - MCAST_DEFAULT_MASK0/MCAST_DEFAULT_MASK1 for multicast frames

Broadcast frames are always sent to all ports listed in the BCAST_DEFAULT_MASK0/BCAST_DEFAULT_MASK1. There are two sets of unicast, multicast, and broadcast default masks. Each mask can be used by a different set of ports by configuring the PORT_XCAST_MASK_SEL register.

Setting any of the default masks to all 0 would lead to discard of the frame.

In all cases, the port, where the frame was received from is automatically removed from the final forwarding port mask.

Note: Flooding and additional frame filtering can be further controlled with the mechanism described in (3) Broadcast / Multicast / VLAN Domain Resolution.

(6) Automatic Learning / Migration

The hardware allows performing automatic learning of source addresses of incoming frames. When a frame is received from any port, its source address is searched in the lookup memory in addition to the destination address lookup described before.

If learning is enabled (see section 29.3.1.83. LK_CTRL : Learning/Lookup Function Global Configuration Register), the following tasks are performed:

- Checks the INPUT_LEARN_BLOCK register. If learning is disabled for a port, the source address of incoming frames is ignored.
- If the frame is marked as a management frame (BPDU, see (7) Bridge Protocol Frame Resolution), the source address is never learned (independent of INPUT_LEARN_BLOCK).
- If the source lookup was successful, the port mask is verified to match the incoming port number where the frame was received. If a mismatch occurs, the forwarding port mask is updated accordingly (migration), but only if the entry is a dynamic entry. Static entries are not changed.
- If the source lookup failed, the new address is added to the lookup memory (if allowed by configuration in LK_CTRL). If there is no more space in the table, the new entry replaces an arbitrary entry with an identical matching hash value (i.e. one of the 8 entries for the hash is deleted).
- The aging bit of the entry is set. The aging function continuously scans the table and clears the bit to detect outdated entries.
- An interrupt is generated whenever a source address is not found in the table (this is independent from learning having entered an address or not). See section 29.3.1.84. LK_STATUS : Status Bits and Table Overflow Counter Register.

If automatic learning is disabled, no inspection of the frame source address is performed. It is then the responsibility of the software to place the addresses in the lookup memory (static configuration). The learning interrupt still indicates when new addresses arrive.

The management frames (BPDU, see (7) Bridge Protocol Frame Resolution) are never learned, and it is the responsibility of the software to add a source address of such frames if necessary.

(7) Aging

Aging refers to deleting old entries within the table. It processes as follows:

1. For each entry, an aging bit is implemented. The bit is set whenever the source address is found within a received frame.
2. When the aging process (timer) inspects an entry, it resets the aging bit to 0. If it finds an entry with the aging bit already being 0, the entry is deleted.
3. Static entries are not affected by the aging process and kept always.

This process runs continuously in the background processing one entry at a time. The aging period is programmable (see section 29.3.1.90. LK_AGETIME : Period of the Aging Timer), giving a range from seconds to several minutes. Aging can also be disabled completely under software control.

(8) Independent VLAN Learning (IVL) Mode

The default switch operation is to perform lookup and learning procedures on MAC addresses only. That is, MAC addresses are shared over all VLANs, if any (Shared VLAN Learning, SVL Bridge).

If independent VLAN learning is enabled (LK_CTRL), it manages MAC addresses together with VLAN identifiers, allowing identical MAC addresses in multiple VLANs.

A VLAN identifier of all 0 means that there is no VLAN associated with the entry (the null VLAN identifier is defined in IEEE 802.1Q Clause 9.6 as a wildcard) and the MAC address is therefore allowed in any VLAN.

(a) Lookup Processing

The lookup process searches the address table for matching MAC addresses.

The configured SYSTEM_TAGINFOn of the port is used as the VLAN ID when an untagged frame, or a priority tagged (VID = 0) is received. The use of the SYSTEM_TAGINFOn setting always applies and is independent from any VLAN input manipulation settings (see section 29.4.3.2. VLAN Input Processing Function).

If the MAC address is found, the VLAN ID of the entry is further inspected and considered as follows:

Table 29.15 Lookup VLAN match decisions

VLAN ID in frame or SYSTEM_TAGINFOn	VLAN ID in table	Match decision
0/untagged	Any	Matches (shared MAC address)
Any	0	Matches (shared MAC address)
!0	!0	Matches only if VLAN id matches (independent VLAN)

Note: If the SYSTEM_TAGINFOn has the null VLAN ID (default) and if multiple table entries for the same MAC address with different VLAN IDs are available, any of these will be matching (which of them is arbitrary and is not predictable).

(b) Learning

When the lookup of the frame source address does not match any entry as previously described, the MAC address is inserted into the table with:

- Untagged frame: The VLAN ID of the entry is set to the value programmed in SYSTEM_TAGINFOn of the port.
- Priority tagged frame (VLAN ID = 0): The VLAN ID of the entry is set to the value programmed in SYSTEM_TAGINFOn of the port.
- Tagged frame: The VLAN ID of the frame is stored.

Note: This use of the SYSTEM_TAGINFOn setting always applies and is independent from any VLAN input manipulation settings (see section 29.4.3.2. VLAN Input Processing Function).

29.4.3.9 Industrial Automation Profile for TSN (TSN-IA-Profile) Forwarding Table

This is a TSN profile designed for applications in industrial environments where time-sensitive streams must have a controlled latency through the network. It is defined by the IEC/IEEE 60802 joint taskforce. Supports the number of required streams (8 K at the time of writing) without increasing the size of the hash-based forwarding database.

(1) General

The IA-TSN-profile requires the switch to be capable of forwarding 8 K or more distinct streams in addition to at least supporting 2 K or more distinct MAC addresses for non-IA traffic. The IA-TSN-profile identifies the different streams by the MAC destination address (see section 29.4.1.7. Industrial Automation — TSN Frames) and up to 4 distinct VLAN IDs.

Supporting these many streams by solely using a hash-based forwarding database poses a challenge because the inherent collisions which limits the maximum number of usable entries in the table. The IA forwarding database (IA-FDB) uses a direct-index address based on the frame's ID and the VLAN ID to allow a packed storage of all the streams, relieving the hash-based FDB from this task.

The IA-FDB is disabled by default and does not have an automatic learning function like the L2 FDB. Therefore, it must be enabled by firmware on a per-port basis and be manually configured as streams are created in the network.

When the IA-FDB is enabled, the port initiates an L2 lookup in both the hash-based FDB and the IA-FDB, and uses the result from the IA-FDB if the lookup results in a found entry in the table such as a hit.

(2) Forwarding Flow Using IA-FDB

As mentioned before, the IA-FDB is used for storing the forwarding information for IA-TSN frames. The IA-FDB is enabled per port using the IALK_CONTROL.IA_LKUP_ENA control field. Figure 29.18 shows the decision process when using the IA-FDB as part of the lookup process. The IA-FDB lookup is executed in parallel to the L2 hash-based FDB. The L2 hash table returns a found (hit) flag if the entry is found in the L2 FDB, along with a port mask. The IA-FDB also returns a found flag with a corresponding port mask and a learn_ok flag that is used to inhibit learning for this frame.

The found flags are used to determine the result, with the IA-FDB having priority over the L2 FDB. When none of the tables return a hit indication the frame is flooded based on it being unicast or multicast.

Note: The port also issues a look up requests to the IA-FDB under the same conditions that it would for a destination lookup to the L2 FDB. This means that BPDU, broadcast, and frames from the management port with a force forward request do not generate a request. Also, the result is also subject to further filtering for exmample, same-port filtering (the source port is excluded from the destination port mask), VLAN domain verification, port mirroring, snooping, and pattern matcher modifications.

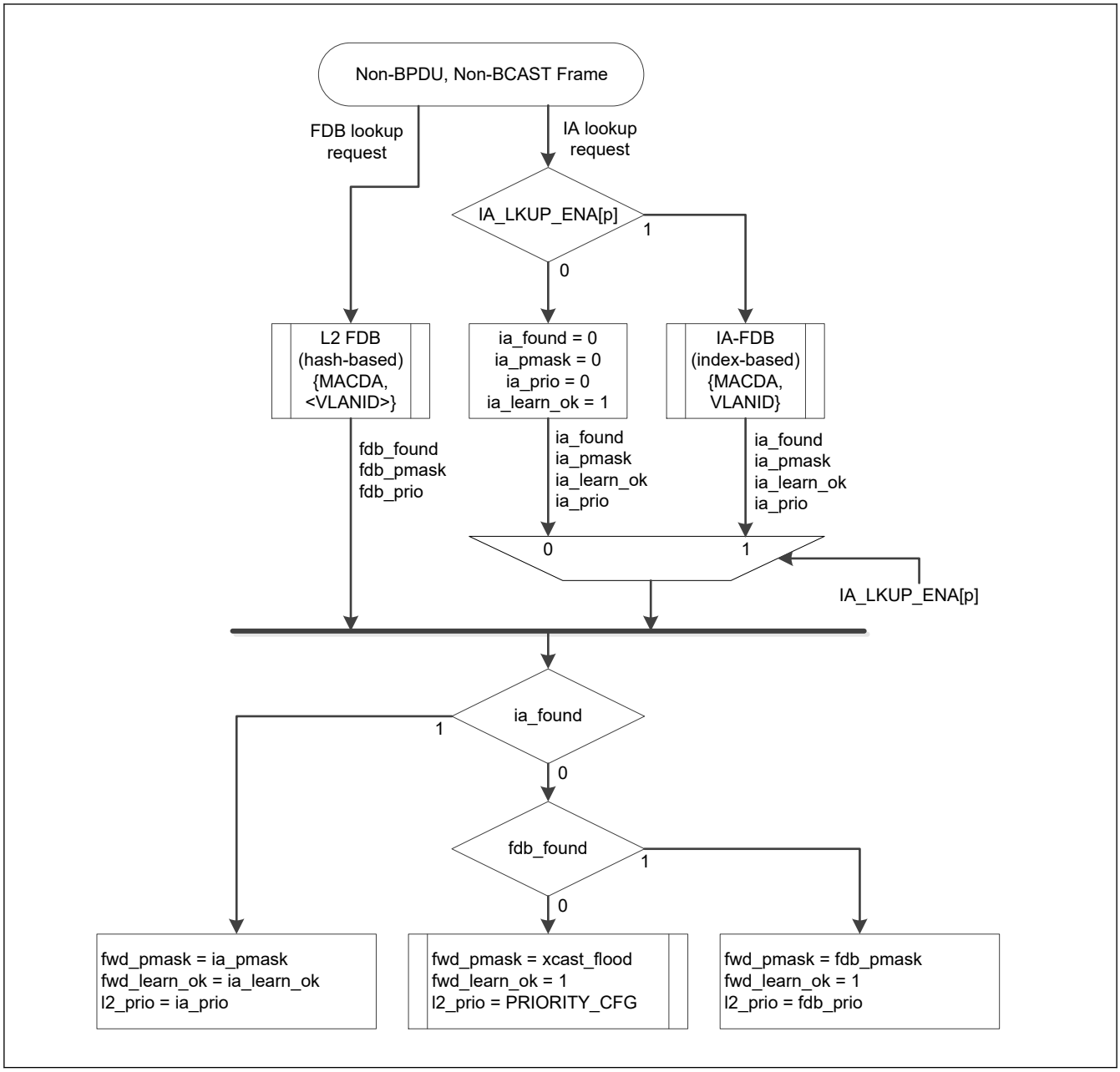


Figure 29.18 Forwarding flow using IA-FDB

(3) IA-FDB Lookup Flow

Figure 29.19 shows the look up process executed by the IA-FDB table for every port request. The flow is controlled MAC destination address of the frame (see section 29.4.1.7. Industrial Automation — TSN Frames for details on the MAC destination address format) and the VLAN ID of the frame (which can be the VLAN ID of the port if the frame has no VLAN tag).

The process can be summarized as follows:

1. Compare the OUI of the MAC destination address of the frame to the value configured in the IALK_OUI register. If it is not a match, return a not-found response.
2. Compare the ID of the MAC destination address of the frame to be within the range defined by IALK_ID_MIN and IALK_ID_MAX. The behavior when the ID is not within the range is configured in the IALK_ID_CONFIG register.
   (a) If the frame is not within the range and INVLD_ID_FLOOD is 0, return a not-found response.
   (b) If the frame is not within the range and INVLD_ID_FLOOD is 1, return a found response using the mask in INVLD_ID_FLOOD_MASK, the learn_ok value in INVLD_ID_LRN_ENA, and the priority from INVLD_ID_PRIO. To force a frame to be discarded the mask INVLD_ID_FLOOD_MASK must be set to 0.

Also, it is recommended to set INVLD_ID_LRN_ENA to 0 to prevent learning the MAC source address of the frame.

3. Calculate a new ID value by subtracting IALK_ID_SUB from the frame ID.
4. Check the number of enabled VLANS in IALK_VLAN_CONFIG.VLANS_ENABLED.
   (a) If VLANS_ENABLED is 0 (no VLANs), use the calculated ID as the entry address for the IA-FDB, and set n to 0 (used later to return learn_ok on a hit).
   (b) If VLANS_ENABLED is 1 or more, check whether the VLAN ID of the frame matches any of the registers IALK_VLANIDn (n = 0 to (VLANS_ENABLED - 1)). A match requires that IALK_VLANIDn.VLANID_ENA is set to 1 and IALK_VLANIDn.VLANID matches the VLAN ID of the frame, or that IALK_VLANIDn.VLANID is 0.
      (i) If there is match, calculate the entry address addr using the LSBits to select the VLAN and the MSBits from the calculated ID. The number of bits used to select the VLAN is $\log_2$(VLANS_EANBLED).
      (ii) If there is no match, the response is determined by the configuration in IALK_VLAN_CONFIG.
      If UNKWN_VLAN_FLOOD is 0, the response is not-found.
      Otherwise, the returned port mask is UNKWN_VLAN_FLOOD_MASK, learn_ok is given by UNKWN_VLAN_LRN_ENA and the priority is taken from UNKWN_VLAN_PRIO. To discard the frames with unknown VLAN set UNKWN_VLAN_FLOOD_MASK to 0.
5. Read from the FDB and return a valid and pmask (port-mask). A valid indicates whether the entry contains a valid port-mask.
6. If the entry is valid (valid is set to 1), return the port mask value in pmask and set learn_ok from IALK_VLANIDn.VLANID_LRN_ENA and the priority from IALK_VLANIDn.VLANID_PRIO. When no VLANS are enabled, n is set to 0.
7. If the entry is invalid, the response depends on whether VLANS are enabled.
   (a) If VLANs are disabled (VLANS_ENABLED = 0), the response is "not-found".
   (b) If VLANs are enabled (VLANS_ENABLED > 0), the response is found even though the entry in the table was invalid. The return port mask is taken from the flooding mask IALK_VLANIDn.VLANID_FLOOD_MASK, learn_ok is taken from IALK_VLANIDn.VLANID_LRN_ENA and the priority from IALK_VLANIDn.VLANID_PRIO. To discard frames whose valid entry is set to 0, set VLANID_FLOOD_MASK to 0.

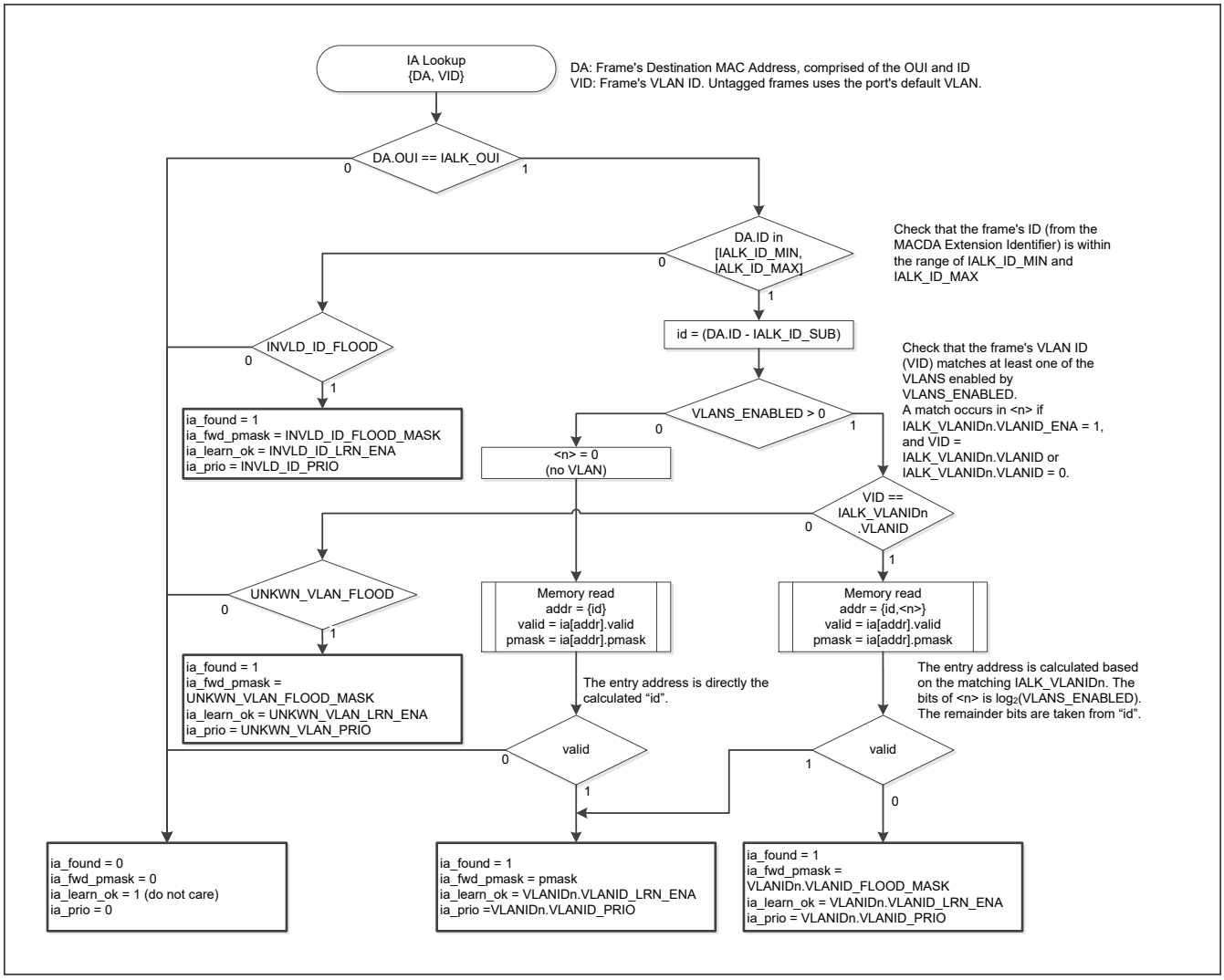


Figure 29.19 IA-FDB look-up flow

(4) IA-FDB Physical and Logical Layout

As previously shown, the IA-FDB is indexed by an entry address, in a direct and linear fashion. The entry address is composed by the index of the IALK_VLANIDn register matching the VLAN ID of the frame, and the adjusted ID from the MAC destination address of the frame.

The table must be configured during initialization for the number of intended VLANs to support using the configuration register IALK_VLAN_CONFIG.VLANS_ENABLED, and this must be done prior to enable the IA lookups in IALK_CONTROL.

When no VLANs are used, the ID is used directly as the address of the entry, taking as many bits as entries supported by the table. On the other hand, when at least 2 VLANs are enabled, the bits given by $\log_2$(VLANS_ENABLED) are used as the LSBits for the entry address, and the remainder bits are taken from the LSBits of the ID.

Figure 29.20 shows an example where 4 VLANs are enabled in VLANS_ENABLED and the total number of entries for the IA-FDB is 8192 where n corresponds to each of the 4 IALK_VLANIDn registers used for comparing the VLAN ID.

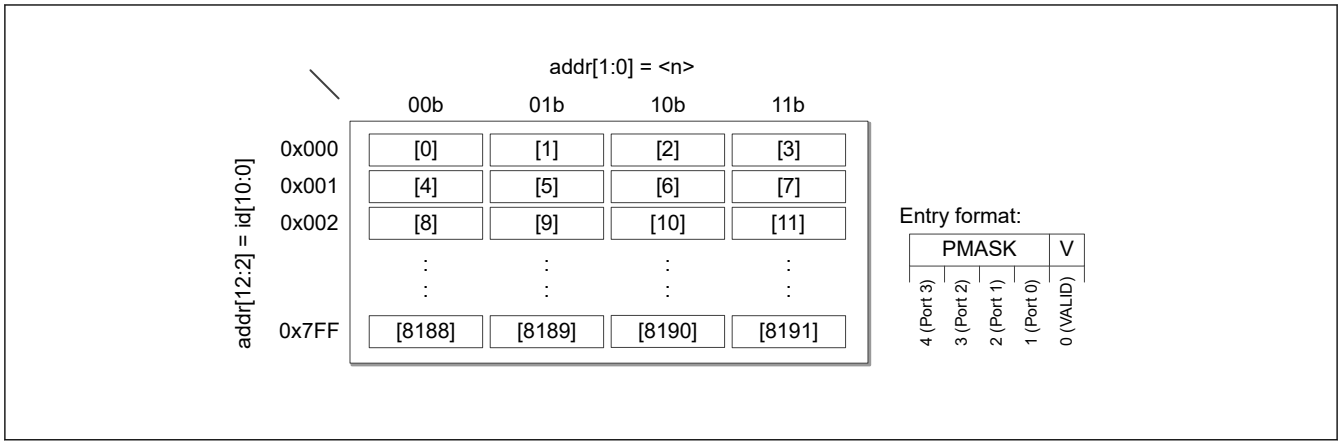


Figure 29.20 IA-FDB 4-VLANs logical geometry example

The IA FDB size is 8192 and supports up to 4 IA_VLANs.

(5) Programming

The IA-FDB initializes with all bits set to 0, which means that all entries are invalid, and all port masks are 0.

The programming of the IA-FDB is done using the control registers IALK_TBL_ADDR and IALK_TBL_DATA. To program an entry, the address must be calculated as previously explained, considering the number of enabled VLANs. This address is programmed in IALK_TBL_ADDR first, and then the entry can be written by writing the entry in IALK_TBL_DATA as indicated in the register description. The entry can also be read from IALK_TBL_DATA.

To facilitate initialization the address in IALK_TBL_ADDR can be configured to auto-increment for every read and write operation to IALK_TBL_DATA. The auto-increment value is a 4-bit field to allow incrementing within the same VLAN (see Figure 29.20). The address value rolls over automatically.

If the number of entries in the IA-FDB is not a power of 2, writing an invalid address in IALK_TBL_ADDR is not defined and should be prevented with software.

(6) Usage Notes

The programming of the registers IALK_ID_MIN, IALK_ID_MAX, and IALK_ID_SUB must be programmed so that the resulting ID, after being combined with the VLAN index n to form the entry address results in a value that is within the range of implemented entries. The hardware does not perform any range checks on the entry address and simply does the calculation as indicated in the previous sub-sections.

To re-iterate, the resulting forwarding port mask from the IA-FDB is still subject to other forwarding checks detailed on section 29.4.3.10. Frame Forwarding Tasks. It is recommended that the VLAN IDs used for IA-TSN traffic are not programmed in the VLAN table to prevent further filtering.

The register IALK_CONTROL allows disabling cut-through forwarding for frames returning a found response from the IA-FDB. The use of this feature is not specified in the IEC 60802. The default value is to disable cut-through, so it is recommended to set the cut-through enable when enabling the IA-FDB for a port.

The IEC/IEEE 60802 IA-TSN profile requires that IA-TSN frames are prevented from automatic hardware learning. To achieve this, various *_LRN_ENA bits must be set to 0. When LRN_ENA is set to 1, learning can occur only if also hardware learning is enabled in the LK_CTRL register.

The priority when a found response is returned can be set in the various *_PRIO fields. This priority is the same priority as the MAC address table priority as indicated in (4) Priority Resolution. The final priority/output queue follows the process described in that section. The entries in the IA-FDB are considered static entries.

29.4.3.10 Frame Forwarding Tasks

(1) Overview

When an input port has been selected, the frame is forwarded to its corresponding output ports as necessary. Output port resolution and switching are based on the information from the 2-stage destination MAC address lookup followed by additional resolution functions to allow frame duplication and flooding control, which are described in the following sections.

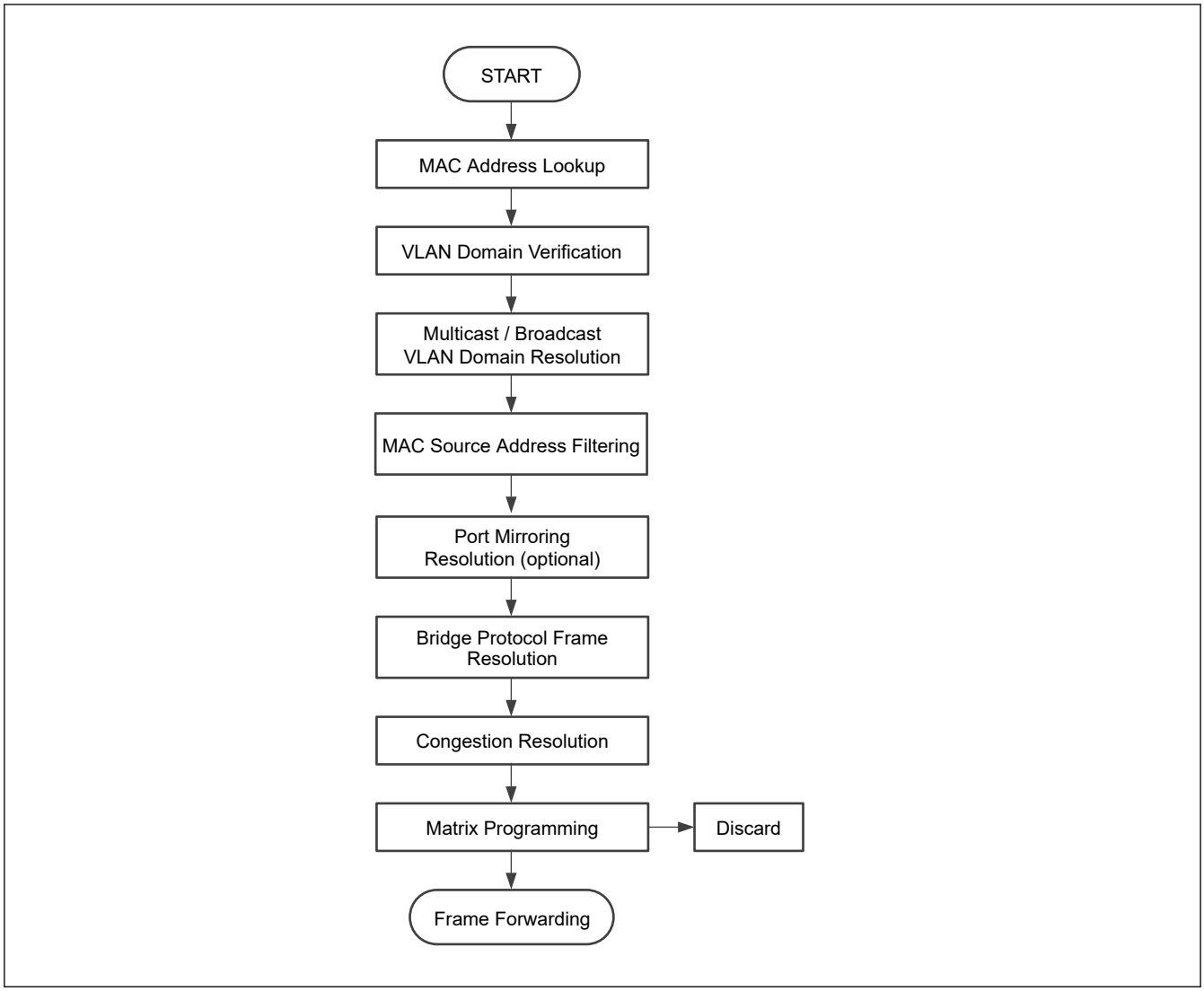


Figure 29.21 Frame forwarding tasks overview

(2) VLAN Domain Verification

When a frame is received at a port, it can be verified to be within the correct VLAN domain. The register VLAN_VERIFY is used to define if input verification should occur. Output verification is always implicitly done with the VLAN resolution function, which forwards a frame only to ports that are a member of the VLAN as defined by the VLAN table (see (b) VLAN Domain Resolution / VLAN Table). The following rules apply if the verify bit within VLAN_VERIFY for the port is enabled:

- If the VLAN ID of the frame is found within the table and the input port is member of the VLAN domain, the frame is forwarded normally.
- If the VLAN ID of the frame is found within the table but the input port is not a member of the VLAN domain, the frame is marked invalid and is discarded eventually.
- If the VLAN ID of the frame is not found in the VLAN table, or the frame has no VLAN tag, the frame is forwarded normally, or, if the discard bit for the port is set to 1 (VLAN_VERIFY.VLANDISC), it is discarded.

Note: When the VLAN_VERIFY does not enforce input verification, the output verification (VLAN domain resolution) is always active. To avoid unexpected behavior if VLAN domains are not used, the complete VLAN_RES_TABLEn should be programmed to have an all 1 port mask in all entries.

(3) Broadcast / Multicast / VLAN Domain Resolution

(a) Overview

To ensure that traffic within VLAN channels are always routed to the correct ports, for example, to avoid the duplication of critical information through a network, the switch implements a resolution mechanism that, for any frame that is switched to multiple ports, checks the VLAN ID provided with the current frame.

The VLAN resolution mechanism searches the VLAN resolution table (see section 29.3.1.67. VLAN_RES_TABLEn : 32 VLAN Domain Entries (n = 0 to 31)), which stores up to 32 unique VLAN IDs, each associated to a port bit mask. It is used to limit forwarding of frames only to ports that are member of a VLAN domain.

(b) VLAN Domain Resolution / VLAN Table

The VLAN resolution table (registers VLAN_RES_TABLEn) provides a unique VLAN ID / port bit mask association for up to 32 VLAN IDs. A default entry (register BCAST_DEFAULT_MASK0/BCAST_DEFAULT_MASK1) provides an additional port bit mask. The port bit mask implements one bit for each port.

Each port bit Mask indicates, if set to 1, that it is member of the VLAN and frames with the corresponding VLAN ID can be switched to the port. If the port bit Mask is set to 0, a frame with the corresponding VLAN ID is not switched to that port. If no VLAN ID matches, the default mask is applied.

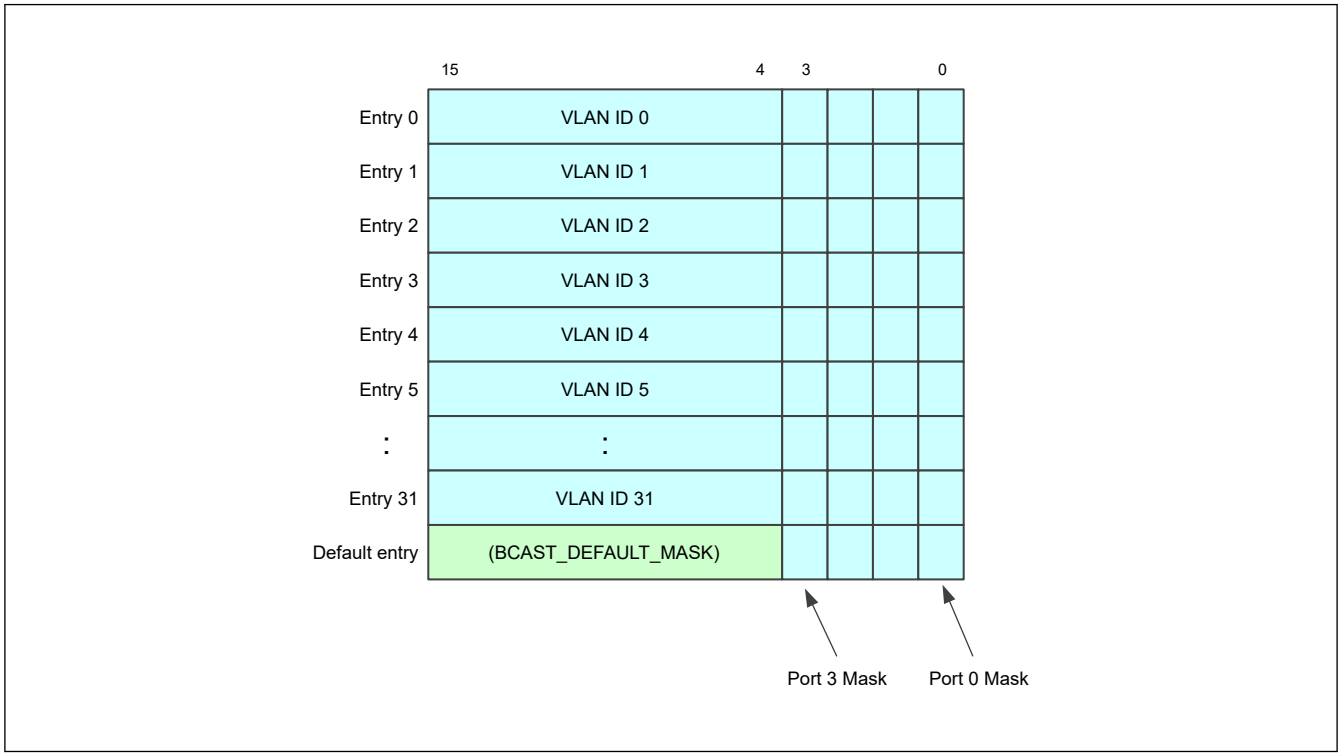


Figure 29.22 VLAN resolution table overview

Note: The port mask is found in bits [3:0] and the VLAN info is found at bits [15:4].

The VLAN table cannot be deactivated. To avoid unexpected behavior if VLAN domains are not used, all entries of the table (all VLAN_RES_TABLEn registers) should be programmed to have an all 1 port mask in all entries (reset default).

(c) VLAN Switching / Resolution Mechanism

The VLAN table is used for both, VLAN domain verification (see (2) VLAN Domain Verification) as well as VLAN resolution. Once the frame has passed any VLAN domain verification (not to be discarded by the verification function already), the forwarding resolution applies.

- If the destination MAC address (unicast or multicast) is found in the MAC address table:
  - If the frame carries a VLAN ID that is found in the VLAN resolution table, the frame can be forwarded only to the ports within the VLAN domain and is discarded if the destination port is not member of the VLAN domain.

  - If the frame carries a VLAN ID that is not found in the VLAN resolution table or does not contain a VLAN tag, it is forwarded as indicated by the lookup table (VLAN domain verification can be configured to discard the frame in this case if enabled).

- If the destination MAC address (unicast or multicast) is not found in the MAC address table, or if the destination address is the broadcast address, the frame is forwarded according to the following rules:
  - The destination port mask is loaded from the respective register U/M/BCAST_DEFAULT_MASK0 or U/M/BCAST_DEFAULT_MASK1 depending on unicast, multicast, or broadcast. Then the following filtering on this mask applies.
    - If the frame carries a VLAN tag, the VLAN resolution table is searched for a matching VLAN ID and the frame is sent only to ports that are associated with the VLAN ID.
    - If the frame carries a VLAN tag and the VLAN ID does not match any entry in the VLAN Resolution Table, or the frame does not carry a VLAN tag, the frame is forwarded to all ports that are enabled by the default mask.
    - If it cannot be associated with any VLAN group and if the default group is set to all 0s, the frame is discarded.

To disable the VLAN resolution, set all VLAN IDs to 0x000 and all Port bit mask to 1 in the VLAN resolution table. If the VLAN resolution is disabled, normal port flooding is implemented as described in (5) Destination MAC Address Lookup. The default entry can still be used to restrict flooding to only dedicated ports, if not programmed to all 1.

(4) MAC Source Address Filtering

The switch can be enabled to perform a source address check on the received frames of the line ports. This check allows limiting a single source MAC address to send frames to one or more destination ports, while blocking all other nodes from sending frames to them. A use case for this filter is to accelerate forwarding of EtherCAT frames when the switch operates as a stream adapter to create a virtual channel between the EtherCAT Master controller and one or more EtherCAT slaves connected to the switch. The filter is configured to operate on the port connected to the EtherCAT master to allow only frames with the source address of the master to be forwarded to the ports where the EtherCAT slaves are connected. Frames from the EtherCAT slaves towards the master must be forwarded first to the management port to do a MAC address replacement before forwarding these frames to the master.

The filter is enabled with the control register SRCFLT_ENA where two fields determine how the filter is applied.

- SRCENA: Defines on which receive ports the check is executed. Ports which are not enabled simply bypass the check. For EtherCAT, a single port would be enabled.
- DSTENA: Defines on which transmit ports the check is applied to. The destination ports that are not enabled are not affected by the filtering function. For EtherCAT, the ports connected to the slaves are enabled.

The MAC source address to accept is programmed in the register SRCFLT_MACADDR_HI/SRCFLT_MACADDR_LO. A 16-bit mask is provisioned to allow masking the last 16 bits of the address.

The filtering check is performed after the L2 lookup is done or flooding is calculated, and after the VLAN domain is verified. This means that the check is applied regardless on whether the destination address is known or if the destination ports DSTENA are present in the flooding masks. Optionally a filtered frame can be forwarded to the management port via the control bit SRCFLT_CONTROL.MGMT_FWD to allow the firmware to inspect the frame.

When a frame is discarded due to this function, the frame is only counted in the counter IDISC_SRCFLT as discarded if the final forwarding mask is zero. If MGMT_FWD is set to 1 then the frame is not counted as a discard, or if the resultant mask after removing the ports in DSTENA is non-zero (the frame is received by other port).

To further support EtherCAT, the filtering function implements a watchdog timer that resets when a frame with a source address matching the filtering address. The watchdog is enabled in the SRCFLT_CONTROL by the control bit WATCHDOG_ENA, and the watchdog expiration time is programmed in WATCHDOG_TIME in the same register. When the watchdog expires the interrupt SRCFLT_WD_INT in the global interrupts asserts. The interrupt is enabled in the INT_CONFIG register and its state can be read in the INT_STAT_ACK register. When the watchdog expires, it stops operating until it is disabled and enabled again using WATCHDOG_ENA.

(5) Port Mirroring

The optional function allows duplicating traffic to a dedicated mirror port. Any one of the ports can be assigned to act as a mirror port (register MIRROR_CONTROL).

The mirror port then is always added to the list of output ports and therefore receives a copy of the frame, if any of the following rules matches with the currently processed frame:

- Ingress Port Number Match
  When a frame is received on port n and the corresponding bit in the register MIRROR_ING_MAP is set to 1, the frame is mirrored.
- Egress Port Number Match
  When a frame is forwarded to port n and the corresponding bit in the register MIRROR_EG_MAP is set to 1, the frame is mirrored.
- MAC Ingress SA Match
  When the Ingress Port Number match succeeded (see above) and the MAC source address matches the MIRROR_ISRC_0 and MIRROR_ISRC_1, the frame is mirrored.
- MAC Ingress DA Match
  When the Ingress Port Number match succeeded (see above) and the MAC destination address matches the MIRROR_IDST_0 and MIRROR_IDST_1, the frame is mirrored.
- MAC Egress SA Match
  When the Egress Port Number match succeeded (see above) and the MAC source address matches the MIRROR_ESRC_0 and MIRROR_ESRC_1, the frame is mirrored.
- MAC Egress DA Match
  When the Egress Port Number match succeeded (see above) and the MAC destination address matches the MIRROR_EDST_0 and MIRROR_EDST_1, the frame is mirrored.

In addition, a counter is implemented (register MIRROR_CNT) that allows specifying that only every Nth (N: count value) frame that matches any of the specified criteria is mirrored. If the counter is set to 1 or 0, every frame that matches any of the specified criteria is mirrored.

(6) Congestion Resolution

(a) Overview

The congestion resolution function is used whenever an output port is not available and data must be sent to that port. An output port is defined to be available if the port is enabled (bit n in register PORT_ENA.TXENA is 1, n = 0 to 3) and its output queue is not full. If a port is deasserted (bit n in register PORT_ENA.TXENA is 0), the port is not available and frames cannot be switched to that port.

The congestion resolution function determines whether the frame should be processed further or discarded according to the following rules.

(b) Unique Destination (One Input to One Output)

If the output port is enabled and can accept, the frame is forwarded normally.

In any other case, the frame is discarded.

(c) Multiple Destinations (Flooding)

After broadcast/flooding resolution, a frame is switched to multiple output ports.

- Output disabled:
  All disabled ports are removed from the list of outputs.
- Output congestion:
  If any of the outputs cannot accept a frame (as indicated by the output queue management for the port), it is removed from the list of outputs also.

If after the removal, no output port is left in the list of outputs, the frame is read from the input and discarded. This event is counted in the port specific ODISCn statistics counter.

(7) Bridge Protocol Frame Resolution

To implement bridge control protocols like the Spanning Tree protocol, the following control functions are performed by the Bridge Protocol Frame Resolution function.

(a) Management Frame (BPDU) Identification

Bridge Protocol Frames are identified by its destination address being any of the following:

- 01-80-c2-00-00-00 to 01-80-c2-00-00-0F (Spanning Tree, IEEE 802.1D, Table 7-9)
- 01-80-c2-00-00-10 (Bridge Management Address, IEEE 802.1D, Table 7-10)
- 01-80-c2-00-00-20 to 01-80-c2-00-00-2F (Generic Attribute Registration Protocol, IEEE 802.1D, Table 12-1)

In addition, the following features are available to define a frame as Bridge Protocol frame which then experiences the same treatment as any other BPDU within the switch.

- Custom Address:
  To support other protocols and standards, the switch allows defining four additional addresses (see registers MGMT_ADDR0_lo/hi through MGMT_ADDR3_lo/hi) which are treated identically to the above, marking a frame as a Management (BPDU) frame.

To support IEEE 802.1X authentication, the destination address 01-80-c2-00-00-03 is used by EAPOL frames and can be treated differently from BPDU frames (see section 29.4.8. Port Based Access Control (802.1X)).

(b) Input Port Blocking

The input port blocking function is used to avoid forwarding of frames after address learning. The firmware can program the register INPUT_LEARN_BLOCK and if a frame is received on a port n which should be blocked (blocking bit n = 1) and the frame is not a bridge protocol frame, the frame is marked for discard and is not forwarded to any output port.

(c) Input Port Learning Disable

To reduce processing load from the firmware, a port can be configured for exclusion from learning (see section 29.3.1.8. INPUT_LEARN_BLOCK : Input Learning Block Register). When learning is disabled on a port, no source address extraction happens for incoming frames.

Incoming BPDU frame source addresses are never extracted and not forwarded to the learning process independent from this setting. The application must add source addresses from received BPDUs to the lookup table for example, as static entries as required.

(d) Management Frame Forwarding

If a frame is coming from any port except the management port and is marked as a management frame (BPDU frame) and the management port is enabled (see section 29.3.1.9. MGMT_CONFIG : Management Configuration Register), the frame is forwarded to the management port only, ignoring any destination address lookups.

If the management port itself receives management frames (host CPU sends management frames to switch), they are forwarded according to the port mask defined in the section 29.3.1.9. MGMT_CONFIG : Management Configuration Register. A handshaking mechanism is implemented that can be used by the protocol software to configure the destination port mask on a frame by frame basis for management frames.

In addition, the port mask given in MGMT_CONFIG can include ports that are disabled for normal traffic (configured in register PORT_ENA.TXENA) allowing transmitting management frames on ports that are disabled for normal traffic.

When forced forwarding is used, management frames are forwarded to the ports given in the forced forwarding mask and the port mask given in MGMT_CONFIG is ignored.

Note: VLAN domain verification/discard (see section 29.3.1.5. VLAN_VERIFY : Verify VLAN Domain Register) should be switched off for the management port to avoid that the switch discards management frames.

MGMT_CONFIG.MGMT_EN must be set to 0 to allow forwarding of BPDU frames from the management port without a management tag.

(8) Forced Forwarding

When the forced forwarding is used, it overrides any forwarding decision and a frame is forwarded to all ports given in the forced forwarding mask by management tagging fields.

This includes forwarding to disabled ports (overriding PORT_ENA) when the force filtering by management tagging fields is disabled (0). Only the port where the frame was received from is removed from the port mask automatically (it is not possible to transmit a frame to where it was received from).

Mirroring and Snooping functions may continue to inspect forced forwarded frames.

(9) Cut Through Forwarding

The switch normally operates in so-called store & forward mode. All frames are received and stored completely before they are forwarded to any output port. This allows error detection and discards of erroneous frames within the switch as well as communication between links of different speeds.

Alternatively, the switch can operate in cut through mode, where a frame is forwarded and transmitted at an output port before it is completely received. This allows for reduced latency and is independent of the frame size. However as the frame is forwarded before error checks can be done, erroneous frames are transmitted (the switch ensures to mark them erroneous and sends it with an invalid CRC).

Cut Through operation is configurable on a per-port basis with register MODE_CONFIG. It allows to define a group of ports for cut through and have other ports using normal store & forward.

When cut through forwarding is enabled, a frame is forwarded using cut through only if the following conditions are all valid. Otherwise, the frame is forwarded using store & forward:

- The source and destination ports both have their cut through mode bit set
- It is not a broadcast, not a BPDU, not an IEEE 1588 PDELAY_REQ frame (auto-response)
- The destination ports operate at the same or slower link speed than the source port
- The frame is not received as a preemptable frame (with an SMD-Sx as a frame delimiter)

A frame can be sent as cut-through to some destination ports and store & forward to other ports if the cut-through criteria are not met.

The cut through forwarding delay depends on link speed and the switch operating clock frequency. For a switch operating clock of up to 200 MHz, the following forwarding delays are possible.

See section 29.4.22.1. Cut-Through Frame Forwarding Delay.

When using cut-through operation the register CT_DELAY must be programmed as indicated in section 29.3.6.17. CT_DELAY_Pn : Port n Cut-Through Delay Indication Register (n = 0 to 2) to prevent frame underflows.

The control bit COMMAND_CONFIG.TX_TRUNCATE can be set to 1 so frames received with errors that started transmission in cut-through mode are shortened by the switch. The shorting is marginal (between 8 to 12 bytes) but it allows the frame to be progressively shortened over network hops, eventually forcing it to be discarded when it becomes less than 64 bytes.

(10) Switching

After the output ports are determined, the switch control enables the corresponding path through the switch matrix and the frame is forwarded to the output queues.

In a similar fashion, if a frame should be switched to multiple ports (broadcast), the switch control enables the corresponding paths through the switch matrix and the frame is forwarded to all the destination output ports.

Duplication is eventually done by the output ports reading the same frame from memory multiple times.

29.4.3.11 Broadcast Storm Protection

To avoid network and memory congestion caused by a broadcast storm, the switch engine can be programmed to limit the number of broadcast frames over a certain time period.

A counter is implemented on each receive port, which increments whenever a broadcast frame is received. The counters of all ports are cleared each time the timeout expires.

When the counter of a port reached the limit, further broadcast frames received at this port are discarded.

The register BCAST_STORM_LIMIT is used to set the periodic timeout and the counter limit for all ports.

29.4.3.12 Multicast Storm Protection

To avoid network and memory congestion caused by a multicast storm, the switch engine can be programmed to limit the number of multicast frames over a certain time period.

The behavior is identical to the broadcast storm protection, however, implements its own per port counters.

The register MCAST_STORM_LIMIT is used to set the counter limit for all ports. The periodic timeout from BCAST_STORM_LIMIT is used to reset the counters.

29.4.3.13 Output Frame Queuing

Figure 29.23 shows a conceptual output queue management function supporting multiple queues.

The implementation offers 8 queues numbered 0 to 7 where 0 is lowest and 7 is highest priority.

The output queues (FIFOs) are implemented separately for each output port. After input processing has completed, the frame is stored in a queue within the corresponding output port, based on the priority resolution result for the frame (classification, see (4) Priority Resolution).

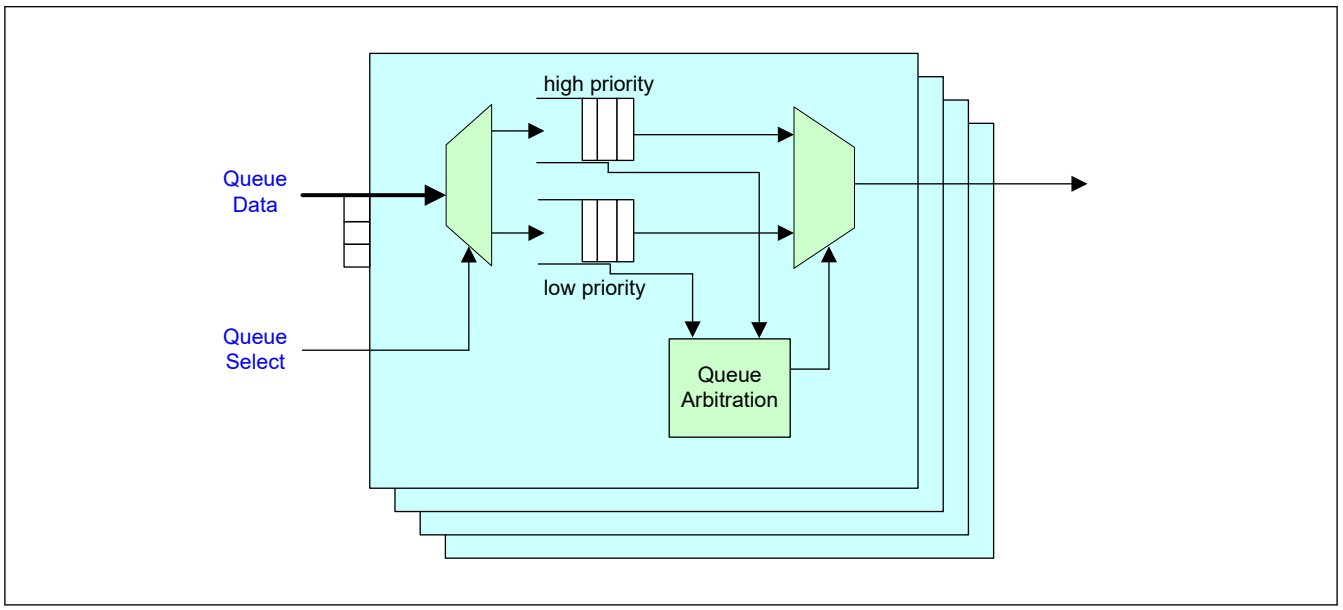


Figure 29.23 Output queuing

The queue arbitration selects the appropriate queue and indicates data availability to the output interface, which eventually transmits the frame. A higher queue number takes precedence over a lower queue number.

The queue arbitration implements a strict priority selection. Weighted fair queuing is also available (see section 29.3.1.41. IMC_CONFIG : Input Memory Controller Configuration Register).

In addition, the optional TDMA scheduler influences arbitration by allowing read from a queue only at specific times (see section 29.4.15. TDMA Operation).

The queues can be configured to transmit express or preemptable traffic. By default, all queues are express queues.

When some queues are configured for preemptable traffic, all express queues have priority over the preemptable queues when preemption is enabled and operational. See section 29.4.18. IEEE 802.3br Traffic Interspersing for details on traffic interspersing and preemption configuration.

29.4.3.14 Output Port Congestion Management

Depending on the output queue management method, a congestion indication is generated to the switch (internally) when a queue becomes full. When a frame must be forwarded to a congested queue, the frame is discarded and the associated error counter (section 29.3.9.5. ODISCn : Port n Discarded Outgoing Frame Count Register (n = 0 to 3)) is incremented.

(1) Tail Drop

By default, frame discard occurs only when a queue becomes full, or if the shared switch memory is exhausted. Frames are dropped as long as the congestion situation persists.

The implementation allows to store up to 58 frames per queue per port. This is independent of the frame size.

(2) Random Early Detection

As an alternative to Tail Drop, the Random Early Detection (RED) active queue management following RFC 2309 is available (see section 29.4.23.3. References, (1)). It can be enable per queue (only queue 0 to 3) but only globally for all ports (see registers QMGR_RED_MIN4*1/MAX4*2/CONFIG*3).

The RED algorithm averages the fill level of a queue over time. Depending on the current average a drop probability is calculated and frames are dropped before the queue becomes full. When filled completely, tail drop occurs. The early dropping of frames may help higher protocols (mainly TCP) to better adapt to the available link bandwidth. The algorithm is configured using two thresholds defining a window where frames are dropped before the queue capacity is reached. No drop occurs below the minimum threshold and all frames are dropped when the maximum is reached (resulting in tail drop behavior). Between minimum and maximum, the probability for dropping a frame increases. That is, the more the queue is filled the more frames are dropped which ideally, avoids reaching the maximum which causes continuous frame drops.

When the current queue fill level drops below the calculated average, the average is immediately set to the current level of the queue. This prevents a delayed reaction where the queue is already emptied again while the average still indicates a higher fill level causing unnecessary dropping of frames.

Note 1. See section 29.3.1.34. QMGR_RED_MIN4 : RED Minimum Threshold Register.
Note 2. See section 29.3.1.35. QMGR_RED_MAX4 : RED Maximum Threshold Register.
Note 3. See section 29.3.1.36. QMGR_RED_CONFIG : RED Configuration Register.

Figure 29.24 shows the behavior.

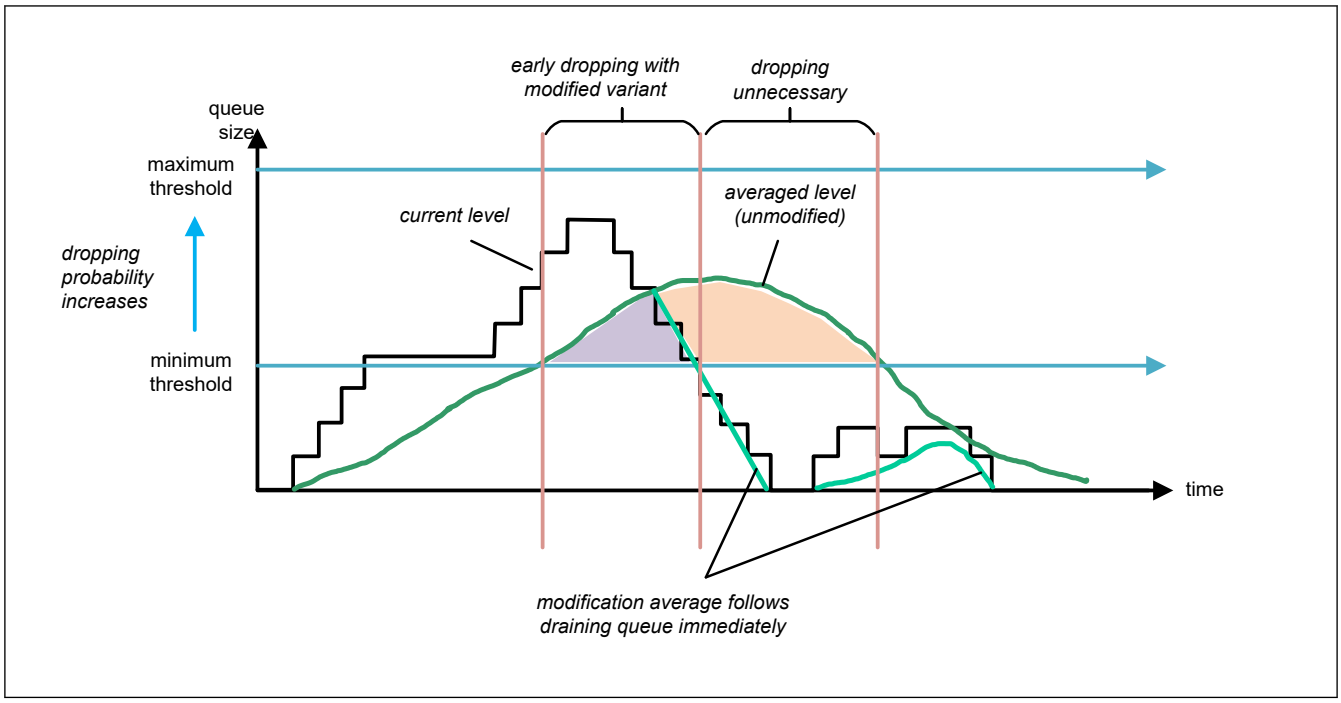


Figure 29.24 Random early detection algorithm implementation

The algorithm avoids dropping of frames when the queue is draining which means the output rate is higher than the input rate and therefore, dropping of frames should not occur. This allows to absorb a short peak in the queue size making use of the available storage capacity, ideally without (unnecessary) dropping activity.

29.4.4 Timestamping Functions (TSM)

29.4.4.1 Overview

The timer and timestamping module (TSM) implements the following functions:

- Collects a timestamp for received frames on all line ports.
- Collects a timestamp for transmitted frames on all line ports with interrupt generation.
- Processes IEEE 1588 Layer 2 event frames updating on the fly (1 step) the correction field with the transient time.

- Up to 2 adjustable timers allowing synchronizing the local time of the timer to a remote master clock through some protocol specific application software to implement, for example, the IEEE 1588 Precision-Time-Protocol (PTP).

29.4.4.2 IEEE 1588 Message Header Structure

Note: The following is provided as a brief reference only. Refer to the standard documents for details and usage scenarios.

An IEEE 1588 message can be transported with various encapsulation protocols for example within an Ethernet frame (L2) payload or a UDP/IP payload. Table 29.16 shows the message header, which is common to all IEEE 1588 version 2 messages. The MAC allows updating the correction field during transmission when using Layer 2 frames which are identified by Ethernet type field being 0x88F7. Other encapsulations are not supported. Correction field updates only occur in event messages (messageId < 4).

All fields follow the network byte order which is the first byte transmitted is the most significant byte (leftmost) of a multi byte field. Table 29.16 and Table 29.17 provide a short overview. For details of any of the fields, refer to the IEEE 1588 specification document (see section 29.4.23.3. References, (2)).

Table 29.16 IEEE 1588v2 message header (PTPv2)

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
transportSpecific				messageId				1	0
reserved				versionPTP = 0x2				1	1
messageLength								2	2
domainNumber								1	4
reserved								1	5
flags								2	6
correctionField								8	8
reserved								4	16
sourcePortIdentity								10	20
sequenceId								2	30
control								1	32
logMeanMessageInterval								1	33
further message specific fields may follow								n	34

The type of message is encoded in the field messageId (informal) as follows:

Table 29.17 PTPv2 message type identification

messageId	Message name	Message
0x00	SYNC	event message
0x01	DELAY_REQ	event message
0x02	PDELAY_REQ	event message
0x03	PDELAY_RESP	event message
0x04 - 0x07	—	reserved
0x08	FOLLOW_UP	general message
0x09	DELAY_RESP	general message
0x0A	PDELAY_RESP_FOLLOW_UP	general message
0x0B	ANNOUNCE	general message
0x0C	SIGNALING	general message
0x0D	MANAGEMENT	general message

29.4.4.3 Timestamp Processing

(1) Receive Timestamp Processing

When a port receives a frame, the timestamp, based on the current time from the timer, is captured when the start of frame delimiter (SFD) is detected at the PHY interface. The timestamp is forwarded together with the frame throughout the switching engine and can be accessed by the management port to implement, for example, the Precision-Time-Protocol (PTP).

The receive timestamp is encapsulated in the proprietary tag information for frames forwarded to the management port (see section 29.4.5. Management Port Specific Frame Tagging).

Only the timestamp from one timer can be encapsulated with the frame. The timer to be used is determined by (from lower to higher priority):

1. The default timer of the port based on the port selection, configured in the COMMAND_CONFIG_Pn register, TIMER_SEL field (see section 29.3.6.2. COMMAND_CONFIG_Pn : Port n Command Configuration Register (n = 0 to 3)). Each port can be configured to select the default timer to use for timestamping frames on receive (and on transmit).
2. PTP auto response timer selection based on the domainNumber of the frame (see section 29.3.2.3. PTP_DOMAIN : Domain Number of PTP Frame) or a default value per-port if no match against domainNumber programmed in the PTP Auto Response register (see section 29.3.6.10. PTPAUTORESPONSE_Pn : Port n PTP Auto Response Register (n = 0 to 2)).
   See section 29.4.4.6. Automatic Response to Peer Delay Request Messages for more information on PTP auto response.
3. An override using the pattern matchers. The pattern matchers can override the default timer to use based on fields on the frame. See section 29.4.14. Receive Pattern Matcher for details on the pattern matchers.
   See section 29.4.22.3. Frame Timestamp Interface Delay for details on how the timestamp captured by the port must be corrected to obtain a precise value. This correction accounts for the delay between the timestamp point and the MII/GMII interface.

(2) Transmit Timestamp Processing

When a frame is transmitted to the PHY on a port, the timestamp is also captured. The outgoing timestamp can be stored in a per-port FIFO to be read by software later using the registers TS_FIFO_READ_CTRL (see section 29.3.2.7. TS_FIFO_READ_CTRL : Transmit Timestamp FIFO Read Control Register) and TS_FIFO_READ_TIMESTAMP (see section 29.3.2.8. TS_FIFO_READ_TIMESTAMP : 32-bit Timestamp Value Read from FIFO). Each port is capable of storing up to 16 timestamps.

The management port, through its management port specific frame tagging has the capability to identify frames for outgoing timestamp capture, as timestamps must only be captured for dedicated event frames, not for all frames, as well as to indicate the timer to use for the timestamp operation and a 7-bit unique ID that gets stored together with the timestamp for further correlation. See section 29.4.5.4. Receive Processing (CPU to Switch) for details on the management tag control bits.

Similarly to the receive timestamp processing, the timer to use for timestamping a frame is determined by (from lowest to highest priority):

1. The default timer of the port based on the port selection, configured in the COMMAND_CONFIG_Pn register, TIMER_SEL field (see section 29.3.6.2. COMMAND_CONFIG_Pn : Port n Command Configuration Register (n = 0 to 3)). Each port can be configured to select the default timer to use for timestamping frames on receive (and on transmit).
2. PTP auto response timer selection based on the domainNumber of the frame (see section 29.3.2.3. PTP_DOMAIN : Domain Number of PTP Frame) or a default value per-port if no match against domainNumber programmed in the PTP Auto Response register (see section 29.3.6.10. PTPAUTORESPONSE_Pn : Port n PTP Auto Response Register (n = 0 to 2)).
   See section 29.4.4.6. Automatic Response to Peer Delay Request Messages for more information on PTP Auto response.
3. Override from the pattern matchers. See section 29.4.14. Receive Pattern Matcher for details on the pattern matchers.
4. Indication from the management port using the management tag (section 29.4.5.4. Receive Processing (CPU to Switch)).

Note: A limitation to the timestamping logic is that when the timer selection is the default of the port and the transmitted frame is a multicast frame, all ports utilize the timer indicated by the port with the lowest index in the destination port mask. For instance, if a frame from the management port is transmitted to ports 0 and 1, both ports timestamp the frame using the timer selection value of port 0.

See section 29.4.22.3. Frame Timestamp Interface Delay for details on how the timestamp captured by the port must be corrected to obtain a precise value. This correction accounts for the delay between the timestamp point and the MII/GMII interface.

29.4.4.4 Transparent Clock Support

(1) General

The hardware implements the necessary functions to implement transparent clocks (TC) in both the end-to-end variant and the peer-to-peer variant.

The end-to-end transparent clock is the default behavior when the register PEERDELAY is set to 0.

When a peer-to-peer transparent clock is implemented, the PTP software must determine the peer delay at the port where the master is connected to (where it receives SYNC messages). This value is then programmed into the PEERDELAY register. The setting must be updated when a new peer measurement has been done or whenever the master port changes.

(2) Implementation of Correction Field Update

The implementation is capable of updating the correction field within Layer 2 PTP frames only (frames with type 0x88F7). PTP messages within UDP/IP frames are not considered.

The update module only processes event messages, which it automatically detects from the messageId type field found within the PTP header (type < 4). This means that it does not process follow-up frames (which are not event frames), hence any correction field updates are found in the corresponding SYNC frame. This allows supporting one step as well as two step-master and -slave nodes connected to the switch.

For an end-to-end transparent clock implementation, the correction field of SYNC and DELAY_REQ messages is updated with the transient time (output time − input time).

For a peer-to-peer transparent clock implementation, the correction field update includes adding the value set in the PEERDELAY register of the port where the frame was received to the transient time. The hardware always implements the addition of this value, hence it only depends on configuring the PEERDELAY to define whether an end-to-end (PEERDELAY = 0) or peer-to-peer (PEERDELAY > 0) behavior is selected. It is the responsibility of the PTP software to exchange the PDELAY_REQ/PDELAY_RESP messages on every port to determine the peer delay values accordingly.

Correction field updates occur only on frames that are exchanged between external (line) ports. Any frame from and to the management port is not modified.

Figure 29.25 shows the operations when a frame is forwarded between line ports.

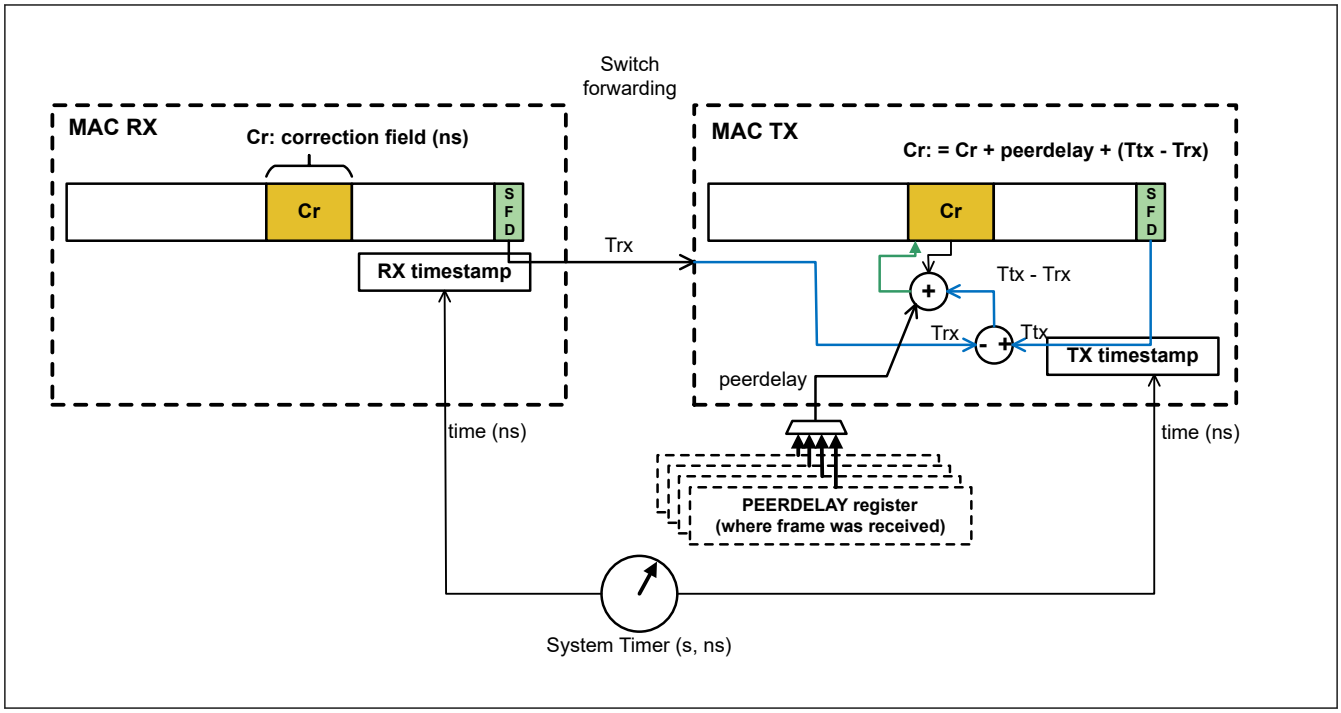


Figure 29.25 One-step correction field update

Note: No specific configuration is necessary (aside from PEERDELAY register and having a correctly running timer). The function is always active and automatically detects IEEE 1588v2 Layer 2 event frames for processing.

Using multiple timers for updating of the correction field of more than one time domain has limited functionality. If the same path is used for multiple domains, a pattern matcher for each extra domain is needed to select a different hardware timer to use.

29.4.4.5 Using One Step Update at the Management Port

When sending frames through the management port to the switch, the normal correction field updates are not available as there is no notion of a received timestamp at this interface.

However, to allow master or boundary clock implementations that uses one-step precise timestamping for CPU generated event frames such as SYNC messages, the management tag one step update feature can be used (see section 29.4.5.4. Receive Processing (CPU to Switch), ControlData[4]).

To use this function, an event frame requires special preparation by the application: The correction field must be set to the same value as nanoseconds value of the originTime field. When preparing a frame for one-step update, the application must perform the following steps:

- Read out the current time (seconds, nanoseconds) from the hardware timer.
- Prepare the event frame (such as SYNC message) and copy the time values retrieved from the timer into the originTime field of the frame. The originTime field contains seconds and nanoseconds fields.
- Set the correction field nanoseconds value of the frame the same as nanoseconds value of the originTime field (seconds are not relevant to the correction field).
- Transfer the frame to the switch using the management tag feature and setting ControlData[4] to 1 (see section 29.4.5.4. Receive Processing (CPU to Switch)) and selecting the corresponding hardware timer.

The MAC transmitter when transmitting the frame rewrites the correction field with the difference of the transmit timestamp and the value in the correction field. This function does not add to the correction field, it reads the value, subtracts it from the transmit time, and writes the result back into the correction field (after performing a 1 second modulo). Any PEERDELAY register settings are ignored. Figure 29.26 shows the principle of operations.

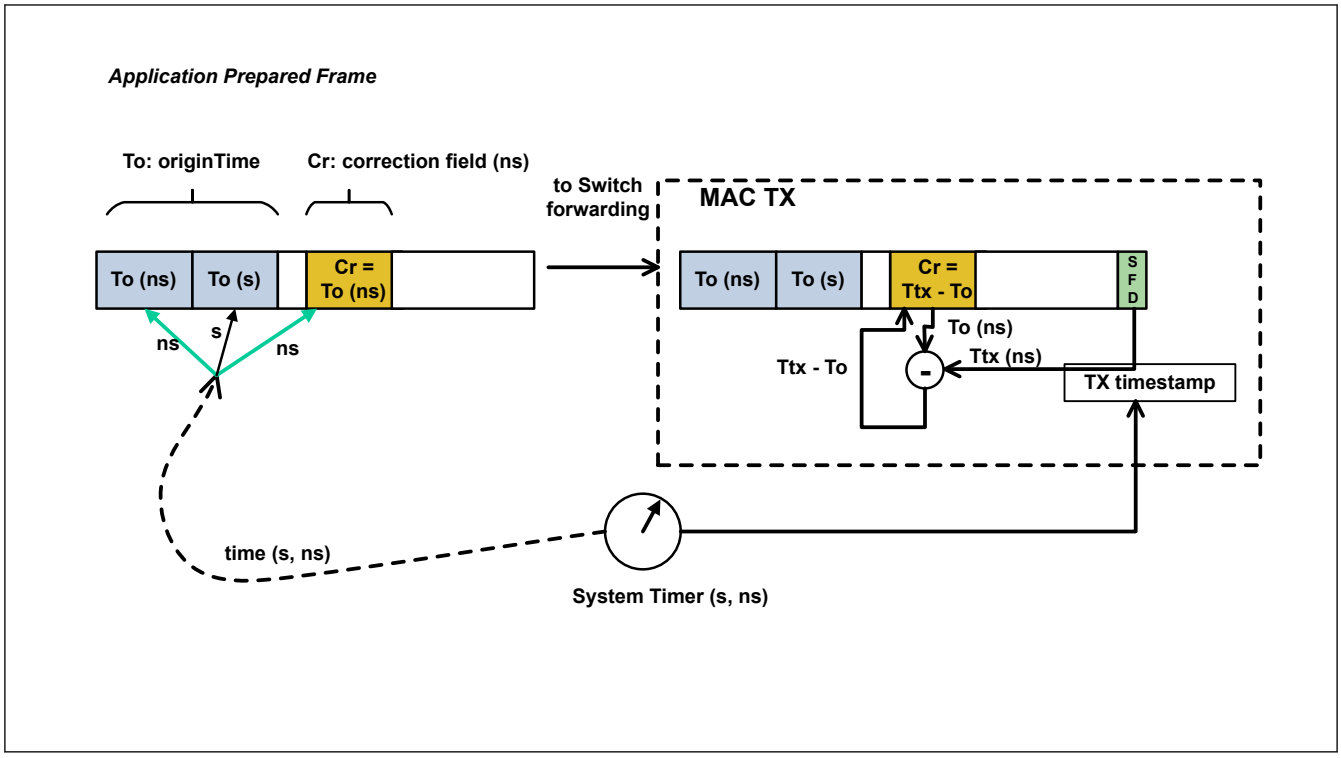


Figure 29.26 Management port one-step transmit feature

A receiver always adds the correction field value to the originTime field value and therefore, can get the precise transmit time of the frame as:

$T = To + Cr$ which is $To + Ttx - To = Ttx$.

Note: As the update function does not modify the seconds value within the frame, the application must ensure the time delay from reading the hardware timer value and the actual frame transmission at the switch output port is less than one second.

29.4.4.6 Automatic Response to Peer Delay Request Messages

(1) General

Each line port MAC can be configured to automatically respond to incoming peer delay request (PDELAY_REQ) messages. These are used in peer-to-peer clock implementations to measure the link delay between neighbor nodes. Both nodes of a link periodically transmit the PDELAY_REQ message to each other to allow both nodes to determine the link delay.

A node when receiving a PDELAY_REQ message responds with a PDELAY_RESP message. This enables the requesting device to calculate the path delay.

The switch management application periodically issues PDELAY_REQ messages to determine the link delay on each of its line ports to set PEERDELAY register of each port. The PEERDELAY register is used by the transparent clock when forwarding SYNC frames through the switch (see section 29.4.4.4. Transparent Clock Support).

In addition, the switch receives PDELAY_REQ from all its connected neighbor nodes and must respond with PDELAY_RESP messages, individually per port. The management application can be used to create such response messages, but as incoming requests have no relevance to the local system, this only represents additional processing overhead.

(2) Response Generation

To reduce overhead, each line port MAC can be instructed to automatically answer with a PDELAY_RESP message to incoming PDELAY_REQ messages. The incoming request is converted to a response and routed through the switch, however it is transmitted only at the port where it was received (emulating a MAC local response). This function can be enabled in the PTPAUTORESPONSE_Pn register.

When auto response is enabled, the following functions are performed:

- A PDELAY_REQ message is identified by an IEEE 1588v2 Layer2 frame (type 0x88F7) with messageId = 2 (first octet in PTP header, see section 29.4.4.2. IEEE 1588 Message Header Structure) and the hardware timer to use is selected by matching the domainNumber in the PTP frame against the values programmed in the PTP_DOMAIN register. If no match is found, a default value per-port is used for PTP functions, programmed in the PTPAUTORESPONSE_Pn register.

The next steps occur only when a PDELAY_REQ has been identified:

- If the destination address of the frame is unicast, the source address of the incoming frame is copied into the destination address of the frame. Otherwise, the destination address is not modified (if the neighbor sends the request with multicast the response returns to the same multicast).
- The source address of the frame is set to the MAC address (MAC_ADDR_0_Pn/MAC_ADDR_1_Pn registers) of the port.
- Address lookup and learning are suppressed for the frame. The frame is transferred to the switch but is marked for exclusive forwarding to the same port (this is an exception of the normal forwarding rules which do not forward to the same port). Therefore, normal statistics occur as with any other frame.
- When the frame is forwarded, it is converted into a one-step PDELAY_RESP by the MAC transmitter according to section 29.4.23.3. References, (2), as follows:
  - The messageId is changed to 3 (PDELAY_RESP).
  - The twoStepFlag bit within the flags field of the header is cleared (bit 1 of first octet, see section 29.4.23.3. References, (2)), to indicate a one-step response.
  - Adds the transient time (not considering the PEER_DELAY register value of the port) to the correction field.
  - The sourcePortIdentity is set according to the values from registers PTPCLOCKIDENTITY1_Pn, PTPCLOCKIDENTITY2_Pn, and PTPAUTORESPONSE_Pn. See Table 29.18.
  - The requestReceiptTimestamp field (10 octets at offset 34) is set to 0.
  - The original sourcePortIdentity field of the frame is copied into the requestingPortIdentity field (10 octets at offset 44).
  - All other bits/fields of the request are left unchanged and therefore are copied unmodified into the response message.
  - The CRC of the message is replaced accordingly.

Table 29.18 PTP auto response header sourceportIdentity mapping

Register[Octet]	Octet offset in PTP header, name
PTPCLOCKIDENTITY1_Pn.CLK_IDENTITY0[7:0]	20, portIdentity.ClockIdentity[0]
PTPCLOCKIDENTITY1_Pn.CLK_IDENTITY1[7:0]	21, portIdentity.ClockIdentity[1]
PTPCLOCKIDENTITY1_Pn.CLK_IDENTITY2[7:0]	22, portIdentity.ClockIdentity[2]
PTPCLOCKIDENTITY1_Pn.CLK_IDENTITY3[7:0]	23, portIdentity.ClockIdentity[3]
PTPCLOCKIDENTITY2_Pn.CLK_IDENTITY4[7:0]	24, portIdentity.ClockIdentity[4]
PTPCLOCKIDENTITY2_Pn.CLK_IDENTITY5[7:0]	25, portIdentity.ClockIdentity[5]
PTPCLOCKIDENTITY2_Pn.CLK_IDENTITY6[7:0]	26, portIdentity.ClockIdentity[6]
PTPCLOCKIDENTITY2_Pn.CLK_IDENTITY7[7:0]	27, portIdentity.ClockIdentity[7]
PTPAUTORESPONSE_Pn.PORTNUM0[7:0]	28, portIdentity.PortNumber[0] (msb)
PTPAUTORESPONSE_Pn.PORTNUM1[7:0]	29, portIdentity.PortNumber[1] (lsb)

(3) Usage Notes

- Auto response generation is available only for Layer 2 frames (not with UDP/IP).
- Auto response forces the MAC RX to strip the CRC. The TX appends a new CRC.

- When using the recommended PTP multicast destination address (01-80-c2-00-00-0e), the register MGMT_CONFIG must have management enabled bit (MGMT_EN) set, or at least the management frame discard bit (MGMT_DISC) cleared. Otherwise, the frame is discarded as this is a BPDU address.
- The normal priority resolution functions apply to the frame when forwarding. The PRIORITY_TYPE1 and PRIORITY_TYPE2 registers can be used to set the priority of Layer 2 PTP messages.
- The destination MAC address change is performed at the MAC RX. Hence if mirroring is active, it would show the frame with the already modified destination address.
- Although not required by PTP, a single VLAN tag is supported for PTP messages.
- If the port has VLAN input/output manipulation functions active, those settings are ignored for the response generation.
- The default timer to use for auto response generation is configured in the register PTPAUTORESPONSE_Pn.

29.4.4.7 Adjustable Timer

(1) Overview

The Adjustable Timer implements the Free Running Counter (FRC). The switch has two hardware timers. These timers are used for:

- Generating the timestamps for received and transmitted frames
- Being the time reference for the TDMA scheduler
- Providing a hardware time accessible to software using registers

Through dedicated correction logic, the timer can be adjusted allowing synchronization to a remote master and provide a synchronized timing reference to the local system. The timer can be configured to cause an interrupt after a fixed period of time to allow synchronization of software timers or perform other synchronized system functions.

The timer is usually used to implement a period of 1 second, hence its nanoseconds value ranges from 0 to $1 \times 10^9 - 1$. The period event can trigger an interrupt and software can then maintain time values or timed actions as required.

These two hardware timers of the switch are connected to four units of PTP Pulse generators (see section 29.4.20. PTP Pulse Generator) whose outputs are connected to external pins (see Figure 29.1).

The timer is programmed via the host interface. section 29.4.22.4. Timer Configuration Delay describes the delays relevant for programming the timer with higher precision, assuming that the system is able to also characterize the delay to the host interface of the switch.

(2) Adjustable Timer Implementation

The adjustable timer consists of a programmable counter/accumulator and two correction counters. The periods of the counters and its increment rate are freely configurable allowing very fine tuning of the timer.

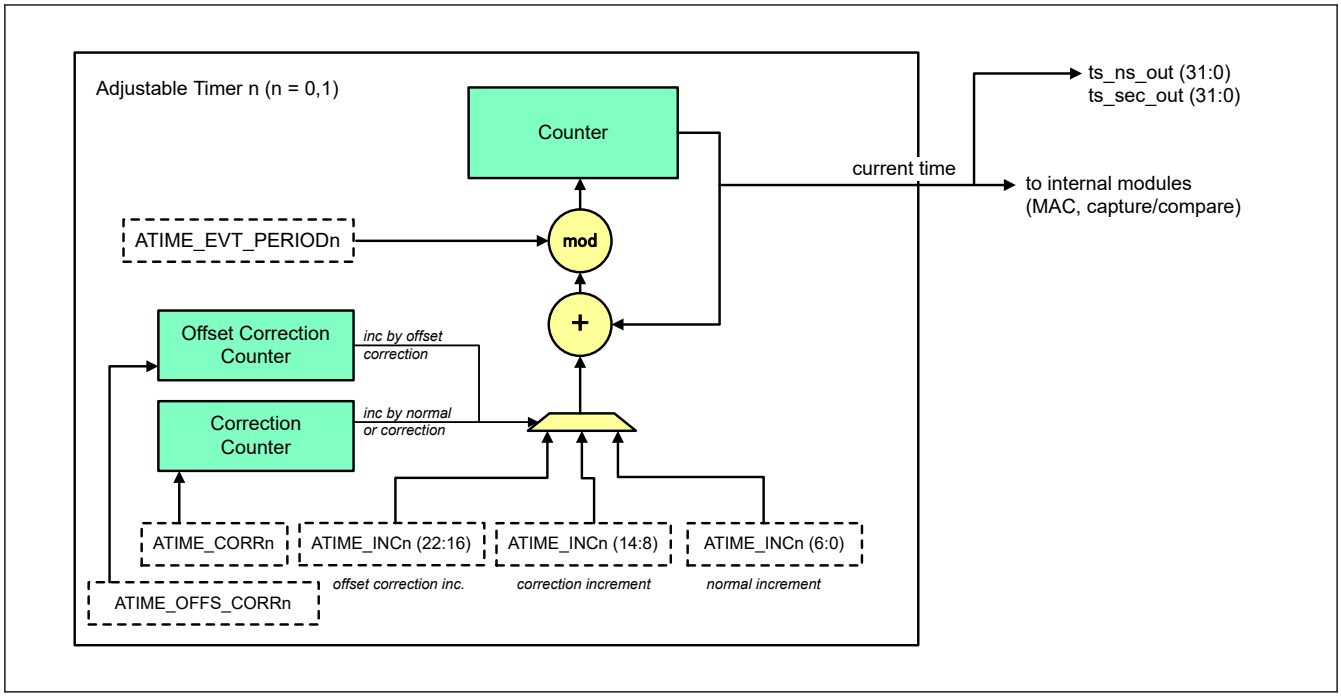


Figure 29.27 Adjustable timer implementation detail

(3) Normal Timer Operation

The counter (timer) is producing the current time. Each clock cycle is ts_clk: Timer module clock. The timer module can operate independently from all other clock domains. A constant value is added to the current time as programmed in the ATIME_INCn register. The ts_clk operates at 125 MHz, so the increment is to set to 8 representing 8 ns.

The period, configured in register ATIME_EVT_PERIODn, defines the modulo when the counter has to wrap around. In a typical implementation, the period is set to 1×10^9 so the counter wraps every 1 second and hence all timestamps represent the absolute nanoseconds within the 1 second period. When the period is reached, the counter wraps to start again according to the period modulo. This means it does not necessarily start from 0 but instead the counter is loaded with the value (current + inc − 1×10^9) assuming the period was set to 1×10^9.

(4) Drift Correction

The correction counter operates fully independently and increments by 1 with each clock cycle (ts_clk). When it reaches the value configured in ATIME_CORRn, it restarts and instructs the timer once to increment by the correction value, instead of the normal value. The normal and correction increments are configured in register ATIME_INCn. To speed up the timer the correction increment would be larger than the normal increment value. To slow down the timer, the correction increment is smaller than the normal increment value. The correction counter only defines the distance of the corrective actions, not the amount. This allows very fine corrections (and hence low jitter) happening in the range of 1 ns independently from the chosen clock frequency.

(5) Offset Correction

The offset correction counter operates fully independently and decrements by 1 with each clock cycle (ts_clk), when loaded. It can be loaded with a value by writing register ATIME_OFFS_CORRn. The timer does not start until an offset correction has been written into ATIME_OFFSETn (the ATIME_OFFS_CORRn must be written before writing the ATIME_OFFSETn value). When a value is written into ATIME_OFFSETn, the offset correction counter is loaded with ATIME_OFFS_CORRn and starts. When it reaches 0, it decrements the ATIME_OFFSETn value and applies the offset correction increment as defined in ATIME_INCn to the time counter. If the ATIME_OFFSETn value is not 0 after it was decremented, the counter is reloaded with ATIME_OFFS_CORRn and the cycle repeats. This repeats until the value in ATIME_OFFSETn reaches 0, after which no more corrections are performed. With this correction, it is possible to shift the timer to another time without causing an immediate jump in time. When the offset correction has completed, the ATIME_OFFSETn register reads out 0 and the offset event interrupt can optionally be triggered.

Alternatively, instead of distributing the offset correction over time using the offset correction timer, it is possible to immediately change the current time by the offset amount. This leads to a jump of the timer to the time current time + offset.

Setting ATIME_OFFS_CORRn to 0 and then writing the offset into ATIME_OFFSETn achieves this. A positive or negative value can be written to adjust the timer accordingly. The value written to ATIME_OFFSETn must be corrected by adding 2 × ts_cyc (timer clock period) to account for an internal delay in the timer update logic.

29.4.5 Management Port Specific Frame Tagging

To support the CPU through a normal networking interface (GMII), some special side band information required for implementing management protocols requires the addition of such information into frames exchanged between the CPU and the switch. This function is available only on the dedicated management port.

Management frames are frames that are identified by their destination address or type such as Bridge Protocol Data Units (BPDU) as defined in (7) Bridge Protocol Frame Resolution (destination 01-80-c2-00-00-xx).

Instead of only affecting management frames, the function can be optionally configured to include the side band information into all frames transferred from the switch to the CPU.

The CPU has the flexibility to insert control information into any frame it transfers to the switch. The hardware inspects all frames received at the management port and remove the extra data as necessary.

29.4.5.1 Tagged Frame Format

The additional control information is added following the frame source address marked by a programmable (proprietary) Ether type (ControlTag). If the tag exists, it is always the very first in the frame (before any VLAN tags).

The following information is added:

- ControlTag:
  An identifier that the control data is present within the frame (default: 0xE001)
- ControlData:
  2 octets following the ControlTag with control information
- ControlData2:
  4 additional octets used to carry 32-bit frame timestamps or destination port masks depending on transmit or receive.

All data following the ControlData2 are from the original frame. For example, any VLAN tags are found in this portion of the frame. The control tag, if present, is always the very first tag of a frame.

Table 29.19 Management port frame tagged frame format

7 octets	PREAMBLE
1 octet	SFD
6 octets	DESTINATION ADDRESS
6 octets	SOURCE ADDRESS
2 octets	ControlTag (default: 0xE001)
2 octets	ControlData
4 octets	ControlData2 (timestamp, portmask)
2 octets	Type/Length
0 to 1500/9000 octets	PAYLOAD DATA
0 to 42 octets	PAD
4 octets	FRAME CHECK SEQUENCE

The frame is extended by 8 bytes and the receiving MAC must be enabled to accept such enlarged frames as necessary.

29.4.5.2 Byte Order

The first octet of ControlData is the MSByte (bits [15:8]) and the 2nd octet of Control Data is the LSByte (bits [7:0]) of the 16-bit data field.

The first octet of ControlData2 is the MSByte (bits [31:24]) and the 4th octet of ControlData2 is the LSByte (bits [7:0]) of the 32-bit data field.

29.4.5.3 Transmit Processing (Switch to CPU)

When the switch forwards a frame to the management port to the host CPU, the following information is added into management (or all) frames.

Table 29.20 Management frame transmit tagging

Field	Bit#	Description
ControlData	[1:0] [3:2] Reserved, unused	Port number where the frame was received from. Ports are enumerated starting from 0 to 2.
	[4]	Timer used for timestamping the frame. This field contains the hardware timer used for timestamping the frame. 0: Timer 0 1: Timer 1
	[5]	Reserved, unused
	[6]	RED period indication. When this bit is 1, indicates whether the start of frame was received during the RED period (as indicated by the TDMA engine).
	[15:7]	Reserved, unused
ControlData2	[31:0]	Received timestamp of the frame. The 32-bit nanoseconds value when the Start Frame Delimiter (SFD) was detected at the port where the frame was received.

29.4.5.4 Receive Processing (CPU to Switch)

When the CPU sends a frame to the switch, the tagging function checks the frame and removes the optional control tag information from the frame if present. The CPU is free to add the tag to a frame as necessary or leave it out for normal frames that do not need special treatment.

When the tagging function detects the control tag (set by MGMT_TAG_CONFIG.TAGFIELD), it removes all 8 octets from the frame before it hands the frame over to the switch.

The control tag information for received frames provides the following information.

Table 29.21 Management frame receive tagging fields (1 of 2)

Field	Bit#	Description
ControlData	[0]	Forced Forwarding If set to 1, the frame is forwarded to all ports as defined in the destination port mask defined in ControlData2[2:0]. If set to 0, the destination port mask is ignored and the frame is forwarded normally.
	[1]	Forced Forwarding, include filtering. If set to 1 together with forced forwarding, normal filtering of the destination port mask applies (disabled ports are removed from the list). If set to 0, the frame is forwarded also to disabled ports. This applies to BPDU frames only. Normal frames are always filtered, meaning it can never be transmitted to disabled ports.
	[2]	Forced Forwarding, use management priority. If set to 1 together with forced forwarding, the priority of the frame is taken from the section 29.3.1.9. MGMT_CONFIG : Management Configuration Register, otherwise, the frame is prioritized as any other frame based on VLAN, IP.
	[3]	Mark frame for transmit timestamping. When set to 1, the frame transmit timestamp is latched into the corresponding transmit timestamp register of the port when the frame is transmitted. See section 29.4.4. Timestamping Functions (TSM) for details on timestamping functions.
	[4]	Mark frame for special one-step correction field update. When enabled, the correction field of the frame is subtracted from the TX timestamp on transmission and the result replaces the correction field. The application needs to prepare the frame specifically to use this update method as described in section 29.4.4.5. Using One Step Update at the Management Port.
	[5]	Suppress PRP trailer append. If set to 1, suppresses the switch adding a redundancy control trailer (RCT) when sending the frame at a port defined in the PRP group (bypassing the RedBox function). The bit has no effect if transmission occurs at a ports other than PRP group. If set to 0, the switch can append the RCT as needed. Must be 0 if ControlData[6] = 1.
	[6]	Force PRP trailer append. If set to 1, indicates that the transmitter must append a redundancy control trailer (RCT) when sending the frame at a port defined in the PRP group. This enforces trailer append, even if the frame is not duplicated to both redundant networks. Depending on bit 7, either the switch generates a sequence number, or the application can provide the sequence number that should be used. The bit has no effect if transmission occurs at a ports other than PRP group. If set to 0, no trailer append request is given. The bit has an effect only, if ControlData[5] = 0 and is ignored otherwise.
	[7]	Use provided sequence number for the PRP trailer. If set to 1, indicates that the sequence number for the frame to be used when appending the RCT is found in ControlData2[31:16]. If set to 0, the switch automatically generates a sequence number when forwarding the frame to the PRP ports. This requires that the source address of the frame is stored in the MAC address table (either a dynamic or static entry). The bit has an effect only, if ControlData[6] = 1 and is ignored otherwise. When using a provided sequence number (bits [7:6] = 11b), the sequence number/history memory is not updated. Therefore, if a following frame is sent without this method, an arbitrary/ older sequence number from the memory is added (this may be useful for testing out of sequence error scenarios).
	[8]	Reserved, (write 0 always).
	[9]	If set to 1, override destination queue. The destination queue is taken from bits [12:10] and any priority resolution is ignored.
	[12:10]	Destination queue. When bit [9] is set this queue is used and any priority resolution is ignored.
	[13]	If set to 1, override timer selection. Enables overriding the timer to be used for timestamp operations with the value in bit [14].
	[14]	Timer selection. These bits indicate the timer to use for timestamp operations when bit [13] is set to 1. 0: Timer 0 1: Timer 1
	[15]	Reserved, (write 0 always).

Table 29.21 Management frame receive tagging fields (2 of 2)

Field	Bit#	Description
ControlData2	[2:0]	Destination port mask. Relevant only if ControlData forced forwarding bit 0 is set. It then presents the list of ports where the frame should be forwarded to. One bit per port. A bit set to 1 causes the frame being forwarded to that port. Bit 0 = port 0, Bit 1 = port 1, and so on.
	[8:3]	Reserved, set 0.
	[15:9]	Transmit timestamp ID. When the frame is marked for transmit timestamping, these bits are a user-defined value that gets stored with the timestamp. This serves as a way of correlating a timestamp to a frame for the software layer.
	[31:16]	When ControlData[7] = 1, the 16-bit sequence number for use with the frame when appending the redundancy control trailer (RCT). Reserved otherwise.

29.4.5.5 Module Configuration

The tagging module is configured by the register MGMT_TAG_CONFIG.

The module can either insert the control information into every frame sent to the host CPU or only into dedicated management frames (BPDUs).

In addition to BPDU frames, two types (PRIORITY_TYPE1/2) can be defined that cause the tag being inserted if it should not be done on all frames. When using this feature, the type field compare can be done only for the first type field in the frame (it cannot skip VLAN tags at this point in the data path where the decision for the tag insertion is done).

When the CPU sends a frame to the switch, it can insert the control information as needed, every frame can be treated individually and software can decide for the best implementation on a frame-by-frame basis.

29.4.5.6 Usage notes

The MAC of the management port removes the management tag but counts the bytes of the management tag in the statistics counters. If the frame from the CPU including the management tag is between 64 and 71 bytes, the resulting frame after the management tag removal is less than 64 bytes. This frame is padded when transmitted by the transmit MACs.

Note that padding bytes added automatically are not always zero value. If non-zero value padding is not desirable, prepare the frame more than 71 bytes including management tag by software.

29.4.6 MAC EEE Support

29.4.6.1 Overview

Energy Efficient Ethernet (EEE) is defined in IEEE 802.3az. It defines energy saving modes for backplane applications allowing the PHY device to power down the link during idle times.

The MAC Core implements the necessary functionality to signal Low Power Idle (LPI) to the PHY and provides the necessary handshaking signals to the (external) user application (LPI Client Application).

The MAC includes the necessary timers and a control register to allow for the following usage scenarios:

1. Fully autonomous operation:
   The MAC integrated timers detect idle and initiate LPI transmission to power down the link when the idle time is reached. It also automatically wakes up and performs the Tw_sys_tx wake up delay before transmitting traffic.
2. Software controlled operation:
   A control register (EEE_CTL_STAT_Pn) within the MAC allows controlling the transmission of LPI (lpi_req) and holding the transmitter during wakeup (lpi_txhold). This allows software to implement idle detection and wake up timers.

Figure 29.28 shows the implementation of the MAC core.

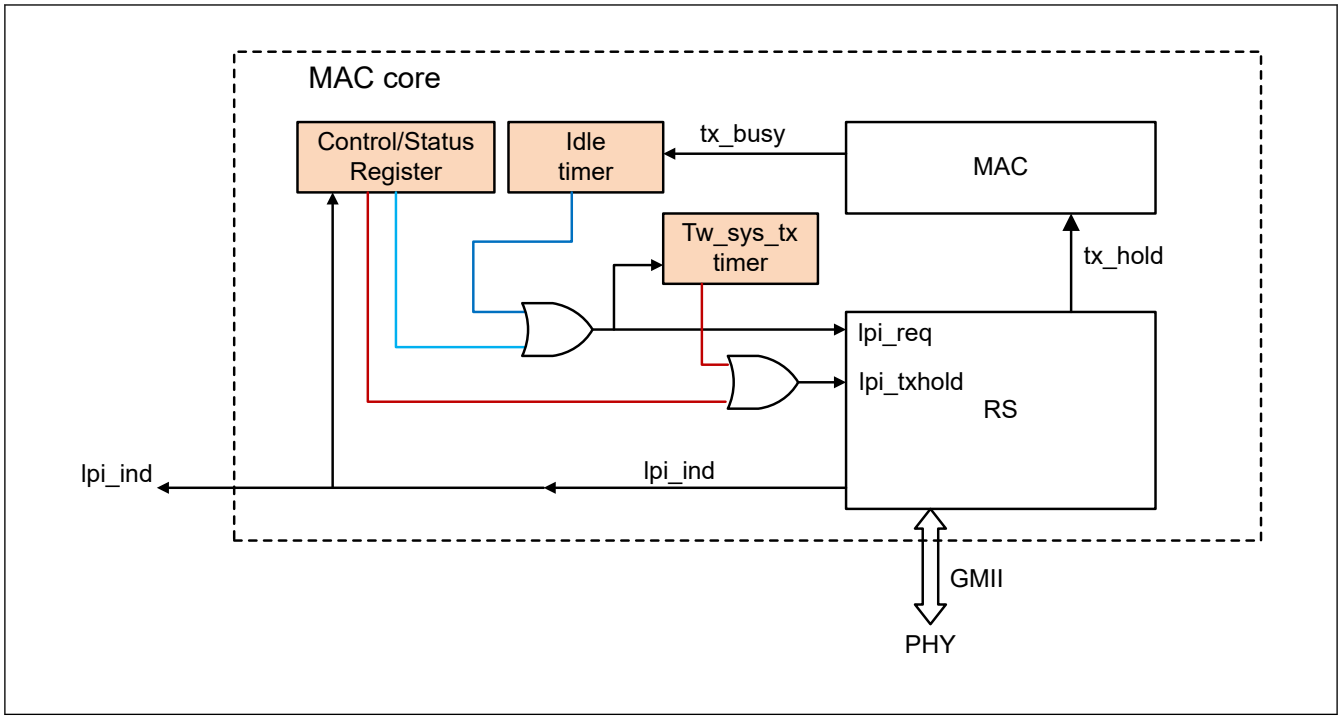


Figure 29.28 MAC extension for autonomous EEE operation

The following three functions are integrated with the MAC core on every line port:

- A control/status register. Allows software to configure the mode of operation (full autonomous with timers, or software controlled) and inspect the received LPI indication status.
- An idle timer. This timer asserts lpi_req when the MAC does not transmit any data for the configured amount of time. It causes the RS layer to transmit LPI.
- A wakeup timer (Tw_sys_tx). This timer prevents the MAC transmitter from transmitting traffic for a certain period after waking up. It keeps the lpi_txhold signal asserted after lpi_req deasserted.

All timers are programmable through registers under software control.

29.4.6.2 PHY Interface Encodings for EEE

The MAC general purpose PHY interface allows interconnecting to any PHY interface using a PHY interface converter as needed. Table 29.22 and Table 29.23 list the encodings for standard GMII and MII to allow transferring the Low Power Idle (LPI) sequences between MAC and PHY.

(1) GMII Encodings for EEE

Table 29.22 shows the interface encodings. When no frame is transferred on the interface, specific idle sequencing can be encoded on the GMII asserting the error signal while data valid is kept low.

Table 29.22 GMII encodings for EEE

rxdv/txen	rxer/txer	rxd/txd[7:0]	Description
1	x	0xXX	Normal frame transfer
0	0	0xXX	Normal interpacket gap (idle)
0	1	0x01	Low Power Idle indication
0	1	0x0F / 0x1F	Carrier Extend / Error during half-duplex operation only (not supported)
0	1	others	Reserved

(2) MII Encodings for EEE

Table 29.23 shows the interface encodings. When no frame is transferred on the interface, specific idle sequencing can be encoded on the MII asserting the error signal while data valid is kept low.

Table 29.23 MII encodings for EEE

rxdv/txen	rxer/txer	rxd/txd[3:0]	Description
1	x	0xX	Normal frame transfer
0	0	0xX	Normal interpacket gap (idle)
0	1	0x1	Low Power Idle indication
0	1	others	Reserved

29.4.7 MAC Transmit Rate Control

29.4.7.1 General

Every MAC implements a rate control function to allow reducing the average transmitted traffic load on the link. When enabled, the transmission bandwidth can be configured to use from 1% to 80% of the available link bandwidth. When disabled, 100% link bandwidth is used.

The implemented algorithm uses the credit based shaper as defined in section 29.4.23.3. References, (3). The algorithm maintains a credit counter for the traffic being transmitted by the MAC. The counter is incremented whenever a frame is waiting for transmission, or if the current count is negative. It decrements when a frame is transmitted. A frame can be transmitted only when the credit counter has a value of ≥ 0. If no frame is waiting and the credit counter is > 0, it is reset to 0.

The increment rate of the counter is defined by the idle slope. The decrement rate of the counter is defined by the send slope. The idle slope is configurable to define the bandwidth that can be allocated for the traffic. The send slope is defined by the link speed.

29.4.7.2 Configuration Settings

The send slope depends on the transmission speed of the link (10 or 100 or 1000 Mbps) and is defined by the current mode of operation of the MAC. The idle slope is defined by configuration (see section 29.3.6.16. IDLE_SLOPE_Pn : Port n MAC Traffic Shaper Bandwidth Control (n = 0 to 3)) depending on the (relative) bandwidth that should be allocated for the traffic.

The idle slope can be configured with a granularity of 1/64 bit. That is, the credit counter increments by 1 byte when the increment has reached 512 (8 × 64). The implementation may have restrictions on the granularity depending on byte to clock ratio of the interface, however, the calculations are always based on the 1/64-bit steps. This means the idle slope allows to specify bandwidth usage in steps of 0.2% (1 / (1 + 512)) of the link capacity. Using an 11-bit value allows to configure the bandwidth allocation from 0.2% to 80%.

The resulting bandwidth is calculated as:

bandwidth [%] = 1 / (1 + 64 / IdleSlope × 8) = 1 / (1 + 512 / IdleSlope)

IdleSlope = (512 × bandwidth) / (1 − bandwidth)

Table 29.24 Credit based shaper bandwidth setting examples (1 of 2)

IdleSlope Value	Allocated Bandwidth	Notes
2	1 / (1 + 32 × 8) = 0.4%	Counter decrements 8-bit per byte transmitted. During idle, it increments 2/64 bit per byte clock → requires 32 × 8 bytes clock cycles to increment by 1 byte.
4	1 / (1 + 16 × 8) = 0.77%	During idle, it increments 4/64 bit per byte clock → requires 16 × 8 bytes clock cycles.
6	1 / (1 + 10.6 × 8) = 1.1%	During idle, it increments 6/64 bit per byte clock → requires 10.6 × 8 bytes clock cycles.

Table 29.24 Credit based shaper bandwidth setting examples (2 of 2)

IdleSlope Value	Allocated Bandwidth	Notes
64	1 / (1 + 8) = 11%	Counter decrements 8-bit per transmit clock cycle. During idle, it increments 1 bit per transmit clock cycle hence requires 8 clock cycles.
57	~ 1 / 10 = 10%	—
73	~ 1 / 8 = 12.5%	—
170	~ 1 / (1 + 3) = 25%	—
256	1 / (1 + 0.25 × 8) = 33%	—
512	1 / (1 + 1) = 50%	—
1536	1 / (1 + 0.04 × 8) = 75%	—
2046	80%	—

29.4.8 Port Based Access Control (802.1X)

29.4.8.1 General

The IEEE 802.1X specification defines procedures to allow a port of the switch to be removed from the network until it is authorized by a higher layer authentication function.

Use of IEEE 802.1X is fully controlled by the application (software) usually connected through the management port. The application exchanges authentication messages through the switch with the devices that connect to the switch line ports. The switch does not interpret the messages aside from detecting the authentication relevant frames (EAPOL) and restricting forwarding if a port is configured as unauthorized.

Authentication of a port is controlled for each port individually through its configuration register AUTH_PORTn.

29.4.8.2 Terms & Definitions

- PAE:
  Port Access Entity — An entity associated with a port that implements the authentication protocols of interest.
- EAP:
  Extensible Authentication Protocol — Messages for encapsulation in frames for exchange between PAEs.
- EAPOL:
  EAP Over LAN — Encapsulation of EAP frames is defined in IEEE 802.1X:
  - Destination MAC address is either the PAE group address 01-80-c2-00-00-03 or the defined unicast address of the port configured in the MAC of the port
  - The Ether type is set to the PAE Ethernet Type: 0x888E
  - The frame shall not be VLAN tagged (VID ≠ 0) but may be priority tagged
- Port Controlled Direction — A port, while unauthorized, can operate with controlled direction in or both:
  - When using controlled direction in, the port only restricts incoming traffic but allows other ports to forward traffic to it for transmission. In controlled direction mode both, forwarding traffic from other ports is not allowed.
  - An exception from this rule is the dedicated EAPOL management port, which must be allowed to forward EAPOL (or other) frames even to ports that have controlled direction both to implement the necessary authentication protocol.

29.4.8.3 Functions

When a port is configured as unauthorized, the MAC/switch port receives behavior changes as follows:

- Accept EAPOL frames and forward them exclusively to a defined port (such as EAPOL port, which is typically the management port)
- Non-EAPOL frames can either be discarded or be allowed for limited forwarding to a specific group of ports (guest VLAN). For example, allowing DHCP before authentication

- Allow or discard non-EAPOL frames received from other ports to be forwarded for transmission on the port (controlled directions in or both)
- Allow or discard BPDU frames

When a port is configured as authorized, normal destination lookup and forwarding resolution occur with additional filtering of unauthorized destination ports which are configured in controlled direction both.

29.4.8.4 Forwarding Decision

The following decision is implemented in each port receive data path to determine the destination ports of a frame or discard it depending on authentication status of a port.

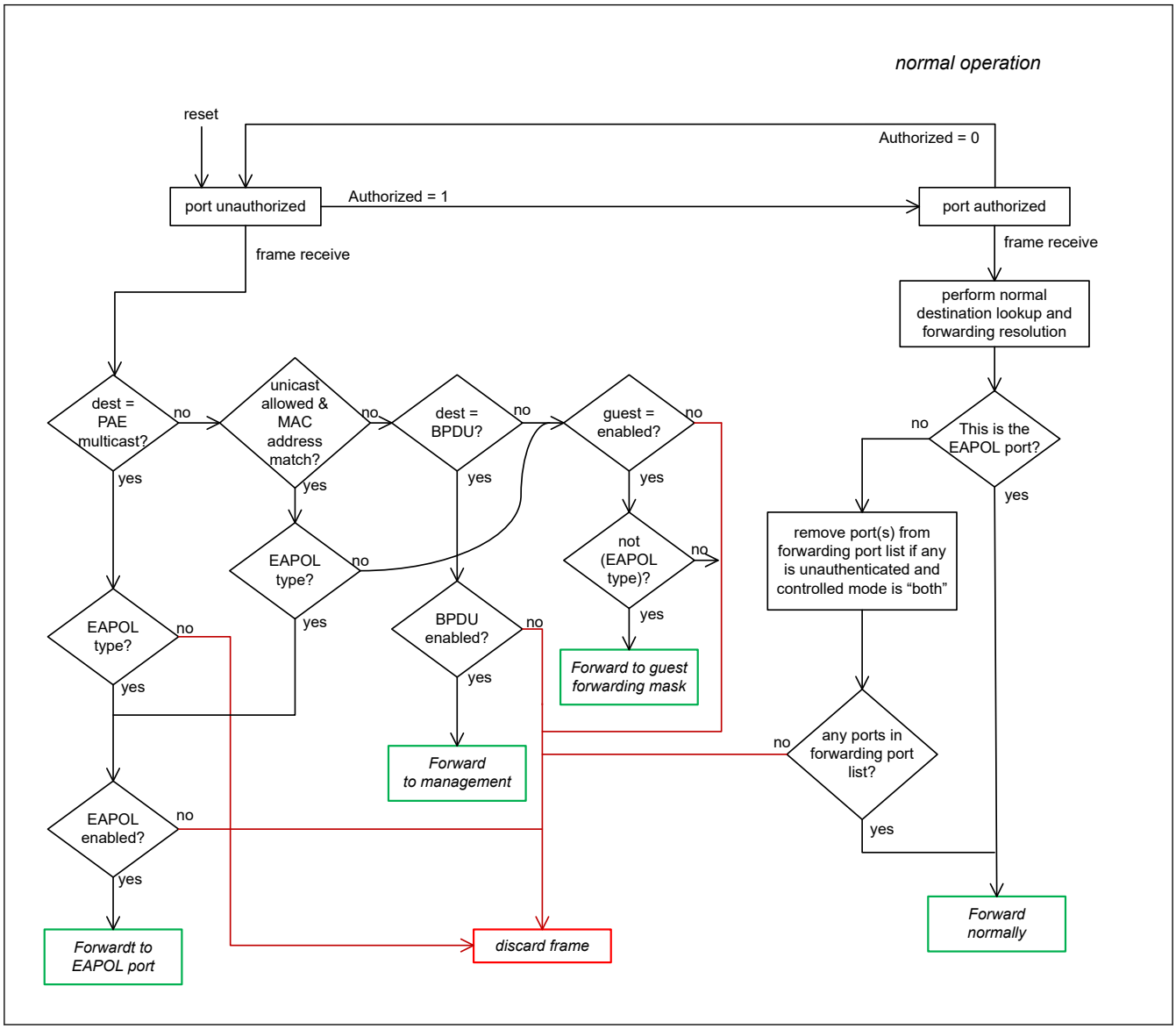


Figure 29.29 IEEE 802.1X forwarding decision flow

29.4.8.5 Usage aspects

The following points should be considered when implementing the authorization application:

- Source address learning should be disabled for an unauthorized port (see section 29.3.1.8. INPUT_LEARN_BLOCK : Input Learning Block Register) to avoid invalid addresses entering or overwriting the MAC address table (MAC address spoofing). The authorization application must enter the address into the MAC address table once the remote device is authenticated.

- With learning disabled and guest traffic allowed for an unauthorized port, the source address is not entered into the address table of the switch. Therefore, all traffic sent to such unauthorized port from another (answering) port is flooded. Flooding is limited to either the guest port mask of the (answering) port if it is unauthorized or the usual U/M/BCast flooding masks if the port is authorized. An exception is the management port, which can use forced forwarding to direct the frame to the intended port only.
- To enable EAPOL reception, the management port functionality must be enabled. That is, register MGMT_CONFIG bit 6 (enable) must be 1 and bit 7 (discard) must be 0. Otherwise, frames are discarded before EAPOL checking can be done.
- If a port is unauthorized, its cut through bit should be cleared to avoid any invalid frames entering the switch. Cut Through should be enabled only on authorized ports.

29.4.9 Interrupts

Global interrupt signal (ETHSW_INTR) and individual module interrupt signals are available.

For global interrupt, the following interrupt sources are available and can be individually enabled through the INT_CONFIG register.

The INT_STAT_ACK register allows inspecting interrupt status and acknowledging the individual pending interrupts by writing a bit with 1 to clear the interrupt latch except sources that have individual configuration and status and acknowledge registers.

Sources that have individual registers need to write individual registers.

Table 29.25 Global Interrupt (ETHSW_INTR) sources

Source	Bit symbol of INT_CONFIG register	Description	Individual Config register	Individual Status/Ack register	Individual interrupt signal
MDIO	MDIO1	MDIO master transaction completion	—	—	—
Lookup Source	LK_NEW_SRC	A frame with an unknown source MAC address was received.	—	—	—
PHY Link Change	IRQ_LINK[2:0]	Change on any of the PHY link status	—	—	—
TX Timestamps	TSM_INT	Per Line Port indication that outgoing frame timestamp is available.	TSM_CONFIG	TSM_IRQ_STAT_ACK	—
MAC EEE	IRQ_MAC_EEE[2:0]	The MAC integrated functions for managing Energy-Efficient Ethernet (EEE) can trigger interrupts from changing power states.	—	—	—
DLR Module	DLR_INT	Events generated by the Device Level Ring (DLR) module	DLR_IRQ_CONTROL	DLR_IRQ_STAT_ACK	ETHSW_DLR
PRP Module	PRP_INT	Events generated by the redundancy network support functions (PRP).	PRP_IRQ_CONTROL	PRP_IRQ_STAT_ACK	ETHSW_PRP
Hub Module	HUB_INT	Events generated by the Hub module.	HUB_IRQ_CONTROL	HUB_IRQ_STAT_ACK	ETHSW_IHUB
Pattern Matcher Module	PATTERN_INT	Events generated by the Pattern Matcher module.	PATTERN_IRQ_CONTROL	PATTERN_IRQ_STAT_ACK	ETHSW_PTRN[11:0]
TDMA Scheduler Module	TDMA_INT	Events generated by the TDMA Scheduler module.	TDMA_IRQ_CONTROL	TDMA_IRQ_STAT_ACK	—
Address Filter	SRCFLT_WD_INT	MAC Address Source Filtering Watchdog	—	—	—
Extended Frame Parser	EFP_INT	Events generated by Extended Frame Parser	*1	*1	—

Note 1. See section 29.4.19.11. Interrupt Source.

Additionally, TDMA_GPIO signals generated in TDMA scheduler module and PTPOUT signals generated in PTP pulse generators are available as interrupts for GIC, DMAC, and ELC as well as external signals.

29.4.10 PHY Management Interface (MDIO Master)

29.4.10.1 Overview

The MDIO management interface is a two wire Management Interface. It provides a standardized method to access PHY device internal management registers. The IEEE 802.3 Clause 22 defines the bus protocol.

29.4.10.2 MDIO Frame Format

A complete frame has a length of 64 bits (32-bit preamble, 14-bit command, 2-bit bus direction change, 16-bit data). Each bit is transferred with the rising edge of the MDIO clock (MDC signal).

Table 29.26 MDIO frame formats (Read/Write)

Type	PRE	Command				TA	Data MSB LSB	Idle
		ST MSB LSB	OP MSB LSB	Addr1 MSB.LSB	Addr2 MSB.LSB			
Read	0xFFFFFFFF	01b	10b	xxxxxb	xxxxxb	Z0b	xxxxxxxxxxxxxxxxb	Z
Write	0xFFFFFFFF	01b	01b	xxxxxb	xxxxxb	10b	xxxxxxxxxxxxxxxxb	Z

Table 29.27 MDIO frame fields description

Name	Description
PRE	Preamble: 32 bits of logical 1 sent prior to every transaction
ST	Start indication: 01b
OP	The opcode defines whether a read or write operation is performed: 1. If set to 10b, a read operation is performed. 2. If set to 01b, a write operation is performed.
Addr1	The PHY device address (PHYAD). Up to 32 devices can be addressed.
Addr2	Register Address. Each PHY can implement up to 32 registers.
TA	Turnaround time. Two bit-times are reserved for read operations to switch the data bus from write to read for read operations. The PHY device will present its register contents in the data phase and drives the bus from the 2nd bit of the turnaround phase.
Data	16 bits of data written to the PHY or read from the PHY.
Idle	Between frames, the MDIO data signal is tri-stated.

29.4.10.3 Turnaround Signalling

For Read operations, the addressed PHY has to drive the data bus. This direction change is done during the two bit-times TA phase of the frame.

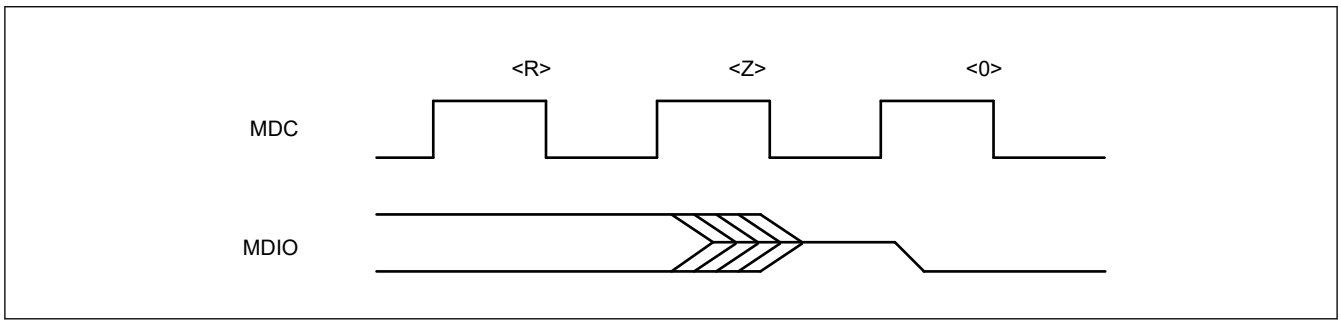


Figure 29.30 MDIO turnaround sequence

When the last bit before the TA phase (<R> in above figure) has been transmitted to the PHY, the transmitter will switch its MDIO data signal into tristate (<Z>). One clock cycle later, the PHY starts driving the MDIO data signal to 0 (<0>) and then shifts out its 16 bits of data from the addressed register. After all 16 data bits have been transferred by the PHY, it again tri-states MDIO.

29.4.10.4 MDIO Control Registers

The configuration register MDIO_CFG_STATUS is used to set the operation parameters for all MDIO transactions and must be initialized before any transactions can occur.

29.4.10.5 Transaction Types

The MDIO provides two transactions. A read transaction and a write transaction. The transactions contain the PHY device address and the register address allowing to address up to 32 devices on the bus with up to 32 registers.

The transactions can occur after the MDIO_CFG_STATUS has been initialized.

(1) Read Transaction

To read a register from the PHY the following sequence is used:

1. Setup MDIO_COMMAND with the PHY device and register address of interest and set bit 15 to trigger the read transaction.
2. Wait for transaction complete (inspecting MDIO_CFG_STATUS.BUSY).
3. Read data from MDIO_DATA.

(2) Write Transaction

To write into a PHY register, the following sequence is used:

1. Setup MDIO_COMMAND with the PHY device and register address of interest.
2. Write data into MDIO_DATA. This triggers the write transaction.
3. Wait for transaction complete (inspecting MDIO_CFG_STATUS.BUSY).

29.4.11 DLR Extension Module

29.4.11.1 Overview

The Device Level Ring (DLR) extension offers beacon frame processing on receive for the two line-ports (0 and 1) of the switch implementation.

The DLR extension module operates between the MAC receive and switch input of a port. Beacon frames are detected and the frame parameters are interpreted and stored locally for software access.

The switch forwards the beacon frames normally (possibly using cut through if available). Using a static address entry in the MAC address table allows limiting beacon frame forwarding to the two DLR enabled ports only. This prevents any beacon traffic sent to the management port or other line ports of the switch avoiding unnecessary processing load.

In addition, a loop filter is inserted into each receive data path to detect if the incoming frame's source address equals the local station MAC address. Such frames are discarded to avoid cycling of frames sent by the local device into the ring during reconfiguration times. The local node's MAC address can be configured with the DLR_LOC_MAClo/hi registers.

Any ring status change for beacon based node implementation is notified to the processor through the interrupt status. It is possible to read the received beacon frame parameters at any time. Statistics counters are implemented counting beacon frames processed.

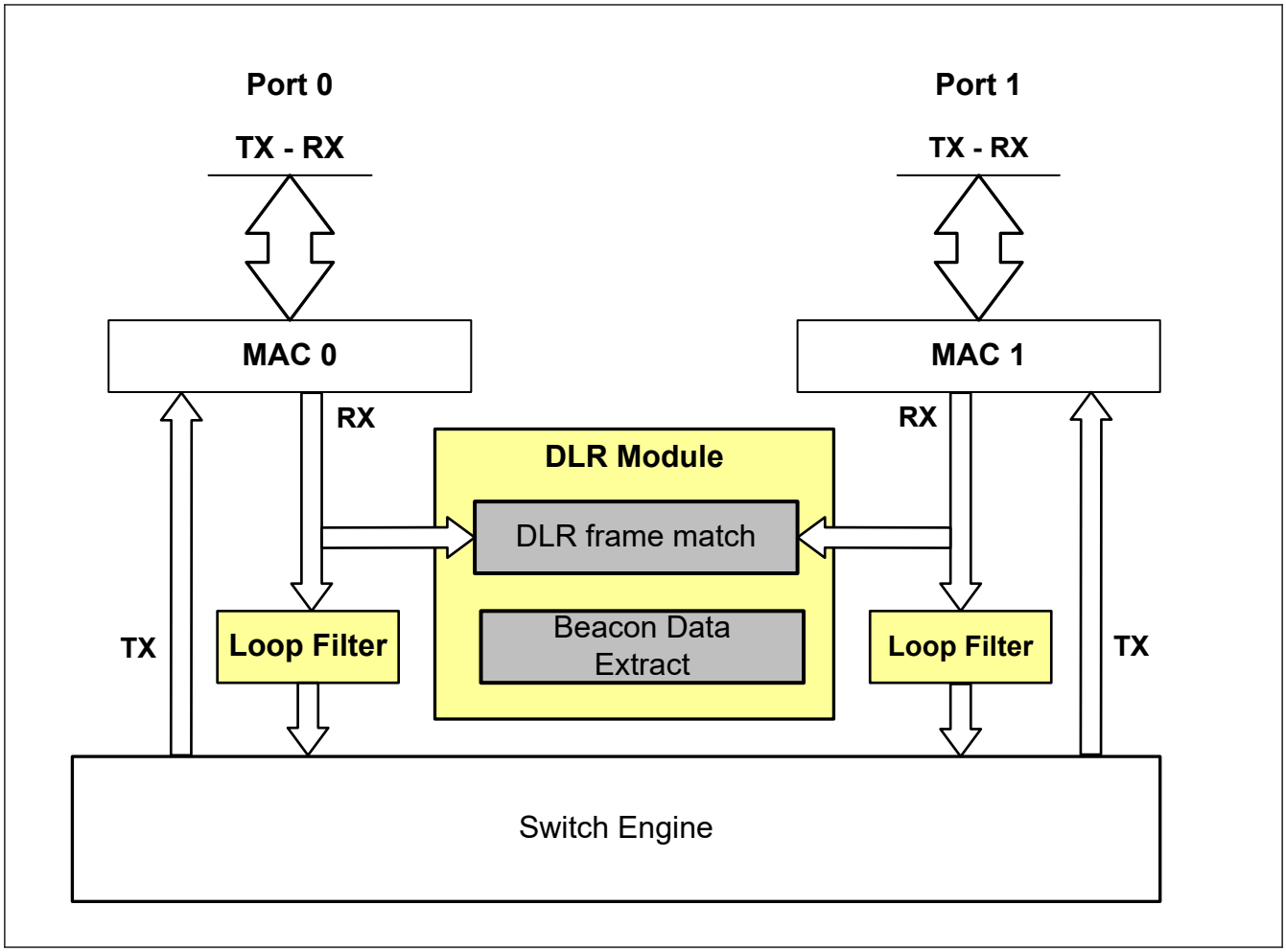


Figure 29.31 Switch with DLR extension

29.4.11.2 Beacon Frame Format

Within a DLR Network, the active ring supervisor transmits beacon frames through both of its Ethernet ports once per beacon interval (400 microseconds by default). DLR frames are using the frame format of 802.1Q. Frames are transmitted with highest priority (7). Beacon frames are DLR frames with a length of 64 bytes, excluding the preamble and the SFD bytes. Beacon frame consists of the following fields:

- Seven bytes preamble
- Start frame delimiter (SFD)
- Six bytes Destination MAC address of 01-21-6C-00-00-01
- Six bytes Source MAC address
- TPID (0x8100) and VLAN Info
- Ring Ether Type of 0x80E1
- Ring Subtype 0x02
- Ring protocol version 0x01
- Frame type 0x01
- Source port 0x00
- Source IP address
- Sequence ID
- Ring state
- Supervisor precedence

- Beacon interval
- Beacon timeout
- Frame check sequence (CRC value)

Table 29.28 Beacon frame format

Frame Length	Common DLR Protocol Header	7 octets	PREAMBLE
		1 octet	SFD
		6 octets	DESTINATION ADDRESS
		6 octets	SOURCE ADDRESS
		2 octets	TPID (0x8100)
		2 octets	VLAN info (0xE000 + VLAN ID)
		2 octets	Ring Ether TYPE (0x80E1)
		1 octet	Ring Sub Type (0x02)
		1 octet	Ring Protocol Version (0x01)
		1 octet	Frame Type (0x01)
		1 octet	Source Port (0x00)
		4 octets	Source IP Address (0x00 if source has no IP address)
		4 octets	Sequence ID
		1 octet	Ring State
		1 octet	Supervisor precedence
		4 octets	Beacon Interval
		4 octets	Beacon Timeout in microseconds
		20 octets	Reserved
		4 octets	Frame Check Sequence

Beacon frames are processed and parameters are stored locally for the software access. The following table shows the beacon frame fields.

Table 29.29 Beacon frame field definitions (1 of 2)

Term	Description	Related register
Destination Address	The destination MAC address for the beacon frame is fixed multicast address of 01-21-6C-00-00-01. This is an exclusive MAC address used only for beacon frames. Hence cut-through forwarding could be used based on this address match.	DLR_DSTlo/hi
Source Address	Source MAC address of the supervisor. 48-bit address is stored in two separate registers.	DLR_SUPR_MAClo/hi
TPID	DLR messages contain 2 octets TPID (0x8100) after the source MAC address according to IEEE 802.1Q.	—
VLAN Info	16-bit information field contains the priority field and the VLAN ID. VLAN ID is configured at the ring supervisor and received by the ring nodes. The default value for the VLAN ID is 0 when there is no VLAN ID available. The default VLAN ID does not need to be changed unless a commercial switch is being used within the ring.	DLR_STATE_VLAN (VLANINFO[15:0])
Ring Ether Type	Ether type for DLR frame is 0x80E1	DLR_ETH_TYP
Ring Sub Type	Ring Sub type value is always 0x02 for the DLR messages	DLR_ETH_STYP_VER (SUBTYPE[7:0])
Ring Protocol Version	Protocol version for the DLR messages	DLR_ETH_STYP_VER (PROTVER[7:0])
Frame Type	Frame Type value for the beacon frame is always 0x01	—

Table 29.29 Beacon frame field definitions (2 of 2)

Term	Description	Related register
Source Port	Source port value is always 0x0 for the beacon frame	DLR_ETH_STYP_VER (SPORT[7:0])
Source IP Address	IP address of the Supervisor. The default value for the IP address is 0 when there is no IP address available.	DLR_SURR_IPADR
Sequence ID	Sequence identification number of the frame	DLR_SEQ_ID
Ring State	State of the Ring network transmitted by the ring supervisor.	DLR_STATE_VLAN (RINGSTAT[7:0])
Supervisor Precedence	The Ring Supervisor Precedence value contains the user assigned precedence value to the ring supervisor. When multiple supervisors are enabled, the precedence value enables the user to select a single supervisor with highest precedence. The ring supervisor's precedence value can be any value within a range 0-255, with numerically higher values indicating higher precedence.	DLR_SUPR_MAChi (PRECE[7:0])
Beacon Interval	Interval in microsecond at which the ring supervisor sends beacon frames. Valid value is in a range between minimum 100 microseconds and maximum 100 milliseconds. Typical value is 400 microseconds.	DLR_BEC_INTRVL
Beacon Timeout	Amount of time in microsecond, nodes shall wait before timing out reception of beacon frames and taking appropriate action. Valid value is in a range between minimum 200 microseconds and maximum 500 milliseconds. Typical value is 1960 microseconds.	DLR_BEC_TMOUT
Frame Check Sequence	CRC value for the frame	—

29.4.11.3 Ring Node Functional Description

Beacon frames are detected and interpreted by the hardware so that the processor is not burdened with processing the beacon frames for ring node implementation. Any ring status change for beacon based node implementation is notified to the processor through the interrupt status. It is also possible to read the received beacon frame parameters at any time. Statistics counters are also implemented to view the number of beacon frames processed.

(1) Configuration

Following are the steps to configure the DLR Module:

- Enter the beacon destination address (01-21-6C-00-00-01) into the MAC address table as a static entry having only bits 0, 1 set in the portmask (see also section 29.4.3.8. Layer 2 Look Up Engine Operational Description). This ensures beacon frames are forwarded only between the two DLR capable ports within the switch.
- Set the beacon destination address (01-21-6C-00-00-01) within the DLR module to allow its detection (see also (2) Beacon Frame Detection):
  - Set register DLR_DSTlo = 0x006C2101 which is also the reset value.
  - Set register DLR_DSThi = 0x0100 which is also the reset value.
- Set the MAC address of the local device which is used by the loop filter.
  - Set the first 4 bytes of the MAC address in register DLR_LOC_MAClo.
  - Set the last 2 bytes of the MAC address in register DLR_LOC_MAChi.
- Set the DLR Ethernet type value of 0x80E1 to the register DLR_ETH_TYP which is also the reset value.
- Enable and control the DLR module through the DLR_CONTROL register. It also defines the timer constant (cycles required to count 1 µs) used by the timeout counters.
- Configure the destination address for Neighbor_Check_xxx and Sign_On messages (01-21-6C-00-00-02) to ensure it is always forwarded to the management port only and not forwarded between any of the line ports. There are 2 alternatives to achieve this:
  - Configure the address into the additional BPDU register MGMT_ADDRn_lo/hi.
  - Add it as static address into the address table with the portmask having only the bit of the management port set to 1.
- Any interrupt bit can be enabled/disabled through the register DLR_IRQ_CONTROL as required.

(2) Beacon Frame Detection

Received frames are inspected by the DLR module only if they match the following rules. Otherwise, the frame is ignored by the DLR module.

- Destination address must match the address provided in DLR_DSTlo/hi registers.
- Ethernet type must match the value given in register DLR_ETH_TYP.

If above rules both match, learning of the source address of the frame is suppressed, even if the port has learning enabled (to avoid the address table is constantly updated with alternating entries for the ring supervisor).

The accepted DLR frame is then forwarded to the beacon frame processing function (within the DLR module), which inspects the frame payload further to extract beacon information. However, this function accepts a frame only if the following rule matches:

- The DLR payload Frame Type field (3rd byte in payload) must have a value of 0x01.

If the DLR payload Frame Type is not matching, the frame is ignored by the beacon processing functions. However, source address learning will still not occur (as it does not depend on payload).

The switch will always process the frame normally (i.e. forwards it between port 0 and port 1), independent of the DLR module function inspecting the frame.

(3) Start Up

On start up ring node will be in IDLE_STATE and presumes the network in linear topology mode. The current state of the local ring node and other status bits are stored within the register DLR_STATUS.

Beacon frame received with invalid timer value will be ignored if ignore enabled through the DLR_CONTROL register. Invalid timeout timer value of the beacon frame will be always stored within the register DLR_INV_TMOUT irrespective of the ignore enable bit within the DLR_CONTROL register. An interrupt will be generated on receiving a beacon frame with invalid timeout timer value if enabled through DLR_IRQ_CONTROL.

On receiving a beacon frame through either port, the node shall transition to FAULT_STATE, which presumes the network in ring topology mode. An interrupt will be generated to the processor indicating a need for flushing the MAC address learning table and a change in state if enabled through the register DLR_IRQ_CONTROL. All ring supervisor parameters will be saved within register. Following parameters will be saved only during transition from IDLE_STATE to FAULT_STATE:

- Supervisor's MAC address and will be stored in the register DLR_SUPR_MAClo and DLR_SUPR_MAChi
- Supervisor's precedence value will be stored in the register DLR_SUPR_MAChi
- VLAN ID will be stored in the register DLR_STATE_VLAN
- Beacon timeout timer value will be stored in the register DLR_BEC_TMOUT

Supervisors IP address is accepted to change at any time. New IP address will always replace the old one and interrupt will be generated indicating the IP address change if enabled through the DLR_IRQ_CONTROL.

If beacon frame is received from a different supervisor with a higher precedence value or higher numeric value for MAC address with same precedence value, new beacon frame parameters will replace all the old values. An interrupt will be generated indicating change of supervisor if enabled through the register DLR_IRQ_CONTROL. Node will stay in FAULT_STATE.

If beacon frame is received from a different supervisor with a lower precedence value or lower numeric value for MAC address with same precedence value, beacon frame will be ignored. An interrupt will be generated indicating ignored beacon frame if enabled through the register DLR_IRQ_CONTROL. Node will stay in FAULT_STATE.

Ring supervisor is not expected to change parameters within the beacon frame. If parameters need to be changed, supervisor is expected to stop transmitting beacon frame for at least two beacon timeout periods before transmitting beacon frames with new parameters.

If the local node state changes back to IDLE_STATE due to beacon timeout timer expiration on both ports, it will generate interrupt if enabled through the register DLR_IRQ_CONTROL. Current interrupt status is available for software access showing beacon timeout timer asserted for both ports, need to flush MAC address learning table and state change within interrupt status register DLR_IRQ_STAT_ACK.

On receiving beacon frame through both ports and after receiving a beacon frame from active ring supervisor with ring state field set to RING_STATE_NORMAL on either one of its ports, Local node shall transition to NORMAL_STATE. Interrupt status bits will show the change in state, need to flush the unicast MAC address learning table and to stop neighbor check timeout timer when active within the software.

Note: Neighbor check time out timer (100 milliseconds) for neighbor check process (see section 29.4.23.3. References, (5)) should be implemented by the software. Software may take advantage of the stop neighbor check timeout timer bit of the interrupt status register to stop the timer.

(4) Fault Detection

One of several possible events shall cause the ring node to transition from NORMAL_STATE to other states:

- Receipt of beacon frame with state parameter set to RING_FAULT_STATE. Interrupt status will show a change in node state change and cause an interrupt if enabled.
- Receipt of beacon frame with different MAC address and higher precedence than the current active ring supervisor. In addition to state change, supervisor change bit will be asserted within the interrupt status register.
- Loss of beacon frame on both ports for a period specified by the beacon timeout period will cause the node to transition to IDLE_STATE. In addition to the state change, beacon timeout timer expire bits for both ports bit will be asserted within the interrupt status register.
- Loss of beacon frame on a single port for a period specified by the beacon timeout period will cause the node to transition to FAULT_STATE. In addition to the state change, beacon timeout timer expire bits for corresponding port bit will be asserted within the interrupt status register.

(5) Error Handling

DLR node module is able to handle several error conditions:

- Beacon frames with CRC error are detected and not interpreted for DLR node implementation and discarded before entering the switch. Beacon frame parameters are not stored for the error frame. Beacon frames with CRC error are counted and stored in the DLR statistics register DLR_RX_ERR_STAT0 and DLR_RX_ERR_STAT1 for the port 0 and port 1 respectively.
- Valid range for the beacon frame timeout timer value is between 200 microseconds to 500 milliseconds. Beacon frame from supervisor with invalid value of the beacon timeout will be ignored for processing (however are still forwarded normally) if configured through the DLR_CONTROL register. Beacon frames with invalid timeout timer are always detected irrespective of the DLR_CONTROL register configuration and invalid timeout value is stored within the register DLR_INV_TMOUT. An interrupt is also generated if enabled through the DLR_IRQ_CONTROL register.

(6) Beacon Frame Parameter Extraction

Parameters are transmitted by the Ring Supervisor through beacon frame and extracted by the node and stored within read only Registers. There are three types of beacon frame parameters depending on time of capture and store.

- Type1:
  Parameters that are not allowed to change during normal operation. This is done to avoid any unexpected change during normal operation. Parameters are always extracted and compared for the state machine evaluation but they are stored only during IDLE state or if the received beacon frame has higher precedence than the previously received beacon frame.
- Type2:
  Allowed to change at any time. Such parameters are always extracted and stored if the beacon frame is received from current Ring supervisor or node is in IDLE state or received beacon frame has higher precedence than previously received beacon frame.
- Type3:
  Always extracted from all beacon frames on any port and stored within the register. Does not have any dependency on state or supervisor or precedence. Useful for debugging purpose only.

(7) Link_Status Message Generation

When a DLR node detects a link failure a Link_Status message is generated to the other port to notify the active ring supervisor node. This feature can be automated by the switch with assistance from software using the queue trigger feature of the memory controller at the expense of sacrificing one output queue in the memory controller.

Each MAC port can enable a trigger event to the IMC for transmitting a frame on the event of a link loss via the register MAC_LINK_QTRIG_Pn. When the DLR ring is formed and the DLR state machine is in the NORMAL_STATE the ENABLE bit in MAC_LINK_QTRIG_Pn can be enabled to send the Link_Status frame in the event of a link loss.

The following steps must be performed to support Link_Status generation using the switch:

- Select a high priority queue for the Link_Status function in the DLR ports, namely QLS. This QLS can only be used for the Link_Status function and the switch must be configured to prevent frames from being sent to this queue.
- Gate the QLS queue using MMCTL_QGATE for the DLR ports by setting to 1 the bits corresponding to the DLR ports in PORT_MASK and setting QUEUE_GATE to 0x2 for the QLS queue.
- Enable the 1-frame mode (buffer mode) using the MMCTL_1FRAME_MODE_Pn for the DLR ports by setting the corresponding bits to the DLR ports in Q_1FRAME_ENA and Q_BUF_ENA. This causes the Link_Status frame to be stored in permanently in the memory.
- Preload the Link_Status frame in the QLS queue in both DLR ports by sending the Link_Status frame (or frames if they are distinct per port) from the firmware using forced forwarding (see section 29.4.5.4. Receive Processing (CPU to Switch).
- Wait until the DLR state machine is in the NORMAL_STATE (by reading the DLR_STATUS register, see section 29.3.11.2. DLR_STATUS : DLR Status Register.
- Enable the MAC link trigger function in MAC_LINK_QTRIG_Pn by setting the ENABLE bit to 1, setting to 1 the QUEUE_MASK to select the QLS queue, and programming the PORT_MASK with the "other" DLR port so the Link_Status frame is sent via the non-failed port.

(8) Dynamic Unicast L2 Entries Hardware Auto-Flush

The DLR can initiate a flush of the dynamic unicast entries in the L2 forwarding table by setting the AUTOFLUSH bit in the DLR_CONTROL register. The auto-flush feature executes whenever the following conditions occur:

1. The DLR state transitions from IDLE to FAULT
2. The DLR state transitions from FAULT to IDLE
3. The DLR state transitions from FAULT to NORMAL
4. The DLR state transitions from NORMAL to FAULT
5. The DLR state is FAULT and a beacon frame is received with a precedence higher than the active supervisor's precedence value.

The auto-flush function executes a lookup engine command LK_ADDR_CTRL.DEL_PORT with a mask (ADDR_MSK field) of 0x03 to delete all dynamic entries in the L2 table that are listed for forwarding to ports 0, 1 (i.e. it deletes all learned unicast that are located within the ring).

The interrupt "flush_macaddr" follows the conditions above as well so alternatively the hardware auto-flush can be disabled and the interrupt enabled to handle the flushing by the firmware.

29.4.12 Parallel Redundancy Protocol (PRP)

29.4.12.1 Overview

The switch can be used within a redundant network infrastructure according to IEC 62439-3 Clause 4 (section 29.4.23.3. References, (6)). The Parallel Redundancy Protocol (PRP) defines a network infrastructure consisting of two fully independent networks which are operated in parallel between all nodes. No communication occurs between the two networks and they are installed so that failure of one network does not affect the operation of the other network. Each network by itself is not aware of its role in a redundant infrastructure and is built from standard Ethernet networking equipment.

Each node that connects to the redundant network infrastructure uses two separate networking interfaces to connect to both networks at the same time. This is called a Double Attached Node (DAN). All DANs inject the same frame into both networks and append a Redundancy Control Trailer (RCT) which allows a receiving node to identify the redundant frames and only process one of the two frames. Hence if one of the networks experiences a failure the communication is not interrupted as all data still passes through the redundant network.

The infrastructure is held flexible to allow attachment of existing network components with only a single network interface (Single Attached Node, SAN) into the network. A SAN cannot benefit from the redundancy and can only communicate with other DAN or SAN on the same network.

If a SAN should also benefit from the redundant network infrastructure, it can be connected through a so-called Redundancy Box (RedBox) device which performs the necessary frame duplication functions to all traffic transferred between the SAN and the redundant networks. From the perspective of the network and all other nodes, a SAN behind a RedBox acts identical to a DAN and is called a virtual DAN (VDAN).

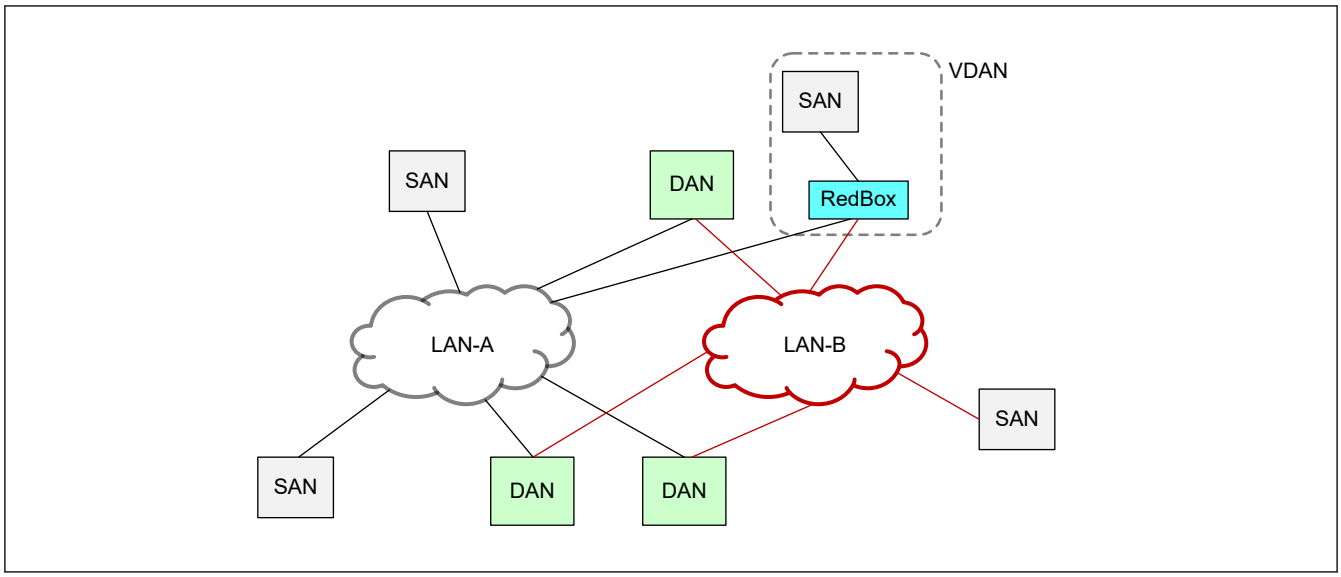


Figure 29.32 PRP network infrastructure overview

29.4.12.2 ETHSW Function Summary

The ETHSW allows implementing a very flexible device by combining two of its ports to a "PRP Group" to connect to such a redundant network infrastructure. All traffic to and from the PRP grouped ports will be subject to redundancy processing like frame duplication, trailer append, and duplicate detection. No traffic forwarding occurs between the PRP grouped ports.

At the same time, the other ports of the ETHSW can connect to regular SANs or other local (non redundant) networks and the switch will perform the function of a RedBox when forwarding traffic between the PRP Group and any other port. The following figure shows the different roles of the ETHSW within some mixed infrastructure example.

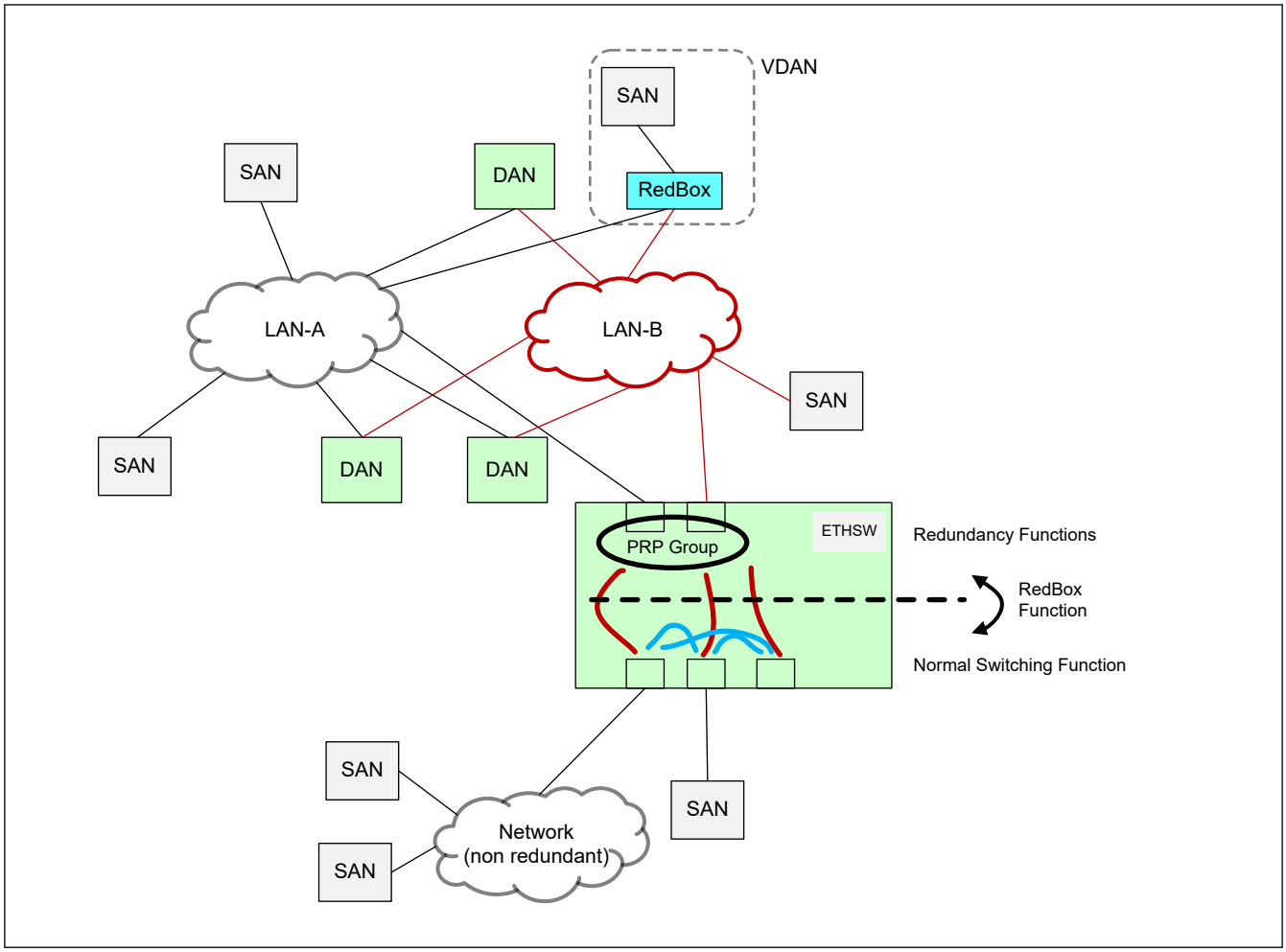


Figure 29.33 ETHSW within PRP infrastructure

The implementation is held flexible and enables the following functionalities:

- Any two ports of the line ports can be grouped for redundancy (one group can exist).
- Perform frame duplication and RCT append to frames directed to double attached nodes (DANs) when transmitting to the PRP group ports.
- Perform duplicate detection on frames received from the redundant networks with optional dropping of duplicates.
- Avoid frame duplication and RCT append if the destination is known to be a single attached node (SAN).
- The switch forwarding function distinguishes DANs and SANs fully automatically using its lookup table. A DAN has both ports of the PRP Group listed as its destination, whereas a SAN only has one of the ports of the PRP Group as its destination.
- The learning function is enhanced to automatically detect DANs when it receives redundant traffic from a node through the PRP grouped ports.
- Alternatively, management can setup the forwarding table to define DANs and SANs.
- Several options exist to configure trailer append and duplicate detection functionalities with optional automatic removal of the RCT when forwarding to non redundant ports.
- The special local switch management port (internal port) which connects the local node to the network(s) allows to operate the necessary network monitoring (software) and management services and can act itself as a DAN attached to the redundant networks.
- Traffic in between the PRP group ports is suppressed. SANs on one of the redundant networks cannot communicate with SANs on the other redundant network.
- Transparent RedBox functionality when forwarding traffic between the PRP grouped ports and the other switch ports.

29.4.12.3 Switch Forwarding Behavior

(1) Forwarding to Redundant Networks

Due to its flexible architecture and advantage of having a full switch in place, no dedicated redundancy layer exists. Instead, the existing switch forwarding mechanisms together with the capability to duplicate frames (used e.g. for broadcast or flooding) is used. When forwarding frames to the PRP grouped ports (as defined in register PRP_GROUP), the switch provides additional information with the frame to instruct the MAC transmitters to attach the redundancy trailer as needed. This allows transparent mixing of redundant and non redundant traffic on the networks. For example, link local traffic like BPDU management frames are forwarded without duplication and without redundancy trailer as are frames destined to SANs.

The following figure shows the flow of a frame and its respective actions along the path when forwarded from a non redundant port to the redundant ports. This effectively performs the RedBox functionality.

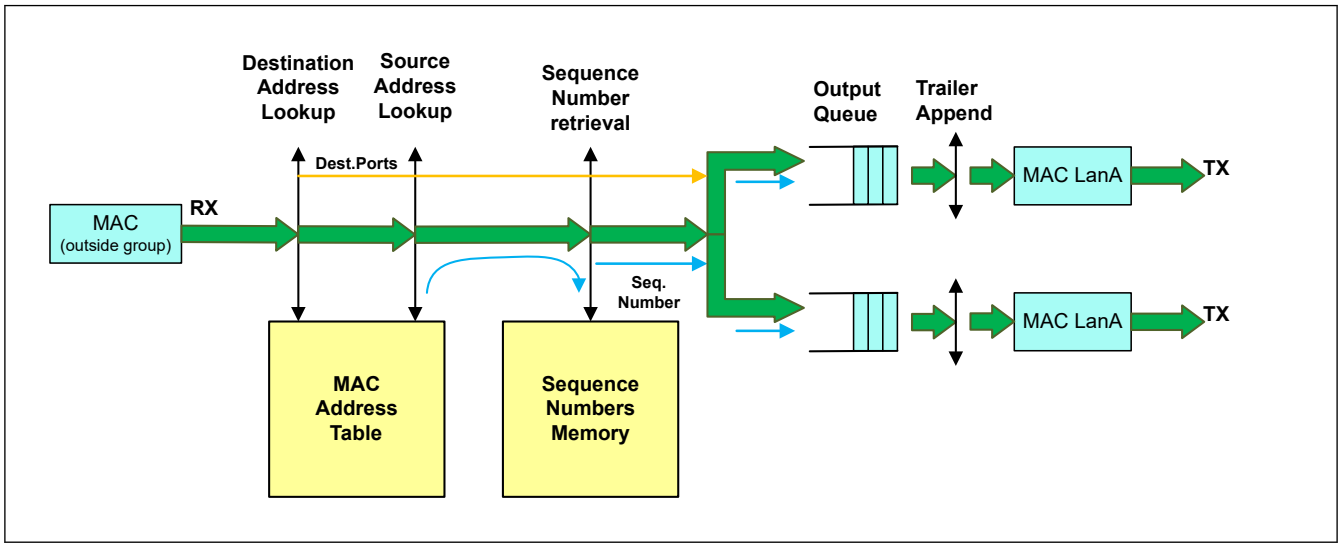


Figure 29.34 Forwarding to redundant ports (RedBox)

This also means that the switch's output queues are by itself already part of the redundant network infrastructure and sending of the duplicated frames into the redundant networks may not occur simultaneously. For example, if the queues of the redundant ports are filled to different levels because e.g. one port is transferring a frame to a SAN. Then the first of the two duplicates is sent immediately while the 2nd frame will have to wait for completion of the ongoing transmission at the other port.

To support automatic sequence number generation required to allow duplicate detection at the receiving DAN, the switch MAC address table is extended with a 2nd memory to store, for each MAC address, the current sequence number. The sequence number is automatically incremented whenever a frame from such a MAC address is forwarded from a non redundant port to the redundant ports and experiences duplication with trailer append. If the same source communicates with a SAN, the sequence number is not retrieved and stays unmodified.

(2) Receive from Redundant Networks

When receiving traffic from redundant networks optionally, a history memory is used to keep track of received sequence numbers. If the history memory shows that the same frame has been received from the same source already, it can be discarded. If it is the first of the two frames, it will be forwarded normally with the option to remove the redundancy control trailer.

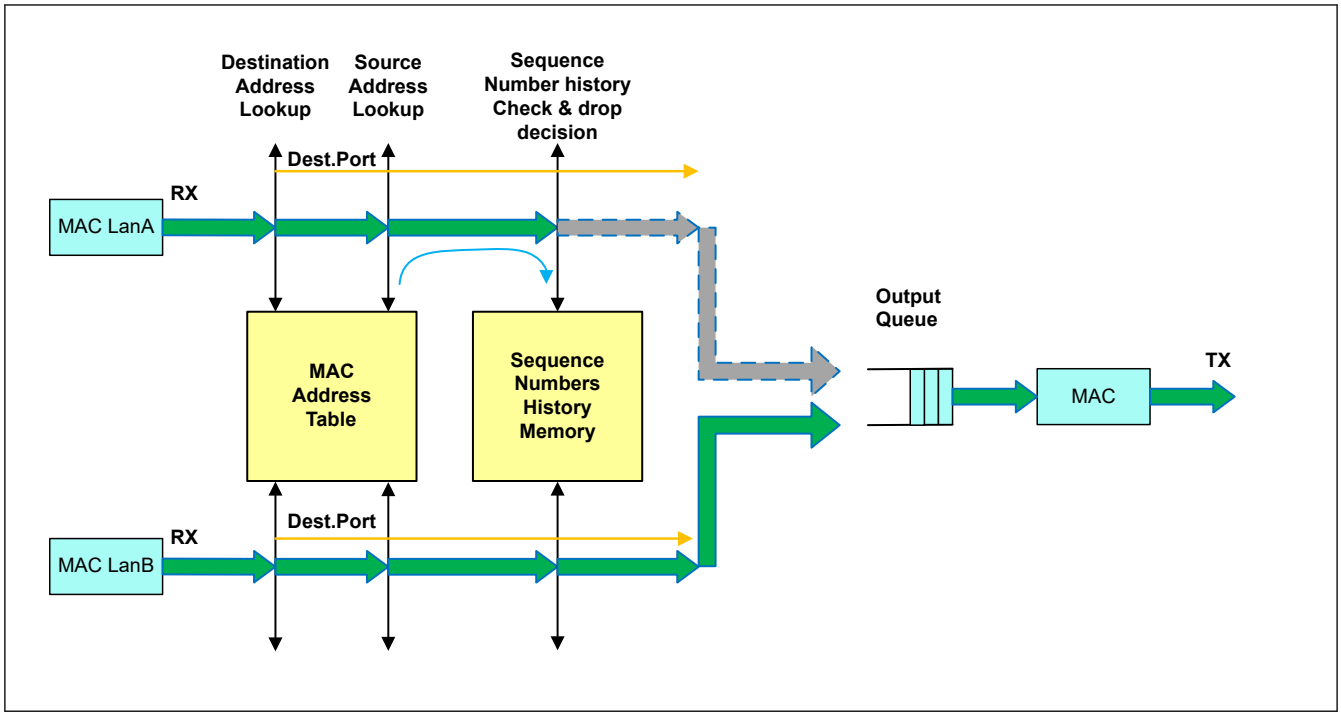


Figure 29.35 Receiving from redundant networks

The figure shows an example where the duplicate is dropped in the upper path (from MAC LanA). The sequence numbers history memory is consulted by every port when it receives frames to identify duplicates. The switch will then mark the frame for discard and not forward it.

The history memory provides specific atomic functions that guarantee that simultaneous reception of frames will always generate a consistent dataset in memory that allows a decision to drop one of the redundant frames only.

If sequence numbers are unknown or out of the expected range, the history is capable to manage, duplicates will not be detected and both frames will be forwarded to the output queue(s). The higher layer protocols therefore will need to be robust against receiving duplicated frames, as there is no guarantee that duplicates will be dropped. Rather in case of doubt the forwarding of both frames tries to ensure that no data is lost.

29.4.12.4 MAC Address Learning Enhancements

The hardware MAC address table management allows automatic learning of source addresses when receiving frames (see also (6) Automatic Learning / Migration). For supporting redundant networks, it is enhanced to allow the same source address to appear on both redundant ports. That is, when the same MAC address is found in the source address field of frames received on both redundant ports, the associated port mask of the entry in the table will contain the same two bits that were defined as a PRP group (register PRP_GROUP).

This causes the switch forwarding to duplicate the frames to both redundant ports when a frame is sent to a node attached to both redundant networks (DAN). This also ensures that no duplication is performed if the address entry would not show both bits within its table identifying the remote device as a SAN.

In addition, dynamic entries indicate reception of frames from either of the two ports in the (otherwise unused) bits 51 and 52. The hardware will only set the corresponding bit when receiving frames from the designated LAN A or LAN B port (as defined by register PRP_GROUP). It is up to the management (software) to deliberately clear those bits as necessary. This can be used to e.g. verify that traffic from a DAN is received on both ports allowing detection of network failures in one of the redundant networks.

The following figure shows the enhanced format of dynamic entries when redundancy operation is enabled (see also Figure 29.16). The normally unused bits 51 and 52 are used to indicate reception from the corresponding port of a redundant network.

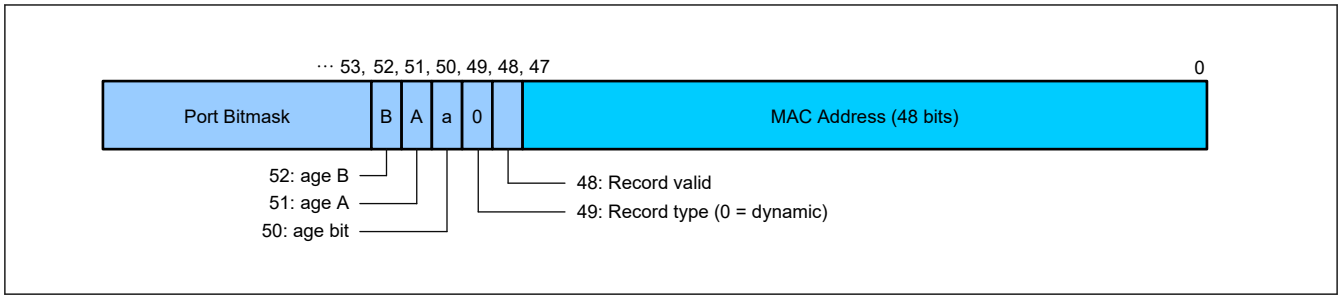


Figure 29.36 Dynamic address table entry with redundancy enhancement

By disabling the learning function (register LK_CTRL), software can control the address table in a static way. However, it should then still use dynamic entries (i.e. Bit 49 = 0) when adding addresses for DANs to enable the automatic updating of Bits 51, 52 for network monitoring purposes. The aging and migration functions can be disabled, hence dynamic entries will not be modified by the hardware except for updating (setting) those two bits of an entry.

Note: The normal age bit (bit 50) is also set always by the hardware independent of the port where the frame was received from.

PRP cannot be used together with memory reservations (see section 29.4.16.2. Memory Reservation). When enabling PRP memory reservations have to be disabled for all queues.

Table 29.30 PRP support relevant registers (1 of 2)

Register	Notes
PRP_GROUP	Defines which two of the line ports should be used to connect to redundant networks. The management port (internal Port 3 of ETHSW) cannot be assigned. When the switch forwards frames to the PRP group ports, it automatically performs frame duplication and appends the redundancy control trailer (RCT). A dedicated memory is used to store the current sequence number for each known (source) MAC address which is automatically incremented whenever such node sends frames to the redundant networks requiring duplication and trailer append. Frame duplication is controlled through the normal MAC address table entries (see section 29.4.3.7. Layer 2 Look Up Engine). If an address is inserted into the table, the port bitmask can be set to contain both ports to identify a DAN. The MAC transmitters use this information to determine if a trailer should be appended or not. On the other hand, if only one port is listed, frames are forwarded to any other port, no duplication occurs and no trailer is appended. Note: When using ports within a PRP group, the VLAN output manipulation function that relies on the per-VLAN tag mask cannot be used any more (i.e. Mode 3 as described in (d) Mode 3: VLAN Domain Mode / Transparent Mode is not available at the redundant ports).
PRP_CONFIG	Enables all redundancy functions and configures when to add redundancy trailer or perform duplicate detection with automatic removal of the trailer.
PRP_IRQ_CONTROL	Allows enabling several interrupts that may be of interest to the management application when implementing network monitoring functionalities.
RM_ADDR_CTRL	Access functions to read and write the history (rx) and sequence number (tx) memory entries for every MAC address. The memory has the same number of entries as the MAC lookup table memory (LK_ADDR_CTRL) allowing to store a sequence number or sequence number history for each individual MAC address. Which type of information is stored depends on the port(s) associated with a MAC address: If it is associated with the redundant ports (i.e. it is a source address of a DAN), a history is stored allowing duplicate detection. If the MAC address is associated with any other port of the switch, a sequence number is stored which is used when appending the RCT to frames forwarded to a DAN. Note: The history and sequence number memory is automatically updated and managed by the switch hardware. The application usually does not modify the entries. It may however decide to e.g. clear the table or perform modifications during initialization or network reconfiguration scenarios.

Table 29.30 PRP support relevant registers (2 of 2)

Register	Notes
LK_CTRL	The lookup function can be used to automatically learn addresses. This includes automatic detection of DAN and SAN within the redundant networks. Dynamic entries are enhanced to provide a (new) indication that allows if frames are received from a node through one or both of the redundant ports. Alternatively, management can set up the address table if the network configuration is known, or should be controlled by other measures. The management (software) then may choose to disable the global learning and aging functions (LK_CTRL bits 1, 2, 3) and can use the table management registers (LK_ADDR_CTRL) to add and remove addresses as needed. Even if using a static table setup, the entries for DANs and SANs should follow the format of dynamic entries (see Figure 29.16) as it allows the hardware to update the age A/B bits. The LAN A/B bits can be cleared by software and will be set by the switch hardware whenever the corresponding MAC source address is found in a frame received from any of the redundant ports. This method allows implementing a software controlled aging or additional monitoring for e.g. detecting faults within one of the networks connecting to a DAN.

29.4.13 Integrated HUB Module

29.4.13.1 Overview

The switch offers an integrated flexible Hub function. The Hub function emulates functionality similar to an IEEE 802.3 Clause 27 Class 2 repeater connecting multiple half-duplex ports to create a shared network infrastructure.

The basic task of the Hub function is to copy all frames cut through manner between all ports. Only one port can receive at any time and data is always copied to all other ports, allowing for half-duplex operation only. If multiple ports receive simultaneously, a collision occurs resulting in corrupted data.

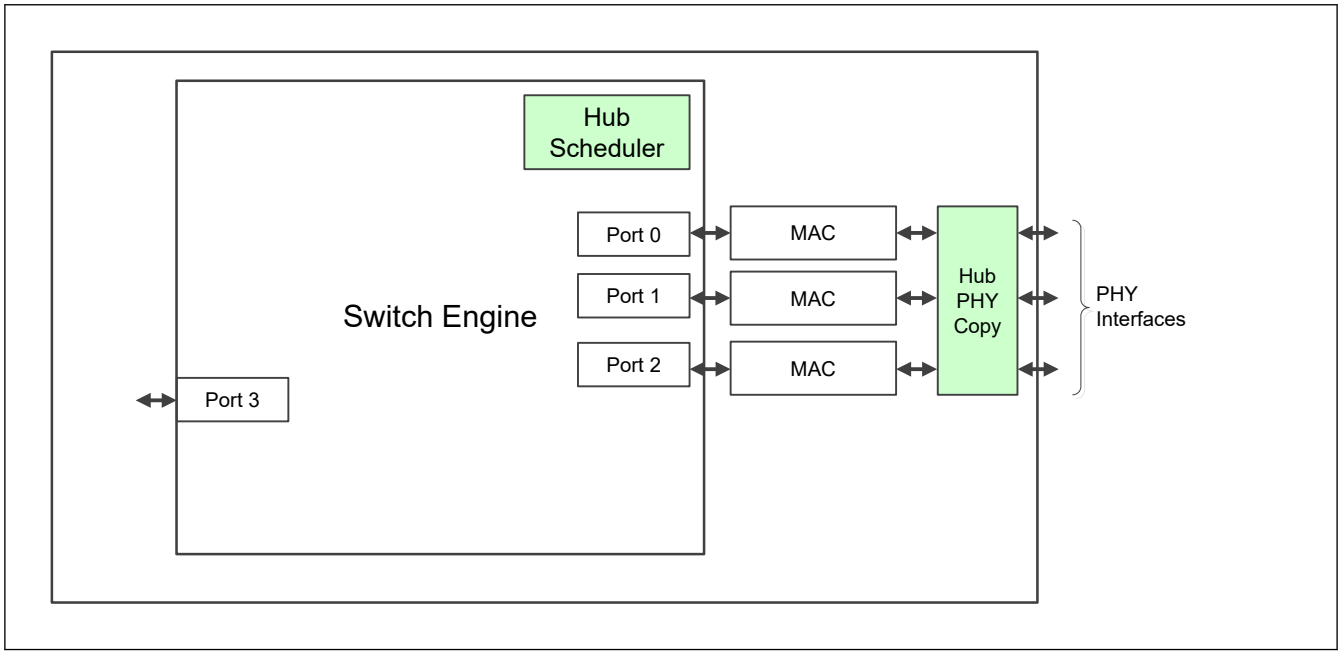


Figure 29.37 Integrated hub function overview

The switch integrated Hub function allows flexible assignment of any line port to create a so-called Hub Group. The Hub Group implements a copying layer inserted between MAC and PHY interfaces with the following functionalities:

- Copying of data in between all ports of the Hub Group at the PHY interfaces.
- Copying of MAC transmit data from a single MAC of the group to all Hub Group PHY interfaces.
- Data received at PHY interfaces is transparently received by the corresponding MAC port (i.e. is not copied to multiple MAC receivers).

Additional switch functionalities available when the hub mode is active are:

- Optional frame retransmission following collisions (when a MAC transmitted).

- Hub scheduler ensuring at all times only one MAC of the Hub group can transmit.
- Additional MAC transmit allowance controllable by application or from received frames (pattern matching).
- Regeneration of the preamble for transmitted frames ensuring that each frame has exactly 7 bytes of preamble plus 1 byte of SFD.

29.4.13.2 Functional Description

The switch can forward (transmit) to the Hub Group with any of the MACs of the Hub Group. When a MAC transmits into the Hub Group, its data is copied to all PHY interfaces of the group. Data received at any of the Hub Group's PHY interfaces is received by the switch through the port's connected MAC layer as normal (i.e. no copying into multiple MAC receivers occurs).

The switch forwarding is modified to allow only one of the Hub Group's MACs to transmit at any time. The MACs of the Hub Group must be configured for half-duplex operation.

(1) Dataflow Example

The following figure shows an example Hub Group configuration consisting of ports 0 and 1.

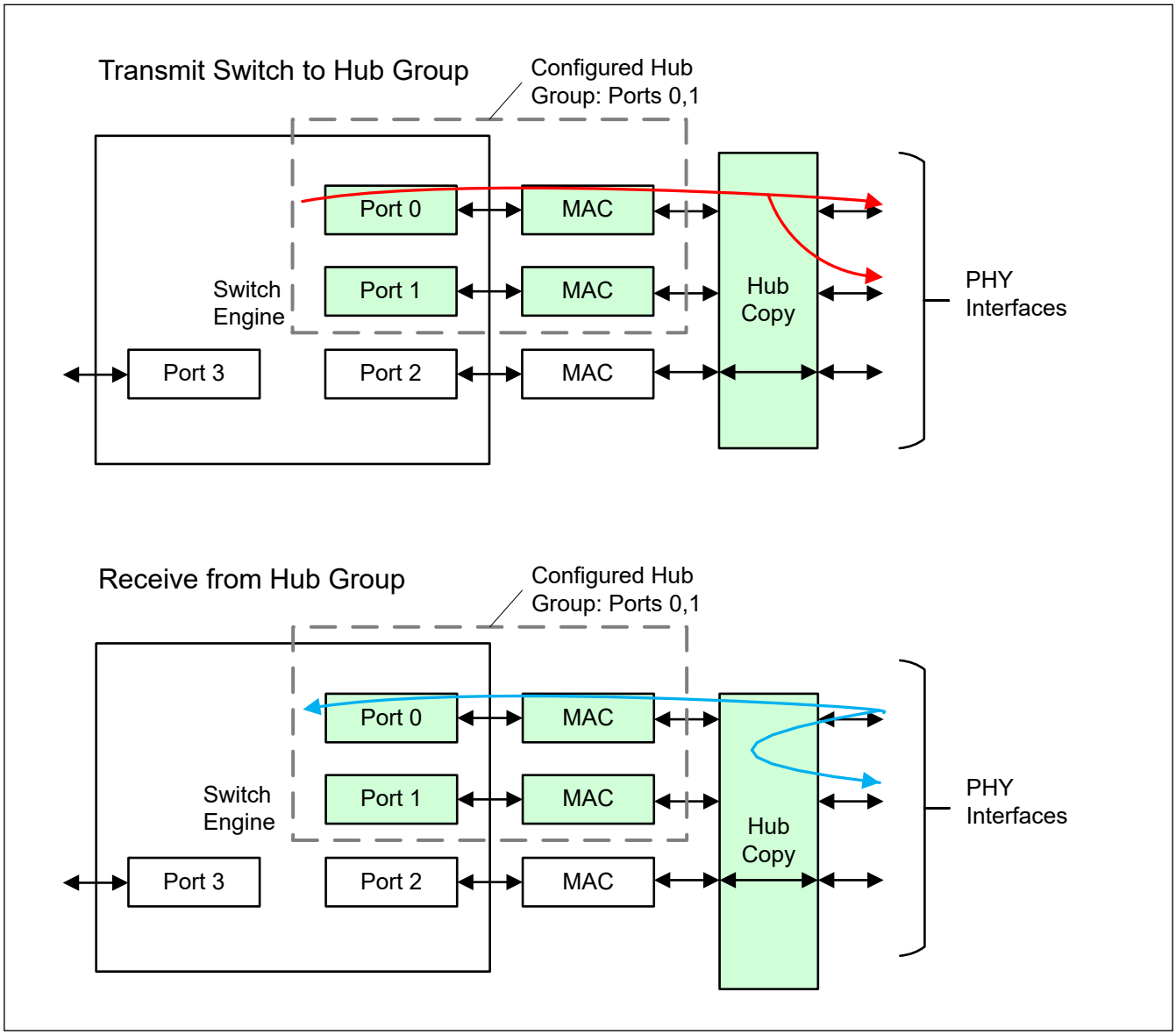


Figure 29.38 Integrated hub copy function examples

(2) Transmit into Hub Group

When a switch port (port 0 in above example) within the Hub Group transmits, its MAC transmit data is copied to all group's PHY interfaces simultaneously. Note that any MAC in the group can transmit independently into the group and be copied to all PHY interfaces. The hub scheduler function within the switch ensures that only one of the output ports can transmit at any time into the group to avoid collisions.

The ports which do not participate in the Hub Group are fully independent. Their PHY interfaces are directly connected to the corresponding MAC, bypassing the copy function, and may operate at different rates.

(3) Hub Group Receive

When a PHY interface within the Hub Group receives data, it is copied to all other PHY interfaces of the group, as well as to the corresponding MAC (i.e. PHY 0 to MAC 0, PHY 1 to MAC 1). The switch forwarding ensures that no frame received at a port of the Hub Group will be forwarded to the other ports of the Hub group to avoid frame duplication (and causing loops).

For learning, it means it will learn a MAC address from incoming frames at the port where it was received (same as normal switch mode). However, forwarding to a MAC learned on any of the Hub ports will still only use the dedicated (configured) Hub Group's default port.

29.4.13.3 Hub Specific Forwarding Rules

The switch forwarding decisions, when the Hub Group exists, needs special consideration for frames that are forwarded between the group and all other ports to avoid frame duplication. As every MAC within the Hub Group will transmit to all PHY interfaces of the group, frames would be sent multiple times if they are stored in multiple switch output ports (e.g. due to flooding or broadcast).

The switch forwarding decisions are modified to send frames always to only one MAC within the hub group (the default group port). When receiving frames from a port of the Hub group, the frame is never forwarded to any other port within the group (as this is already performed by the hub copy).

The following forwarding rules apply when the Hub is enabled:

- If frame received from within the Hub group
  - If isolate mode (HUB_CONFIG.HUB_ISOLATE = 1), set the management port as the only destination.
  - Otherwise, normal destination lookup defines the destination (incl. flooding) and all hub group ports are removed from final destination map.
- If frame received from the management port
  - If destination lookup fails or broadcast or BPDU: allow flooding only to the hub default port.
  - If destination lookup successful and a static entry: forward normally (can use static table entries to have full control to any hub group port, unfiltered!).
  - If destination lookup is successful and a dynamic entry:
    - If destination is a port of the hub group, forward to hub default port only;
    - Otherwise, forward normally (i.e. traffic to non hub ports).
  - Forced forwarding is unrestricted (application is responsible not to send to two ports within hub group).
- If frame received from another port (i.e. non hub group and not from management)
  - If destination lookup resolves non hub only, forward normally.
  - Otherwise, if destination is inside hub group (either from lookup or flooding),
    - If isolate mode: remove all hub group, add management port;
    - Otherwise, remove all hub group, add default hub port only.

The basic concept behind these rules is that all traffic from a port outside the Hub Group is sent only through the output port of a single MAC (the default hub port) into the hub infrastructure. Hence the other port(s) within the Hub group are not used for transmitting by default. They are however fully operable and the switch integrated hub scheduler will ensure that at any time only one of the MACs within the Hub Group will transmit into the hub infrastructure.

Those unused transmit MACs are available for use by the management port (application), which can use either forced forwarding or static address table entries to direct traffic to those ports.

For example, when used in a PowerLink environment, the default port would be used for all asynchronous traffic, while the management port can use the alternate port for all isochronous traffic and the protocol specific frames that require guaranteed response times.

Note: For Powerlink it is better to use one or more queues for isochronous traffic in the same port and to use the TDMA engine to trigger traffic from these queues, or to use the pattern matcher to trigger frames when requested by the controller.

29.4.13.4 HUB Group Clocking

When using Hub mode, set 100 Mbps and Half-duplex mode to CONVCTRLn register. (See section 27.3.5. CONVCTRLn : RGMII/RMII Converter n Control Register (n = 0 to 3)).

The hub copy logic operates in the switch operating clock domain. It is required that all Hub ports generate at least a valid tx_clk at a rate consistent with the port speed (either 100 Mbps or 10 Mbps) even when the link is down. If the clock is disabled, then the port must also be configured to not be in the Hub group. The frame data is first copied over to the switch operating clock domain and there it is replicated to all the other Hub member ports.

29.4.13.5 PHY Requirement

A PHY that should be used with a port that is intended for use within a Hub Group requires the following functionality for proper operation when hub mode is active.

1. The PHY, when operating in half-duplex, must not return transmit data to its receive interface.
2. The carrier sense (CRS) signal must assert for any traffic activity (standard behavior).
3. The PHY must operate in 10 Mbps or 100 Mbps at all times. Especially, when it has no link, it must be avoided that it would fall back to operate at 10 Mbps. A misconfigured speed on a port can prevent the hub from becoming operational hence affecting all ports of the group (see also section 29.3.13.9. HUB_STATUS : HUB Status Register).
4. The PHY must report link-up to the switch via the input ETHSW_PHYLINKn (n = 0 to 2) input pin.

Note that the first two requirements are general requirements for any PHY that is used for half-duplex operation attached to any of the switch ports.

29.4.13.6 Usage Information

- The Hub Group can be enabled at any time, however due to the changing clocking requirements special care on possible reset sequences and PHY initialization may be needed.
- A Hub Group should not use ports used in other grouping modes at the same time (e.g. DLR, PRP grouped ports).
- All ports used within the Hub Group must be configured for 10 Mbps or 100 Mbps and half-duplex operation (COMMAND_CONFIG_Pn).
- A frame transmitted by a port in the Hub Group will regenerate the preamble and SFD, ensuring that exactly 7 bytes of preamble plus one byte of SFD are transmitted with every frame, regardless of the number of preamble bytes on the received frame.

29.4.14 Receive Pattern Matcher

The switch has 12 pattern matchers that can be used to match against fields in the frame to perform override actions on the normal forwarding process.

The pattern matchers support fixed pattern matching and a 2-byte matching at an offset of up to 256-bytes after the frame's source address field. The fixed mode is limited to up to 12 bytes following the frame's source address field.

29.4.14.1 Functional Description

(1) Fixed Pattern Matcher

Every port implements additional receive pattern matchers which allow pattern search within up to 12 bytes following the frame's source address field.

The pattern comparison starts at the 13th byte of a frame and can inspect up to the 24th byte (not considering Preamble/SFD). Hence it covers the length/type field of the frame and up to 10 bytes of payload (see also Figure 29.3 for frame formats). For each byte an individual compare value and bit mask is available to define individual length patterns.

A pattern defines a compare value (see registers PTRN_CMP) and a mask value (see registers PTRN_MSK). The mask value is AND-ed with the frame's data and the result compared with the compare value to determine a match. Hence all bits that should be treated as not relevant must have a 0 in the mask as well as in the compare value itself.

The pattern matchers can be used in combination with the integrated hub module (see section 29.4.13. Integrated HUB Module) to allow triggering a MAC transmission when a specific frame is received. This allows implementation of protocols like e.g. Powerlink where a node's transmission into the network is controlled by receiving specific frames from a central master node.

However its use is not limited to the hub module but can be used for any applications.

(2) Flexible Pattern Matcher

These pattern matchers build on top of the fixed pattern matchers by having an additional mode selection between the following four operating modes:

1. Default 12-byte compare with mask (same as fixed pattern matcher).
2. 2-byte table look-up. Compares a single 2-byte field from the frame at a configurable offset with 8 possible values, with port and queue mask for frame triggering and optional Length/Type matching.
3. 2-byte range compare by using a minimum and maximum range, at a programmable offset, with port and queue mask for frame triggering and optional Length/Type matching.
4. 2-byte inverted range compare by using a minimum and maximum range, at a programmable offset, with port and queue mask for frame triggering and optional Length/Type matching. This is the same as the previous mode but a match is indicated if the range is outside the min/max.

Following Table depicts the utilization of the current compare and mask bits in the new modes. In addition, the following control bits need to be added to the PATTERN_CTRLn register (see section 29.3.15.2. PATTERN_CTRLn : RX Pattern Matcher Function Control for Pattern n (n = 0 to 11)).

- MODE: Selects the operating mode between modes 1, 2, 3, and 4 above.
- HUBTRIGGER: Triggers a frame from the ports enabled in PORTMASK (only one port must be enabled). This generates a transaction to the output memory controller to output a single frame from one of the enabled queues. The queues must be in the closed state for this to have any effect. This trigger goes to the integrated HUB module.
- IMC_TRIGGER: Triggers a frame from the ports indicated in PORTMASK and from the queues indicated by QUEUESEL. A single frame per-port among the selected queues is triggered. This request goes to the memory controller.
- IMC_TRIGGER_DLY: Same as IMC_TRIGGER but the event is delayed by the value programmed in MMCTL_DLY_QTRIGGER_CTRL. This allows a response frame to be delayed by a defined value before being transmitted.
- FORCE_FORWARD: Forwards the frame to the ports enabled in PORTMASK.
- MATCH_RED: Enables the matcher only if in the Profinet IRT RED period.
- MATCH_NOT_RED: Enables the matcher only if not in the Profinet IRT RED period.
- QUEUESEL (4-bit): A queue selector which selects the queue to trigger a frame. Can be set to 0x8 to select all queues.
- TIMER_SEL: Selects a timestamp from the specified timer when TIMER_SEL_OVR is set to 1.
- TIMER_SEL_OVR: Enables overriding the timer selection by TIMER_SEL when set to 1.
- LEARNING_DIS: Disables MAC source address learning when set to 1.
- MATCH_LT: Applicable to modes 2, 3 and 4. Compares the Length/Type field with value in the COMPARE register (See table below: length_type[15:0]). The offset is counted after the Length/Type if it matches.
- VLAN_SKIP: Applicable to modes 2, 3 and 4. When set to 1, the first 2 bytes after the MAC Source Address are compared against 0x8100. If this matches a VLAN tag is assumed and the Length/Type is checked after the tag if enabled. If MATCH_LT is not enabled then 4 bytes are added to the programmed offset to bypass the VLAN tag.

- SWAP_BYTES: Applicable to modes 2, 3 and 4. Control the byte order for comparison: when 0 the first byte received is the least significant byte (index 0), and vice versa when 1.

Table 29.31 Flexible pattern matcher PTRN_CMP and PTRN_MSK functionality

2-byte lookup		2-byte range compare	
Field	**Maps to**	**Field**	**Maps to**
cmp0[15:0]	PTRN_CMP_30[15:0]	min[15:0]	PTRN_CMP_30[15:0]
cmp1[15:0]	PTRN_CMP_30[31:16]	max[15:0]	PTRN_CMP_30[31:16]
cmp2[15:0]	PTRN_CMP_74[15:0]	<unused>	PTRN_CMP_74[15:0]
cmp3[15:0]	PTRN_CMP_74[31:16]	<unused>	PTRN_CMP_74[31:16]
cmp4[15:0]	PTRN_CMP_118[15:0]	<unused>	PTRN_CMP_118[15:0]
cmp5[15:0]	PTRN_CMP_118[31:16]	<unused>	PTRN_CMP_118[31:16]
mask[15:0]	PTRN_MSK_30[15:0]	mask[15:0]	PTRN_MSK_30[15:0]
length_type[15:0]	PTRN_MSK_30[31:16]	length_type[15:0]	PTRN_MSK_30[31:16]
offset[7:0]	PTRN_MSK_74[7:0]	offset[7:0]	PTRN_MSK_74[7:0]
<unused>	PTRN_MSK_74[15:8]	<unused>	PTRN_MSK_74[15:8]
and_mask[15:0]	PTRN_MSK_74[31:16]	and_mask[15:0]	PTRN_MSK_74[31:16]
cmp6[15:0]	PTRN_MSK_118[15:0]	<unused>	PTRN_MSK_118[15:0]
cmp7[15:0]	PTRN_MSK_118[31:16]	<unused>	PTRN_MSK_118[31:16]

The functions of the fields from Table 29.31 are:

- cmp[15:0] – Compare value
- min[15:0], max[15:0] – Minimum and maximum length to compare the 2 bytes at the offset. The match is successful is the value is within the range of min and max for MODE 3, or when not within the range of min and max for MODE 4.
- mask[15:0] – Masks bits set to 0 from the range comparison. Can be used to do a 1-byte range check by setting mask[15:0] to 0x00FF. The logical comparison done by the pattern matcher is:
  range_match = ((frame_2bytes & mask) ≥ min) & ((frame_2bytes & mask) ≤ max)
  When using a mask other than 0xFFFF, the minimum and maximum values must be programmed so the logic expression above yields the intended result.
- length_type[15:0] – Value to compare against the frame's Length/Type when MATCH_LT is set in the config register. If VLAN_SKIP is set and the frame contains a VLAN tag with a Length/Type field of 0x8100, the value in length_type[15:0] is compared against the next Length/Type after the VLAN tag.
- offset[7:0] – Byte offset to do the match after the MAC source address of the frame. If VLAN_SKIP is set and the frame contains a VLAN tag with a Length/Type field of 0x8100, the offset is internally incremented by 4 bytes to skip over the tag. In this case the maximum value valid for offset[7:0] is 251 bytes.
- and_mask[15:0] – Indicates that this pattern matcher matches if its own matching conditions match AND the pattern matchers which have their bits set to 1 in and_mask[15:0] also match. The action performed is that of the pattern matcher with the highest index. For example:
  - if pattern matchers 2 and 3 have both and_mask[15:0] set to 0x0C, then a match occurs only if both pattern matcher's matching conditions succeed, and the action performed is the one programmed for pattern 3.
  - if pattern matcher 3 has and_mask[15:0] set to 0x0C, then a match occurs only if pattern matcher 2 and pattern matcher 3 succeed, and the action of pattern matcher 3 is performed. If only pattern matcher 3 matches, then the match fails. If only pattern matcher 2 matches, then the action of this pattern matcher is performed.

Note: and_mask[15:12] is reserved.

29.4.14.2 Usage Information

- The pattern matcher module is configured with registers shown in section 29.3.15. RX Pattern Matcher Registers.
- Up to 12 patterns are globally available (see registers PTRN_CMP_xx, PTRN_MSK_xx and PATTERN_SEL).

- The executed function if a pattern matches a received frame, is configured with the per pattern control register PATTERN_CTRLn (n = 0 to 11).
- If and which pattern(s) is applied to traffic received at a specific port can be configured individually per port with the port specific RXMATCH_CONFIGn (n = port) register. Multiple patterns can be enabled for a port. When multiple pattern matchers succeed and report a match, the pattern matcher with the highest index is executed (i.e. 0 is the lowest priority pattern matcher and 11 is the highest priority).
- For each pattern an interrupt can be generated when a match occurs. This is globally per pattern, not per port, however still requires that a pattern is enabled at any one port at least. The pattern interrupts are enabled with the module's specific PATTERN_IRQ_CONTROL and PATTERN_IRQ_STAT_ACK registers.
- When using modes 2, 3, or 4 in the flexible pattern matcher the offset can be configured up to 255. However, if memory reservations are enabled then the offset must be limited to less than 82 without VLAN skipping (VLAN_SKIP set to 0), or 78 bytes with VLAN skipping (VLAN_SKIP set to 1).
- When a port is configured as blocked (via register INPUT_LEARN_BLOCK) the pattern matcher results are ignored except for BPDU frames. BPDU frames that are received by the external ports can have their port mask overridden using the pattern matchers. This can be used for instance to redirect MRP TEST frames to be forwarded to the other port when received on a blocked port.
- See section 29.4.22.6. Pattern Matcher Delayed Frame Trigger Delay for details on the delay when using the delayed frame trigger function (IMC_TRIGGER_DLY). This delay should be accounted for when programming the time offset to ensure a more precise event.

29.4.15 TDMA Operation

29.4.15.1 Overview

To support network infrastructures that define a time multiplexed access to reserve bandwidth for different traffic classes, the switch can be configured to operate some or all ports in a TDMA (Time Division Multiple Access) fashion.

Typical usage of such a scheduling is to reserve bandwidth and define times (time slot) for transmitting e.g. real-time or delay sensitive traffic. Other traffic can use the network only at other times. This allows implementation of isochronous (real-time) traffic channels with deterministic delays while sharing the network with other best-effort traffic.

Proper operation of such an infrastructure requires that all nodes and switches connected in the network operate with synchronized clocks (e.g. using IEEE 1588) and agree on the time slot assignments through some network management protocol.

The TDMA scheduler is controlled by a sequence table which determines the sequence of events programmed in a data table, as shown in following figure. A Time Control Vector (TCV) is the combination of a sequence and data vector that generates an event or change in the switch. Multiple sequence entries can refer to the same data entry, allowing for more flexibility in defining the tables sizes.

The TDMA scheduler operates in cycles. The cycle duration is determined by the application and programmed in the register TDMA_CYCLE. When the scheduler is operating the sequence table is read in order up to a maximum programmable entry. Each TCV has an associated offset in the TCV data entry which determines the event execution point relative to the start of the cycle.

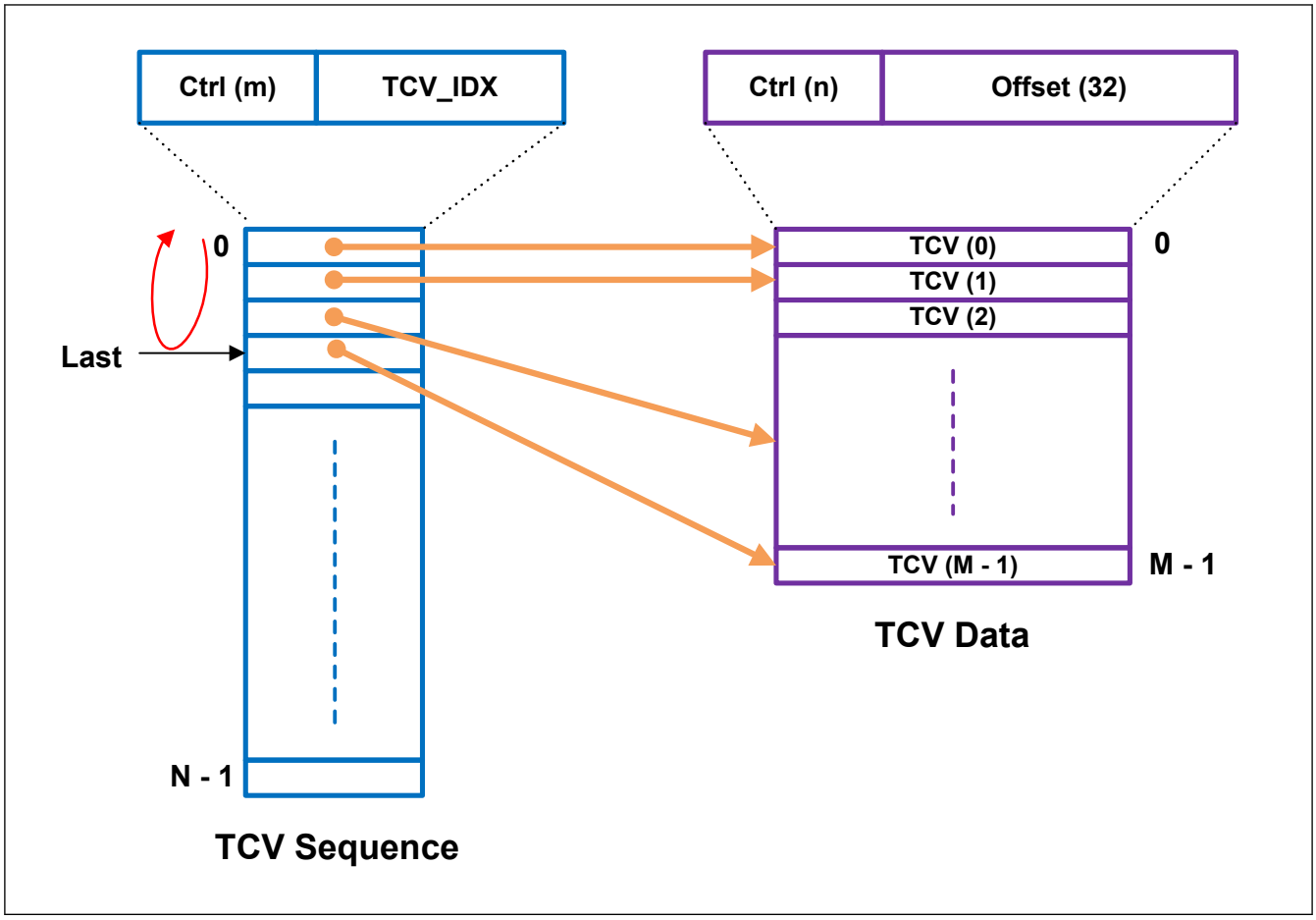


Figure 29.39 TDMA TCV control structure

Following tables below describe the TCV sequence and TCV data fields respectively. The size of TCV Sequence table is 4096 and the size of TCV Data table is 512.

Table 29.32 TDMA timer control vector sequence table

Bits	Field Name (size)	Description
0	START (1)	When set identifies this TCV to be a cycle start vector. The OFFSET in the TCV is taken as the offset after the next cycle starts. The offset value can be zero. This bit should be always set to 1 for entry 0.
1	INT (1)	When set to 1, an interrupt is generated to the CPU when this TCV is executed (after the indicated OFFSET relative to the cycle start).
10:2	TCV_D_IDX (9)	Index in the TCV Data Table. Selects which entry in the TCV data to use for the time offset and control information.
18:11	GPIO	Output value for the output pins ETHSW_TDMAOUTn. The value of these bits is propagated to ETHSW_TDMAOUTn when the vector is executed. The output pin can be configured for level, toggle, or strobe operation with the TDMA_GPIO register.

Table 29.33 TDMA timer control vector data table (1 of 2)

Bits	Field Name (size)	Description
31:0	OFFSET (32)	Time offset from the TDMA Cycle Start. Defines the activation of this TCV as an offset from the current cycle start. The units are nanoseconds (maximum value is ~ 4 ms). A value of 0 can be used to perform back to back operations, limited to the speed at which the TCV sequence and data tables can be read and commands executed. If the TCV sequence table has the START bit set to 1, the activation time is at OFFSET time from the next cycle start.

Table 29.33 TDMA timer control vector data table (2 of 2)

Bits	Field Name (size)	Description
32	INC_CTR0 (1)	Increments counter 0. This counter can be used to keep track of the number of cycles since the start of the TDMA operation. For Profinet, it tracks the SendClock number so it is possible for software to recover the current state after a lock up or software reboot. This may not be needed as the current cycle number can be determined by reading the current absolute time.
33	INC_CTR1 (1)	Increments counter 1. This is a smaller counter that can trigger periodic interrupts. For Profinet it can be used to keep track of the Phase when the Reduction Ratio is greater than 1.
34	RED_PERIOD (1)	Indicates to the ingress RED_GUARD and MAC bridge functions that the RED period is active. This control is passed to the pattern matchers. Set this bit to 0 if period color identification is not required.
35	OUT_CT_ENA (1)	Enables cut-through operation in the output memory read controller. This can be used to disable CT in the YELLOW period and initiate the YELLOW port byte counter. When GATE_MODE and TRIGGER_MODE are both 0 and OUT_CT_ENA is set to 1, the cut-through state for each port is set to the value in PMASK (1: cut-through enabled; 0: cut-through disabled). If GATE_MODE is set to 1 then the cut-through state is changed for the ports selected in PMASK(P) to the value in OUT_CT_ENA. If GATE_MODE is set to 0 and TRIGGER_MODE is set to 1 the cut-through state is not changed. Set this bit to 1 if control on the Cut-Through mode of the switch is not required.
36	IN_CT_ENA (1)	Allows changing the ingress cut-through state in the input frame receiver. The IN_CT_ENA flag is effective when GATE_MODE and TRIGGER_MODE are both 0. When GATE_MODE and TRIGGER_MODE are both 0, and IN_CT_ENA is set to 1, the ingress cut-through state is changed for all ports to the values in PMASK(P), where a 1 enables cut-through, and 0 disables it. Set this bit to 1 if control on the Cut-Through mode of the switch is not required.
37	TRIGGER_MODE (1)	When set to 1, QGATE causes a single-frame transmission between the closed queues selected QGATE on the ports selected by PMASK. GATE_MODE must be set to 0 for trigger operation.
38	GATE_MODE (1)	When set to 1, changes the state of the queues indicated in QGATE for the ports selected by PMASK.
39	HOLD_REQ (1)	When set to 1, a hold request operation (see section 29.4.18.7. MM_CTL.request (hold_req)) can be generated for the ports selected in PMASK. TRIGGER_MODE must be set to 0 for HOLD_REQ to be executed.
47:40	QGATE (8)	Bit mask that controls which queues are gated or triggered in the current time slot. A value of 0 closes (disables) the queue when GATE_MODE is 1. A value of 1 opens (enables) the queue if GATE_MODE is set to 1, or triggers a frame if TRIGGER_MODE is 1 and GATE_MODE is 0. PMASK selects which ports are affected.
50:48	PMASK (3)	Bit mask (one per output port) that controls which ports' queues are gated or triggered, hold request generation and which ports change their CT mode setting. A value of 1 enables the port.

29.4.15.2 Output Queue Scheduling

The main use of the TDMA scheduler is to control the transmission from the output queues. Each TCV can define via a port mask (PMASK) and queue mask (QGATE) which queues in which ports are allowed to transmit in every time slot. Two modes of controlling the queues are supported:

- Queue gating (GATE_MODE set to 1): In this mode the QGATE and PMASK control which queues in the enabled ports are allowed to transmit frames. Among the enabled queues the selected priority algorithm is used to select the next frame.
- Queue triggering (TRIGGER_MODE set to 1): In this mode the QGATE and PMASK control from which queues in the enabled ports a single frame is transmitted. The queue must be in the closed state (gated) and among the selected queues in QGATE a single frame per selected port in PMASK is transmitted.

Note that receive is not restricted in any way when TDMA is active. Frames can arrive completely asynchronously at the switch but will be stored and sorted into timeslots for transmission according to their classification (i.e. priority resolution).

29.4.15.3 Other functions in the TDMA scheduler

Other functions supported in the TDMA scheduler are listed below:

- Interrupt generation: every TCV sequence entry can optionally interrupt the CPU.
- Two hardware counters incremented via TCV control. This allows for a hardware tracking of the current cycle and sub-cycle. Counter 0 is a 32-bit counter which can be initialized by software. Counter 1 is an 8-bit counter, also initialized by hardware, but also capable of generating interrupts on a specific value and with a configurable maximum value.
- RED period indication: This is specific for Profinet IRT. It indicates to the pattern matchers and the MAC IPG calculation logic the current Profinet period color. It can be used however to indicate enable pattern matchers in certain time periods more generically.
- Output Cut-Through enable: Also for Profinet IRT, allows switching from Store-and-Forward in the YELLOW period for the ports enabled in PMASK. This feature requires also that the ports and queues enable the CTFL (Cut-Through Frame Length) function (see section 29.4.16.3. Egress Cut-Through Selection).
- Input Cut-Through enable: Controls the input cut-through enable function. Overrides the default enable for Cut-Through on the receive data path for the ports enabled in PMASK.
- Hold request generation for 802.3br frame preemption (see section 29.4.18.7. MM_CTL.request (hold_req)). This allows generating a hold request to a port with traffic interspersing enabled and operational. The hold request state can be cleared via the TDMA_CONFIG register.
- Second shadow table: To allow atomic for software reconfiguration the TCV sequence table can be split into two logic tables using the register TDMA_TCV_START. The register TDMA_TCV_START indicates the first entry on the TCV sequence table to be executed when a cycle starts (either when the TDMA is first enable, or after the last TCV sequence is executed, as defined by TCV_SEQ_LAST). A second table can thus be created by logically partitioning the TCV sequence and changing TDMA_TCV_START to the start of each table to switch between them. This way a table is effectively "online" while the other table is "offline".
- Gate control for CQF and GPIO pins. See section 29.4.15.7. Gate Control and GPIO Pins.

29.4.15.4 Scheduling Example: Profinet IRT

The following figure shows an example of configuring the TDMA scheduler for Profinet IRT. In this example the switch is defined to have 4 queues, and queues 3 and 2 are reserved for Profinet isochronous traffic (RTC3). Queues 1 and 0 are used for all other traffic. The cycle is divided in three periods: RED where only isochronous traffic is allowed, GREEN where non-isochronous traffic is transmitted in Cut-Through, and a YELLOW period where non-isochronous traffic is transmitted only if the frame can be finished before the start of the next cycle (this is a guard time for the next RED period).

Note: This example does not consider other switch configurations required for Profinet operation.

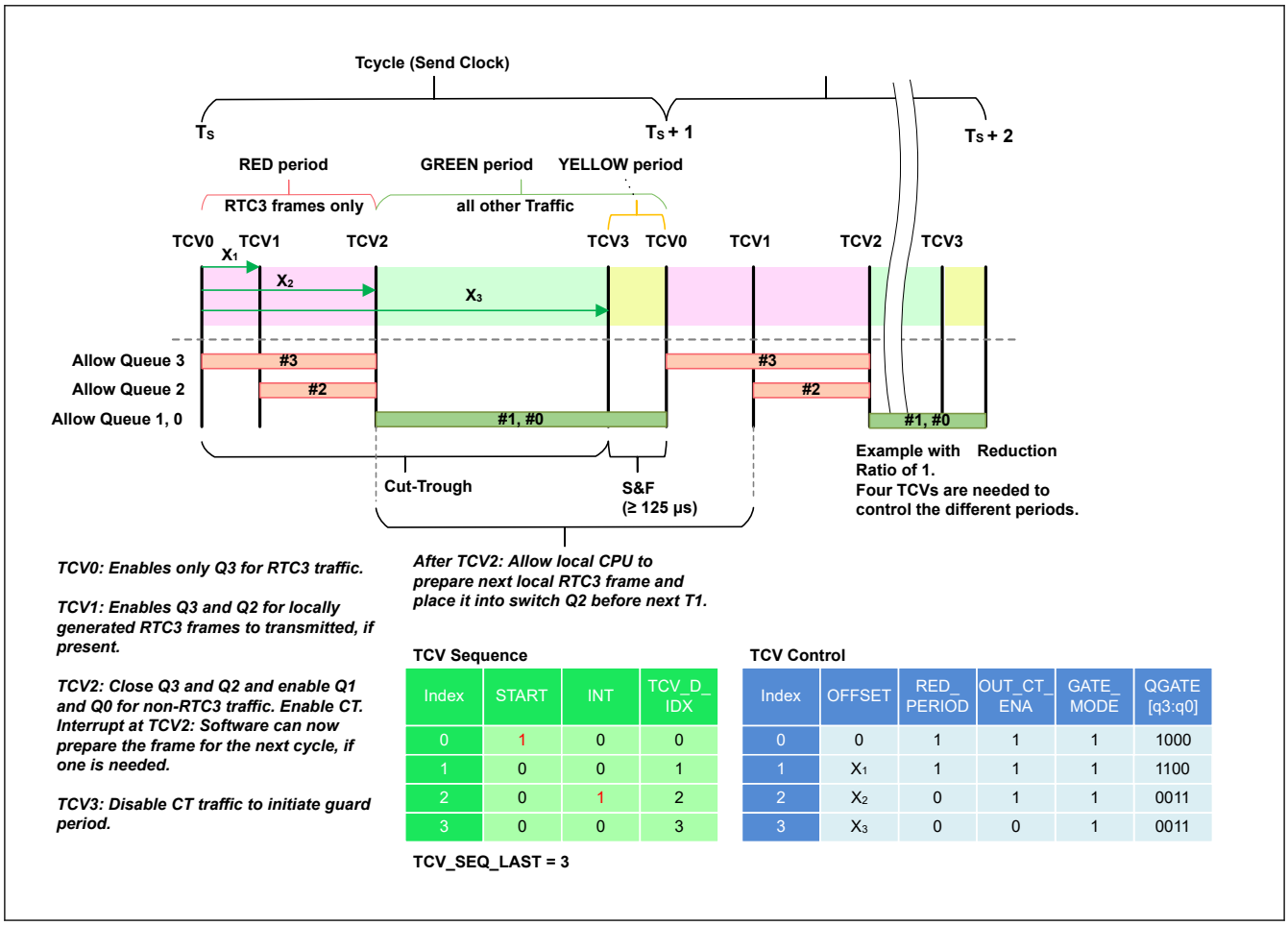
The figure defines a single cycle using 4 TCV. The first TCV enables traffic only from queue 3. The START bit in the TCV sequence entry is set to 1 and the offset in the TCV data entry is programmed to 0 so it is executed on every cycle start.

The second TCV (TCV1) enables a second queue within the RED period. Queues 1 and 0 are kept disabled as this is the isochronous phase. Queue 2 is used in this case to transmit a locally generated frame at a precise time within the RED period. If more frames would need transmission more TCVs can be provisioned for this purpose, and the TRIGGER function can be used to trigger a single frame from one queue instead of enabling the queue.

On TCV2, the third TCV, the GREEN period begins where non-isochronous traffic is generated. The queue gating is inverted (queues 3 and 2 are closed, and queues 1 and 0 are enabled). Also, an interrupt is generated to software to indicate that the traffic from Queue 2 is finished, and more frames can be scheduled if needed.

Finally, TCV3 starts the YELLOW guard period by setting OUT_CT_ENA to 0. Frames here are transmitted from the non-isochronous queues but only if their length is within the remaining of the YELLOW period (MMCTL_YELLOW_BYTE_LENGTH_Pn register in the memory manager configures at the start of this period the number of bytes that can be transmitted).

When TCV3 is executed, TCV0 is read again as the last vector from the sequence table is TCV3. Since TCV0 has the start bit set to 1 the scheduler will wait until the cycle finishes to execute this TCV again.



Index	START	INT	TCV_D_IDX
0	1	0	0
1	0	0	1
2	0	1	2
3	0	0	3

Index	OFFSET	RED_PERIOD	OUT_CT_ENA	GATE_MODE	QGATE [q3:q0]
0	0	1	1	1	1000
1	X_1	1	1	1	1100
2	X_2	0	1	1	0011
3	X_3	0	0	1	0011

Figure 29.40 TDMA scheduling example for Profinet IRT

For more details on the configuration see section 29.3.14. TDMA Scheduler Registers. The initial state of the queues can be configured from the application using the action registers in the management controller (see section 29.4.16.4. Egress Queue Controls, and section 29.3.10.8. MMCTL_QGATE : Queue Gate State Register.

29.4.15.5 Startup Sequence

When the application has determined all information for the cycle and slots to use and the system timer has been locked to some master clock within the network infrastructure (and hence runs precise) it can start the TDMA scheduler.

The hardware timer to use for the scheduler must be selected in TDMA_CONFIG.TIMER_SEL. To start the TDMA scheduler the application defines the absolute time when to begin with the first cycle setting register TDMA_START. It then enables the scheduler writing register TDMA_CONFIG (bit 0 = 1).

Note: The application must ensure to write the TDMA_CONFIG before the value in TDMA_START can be reached by the timer. If it would enable the scheduler after the timer has already passed the start value, the scheduler will wait until the timer wraps around and reaches the value again one full timer period later, which may not be intended.

This will cause the TDMA scheduler to begin its operation as soon as the timer reached the configured start time. While waiting for the first cycle the queues can be enabled or disabled (open or gated) via the register MMCTL_QGATE.

When the first cycle has started, the interrupt occurs if the INT bit is enabled in the corresponding TCV sequence entry (See Table 29.32) and interrupts are also enable (see register TDMA_IRQ_CONTROL).

The TDMA scheduler can be disabled at any time by clearing the TDMA_CONFIG register. This has an immediate effect and will enable all queues in the current state. The queues' states can be checked via register MMCTL_QCLOSED_STATUS_P0_3 and changed using the register MMCTL_QGATE.

If two logic tables are used, a reconfiguration can be done on the table which is offline, and once the reconfiguration is done the tables can be switched by writing the proper TCV sequence entry in TDMA_TCV_START.

29.4.15.6 Usage Information

- The TDMA scheduler is configured through registers given in section 29.3.14. TDMA Scheduler Registers.
- When using the TDMA scheduler, the strict priority selection scheme must be configured (see section 29.3.1.41. IMC_CONFIG : Input Memory Controller Configuration Register). Other schemes are not allowed and may yield unpredictable results.
- The MAC rate limiter for all TDMA enabled ports must be disabled to avoid delaying transmissions unexpectedly (see section 29.3.6.16. IDLE_SLOPE_Pn : Port n MAC Traffic Shaper Bandwidth Control (n = 0 to 3)).
- To classify received traffic into corresponding output queues several methods can be used and combined (see (4) Priority Resolution):
  - Classification based on settings per port with register PRIORITY_CFGn.
  - Define mapping of VLAN priorities per port with register VLAN_PRIORITYn.
  - Using the Pattern Matcher to define priority for a matching frame.

  As classification occurs at frame reception, all ports of the switch must be configured consistently, not only the TDMA enabled ports.
- The transmit MAC has a small buffer to compensate for the latency from the Input Memory Controller (IMC) to the transmit MAC data path. Due to this buffer the time between a queue gating event and the queue actually stopping can be imprecise. The control bit TDMA_PREBUF_DIS in the COMMAND_CONFIG_Pn allows disabling this buffering to increase the precision of the event. However, setting this control bit can cause the minimum IPG on the transmit MAC to be greater than the programmed value in TX_IPG_LENGTH when operating at gigabit speeds.
- See section 29.4.22.5. TDMA Action Delays for details on the delays in the event execution internally and how to correct the time offsets to ensure a more precise operation.

29.4.15.7 Gate Control and GPIO Pins

To monitor the TDMA a set of output pins are provisioned: ETHSW_TDMAOUTn. The number of pins is 8. These pins are controlled via the TCV sequence table (see Table 29.32) in a one to one match (i.e. bit 0 of the TCV.GPIO field controls the output pin ETHSW_TDMAOUT0). The value in the TCV sequence entry is applied when the vector is executed (i.e. at the time programmed in the TCV control vector).

The output pins can be used to monitor the current cycle. For instance in IRT, they can be used to indicate the start of cycle and whether the period is red, green or yellow. The period can be encoded into two bits also and a fourth pin used to indicate the start of the phase (i.e. when the reduction ratio is > 1).

The output behavior of the pin is configured in the register TDMA_GPIO (see section 29.3.14.18. TDMA_GPIO : TDMA GPIO Register). Each pin can be independently configured to operate as:

- Level mode: the output pin changes according to the value in the TCV sequence GPIO field, and holds the value until the next sequence is executed.
- Strobe mode: the output pin changes to 1 if the value in the TCV sequence GPIO field is 1, and holds the value for 16 switch operating clock cycles, and then returns to 0.
- Toggle mode: the output pin toggles its value when the value in the TCV sequence GPIO field is 1, and holds the value until the next TCV sequence with a value of 1 for the GPIO pin is executed.

In the same TDMA_GPIO register the value of the ETHSW_TDMAOUTn (n = 0 to 7) pins can be read.

These pins are also used for Cyclic Queuing and Forwarding (CQF) (see section 29.4.17. Cyclic Queuing and Forwarding (CQF) and Annex T in IEEE 801.Q). When used for CQF the same GPIO pins can be selected via per-port registers in the MMCTL_CQF_CTRL_Pn as the gate control for selecting the ODD/EVEN queue.

29.4.16 Memory Controller

The switch memory is architected as a single unit that is shared among all ports. The memory size is 128 KB. The memory is used as an egress memory buffer for all ports. The memory can store frames of arbitrary size from multiple

sources to multiple destinations (ports). The memory controller offers a set of time-multiplexed write input ports and time-multiplexed output ports with the capability to perform virtual frame duplication on the output ports (multiple reads of a single instance). The memory is partitioned in cells to balance efficiency between administration resources (linked lists pointers) and inefficiencies due to unused space, particularly for small frames.

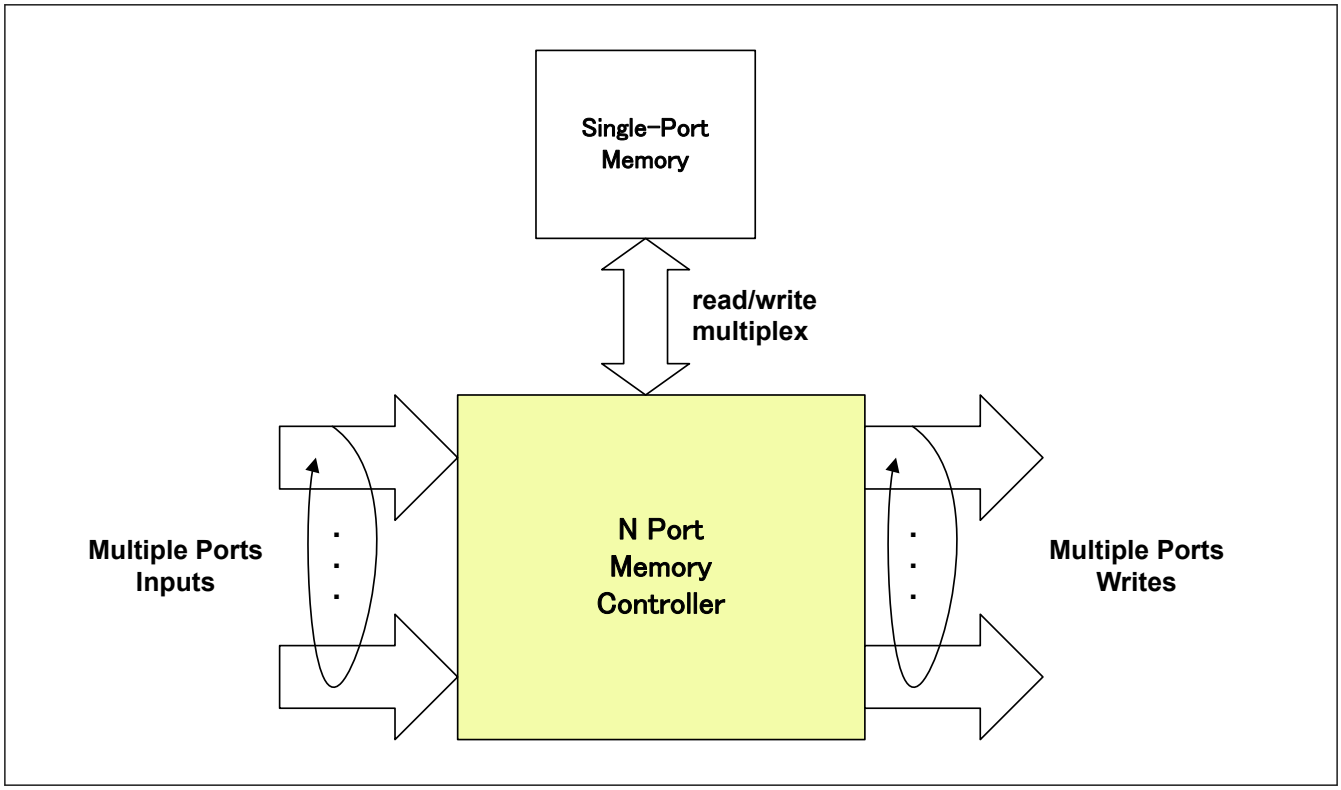


Figure 29.41 Memory controller overview

29.4.16.1 Cells and Output Queues

The complete shared memory is partitioned in 128 byte cells. The total number of cells is 1024. Incoming frames are stored partitioned in 128 byte blocks (cells) within the shared memory. The cell number (cell #) of the first cell of a frame identifies the frame and is passed to the destination queues to be stored in a FIFO to ensure orderly transmission of frames.

Each cell has metadata associated which includes a pointer to another cell (next-cell), number of output queues where the cell is listed (use-count), and whether the cell is the last cell of a frame (end-of-packet) and bytes used in that cell (modulo). As shown in following figure, a frame is tracked via its first cell number in the output queue FIFOs, and successive cells of a frame are linked via the next-cell pointer. The last cell provides end-of-packet indication and modulo information.

Cells can be anywhere within the shared memory, and therefore a single frame must not necessarily be stored in consecutive cells but instead can be scattered over the complete memory at arbitrary cells as shown in following figure.

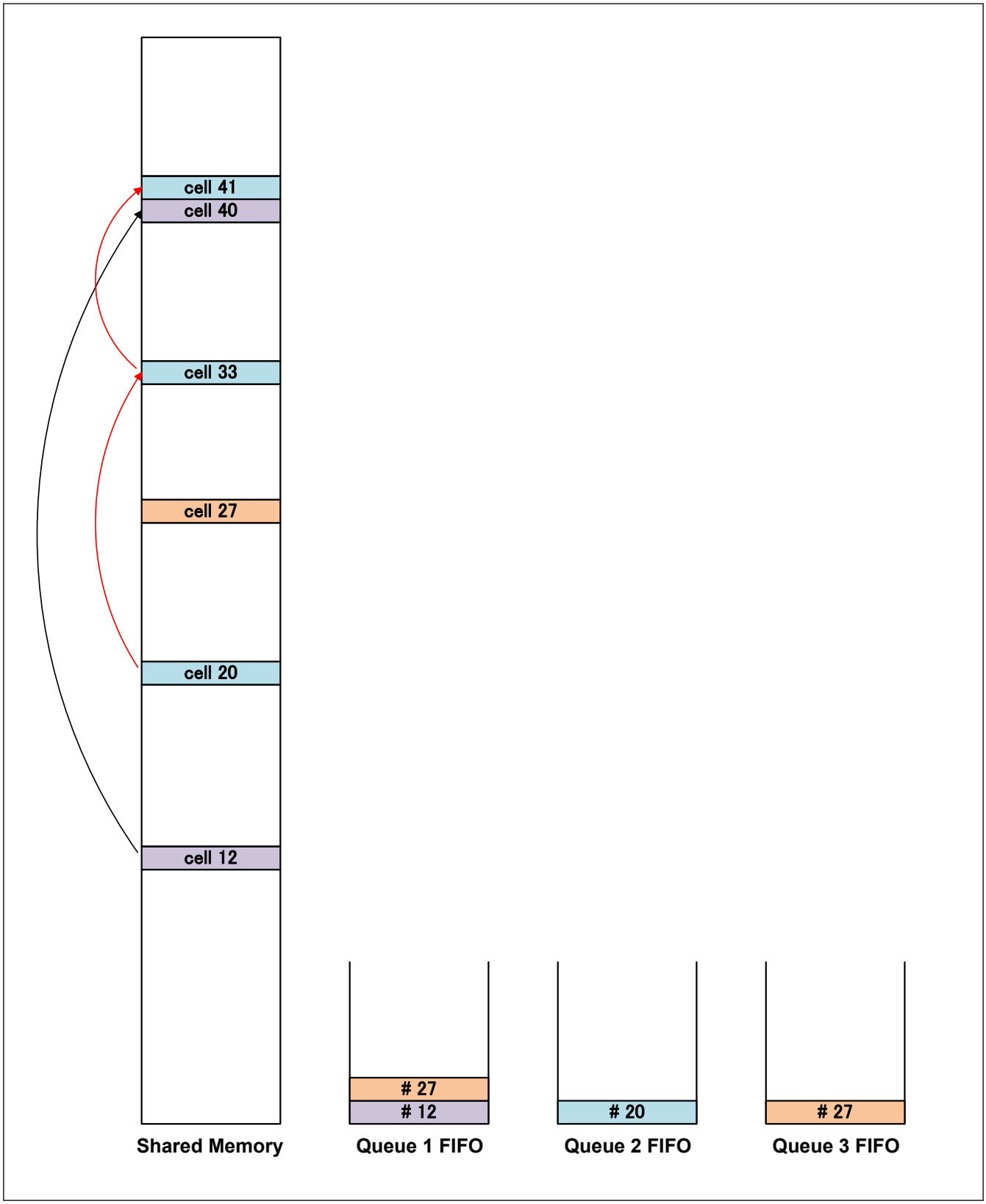


Figure 29.42 Cell storage concept

A central part of the memory controller is the cell allocation and de-allocation functions. When a frame is starting to be received, the allocation module selects a free cell to initiate the storage of the frame from a free-cells tracking FIFO. As each frame has a list of destinations determined by the forwarding function, the cell number of the first cell (which identifies the frame in the system) is stored in each of the output queue FIFOs to where the frame is destined, and the number of queues using the frame is tracked in the use-count. As frames are transmitted by the output ports, the de-allocation function

receives the indication for deallocating the frame from memory. Once the use-count is zero, meaning all ports completed transmission of the frame, the cells used by that frame are returned to the free-cells tracking FIFO for reuse.

29.4.16.2 Memory Reservation

The memory controller is capable of limited memory reservation for guaranteeing that certain traffic classes can have a guaranteed minimum memory portion. This is useful for applications where it is known that some memory reservation is required for the operation of a protocol. One specific example is Profinet IRT where the isochronous traffic requires the use of dedicated queues that must be able to operate during the isochronous phase without interference from non-isochronous traffic.

The memory reservation is achieved by using "pools" of memory cells that are assigned for use on specific queues across all ports. The limitation in the reservation capabilities is that reservations are shared among all ports, but this should be sufficient for applications like Profinet IRT and other forms of isochronous traffic where congestion is not expected.

The memory controller implements to kinds of pools:

- A global shared pool that is used by all queues first if there are cells available.
- Two dedicated pools that can be assigned for use by one or more queues.

The global shared pool allows efficient sharing of a bigger portion of the memory, while the dedicated pools allows for memory guarantees for a traffic class or traffic priority. Each dedicated pool can be assigned to one or more queues, and not all dedicated pools need to be used.

For details on the registers for configuring the memory reservations see section 29.3.10. Memory Controller Extended Registers. The configuration procedure in general can be outlined as follows (after having defined by the particular application the traffic classes requiring memory reservation):

1. Enable memory reservations by writing to RSV_ENA in the register IMC_CONFIG.
2. For each traffic class requiring memory reservations, select one memory pool Pi (i: pool number).
3. In the register MMCTL_POOL_QMAP assign the queues Qi (i: queue number) destined for the traffic class to the memory pool Pi.
4. Program the number of cells reserved in MMCTL_POOLn_CTR, where "n" is the register for configuring pool Pi.
5. Repeat steps 1. through 3. for other traffic classes requiring memory reservations.
6. The remaining memory which is shared among all traffic classes minus one block per input port is programmed in MMCTL_POOL_GLOBAL.

Following figure illustrates the concept of the memory reservation. Each pool is basically a counter for the number of cells utilized. Each counter tracking each pool's utilization is decremented when a cell is requested by the memory writer on allocation, and incremented when a frame is completed and the cells are no longer needed.

Note: It is not possible to reconfigure the memory reservations while the memory is in use.

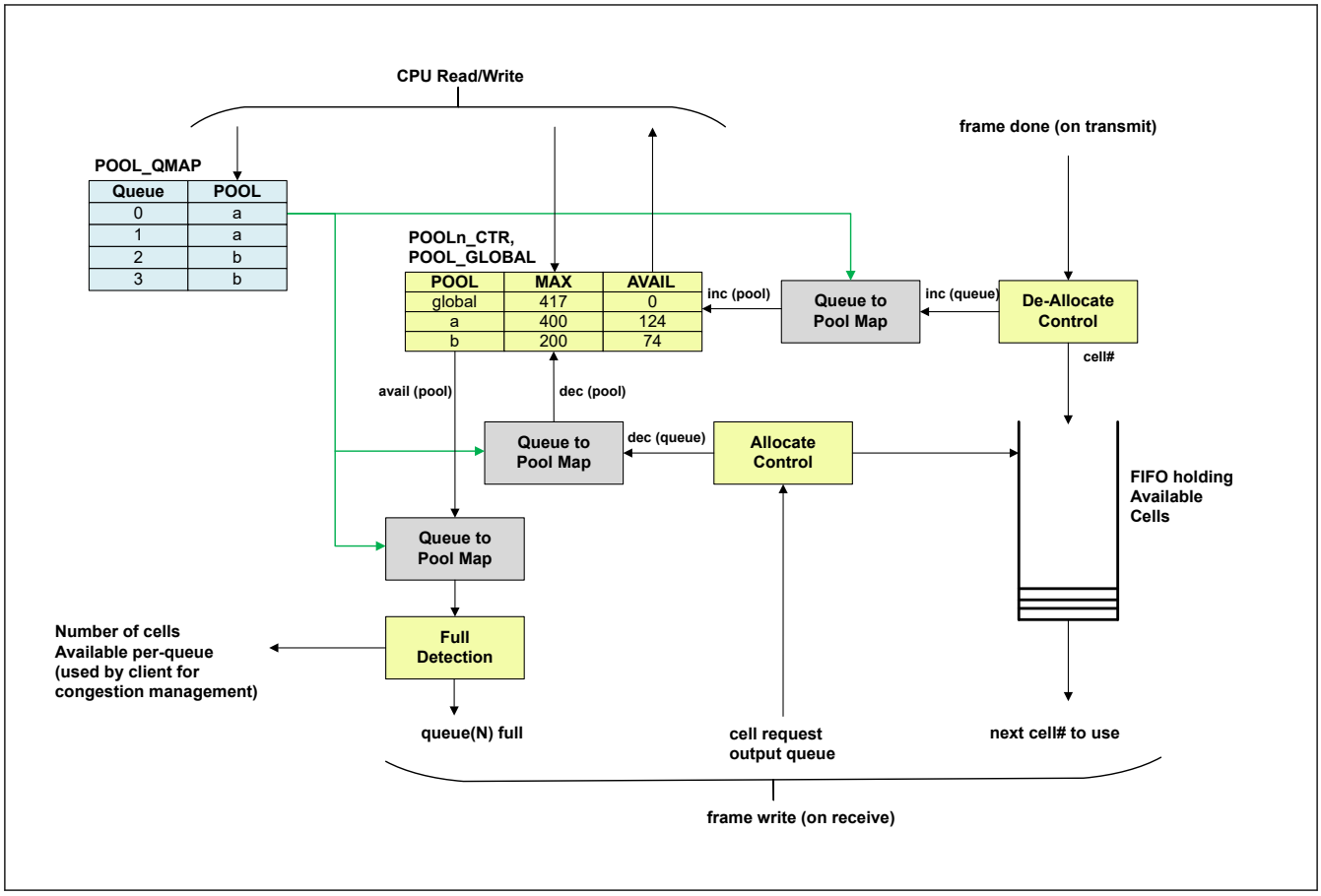


Figure 29.43 Memory reservation functional description

Note: Memory reservations cannot be used if PRP is enabled.

29.4.16.3 Egress Cut-Through Selection

Each port in the switch can be configured for Cut-Through operation which reduces the bridging latency. However, some protocols like Profinet IRT require when switching between non-isochronous to isochronous phases that the non-isochronous phase does not interfere with the isochronous phase, that is, all traffic from the non-isochronous phase must have terminated. A guard-band is then provision sufficiently large to allow a maximum-length frame to be transmitted, and the switch is required to switch into Store-and-Forward mode to ensure that the frame length is known prior to initiating a transmission in this guard-band phase.

The switch between Cut-Through (CT) to Store-and-Forward (SF) can be done at the memory controller when starting a frame transmission by enabling the "Cut-Through Frame Length" (CTFL) feature for the queues which must switch between CT and SF dynamically.

The CTFL feature is enabled via MMCTL_CTFL_P0_3_ENA (see section 29.3.10.2. MMCTL_CTFL_P0_3_ENA : Cut-Through Frame Length Enable Register). Every queue in every port where this feature is required must be enabled via these registers. When enabled a special structure is used for storing the frame length of Cut-Through frames so once the queue is switched into SF mode the frame is held from transmission until the frame length information is known.

The switch between CT and SF can be controlled by the TDMA scheduler (see section 29.4.15.3. Other functions in the TDMA scheduler) or from software via the register MMCTL_OUT_CT (see section 29.3.10.1. MMCTL_OUT_CT : Cut-Through Register).

The CTFL is a limited resource shared between all ports and queue that have it enabled. When a frame is received and there is no space in this table, the switch can be configured to act in either of two ways via the register IMC_CONFIG.CTFL_EMPTY_MD (see section 29.3.1.41. IMC_CONFIG : Input Memory Controller Configuration Register):

1. Receive the frame in the memory as store-and-forward frame (CTFL_EMPTY_MD set to 0).

2. Receive the frame in the memory indicating that the frame length information is not available but still in cut-through mode if the other cut-through requirements are met (CTFL_EMPTY_MD set to 1). In this case the frame may not be transmitted if the YLEN_EN control is set to 1 (see section 29.3.10.3. MMCTL_YELLOW_BYTE_LENGTH_Pn : Port n Yellow Period Byte Length Register (n = 0 to 2)).
   The frame will not be transmitted until CT mode is enabled again (via the TDMA or the register MMCTL_OUT_CT), or the control YLEN_EN is set to 0. Note that a frame in this state will potentially cause a head-of-the-line blocking in the queue, so this mode is only recommended when SF mode is used for short periods (such as with cyclic networks like Profinet IRT).

29.4.16.4 Egress Queue Controls

The memory controller provisions mechanisms to control the state of a queue and its mode of operation. This is useful for protocols like Profinet and Powerlink. The controllability of the queues comes from three sources:

1. From the application via register controls (see section 29.3.10. Memory Controller Extended Registers).
2. From the TDMA scheduler (see section 29.4.15.1. Overview)
3. From the receive pattern matchers (see section 29.4.14. Receive Pattern Matcher)

(1) Queue Gating

A queue gate control allows enabling and disabling (gating) the queue for transmission. While the queue is disabled (gated) the output queue FIFO is not selected for transmission by the queue selection algorithm. Frames can be stored in the queue while the queue is gated.

section 29.4.22.7. Transmit Frame Selection to Frame's FB Delay describes the delay from a queue transitioning from closed to open and the frame transmitted on the MII/GMII interface.

(2) Queue Triggering

Queue triggering allows transmitting a single frame from a group of queues. When a triggering operation is performed, a port and a queue mask are given to the memory controller, and the queue selection algorithm in every selected port will transmit one frame among the indicated queues which are in the gated (disabled) state. This allows for protocols like Powerlink to trigger a frame from an output queue via the pattern matchers when a send request from the controller is received, and Profinet IRT to transmit isochronous RTC3 frames at a deterministic time.

section 29.4.22.7. Transmit Frame Selection to Frame's FB Delay describes the delay from a queue trigger event (queue ready) to the frame transmitted on the MII/GMII interface.

(3) Queue Flushing

Queue flushing allows for flushing the port queues. The basic application is to clear the frames stored in memory under software control, but it can be also be used for implementing special protocols where frames need to be discarded (like PowerLink and Profinet IRT). The following flushing modes are supported:

- Flushing disabled: normal operation when no frame discarding is required.
- Flushing on close: flush the frames queued in the output queue when the queue state transitions from open (enabled) to closed (disabled) until the output queue FIFO is empty, and then flushing is disabled and more frames can be queued up.
- Flushing when closed: flush all the frames in the output queue when the queue state transitions from open (enabled) to closed (disabled) and keeps flushing until the queue state transitions back to open.
- Trigger flush: flushes all frames in the output queue until empty, and then stops flushing, regardless of the queue state.

Whether a queue is flushing frames or not can be read in the registers MMCTL_P0_3_FLUSH_STATUS.

(4) 1-Frame and 1-Frame-Buffer Modes

This mode is static and can only be configured via registers (see section 29.3.10.12. MMCTL_1FRAME_MODE_Pn : Port n 1-Frame Mode Configuration Register (n = 0 to 2)).

The 1-frame mode limits the number of frames queued to 1. If a new frame is received while another frame is still queued, the oldest frame is discarded. Note that a frame may have initiated transmission while a second frame is queued. In this case this second frame is not discarded.

The 1-Frame-Buffer mode also limits the number of frames queued to 1, but on frame transmission the frame is not removed from the queue. If a new frame is received, the old frame is removed from the queue. To use this mode the queue must be in the closed state (disabled, otherwise the frame would be transmitted infinitely). This mode is applicable for instance in implementing Powerlink, where a response can be kept queued at all times so the controller can poll it at any time, and when a newer response is ready, the frame can be sent from software and it will automatically replace the current frame in the egress queue.

29.4.16.5 Cyclic Queuing and Forwarding Configuration

The control register MMCTL_CQF_CTRL_Pn controls per-port the operation of CQF. The configuration is done on a per-egress port to allow configuring only the ports that participate in CQF. For details on CQF refer to section 29.4.17. Cyclic Queuing and Forwarding (CQF).

To enable CQF in an egress port the field PRIO_ENABLE0 must be programmed with a '1' for each of the priorities that will be used for CQF. The priority of a frame is determined as described in section 29.4.3.5. Frame Classification and Priority Resolution. It is recommended that the resolved priority for CQF traffic matches one of the two queues used for CQF, and that the priority for non-CQF traffic is never a priority corresponding to a CQF queue. More than 2 priorities can be enabled for CQF but because all are queued in the same egress queue the transmission order is the order in which they are enqueued, and not their relative priority. It is recommended also to use the two highest-priority queues for CQF to avoid non-CQF traffic to cause interference.

The queues use for CQF are programmed in QUEUE_SEL0. The two CQF queues are always contiguous with the lowest queue number programmed in QUEUE_SEL0.

The control or gating signal for determining which is the buffering queue is selected via GATE_SEL0, and each port and CQF group can use a different gate signal. Moreover, the control bit REF_SEL0 configures whether the gate signal used is the value in GATE_SEL0 for the egress port(s) or for the ingress port. These signals are generated by TDMA scheduler, which can produce up to 8 independent signals. And these are the same source for the output pins ETHSW_TDMAOUTn (n = 0 to 7). For CQF the gate signal must be a level indication (i.e. not strobing) and be kept stable for the duration of the CQF cycle. For details on configuring the TDMA refer to section 29.4.15. TDMA Operation.

Finally, the queuing function can be configured to select the buffering queue with the start of frame or with the end of frame via USE_SOP0. The end of frame is the default behavior and the one specified in 802.1Q. The start of frame is allowed for flexibility, but it should not be used for standard operation.

29.4.17 Cyclic Queuing and Forwarding (CQF)

CQF is described in Annex T of the IEEE 802.1Q-2018 specification. CQF is intended to allow delivery of time-sensitive traffic in a deterministic latency that is dependent only on the number of hops through the network. To remove the hop-to-hop latency and switching delay, each intermediate device is required to buffer the frames received in a time period, while transmitting frames that were received the period before. This results in a hop-to-hop latency equal to one cycle if the network and switch delay allows for the frames to be fully transmitted and received within the time period.

CQF is implemented by selecting two egress queues as CQF queues and alternating their state between "buffering" and "transmitting" as shown in the figure below. This figure shows a simple example where two queues in an egress port alternate between these two states in periodic cycles named EVEN and ODD. Frames received during the EVEN cycle indicated by the control signal (called gate signal as well) being "low" are not transmitted immediately and queued up in Queue A, while frames received during the ODD cycle indicated by the control signal being "high" are queued up in Queue B. When a queue is in the "transmitting" state the frames which were received during the previous cycle are now transmitted in the order they were received (and not their original priority).

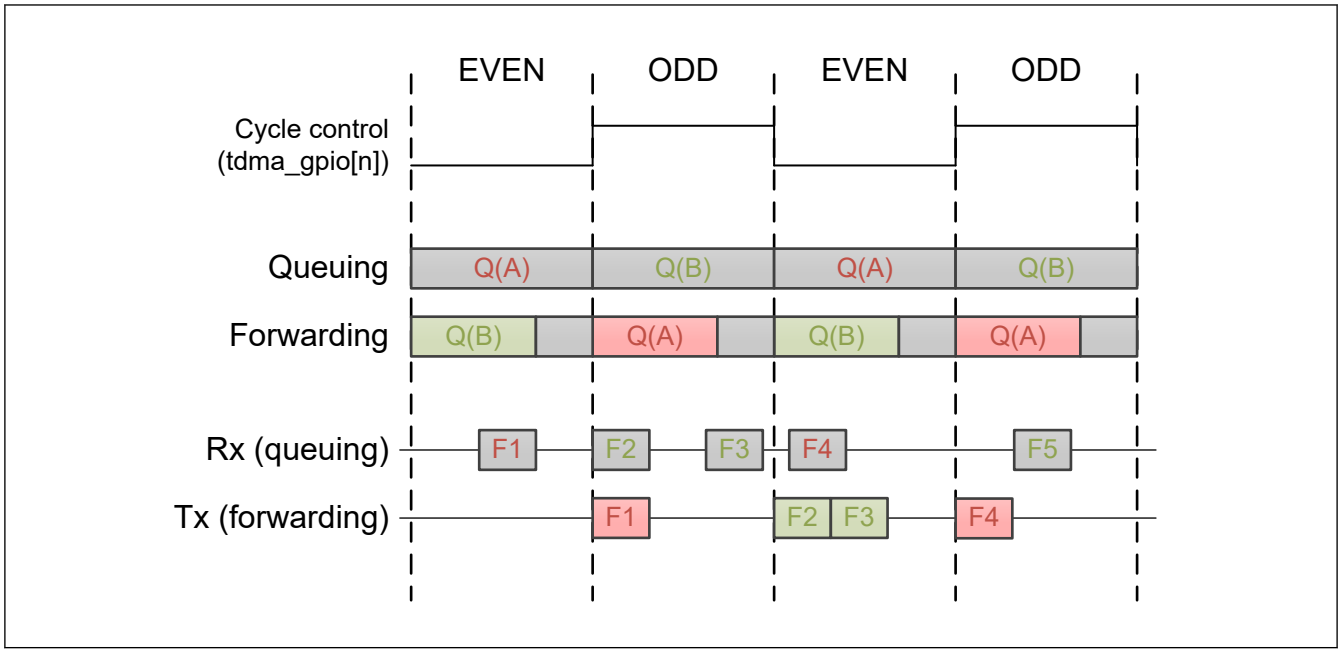


Figure 29.44 Cyclic queuing and forwarding (CQF) using 2 egress queues

To use the switch with CQF the process can be viewed as two separate functions: queueing and forwarding. Queuing determines based on the current cycle, frame priority and register configuration in which queue the frame will be placed, while forwarding controls which queue is allowed to transmit during each cycle. The figure below shows a functional diagram where the two functions are depicted, description of each function follows in the sections below.

The configuration for CQF is mostly configured via the register MMCTL_CQF_CTRL_Pn. Both functions are controlled by the TDMA scheduler which generates the control signal (ETHSW_TDMAOUT) that determines which CQF queue is used in each cycle, and the open/close queue controls to select the queue that is in the forwarding state (see section 29.4.15. TDMA Operation for details on the TDMA operation). section 29.4.17.3. CQF Configuration Example shows a brief example where the switch is configured for CQF.

CQF also requires that the priority resolution is configured so time-sensitive traffic is mapped to the CQF queues, and prevent other traffic from being queued into the CQF queues.

CQF time-sensitive traffic can coexist with non-CQF best-effort traffic. To ensure timely forwarding of the time-sensitive traffic CQF must be configured to use the highest priority queues to ensure timely forwarding of these frames in the corresponding time period.

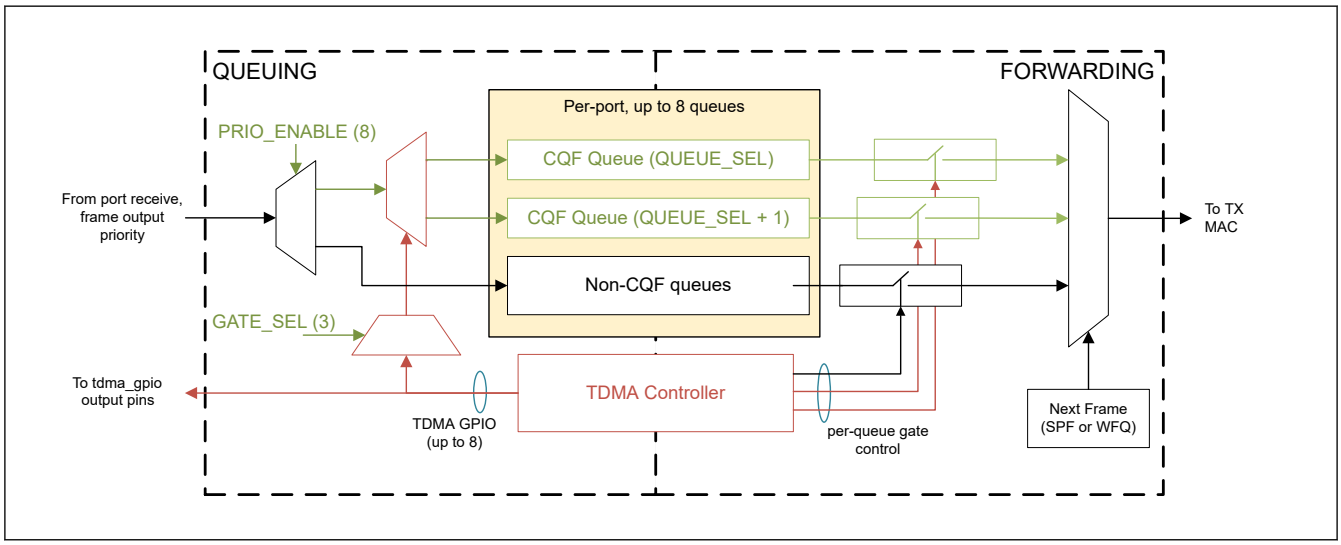


Figure 29.45 Cyclic queuing and forwarding functional diagram

29.4.17.1 Frame Queuing Function

The frame queueing function is based on the resolved priority in the frame receive data-path (section 29.4.3.5. Frame Classification and Priority Resolution). The resolved priority is the result of port configuration, the frame's VLAN priority, the DSCP if present, and it can also be modified via the pattern matchers or by the management port specific tag when the frame is sourced by the local CPU.

In normal operation when CQF is disabled the priority is used directly to select the egress queue for the frame. Enabling CQF via MMCTL_CQF_CTRL_Pn.PRIO_ENABLE0 causes that the frames matching the enabled priorities to be queued in the CQF queues selected via MMCTL_CQF_CTRL_Pn.QUEUE_SEL0, while frames whose priorities are not enabled for CQF are queued in the directly corresponding egress queue.

Note: It is important then to be aware that the hardware provides no protection when the CQF queues match priorities that are not enabled for CQF in MMCTL_CQF_CTRL_Pn.PRIO_ENABLE0. If a frame is received in such priority it will be queued in the corresponding queue.

The CQF queue to use is determined by the selected control signal from the TDMA scheduler (ETHSW_TDMAOUT) via MMCTL_CQF_CTRL_Pn.GATE_SEL0. The programming of the TDMA must ensure that the selected signal is periodic and does not wiggle during the cycle. This can be achieved by programming the signal to operate in either level or toggle modes (details in section 29.4.15.7. Gate Control and GPIO Pins). When the control signal is "low", the CQF queue used is MMCTL_CQF_CTRL_Pn.QUEUE_SEL0, and when "high" frames are queued into QUEUE_SELx + 1. The control MMCTL_CQF_CTRL_Pn.REF_SEL0 controls for each ingress port whether the gate signal used is the one programmed for the egress port(s) or for the ingress port.

To synchronize the CPU the TDMA scheduler the interrupts from the TDMA scheduler can be used to wait for a cycle to being. The state of the gate signals can then be read from the TDMA_GPIO register.

29.4.17.2 Frame Forwarding

Frame forwarding for CQF is controlled also by the TDMA scheduler using the queue gating function to open and close the CQF queues in an alternating fashion such that the buffering queue is in a closed state and the forwarding queue is in an open state as shown in the table below. This is decoupled from the queueing function to provide more flexibility in case guard bands need to be added.

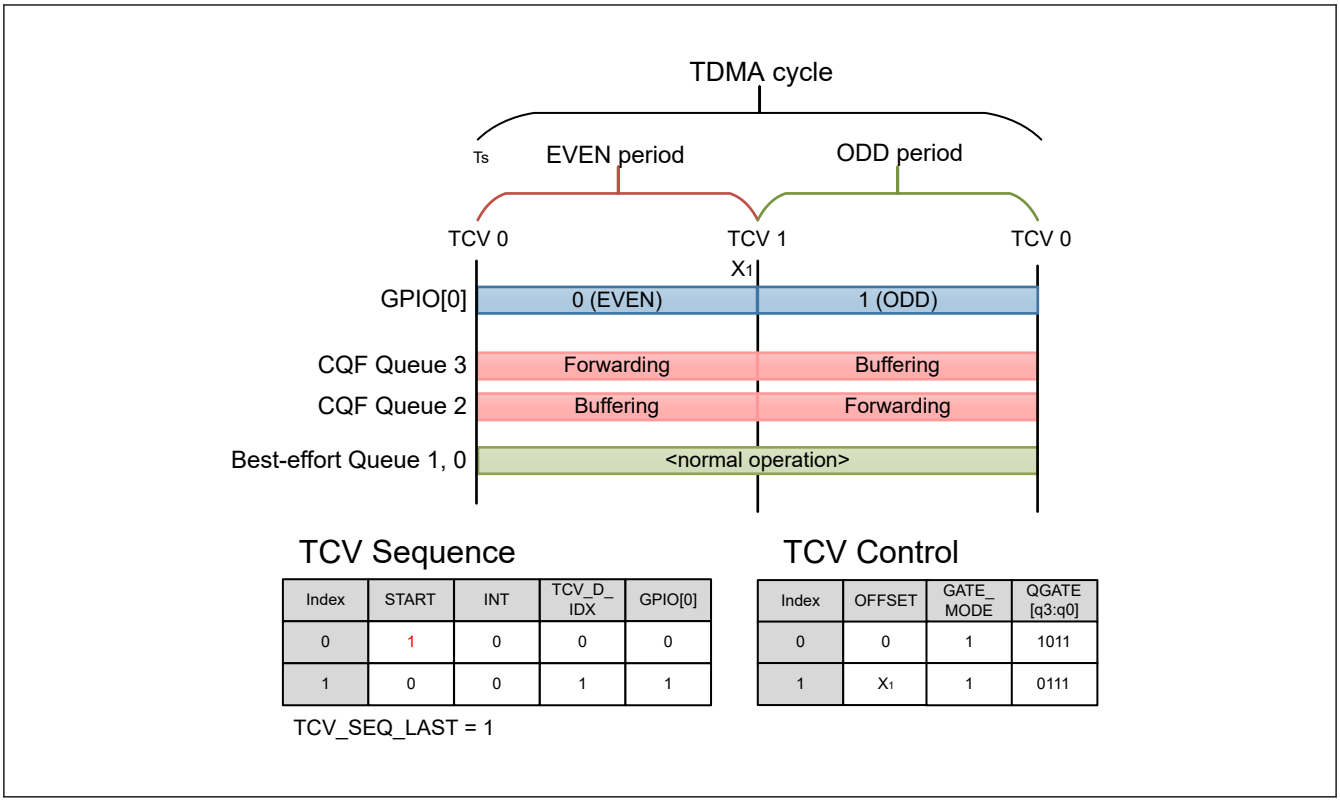
Table 29.34 Queue state programming for CQF

Cycle	CQF Queue A	Queue A State	CQF Queue B	Queue B State
EVEN	Buffering	Closed	Forwarding	Open
ODD	Forwarding	Open	Buffering	Closed

For a proper operation, all traffic received in a cycle must be forwarded in the following cycle such that by the end of the cycle the CQF queue is empty. It is the job of the firmware to configure the TDMA scheduler appropriately and provide enough margins to ensure this. For debugging, the register MMCTL_P0_3_QCLOSED_NONEMPTY can be used to monitor when a queue is closed with frames still in the queue.

29.4.17.3 CQF Configuration Example

The following example illustrates the use of the TDMA scheduler to configure the switch for CQF. The example uses 4 egress queues for frame queueing per port in total and uses only ports 0 and 1 for CQF. Queues 3 and 2 are configured as CQF queues, and queues 1 and 0 are used for best-effort traffic. The figure below show the queue states in both CQF cycles. The best-effort queues are always enabled since the CQF queues are of higher priority, so they are always given precedence over best-effort queues.



Index	START	INT	TCV_D_IDX	GPIO[0]
0	1	0	0	0
1	0	0	1	1

Index	OFFSET	GATE_MODE	QGATE [q3:q0]
0	0	1	1011
1	X1	1	0111

Figure 29.46 CQF configuration example

(1) Priority Classification

The switch must be properly configured to resolve the egress priority of the received frames into the enabled egress queues. To ease the configuration, the priority resolves to queues 0 or 1 for best effort traffic, and queues 3 and 4 for CQF traffic. See section 29.4.3.5. Frame Classification and Priority Resolution for details on frame priority resolution.

Table 29.35 CQF configuration example – Priority resolution

Register	Value	Description
VLAN_PRIORITYn (n = 0 to 3)	o33221100 (octal)	Remap VLAN priorities into the 4 egress queues. Both ports are configured the same.
IP_PRIORITYn (n = 0 to 3)	Map all to queue 0-3	Ensure all possibly used IP TOS/COS field mappings are cleared. If TOS/COS field priorities should be used, the application must ensure they map only to queues 0-3.
PRIORITY_CFGn (n = 0 to 3)	0x01	Enable VLAN priority resolution (bit 0) to respect above VLAN_PRIORITY. Also set the default priority to 0 to avoid interference with CQF traffic. If required by the application it may choose to also enable other additional priority resolutions here ensuring they only map into queues 0-3.

(2) TDMA Scheduler

The TDMA scheduler needs to be configured to toggle the ETHSW_TDMAOUT[0] signal as shown in Figure 29.46 and to open and close queues 2 and 3 to match the forwarding state in the same figure. For the configuration example a cycle time of 4 ms is assumed.

Table 29.36 CQF configuration example TDMA scheduler (1 of 2)

Register	Value	Description
TDMA_CONFIG	0x04, 0x05	TDMA scheduler enable (bit 0 = 1) and timer selection (bit 2 = 1). This register must be written last after all other configuration registers are set correctly.
TDMA_ENA_CTRL	0x3	Indicate that ports 0 and 1 will use the TDMA functions and that the memory controller shall not do any frame prefetching.
TDMA_START	40,000,000	Set a time value of 40 ms for the start (this is arbitrarily chosen here).

Table 29.36 CQF configuration example TDMA scheduler (2 of 2)

Register	Value	Description
TDMA_MODULO	1,000,000,000	Set timer modulo of 1 second. Assumes the local system's timer wraps around every 1 second.
TDMA_CYCLE	8,000,000	Set the cycle time to 8 ms, which is twice the CQF cycle time
TCV_SEQ_LAST	0x1	Last entry in the TCV sequence table. As shown in Figure 29.46, only two entries are used.
TDMA_GPIO	0x0	Configure the GPIO signals to operate in level mode. Here only gpio[0] is used, so the other signals can be configured as required.

Table 29.37 CQF configuration example – TCV sequence

Entry	TCV_SEQ_CTRL	Notes
0	0x00000001	START (bit 0) = 1 INT (bit 1) = 0 TCV_D_IDX (bits [10:2]) = 0x000 GPIO[0] = 0 Set the START bit to 1 so this TCV is executed when a new cycle starts. GPIO[0] is set to 0 here. This is the first CQF cycle.
1	0x00400004	START (bit 0) = 0 INT (bit 1) = 0 TCV_D_IDX (bits [10:2]) = 0x001 GPIO[0] = 1 Second CQF cycle. Toggle GPIO[0].

Table 29.38 CQF configuration example – TCV Data

Entry TCV_D_ADDR	TCV_D_OFFSET[i]	TCV_D_CTRL[i]	Notes
0	0	0x00030B58	INC_CTR0 = 0 INC_CTR1 = 0 RED_PERIOD = 0 OUT_CT_ENA = 1 IN_CT_ENA = 1 TRIGGER_MODE = 0 GATE_MODE = 1 HOLD_REQ = 0 QGATE = 0xB PMASK = 0x3 Set queue 2 in closed state as it is buffering, and queue 3 in open state as it is forwarding
1	4,000,000	0x00030758	INC_CTR0 = 0 INC_CTR1 = 0 RED_PERIOD = 0 OUT_CT_ENA = 1 IN_CT_ENA = 1 TRIGGER_MODE = 0 GATE_MODE = 1 HOLD_REQ = 0 QGATE = 0x7 PMASK = 0x3 Set queue 2 in open state as it is forwarding, and queue 3 in closed state as it is buffering. This TCV is executed 4 ms after the TDMA cycle start.

(3) Memory Controller

The register MMCTL_CQF_CTRL_Pn configures per-port the queues and control signal to be used. Ports 0 and 1 are used for CQF with priorities 2 and 3 being used for time-sensitive traffic, and queues 1 and 0 for best-effort traffic. The following table shows the configuration for this example for ports 0 and 1. Other ports keep this register programmed to 0x0 to disable CQF.

Table 29.39 CQF configuration example – MMCTL_CQF_CTRL_Pn

Register	Value	Description
MMCTL_CQF_CTRL_P0	0x0000020C	Set priorities 2 and 3 for CQF in PRIO_ENABLE0. Program queue 2 to be the CQF base queue in QUEUE_SEL0. Select ETHSW_TDMAOUT[0] for selecting the buffering queue in GATE_SEL0.
MMCTL_CQF_CTRL_P1	0x0000020C	Set priorities 2 and 3 for CQF in PRIO_ENABLE0. Program queue 2 to be the CQF base queue in QUEUE_SEL0. Select ETHSW_TDMAOUT[0] for selecting the buffering queue in GATE_SEL0.

29.4.17.4 Additional Usage Notes

- When egress queues are enabled for CQF, these queues will have cut-through disabled. Frames are not allowed for transmission until the full frame is received. If a received frame results in the frame to be sent to a CQF queue in port A and a non-CQF queue in port B, for port B the frame can be transmitted in cut-through mode if possible.
- A frame received on a port that is time-sensitive can be forwarded to another port with CQF enabled and to a port without CQF (for instance, a copy to the local CPU). For the non-CQF enabled port the egress queue used is the priority indicated by the priority resolution process, while the CQF port will store the frame in the queue indicated by QUEUE_SEL0 and the state of ETHSW_TDMAOUT. This means that a frame can be queued up in different queues in each port. This is not a problem except when memory pools are enabled in any of the involved CQF queues. To ensure proper operation of memory reservations in this scenario, the priority resolution must be configured so all time-sensitive traffic intended for CQF has the same priority as the two CQF queues, and the memory pool must include both CQF queues.
- To implement something like traffic interleaving as described in T.5.2 of IEEE 802.1Q, the queueing function must be based on the ingress port. This is achieved by setting REF_SEL0 to 1 so the gate control signal is based on the input port. The forwarding function is independent of the gating signal and it is simply implemented by opening and closing the queues using the TDMA scheduler of each port independently.
- Guard-bands can be provided by using additional TCVs to shift the toggling of the ETHSW_TDMAOUT signals and queues closing and opening. These guard-bands can delay the switch of queues for a port to compensate for switch-to-switch delays.
- Traffic interspersing can be enabled in the best-effort queues. This reduces the interference caused by best-effort traffic when transitioning between CQF periods.
- The switch provides two flooding masks that can be used to prevent traffic from ports not participating in CQF to transmit into the CQF ports. If these ports need to generate traffic into the CQF ports care must be taken in the programming of the priority resolution so none of the frames forwarded into the CQF ports matches the CQF priorities.

29.4.18 IEEE 802.3br Traffic Interspersing

29.4.18.1 Overview

The switch supports traffic interspersing as defined by the IEEE 80.3br specification. The goal is to reduce the latency to the medium for latency-critical traffic (referred to as express traffic) on transmission by interrupting any best-effort frame (referred to as preemptable traffic) that is being transmitted.

Each port's next-frame selection controller monitors the availability of traffic in the up to 8 egress queues. These queues are configured via a control register into preemptable queues (low priority queues) and express queues (high priority queues). The controller then gives priority to the express queues over the preemptable queues, which is no different than using regular strict-priority arbitration. However, when traffic interspersing and preemption is enabled and operational, the controller can interrupt an active frame from a preemptable queue as needed to prioritize express frames. The controller will use strict-priority or weighted round-robin arbitration amongst the preemptable and express-defined queues but note that express queues have priority over preemptable queues regardless of its queue number.

The diagram in the figure below illustrates traffic transmission without interspersing. Two queues are represented with queued frames at different points, with each bar representing the frame transmission length and the additional queuing time as a shaded area. In this scenario the next-frame selection controller uses strict priority arbitration to select the next frame to be transmitted. The time it takes for high priority frames to be transmitted is greatly affected by the transmission time of lower priority frames that get selected when there are no high priority frames in the queue.

When traffic interspersing and frame preemption is enabled, the high priority traffic can access the media much faster as seen in Figure 29.48. The next-frame controller will preempt a preemptable frame when another express frame appears ready for transmission. The queueing times, and therefore the latency of the high priority frames is greatly reduced, as well as the jitter of such latency. The side effects are of course that the low priority traffic takes longer in being transmitted, and the link utilization goes slightly down as the low priority frames are divided in ethernet segments which must be each individually framed for transmission (IPG, preamble, SFD and FCS for each segment).

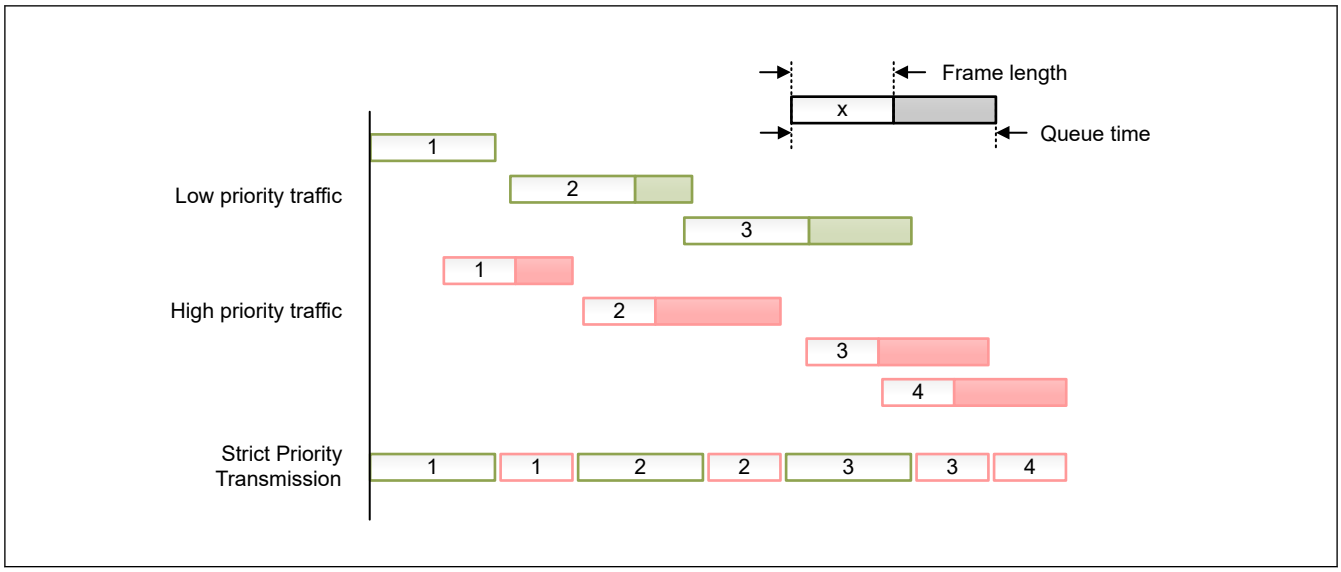


Figure 29.47 Transmission without traffic interspersing

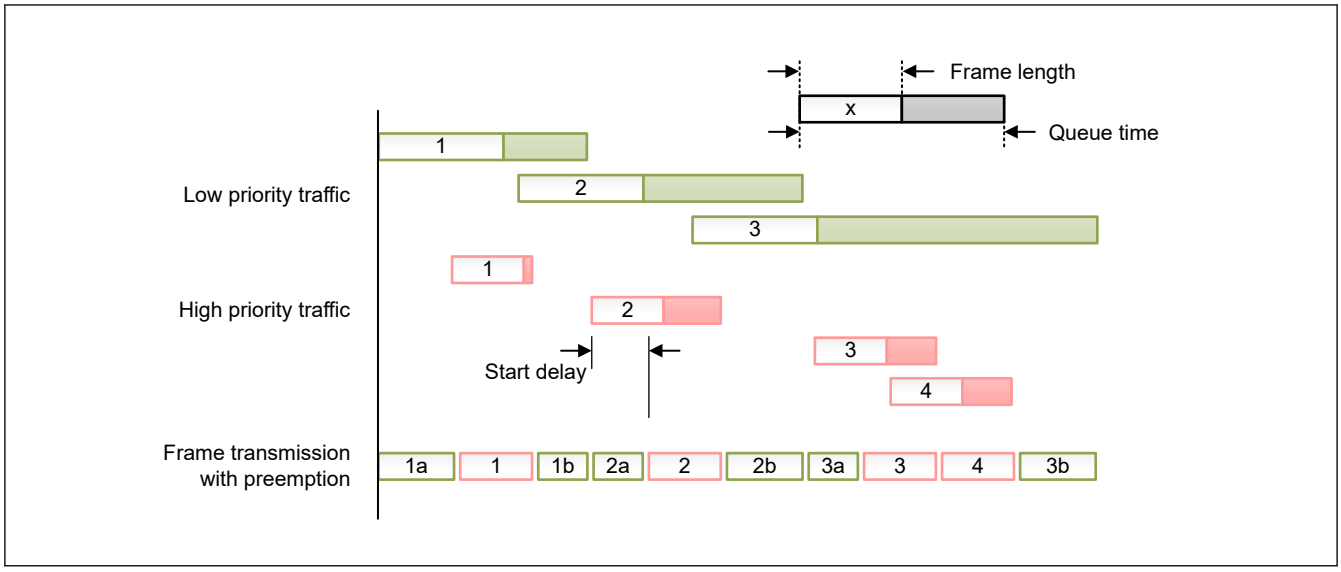


Figure 29.48 Transmission with traffic interspersing and frame preemption

On receive the traffic interspersing function performs the reassembly of preemptable frames. Once reassembled these frames are no different from express frames and they follow the standard forwarding process. One exception however is that reassembled frames are never forwarded in cut-through mode as a preemption event would cause an underrun. Express frames on the other hand have no special treatment as they are indistinct from normal Ethernet frames and can be forwarded cut-through if it is enabled.

The MAC also generates and receives response and verify frames to validate that both ports support preemption, as required by the IEEE 802.3br. This however can be overridden, and preemption can be forced to operate without this verification process.

Preemptable traffic can be stopped and preempted to allow an external agent application to generate an idle window for express traffic. This causes the transmission of preemptable traffic to stop even if there is no express traffic queued. This

action can be performed by the TDMA controller to allow periodic windows of express-only traffic, and from control registers to allow for direct control from the firmware.

29.4.18.2 Operation with Traffic Preemption Disabled (TX and RX)

Traffic preemption must be first enabled, and the verification process successfully completed or be disabled for it to operate. At a higher layer, the support for traffic preemption is auto-negotiated by communicating additional Ethernet capabilities using LLDP frames (see clause 79.3.7 of the IEEE 802.3br standard).

When traffic preemption is non-operational all queues are treated as express queues and the MAC sends all frames using the standard SFD (0xD5).

On receive preemptable frames are still accepted and reassembled even when preemption is disabled. This can be changed by setting the control bit BR_CONTROL_Pn.RX_BR_SMD_DIS to disable the acceptance of frame delimiters other than 0xD5. This can be useful to prevent spurious frame drops due to corruption of the preamble bits when preemption is not used.

The TDMA controller's hold_req function has no effect when preemption is disabled as all egress queues are configured as express queues.

29.4.18.3 Operation with Traffic Preemption Enabled (TX)

Traffic preemption is enabled via the register BR_CONTROL_Pn.PREEMPT_ENA. Enabling preemption must be done by a higher layer after having auto-negotiated it via LLDP (see clause 79.3.7 of the IEEE 802.3br standard), or if it is a closed network where it is known that the link partner supports traffic interspersing. Also, at least one queue must be programmed to transmit preemptable traffic via the control register MMCTL_PREEMPT_QUEUES. Note that a single register for all ports is provisioned, so all ports that have preemption enabled will use the same queues. The ports that have preemption disabled, or for which preemption is enabled but not operational will ignore the configuration in MMCTL_PREEMPT_QUEUES and treat all queues as express queues.

For frame preemption to operate the preemption verification process must successfully complete, or it must be disabled via the control register BR_CONTROL_Pn.VERIFY_DIS (see section 29.4.18.5. Preemption Verification Process). The verify process is a hardware mechanism to ensure that both peers can receive preemptable frames before transmitting preempted frames. The operational state of the port's preemption capability can be observed in the MAC STATUS_Pn.BR_VERIF_ST register.

The IEEE 802.3br standard specifies how the frames must be preempted. Since the frames must always be valid Ethernet frames when leaving the MAC, the minimum frame length must be respected. The Figure 29.49 shows an example where two express frames are interspersed in one long preemptable frame. The process is as follows:

1. A frame is queued in a preemptable queue and no frames in the express queues are available. The next-frame controller initiates the transmission of the preemptable frame.
2. The preemptable frame is started with SMD-Sx to indicate that this is a start of frame.
3. A frame is queued in an express queue.
4. The next-frame controller checks whether the number of bytes transmitted is greater than minFragLength = 64 × (1 + addFragSize) − 4 (4a) and whether the remaining bytes for the frame is ≥ 60 (4b). In this example it is, so current frame is preempted.
5. The MAC adds the mCRC to the frame to indicate that this is an mPacket and that more segments are remaining.
6. After the IPG, the express frame is transmitted.
7. The express frame uses the SMD-E as the identifier and the standard FCS for ethernet frames.
8. Once the express frame transmission is completed, the preempted frame resumes if there are no more frames queued in the express queues.
9. The continuation mPacket is identified with SMD-Cx to indicate that this is a continuation frame, and the fragment count is zero to indicate this is the first continuation mPacket for this frame.
10. Another frame is queued in an express queue, but preemption is not yet executed until the minFragLength is met.
11. After minFragLength bytes are transmitted and more than 60 bytes remain for the frame, the preemptable frame is preempted a second time.

12. The continuation mPacket is terminated by the MAC with an mCRC to indicate that there is still more data pending for this frame.
13. The second express frame is then transmitted after the IPG has elapsed. The same format as the previous express frame is used.
14. Once the express frame is transmitted, the preemptable frame resumes transmission.
15. The second continuation mPacket is identified by using SMD-Cx, and a fragment count of 1.
16. The final mPacket of the preemptable frame is terminated with the standard FCS so the receiving port can detect that there are no more segments remaining. The FCS may be bad if the frame must be discarded (or got corrupted during transmission), but as long as the FCS does not match the mCRC, the frame is deemed complete by the receiving port.

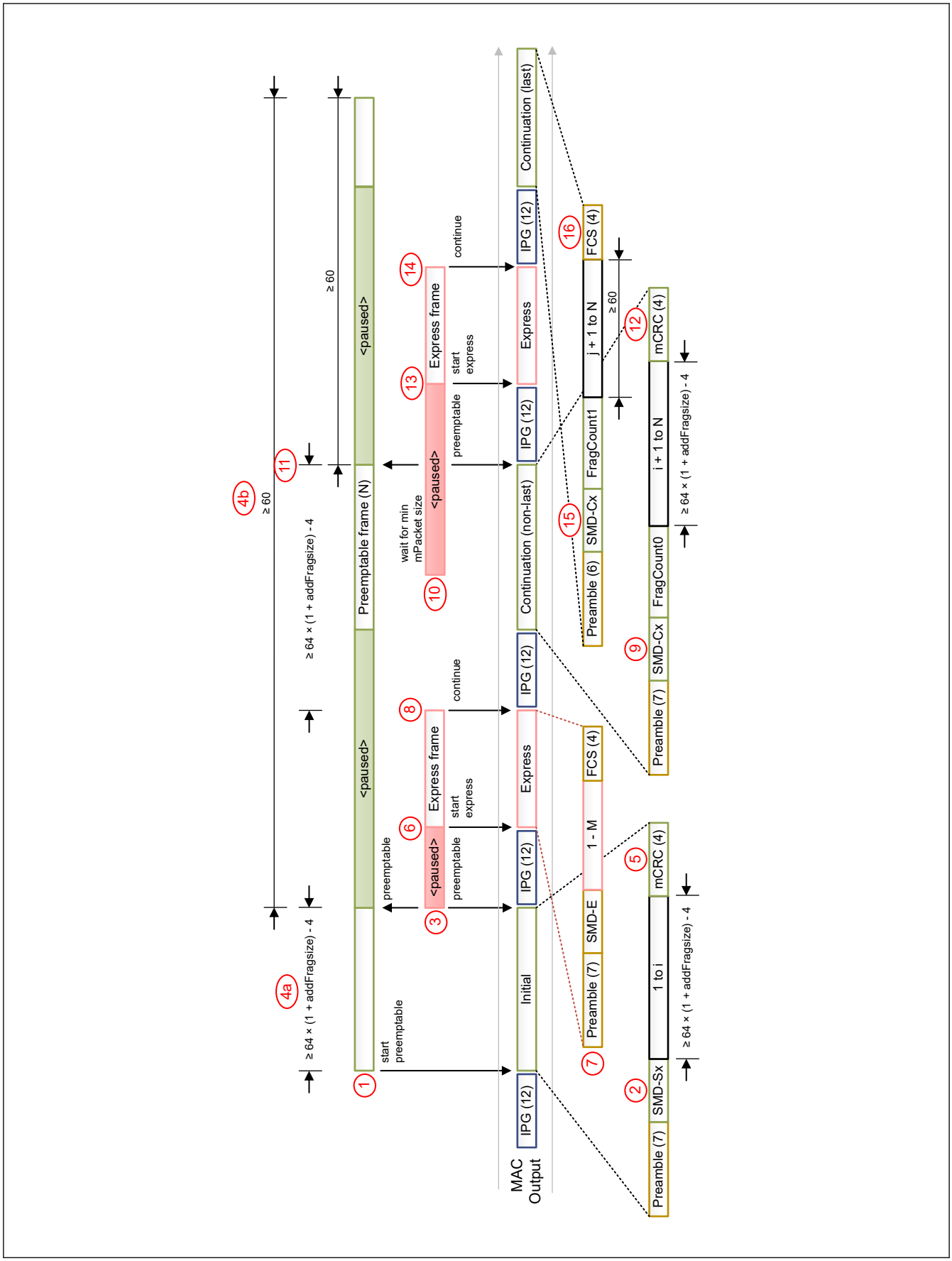


Figure 29.49 Example of frame preemption

In summary, these are the rules that govern how frames are transmitted once traffic preemption is enabled:

1. Express traffic has always precedence over preemptable traffic. The preemptable traffic is transmitted from the queues configured as such in the register MMCTL_PREEMPT_QUEUES. This supersedes the strict-priority arbitration done by the next-frame selection controller.
2. Express frames are transmitted as a single frame delimited by using SMD-E, which is identical to the SFD (0xD5). These mPackets are never interrupted and use the standard CRC32 FCS calculation.
3. Preemptable frames are transmitted with one or more mPackets (see Table 29.7 for SMD encodings).
   (a) The initial mPacket is delimited with SMD-Sx, where x goes from 0 to 3 and increments for every new packet.
   (b) The subsequent (continuation) mPackets are delimited with SMD-Cx, where x goes from 0 to 3 and is the same number as the initial mPacket SMD-Sx. Also, each continuation mPacket indicates the fragment count after SMD-Cx via the FRAG_COUNT field which encodes four different values (see Table 29.8).
   (c) mPackets for express frames use the standard Ethernet CRC32 FCS.
   (d) mPackets for preemptable frames which are not the final mPacket use an mCRC, which is the standard Ethernet CRC32 FCS from the first byte of the full frame up to the last octet of the current mPacket, but with an XOR mask of 0x0000FFFF (see clause 99.3.6 of the IEEE 802.3br specification).
   (e) The last mPacket for a preemptable frame (which can be also the first one if the preemptable frame is not preempted) uses the standard Ethernet CRC32 FCS calculated from the first octet of the full frame.
4. Preemptable frames can be preempted to transmit an express packet if the following conditions are true:
   (a) The current mPacket is at least minFragLength = 64 × (1 + addFragSize) − 4 octets. addFragSize is configured in BR_CONTROL_Pn.ADDFRAGSIZE.
   (b) The preemptable frame has at least 60 octets of data to be transmitted.
5. If tx_hold_req (mapped to MM_CTL.request (hold_req) in the IEEE 802.3br specification) asserts, preemptable traffic is stopped. If a preemptable frame is currently being transmitted and traffic preemption is enabled, the frame is preempted if the conditions in point 3 are met.

29.4.18.4 Operation with Traffic Preemption Enabled (RX)

The receive side processes the preamble of every frame to determine whether it is an express frame, an mPacket part of a preemptable frame, or a "verify" or "response" frame. The port must also be configured to accept SMD-xx as frame delimiters by setting the control bit BR_CONTROL_Pn.RX_BR_SMD_DIS to 0.

Express frames are passed directly to the forwarding processing. Preemptable frames are passed also to the forwarding process, but this process may be suspended if the frame is preempted. For "verify" and "response" frames the port discards these frames and uses them to advance the verification process as needed. Verify and Response frames are not counted in the standard MAC statistics. A separate set of counters track the receive (and transmit) of these frames (see section 29.4.18.8. Counters).

As part of the preamble processing, the port checks for preempted frames that the fragment sequencing is correct (indicated by FRAG_COUNT) for every received mPacket.

29.4.18.5 Preemption Verification Process

This process sends a "verify" frame to the link partner, and expects a "response" frame in return to validate that both ports support frame preemption (see section 29.4.1.6. Preemption Verify and Response Frames for more detail on the frames format, and clause 99.4.3 of the 802.3br IEEE standard).

When preemption is enabled, the MAC initiates the process by sending a "verify" frame to the link partner, and waits 10 ms (or the configured time, which can be between 1 ms to 127 ms in register BR_CONTROL_Pn.TX_VERIFY_TIME) for a "response" frame to be received. If no frame is received, the process is attempted two more times, and on the third failure the MAC stops attempting and inhibits frame preemption.

This process can be disabled via BR_CONTROL_Pn.VERIFY_DIS if it is known that the system supports preemption (i.e. a closed network). When verification is disabled the MAC will still respond to "verify" frames. Moreover, "verify" frames are always replied to unless BR_CONTROL_Pn.PREEMPT_ENA is set to 0 and BR_CONTROL_Pn.RESPONSE_DIS is set to 1.

The verification process is also automatically restarted when the link fails as indicated by the input phy_link.

Note: Setting BR_CONTROL_Pn.TX_VERIFY_TIME to less than 5 ms violates the 20% precision required by the 802.3br specification due to the timer implementation. The precision of the timer is +/− 1 ms.

29.4.18.6 PAUSE Operation with Traffic Interspersing

When PAUSE frames are enabled, the transmit MAC can be paused if a pause request is received from the peer port. When this happens the transmission of express and preemptable frames is stopped. If the pause causes the MAC to stop in the middle of a preempted frame, the frame does not resume transmission until after the PAUSE is cleared.

When the MAC is PAUSE it is still allowed to generate PAUSE frames, as well as Verify and Response frames.

29.4.18.7 MM_CTL.request (hold_req)

The IEEE 802.3br specification defines MM_CTL.request (hold_req) as a mechanism to allow the higher layers to stop the preemptable traffic in advance for express traffic, ensuring that the express traffic will have access to the media as soon as it gets to the output queues. The figure below details the operation of hold requests. When the tx_hold_req asserts (via the TDMA controller's hold_req function), a hold request is given to the next-frame controller, which causes it to stop the preemptable traffic (preempting the frame if needed). Afterwards, the link remains idle unless there are express frames queued for transmission. Express frames will have immediate access to the media, so they get transmitted with almost no delay as soon as they are queued for transmission. A hold request can also be issued by software via the control register MMCTL_HOLD_CONTROL. The preempted state and hold request indication can be checked in real-time via the register MMCTL_PREEMPT_STATUS.

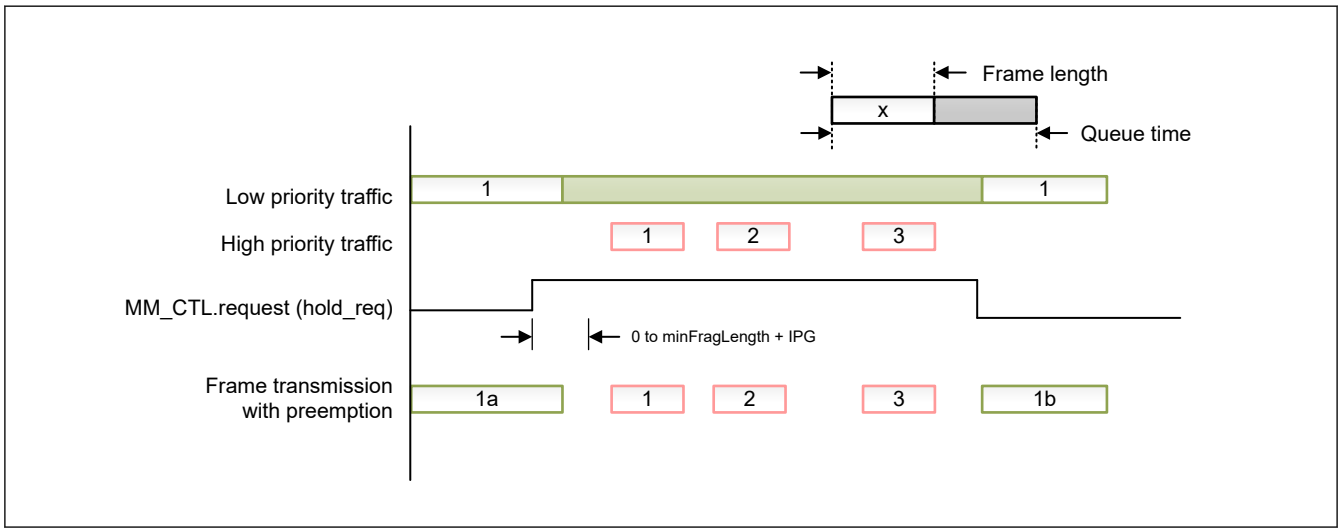


Figure 29.50 Operation of hold_req (MM_CTL.request (hold_req))

section 29.4.22.7. Transmit Frame Selection to Frame's FB Delay describes the delay relevant for hold_req transitioning from high to low and for a frame ready for transmission to be transmitted over the MII/GMII interface.

29.4.18.8 Counters

The table below lists the internal counters supported for 802.3br. Some of them are specified by the IEEE 802.3br standard, while others are custom and meant for debugging.

These counters are accessible in the global discard and 802.3br statistics memory page (see section 29.3.9. Global Discard and 802.3br Statistics Registers).

Table 29.40 Counters related to IEEE 802.3br (1 of 2)

Counter (width)	Counter Name	802.3br clause	Description
aMACMergeFragCountTx (32)	TX_FRAG_CNTn	30.14.1.12	Additional mPacket transmitted due to preemption.
aMACMergeFragCountRx (32)	RX_FRAG_CNTn	30.14.1.11	Increments for every continuation mPacket received.

Table 29.40 Counters related to IEEE 802.3br (2 of 2)

Counter (width)	Counter Name	802.3br clause	Description
aMACMergeFrameSmdErrorCount (16)	RX_SMD_ERR_CNTn	30.14.1.9	Received frame with an SMD-C when no frame is in progress.
aMACMergeFrameAssErrorCount (16)	RX_ASSY_ERR_CNTn	30.14.1.8	Assembly error seen (due to FRAG_COUNT)
aMACMergeFrameAssOkCount (32)	RX_ASSY_OK_CNTn	30.14.1.10	Frame successfully assembled.
aMACMergeHoldCount (32)	TX_HOLD_REQ_CNTn	30.14.1.13	Increments when hold_req transitions from FALSE to TRUE.
ResponseRxGoodCount (8)	RX_RESPONSE_OK_CNTn	N/A	Good Response frames received.
ResponseRxBadCount (8)	RX_RESPONSE_BAD_CNTn	N/A	Bad Response frames received.
VerifyRxGoodCount (8)	RX_VERIFY_OK_CNTn	N/A	Good Verify frames received.
VerifyRxBadCount (8)	RX_VERIFY_BAD_CNTn	N/A	Bad Verify frames received.
ResponseTxCount (8)	TX_RESPONSE_OK_CNTn	N/A	Response frames transmitted.
VerifyTxCount (8)	TX_VERIFY_OK_CNTn	N/A	Verify frames transmitted.

Note: When BR_CONTROL_Pn.RX_STRICT_BR_CTL is set to 1, the VerifyRxBadCount and ResponseRxBadCount will only increment for frames with the corresponding SMD, a length of 64 bytes, a frame content of all 0's, and a bad mCRC.

29.4.18.9 Limitations and Observations

- The IEEE 802.3br standard defines the minimum fragment length as minFragLength = 64 × (1 + addFragSize) − 4. However, this MAC uses the following size as the minimum length minFragLengthReal = 64 × (1 + addFragSize). The net effect is that the minimum fragment can be larger by 4 bytes than what the specification states, but this is not a violation since minFragLengthReal is greater than minFragLength.
- Frames received as express frames can be sent over preemptable queues due to the priority remapping process. If this is expected to happen then the CTFL function must be enabled, otherwise these frames will not be preemptable on transmission. The CTFL is a frame length storage table for frames that are received when cut-through mode is enabled as the frame length is not known until the frame is fully received.
- When a preemptable frame is transmitted without CRC (i.e. the MAC will insert the CRC), the minimum length of a continuation mPacket is 68 bytes. If the application is providing the CRC, then the minimum length of the last continuation mPacket is 64 bytes, but still 68 bytes for the initial or non-last continuation mPackets.
- IEEE1588 1-step PTP frames should never be preempted. The MAC requires these frames to be contiguous to properly update their fields. The application must ensure that PTP frames are transmitted as express traffic.
- When cut-through operation is enabled in the switch the CTFL table must be enabled when frame preemption is enabled. This is required as the CTFL stores the frame's length which is needed to determine preemption. If the CTFL is disabled it could yield to a live-lock condition as a frame of unknown length blocks the preemptable queues.

29.4.18.10 Additional 802.3br Options

The following options are provided as non-standard options that can be enabled or disable to provide additional robustness.

- Strict preamble
  This mode enables checking for the preamble bytes in the SFD detection logic. The mode is configured in the BR_CONTROL_Pn.RX_STRICT_PRE field. When set to 0, the SFD detection logic will look for the SFD or any of the SMD-xx characters to accept a frame. When set to 1, the frame will be ignored if a character other than 0x55 is received prior to an SFD or SMD-xx.
- Disable 802.3br SMD-xx decoding
  This control is set in BR_CONTROL_Pn.RX_BR_SMD_DIS. When set to 1, it instructs the SFD detection logic to accept frames only based on the standard SFD (0xD5). This can be enabled when preemption is not supported to avoid false frame acceptances due to preamble corruption. Note however that even if a false acceptance occurs, the CRC checker will mark the frame as bad as the frame's FCS will not be calculated correctly.

- 802.3 verify process disable
  The verify process required for operating in preemptable mode can be disabled with the register BR_CONTROL_Pn.VERIFY_DIS. This can be set to 1 in a closed system well it is known that the peer port supports preemption, or for debugging the port.
- Strict verify and response frame decoding
  This mode can be set via BR_CONTROL_Pn.RX_STRICT_BR_CTL. When set to 1, the Response and Verify frames must have a correct mCRC, a total length of 64-bytes and a payload of all 0x00. When set to 0, only the mCRC is checked to validate that the frame is a Verify or Response frame. A value of 0 is required to comply with IEEE 802.3br.

29.4.19 Extended Frame Parser (EFP)

29.4.19.1 Overview

It supports ingress filtering and Ethernet frame header manipulation according to the following standards:

- IEEE 802.1CB, Clause 6.6: Active Stream Identification
- IEEE 802.1Q, Clause 8.5 and 6.9: TCI.PCP remapping / TCI.VID assignment.
- IEEE 802.1Q, Clause 12.20.3: Priority Regeneration Override
- IEEE 802.1Q, 8.6.5 and MEF 10.3: Flow classification and metering

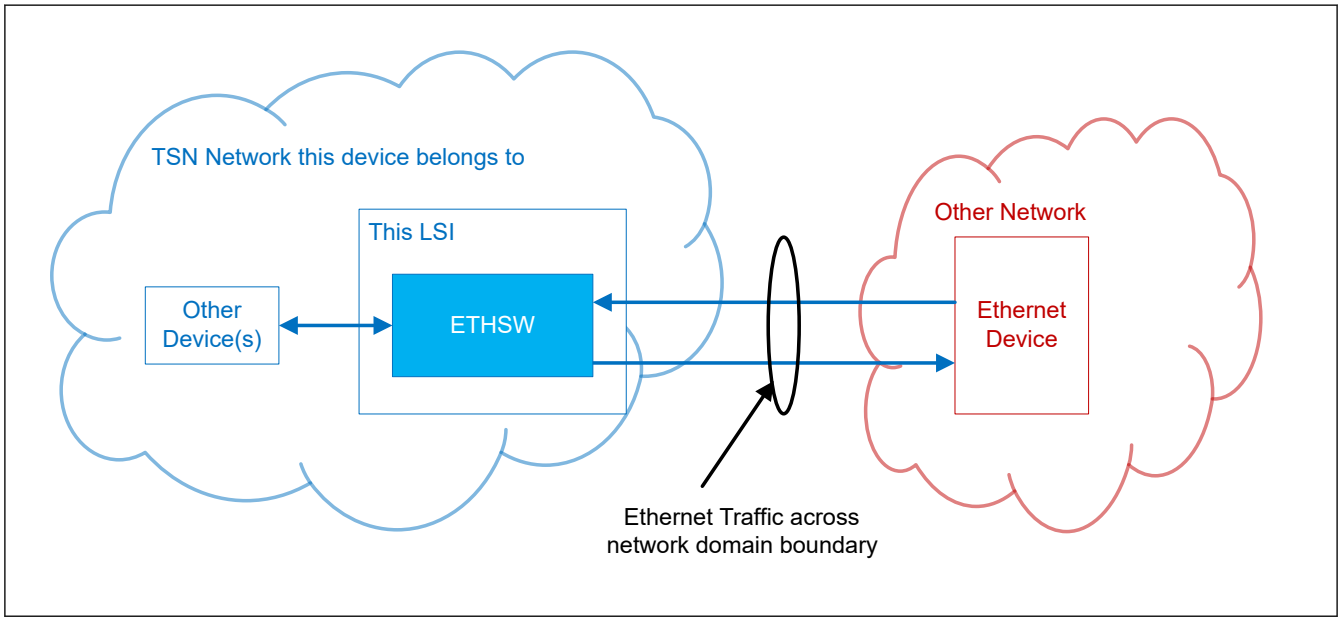


Figure 29.51 Network domain boundary

Configuration:

- 3 independent channels
  - one channel per ingress port of the switch
- 8 Filter per channel
- Priority regeneration 8; 8 per channel
- Active Stream Identification for 128 stream IDs (DA & VID & PCP) per channel
- Max SDU check per channel
- 8 Flow Meters per channel
  - Green Bucket only
  - Envelope Mode (concatenation) of meters for up to 3 buckets

If concatenating more than 3 buckets, overflow token may get lost.

29.4.19.2 Frame Processing Overview

This function processes only ingress frames of line ports (not for the management port). It shall handle incoming traffic from other network domains at the boundary ports of an industrial Ethernet/Profinet network domain (see Figure 29.51). The ingress processing has two purposes:

1. Protection of internal network resources from overflow due to excessive traffic from outside.
2. Header modification to align incoming frames to the internal traffic scheme.

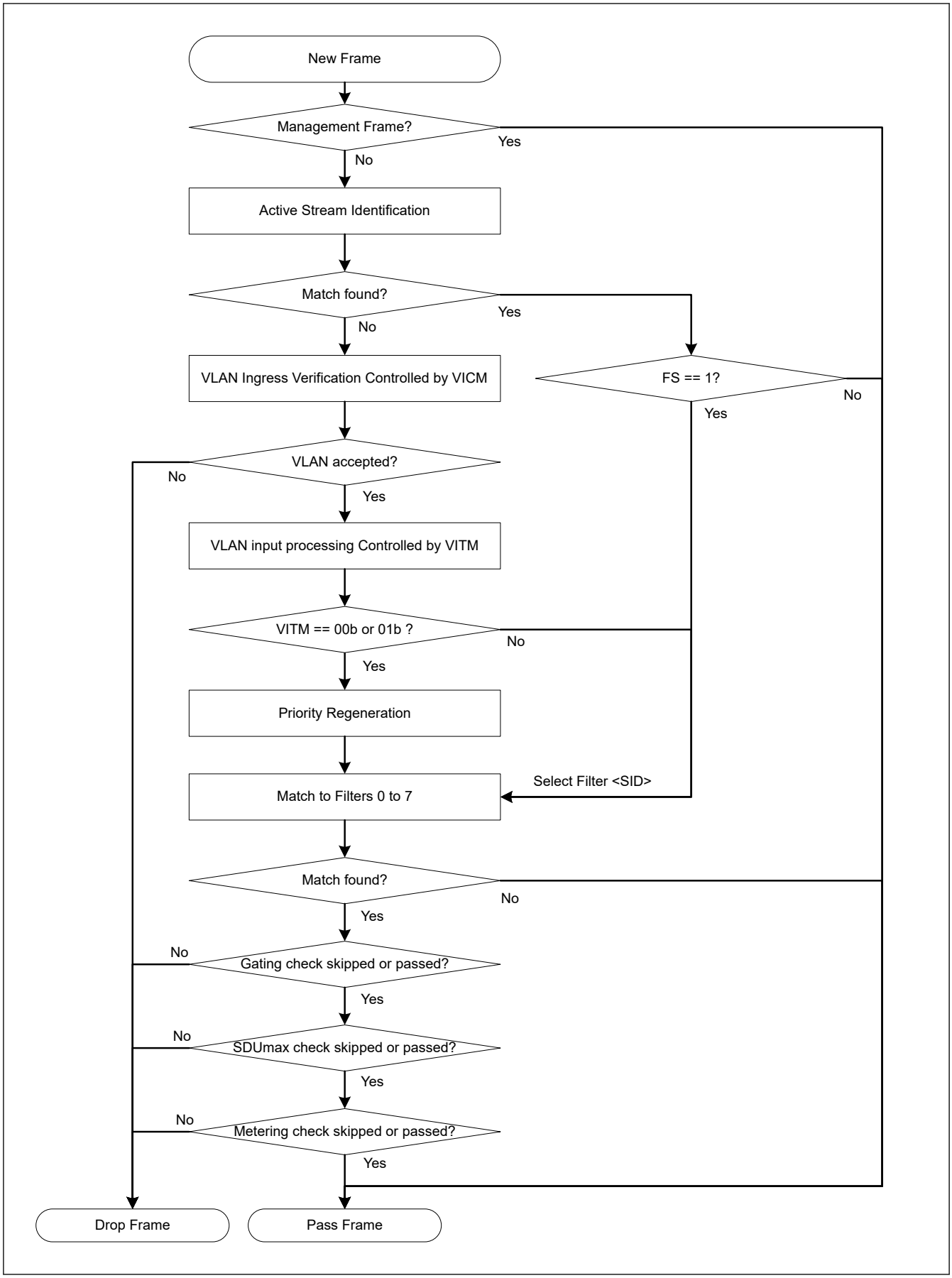


Figure 29.52 Frame processing flow

Figure 29.52: Frame processing flow shows the logical flow of frame processing. To stream processing of the frames, timing of the steps needs to be considered carefully. Features like pre-emption and cut-through cause technical limitations and corner cases for the function. Conditions like broken frames, i.e. cancellation of reception at any time, pre-emption without resuming rest of frame transmission, multiple pre-emptions of a frame (sequentially, not stacked) etc. shall be taken into account for a robust design. Software reset of the processing state machines shall be possible to recover function in case of deadlocks without full reset of the total system.

Timing wise the frame processing can be split into two sections:

1. Ethernet header based
   These processing steps operate on header information and immediately available information. For example, classification of the frame, VLAN tag modifications, active stream identification, gating check. All of these operations can be executed immediately after receiving the Ethernet header information, i.e. the first 14 octets of the frame. The maximum supported line rate is 1 Gbps. The minimum length of a valid Ethernet frame is 64 Octets. Considering preamble and inter frame gap of 8 octets each, this translates to approximately 640 ns for processing two frames back-to-back.
2. Ethernet frame length based
   These processing steps require the information about frame length. Since the frame length of an Ethernet II frame corresponds to the number of received octets, this information becomes available at the end of a frame reception. Consequently, forwarding or dropping decision of cut-through frames cannot be made depending on the frame length (SDUmax checking). However, frames exceeding the maximum length could trigger all consecutive frames of the same class to be dropped. This is an Ethernet header-based check as described above.

COMMAND_CONFIG.EFPI_SELECT must be set to 1 to process of frames. However, it can inspect ingress frames even if COMMAND_CONFIG.EFPI_SELECT is set to 0.

29.4.19.3 Frame Processing Details

(1) Management Frame Identification

Link local management frames are exchanged between neighbored bridges. They are only forwarded to the management port. Management frames are not automatically forwarded to other external ports of the switch. Management frames shall not be subject to ingress processing. According to IEEE802.1Q-2018 Table 8-1 management frames are identified by a Destination MAC Address (DMAC) in the range of 01-80-C2-00-00-00 to 01-80-C2-00-00-3F.

(2) Active Stream Identification

The Active Stream Identification function is defined in IEEE802.1CB-2017 Clause 6.6. It is executed first to determine if the frame is belonging to a defined TSN stream. If the frame matches to an entry in the Active Stream Identification table, the header information of the frame is updated as defined in IEEE802.1CB-2017 and the stream handle is derived from an entry of the Active Stream Identification table. This frame handle then directly refers into the filter table, it means bypassing the frame identification by the MAC/VLAN comparison in the filter entries. The frame conformance checking parameters of the respective filter entry are applied.

(3) VLAN Verification

In accordance with IEEE802.1Q-2018 section 6.9 this function checks the VLAN tag of the frame for conformance to the VLAN assignments of the port. Following settings are supported:

Table 29.41 VLAN input verification mode

Mode	Pn_VLAN_MODE.VICM	Description
0	00b	Disabled, all frames pass
1	01b	Admit Only VLAN-tagged frames
2	10b	Admit Only Untagged or Priority-tagged frames
3	11b	Do not use (drop all frames)

Note: VLAN verification is only done for frames not identified as TSN streams.

Note: This function is already implemented in Switch Function. But due to its functional linkage to the Active Stream Identification function, so use this function here.

(4) VLAN Ingress Processing

This process step manipulates the VLAN tags of ingress traffic. It is only performed, if the Active Stream Identification did not find a match for the ingress frame. Therefore, and since the VLAN tag may be subject to the following filters, the already existing VLAN processing function cannot be used here.

Table 29.42 VLAN input tagging mode

Mode	Pn_VLAN_MODE.VITM	Description
0	00b	Disabled, VLAN tag is neither added nor overwritten.
1	01b	Single Tagging with Passthrough / VLAN ID (VID) Overwrite. Insert Tag if untagged frame. Leave frame unmodified if tagged and VID > 0. If tagged with VID = 0 (priority tagged) then the VID will be overwritten with the VID from Pn_VLAN_TAG and priority is kept
2	10b	Single Tagging with Replace If untagged, add the tag Pn_VLAN_TAG, if single tagged, overwrite the tag.
3	11b	Tag always Insert a tag Pn_VLAN_TAG always. This results in a single tagged frame when an untagged is received, and a double tagged frame, when a single tagged frame is received (or triple tagged if double-tagged received etc.). Note: This mode is prohibited for switch ports running at 1 Gbps line rate.

(5) Priority Regeneration

In accordance with IEEE802.1Q-2018 clause 6.9.4 this function reassigns the PCP value of the VLAN tag. All 8 possible PCP values are replaced by the respective table entry. To bypass priority regeneration, replacement values are same as the input value (default setting after reset release).

(6) Filter Function (Match for Filters)

Supports Per-Stream Filtering and Policing (PSFP) function as defined in IEEE802.1Q-2018 clause 8.6.5.1.

In the Profinet TSN use-case, PSFP application is generalized from TSN streams to all classes of Ethernet frames like broadcast, multicast, and unicast traffic.

The filter function identifies Ethernet frames by their Destination MAC address (DMAC), Source MAC address (SMAC) or VLAN tag and assigns them to a SDUmax, flow gate and flow meter. Each channel provides 8 filters.

Each frame is matched against the filters sequentially starting at filter 0. The first matching filter determines the frame policing parameters. If no filter matches, the frame will be forwarded unconditionally, i.e. PSFP is not applied to the frame.

Frames identified and relabeled by the Active Stream Identification function can be assigned to one of these filters directly. The frame policing parameters of the respective filter are applied without prior frame re-identification by the filter.

Note: Frames processed by the Active Stream Identification function shall get a "frame handle" assigned as lookup value from the table. This frame handle shall be an index (0..7) to the Filter table.

By the frame handle, the respective entry of the filter table can be immediately selected for the respective frame without need to search and match the header information again. Since no new search is needed, the MAC/VLAN entry in the filter table may not match to the frame header. This is no functional limitation. Filtering is optional for these frames, it means, along with the stream ID, a control bit (FS) can enable/disable the PSFP function for these frames.

(7) Gating Check

The gating check is enabled by Pn_QSFTBLs.GAIDV set. The frame can pass the gate, if the tdma_gpio signal for switch selected in Pn_QSFTBLs.GAID is set to 1. In this case the Pn_QGPPC counter is incremented. If not, the frame is dropped, the error interrupt is signaled by the respective flag in Pn_QGEIS.QGMOIS and the counter of dropped frames Pn_QGDPCg is incremented.

If the Pn_QGMOD.QGMOD[g] bit for the respective gate is set and if the error interrupt flag Pn_QGEIS.GQMOIS[g] corresponding to the gate is set, all frames assigned to the respective gate are dropped independently. The software must clear the Pn_QGEIS.QGMOIS[g] flag to re-enable frames passing the gate.

(8) SDUmax Check

The SDUmax check is enabled by Pn_QSFTBLs.MSDUE set. The frame length (FL) is compared against the SDUmax value Pn_QSFTBLs.MSDU defined in the filter entry.

If the FL is less or equal to SDUmax, the frame passes the filter.

If the FL is higher than SDUmax

- The frame is dropped and the SDUmax error interrupt is generated
- If Pn_QSFTBLs.QSMSM is set, all consecutive frames matching this filter are dropped if the error interrupt flag is still set.

Note: The FL is known at the end of the frame. While frames in S&F mode can still be canceled, frames transmitted in cut-through mode cannot be prevented from transmission. However, for such frames the error interrupt and the optional permanent blocking function would be available.

29.4.19.4 Active Stream Identification

It provides a table with 128 entries per channel for Active Stream Identification. The table is organized as hashed table with 64 hashes with 2 entries per has value to mitigate hash collisions. The Address is a hash of lookup {DA, VID, 0000b (padding)}: CRC-7 Polynomial (IEC 60870-5) 0x65 = $x^7 + x^6 + x^5 + x^2 + 1$. It needed to be reset to 7'h3F on a SOP.

Note: The Active Stream Identification table is stored in RAM. This RAM is not initialized by the hardware. The software driver must initialize the memory before enabling the ingress policing function.

Table 29.43 Active Stream Identification table entry

Item	Width	Description
DMACI	48	DMACI/VIDI are the DMAC and VID of the incoming frame, which are matched to this table entry. A full match is required.
VIDI	12	
DMACO	48	If a frame matches this entry, its DMAC, VID, and PCP are replaced by these values. The DEI bit keeps its value.
VIDO	12	
PCPO	3	
SID	3	This is the frame handle assigned to the matched frame. In case of this implementation, it directly refers to a filter entry in the filter table.
FS	1	If set, the filter referenced by SID is applied. If cleared, no filtering is applied.
EV	1	If set, this table entry is valid. If cleared, this table entry shall be ignored.

Note: 16 bytes (128 bits) per entry

Table 29.44 Active Stream Identification table entry bit fields

DMAC Ingress	VID Ingress	DMAC Translate	VID Translate	PCP Translate	SID	FS	EV
Bit [127:80]	Bit [79:68]	Bit [67:20]	Bit [19:8]	Bit [7:5]	Bit [4:2]	Bit [1]	Bit [0]

The SID obtained from the active stream identification table is a direct index into the filter table. It is enabled by setting the FS bit. By this link, the gating and metering features can be utilized for each stream without specific MAC/VLAN filter definition in the filter table.

Note: For frames identified by the Active Stream Identification, as the further steps on VLAN verification, Priority Regeneration and Null stream identification are not applied. Gating, Metering and SDUmax checks are applied only by a linked filter entry.

Following is a flow of Active Stream Identification.

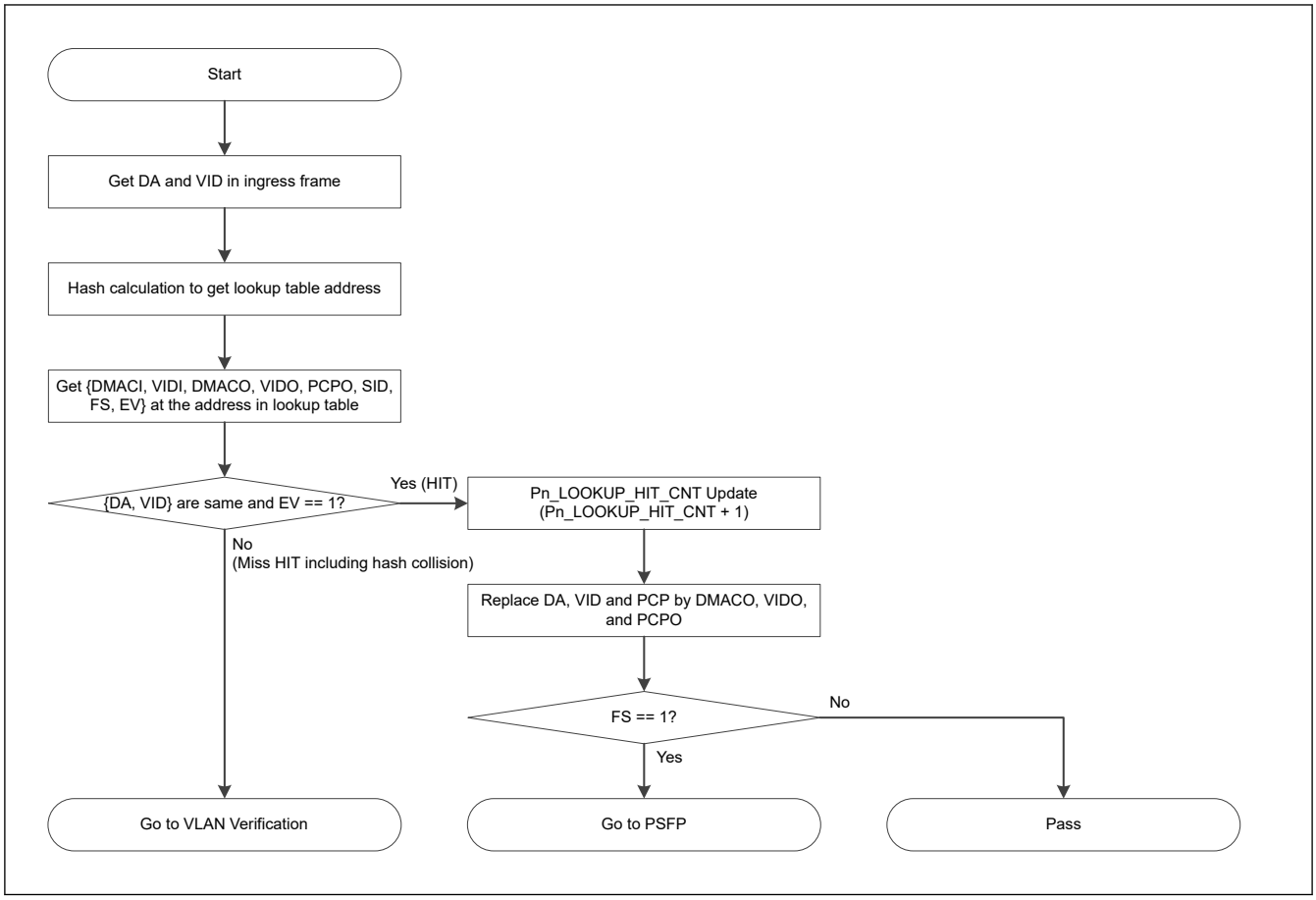


Figure 29.53 Active Stream Identification flow

Hash calculation and comparison image is as follows.

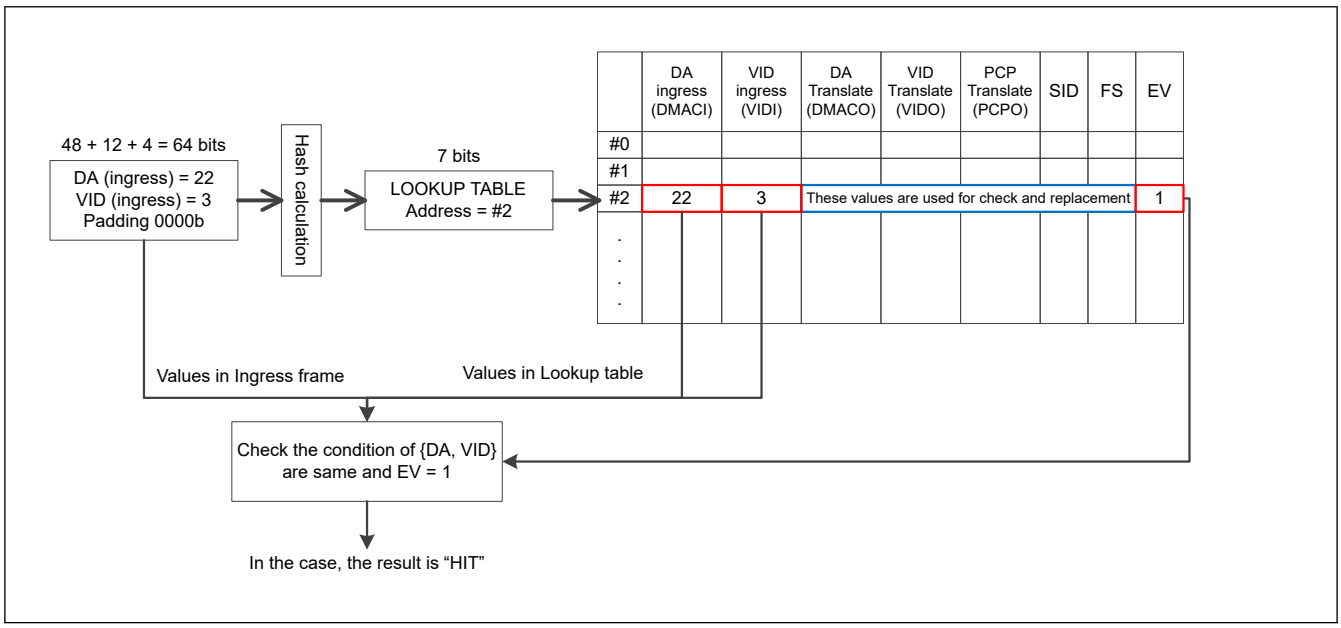


Figure 29.54 Hash calculation and comparison image

Before using the EFP functions, the Active Stream Identification Table should be initialized using the flow in Figure 29.55. This flow will initialize the Active Stream Identification tables of all three channels.

Note: The Active Stream Identification Table is not initialized by hardware function. Thus, it MUST be initialized before the respective channel operation is enabled by triggering the CHANNEL_ENABLE register bit of the respective channel.

Initialization of all three memories can be done in parallel, it means all three ASI_MEM_ADDR.MEM_REQ control bits can be set simultaneously.

Note: Initialization of the Active Stream Identification Table is mandatory. It cannot be skipped even this function is not used.

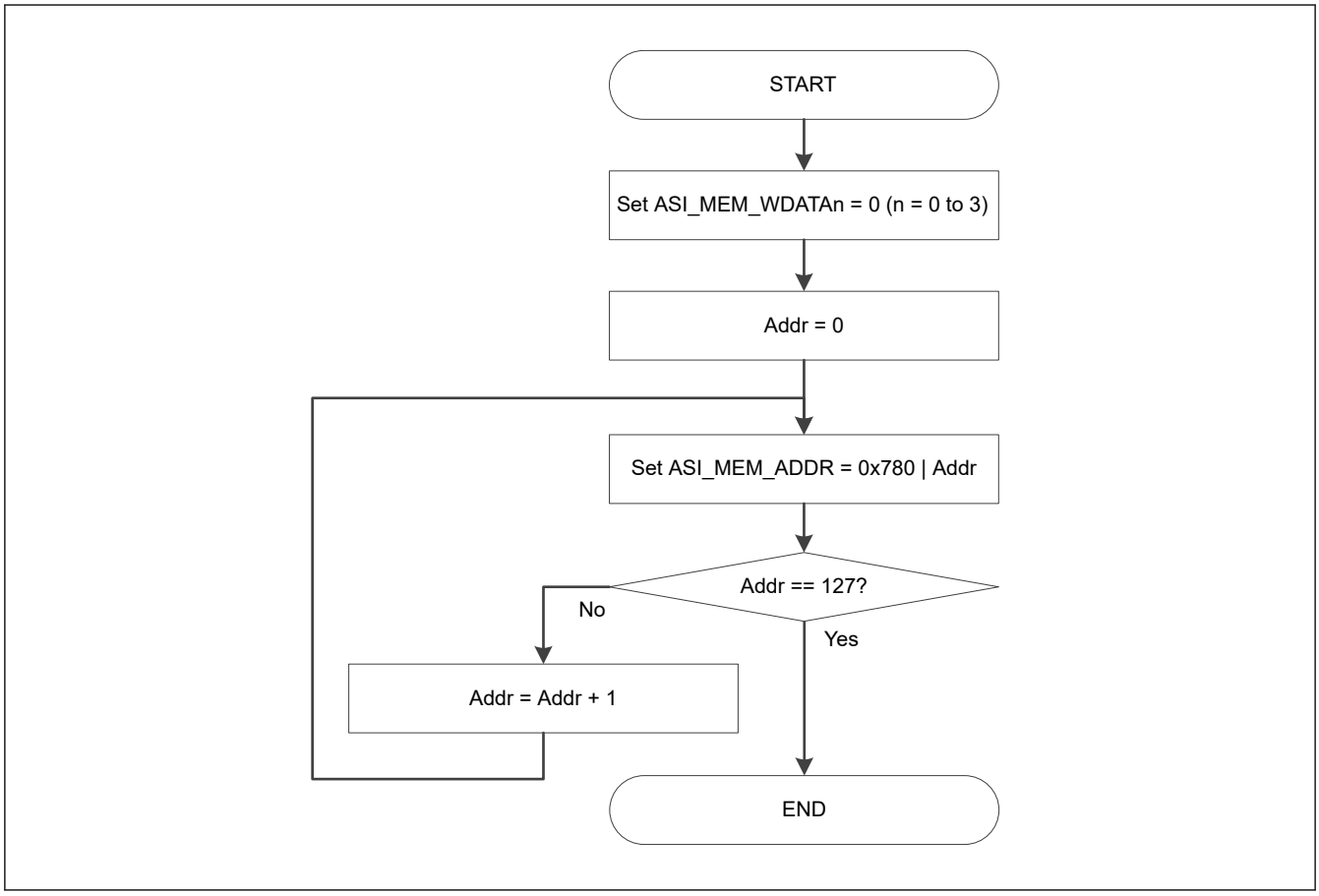


Figure 29.55 Active Stream Identification initialization

Setting an entry into the Active Stream Identification Table shall follow the flow in Figure 29.56.

Deleting an Entry shall follow the flow in Figure 29.57.

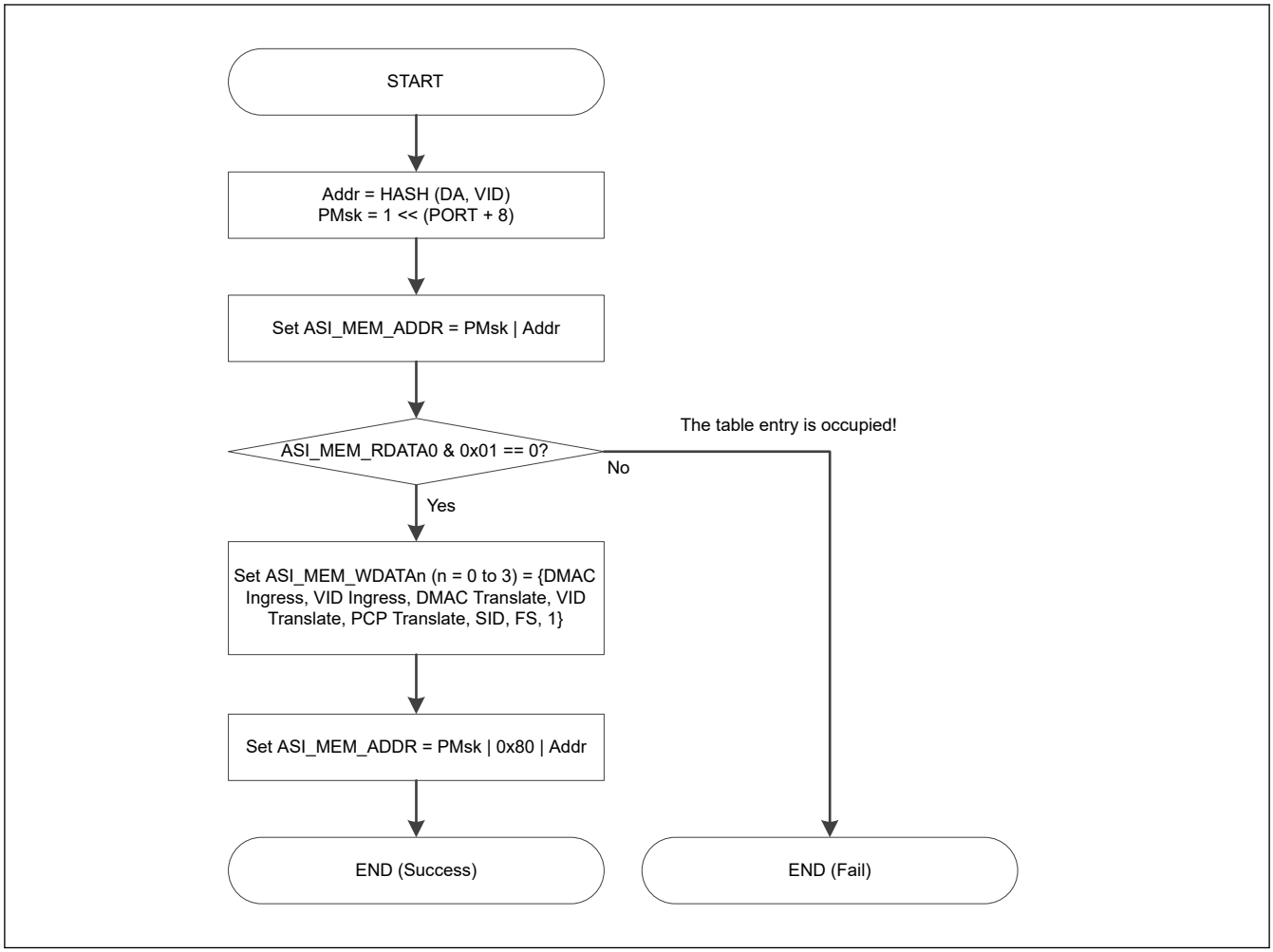


Figure 29.56 Active Stream Identification entry setting

Note: PMsk contains the respective ASI_MEM_ADDR.MEM_REQ bit of the port's memory set.

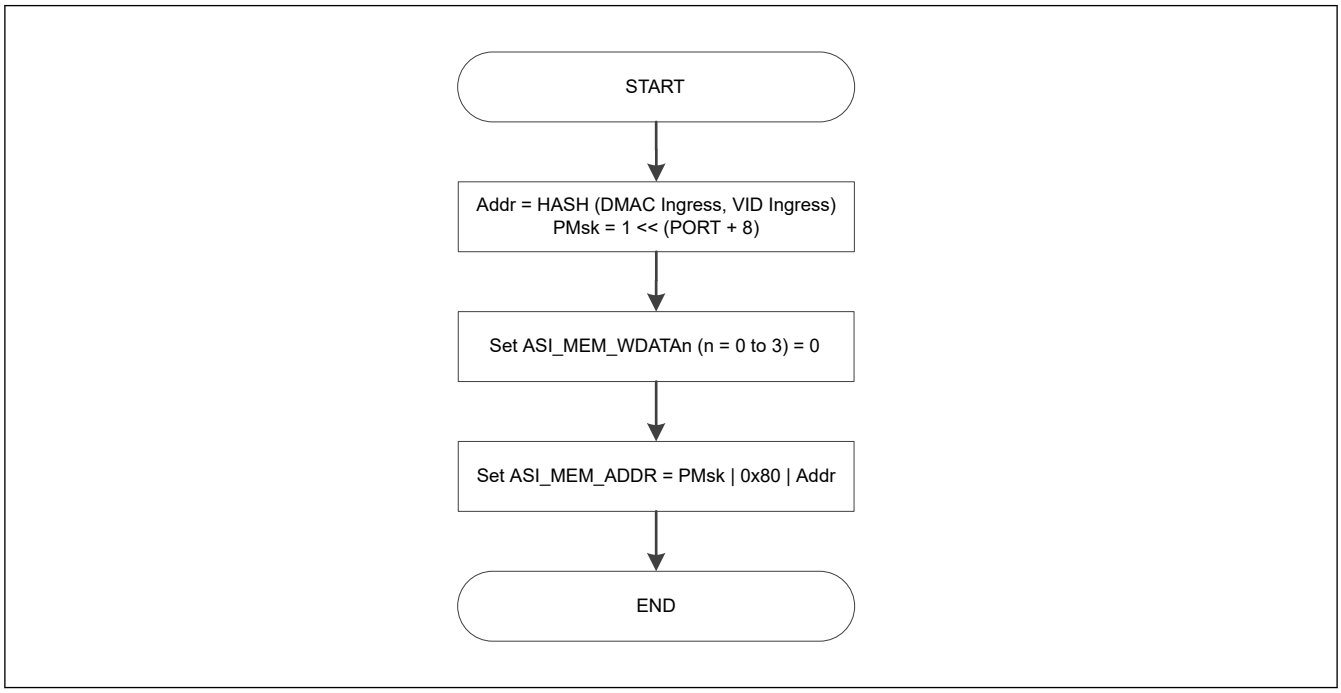


Figure 29.57 Active Stream Identification entry deletion

29.4.19.5 VLAN Ingress Verification

VLAN ingress verification checks presence of a VLAN tag and its VID of an incoming frame.

All frames violated the VLAN ingress verification check are dropped. Following is a flow of VLAN verification.

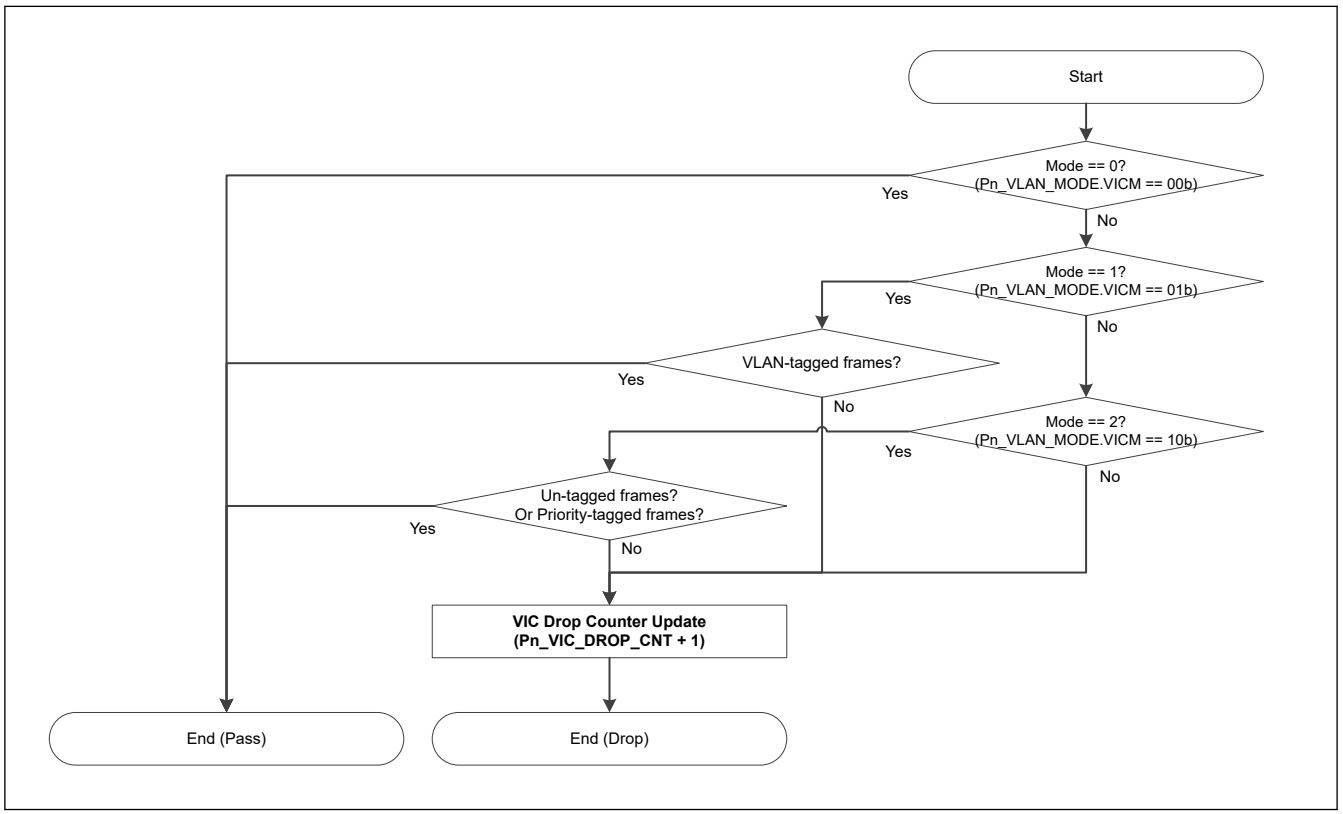


Figure 29.58 VLAN Verification flow

29.4.19.6 Priority Regeneration

The priority regeneration function reassigns PCP values of incoming, VLAN tagged frames. A 1:1 lookup table assigns a new PCP value for each incoming PCP value. The VID and DEI flag are not modified.

Priority regeneration is always executed for VLAN tagged ingress frames, which PCP values have not been changed by the VLAN processing (see VITM setting). However, by default the PCP replacement values correspond to the received PCP, thus the frame is effectively not changed.

The replacement table assigns a new PCP value [0..7] for each permitted PCP ingress value. The same PCP output value can be assigned for multiple PCP ingress values.

For configuration of PCP reassignment, write the respective PCP replacement values into Pn_PCP_REMAP.

Note: Priority regeneration will be skipped, if the PCP value has been assigned by the ingress VLAN tagging flow. See VITM setting in Pn_VLAN_MODE register:

- VITM = 00b or 01b: Priority regeneration is applied.
- VITM = 10b or 11b: Priority regeneration is skipped, PCP value set by Pn_VLAN_TAG is maintained.

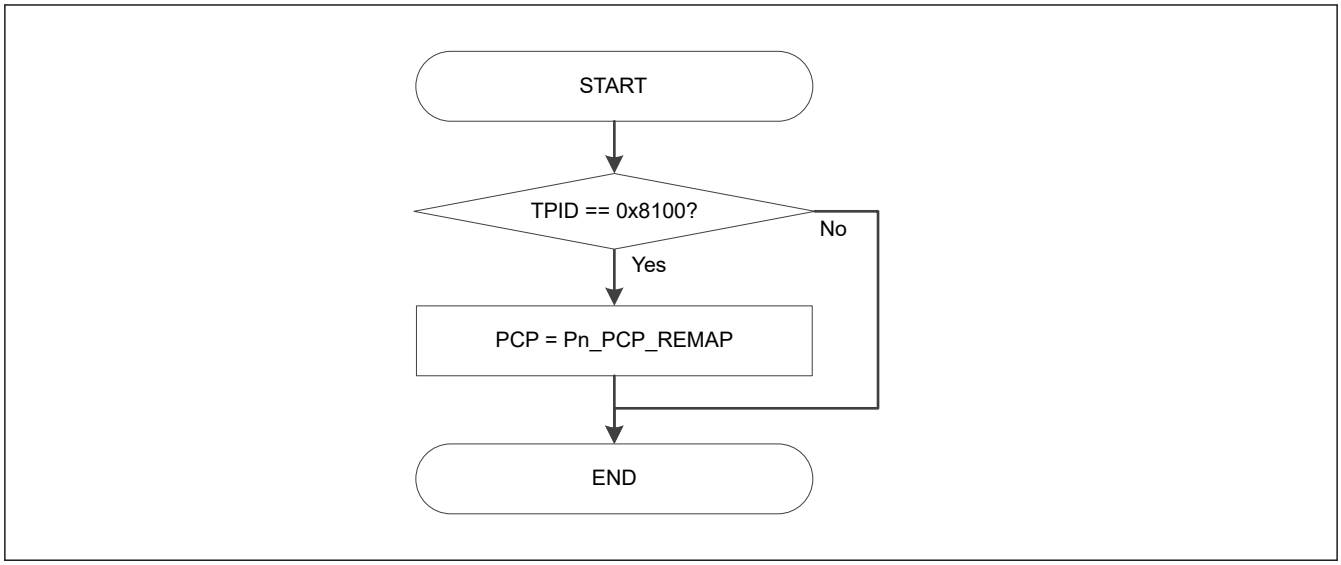


Figure 29.59 Priority Regeneration flow

29.4.19.7 Filtering Function

The filtering function matches the newly received frame against a table of 8 filter entries. The table entries are processed sequentially by comparing the new frame against the filter parameters. Once a match occurs, the comparison terminates and the further frame processing (gating check, SDUmax verification, Flow Metering) is determined by the parameters assigned to the respective filter entry <S>. A match could also be forwarded from the Active Stream Identification table, parameter SID.

If no match is found for a new frame, it means no filter entry matches the frame, the frame is forwarded without any further constraints.

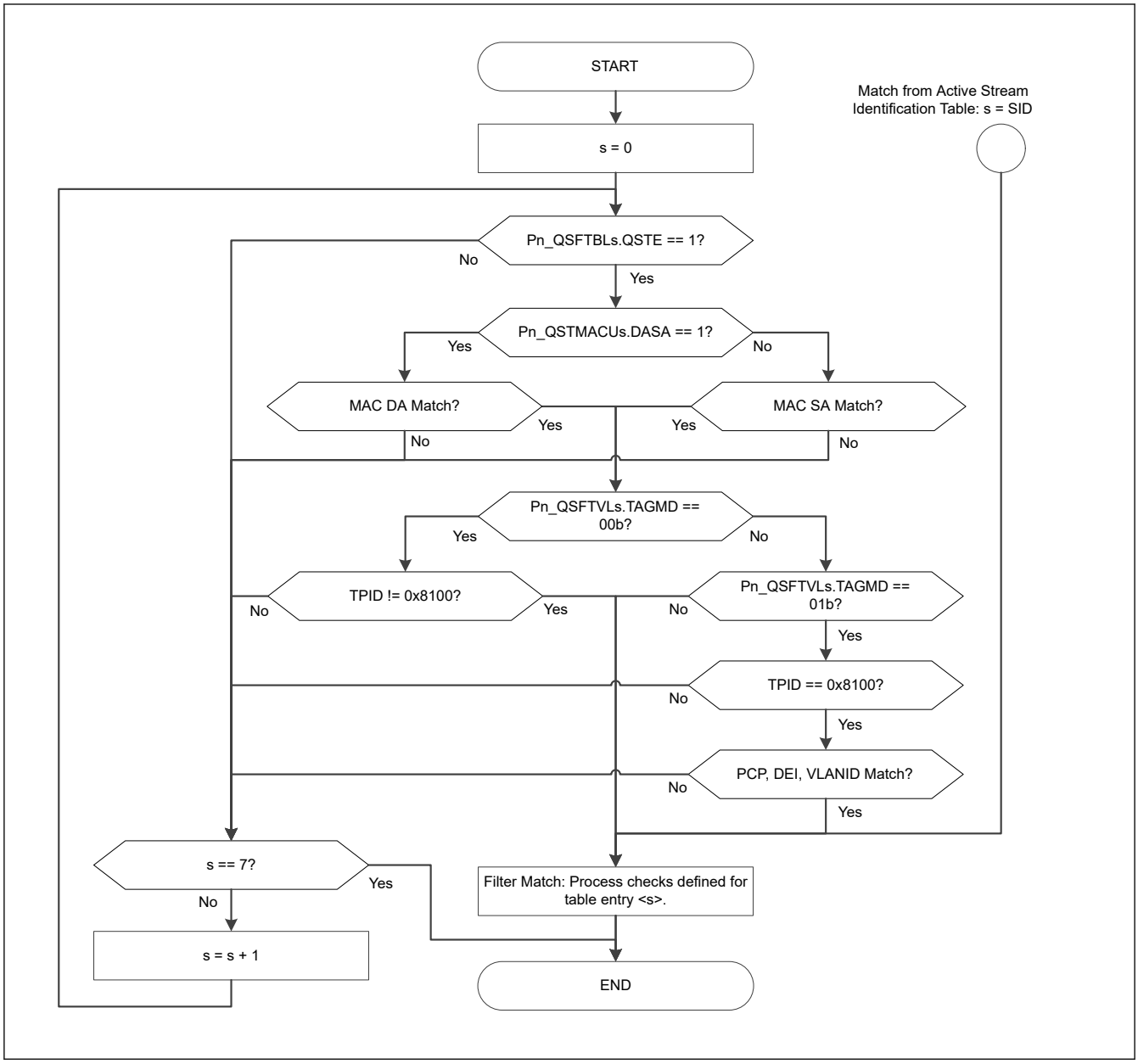


Figure 29.60 Qci filtering flow

Qci stream filter table setting flow is shown in Figure 29.61.

Be sure to set the corresponding Pn_QSTMACUs, Pn_QSTMACDs, Pn_QSTMAMUs, Pn_QSTMAMDs, Pn_QSFTVLs, and Pn_QSFTVLMs first and set Pn_QSFTBLs continuously.

If set in any other sequence, operation is not guaranteed.

Writing of Pn_QSTMACUs, Pn_QSTMACDs, Pn_QSTMAMUs, Pn_QSTMAMDs, Pn_QSFTVLs, Pn_QSFTVLMs, and Pn_QSFTBLs is reflected when is written.

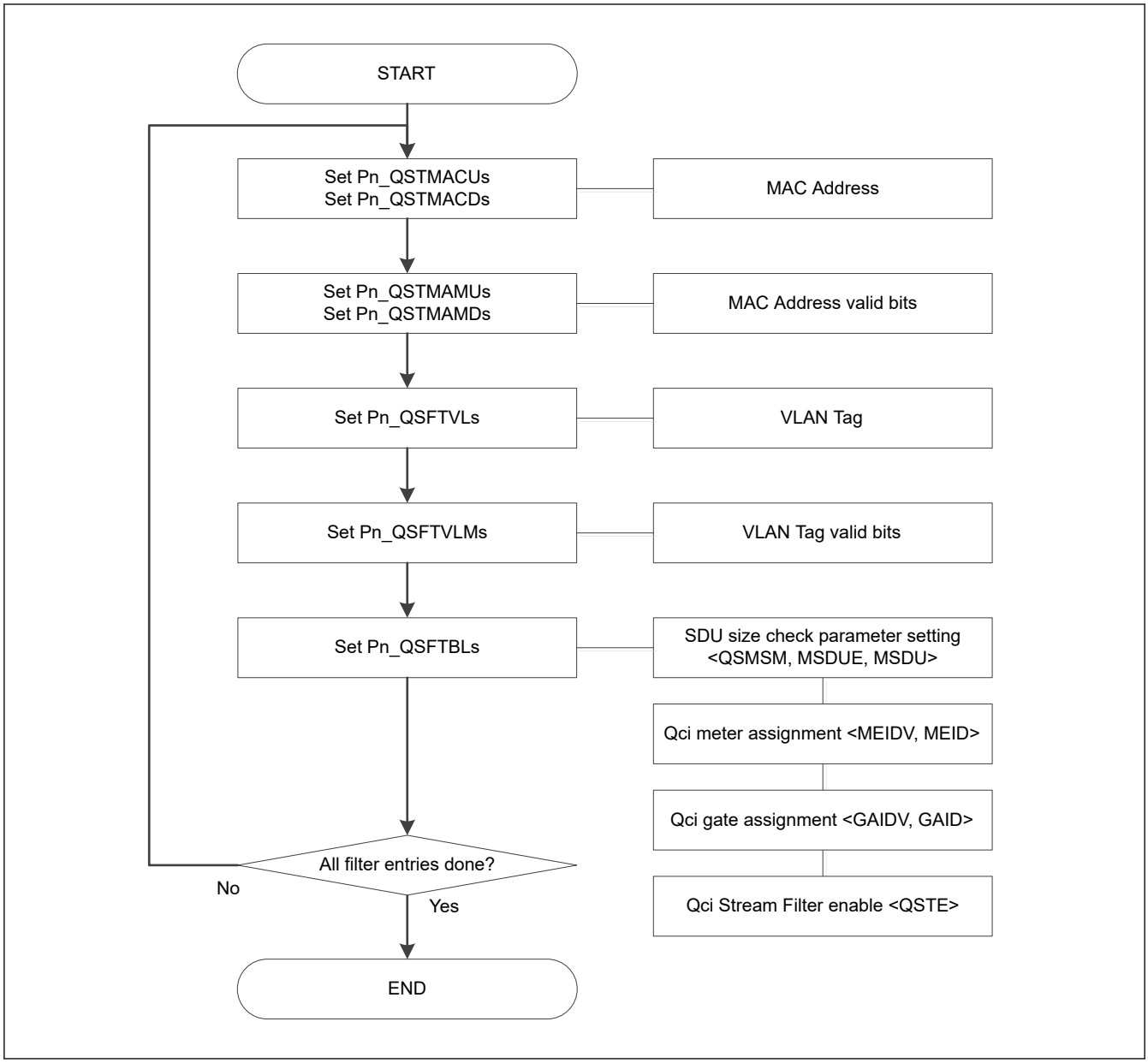


Figure 29.61 Qci stream filter table setting flow

Note: When reconfiguring a filter entry, be sure to set Pn_QSFTBLs.QSTE = 0 before starting to change any other setting of the respective filter entry.

29.4.19.8 Gating Check

A frame is applied for gating check, if the Pn_QSFTBLs.GAIDV control bit is set. In this case, Pn_QSFTBLs.GAID[2:0] determines, which of the tdma_gpio[7:0] signals should be checked. If the signal is low, the frame is dropped and the interrupt event is set.

Depending on the setting of Pn_QGMOD.QGMOD bit, the consecutive frames are not forwarded as long as gating event interrupt is pending.

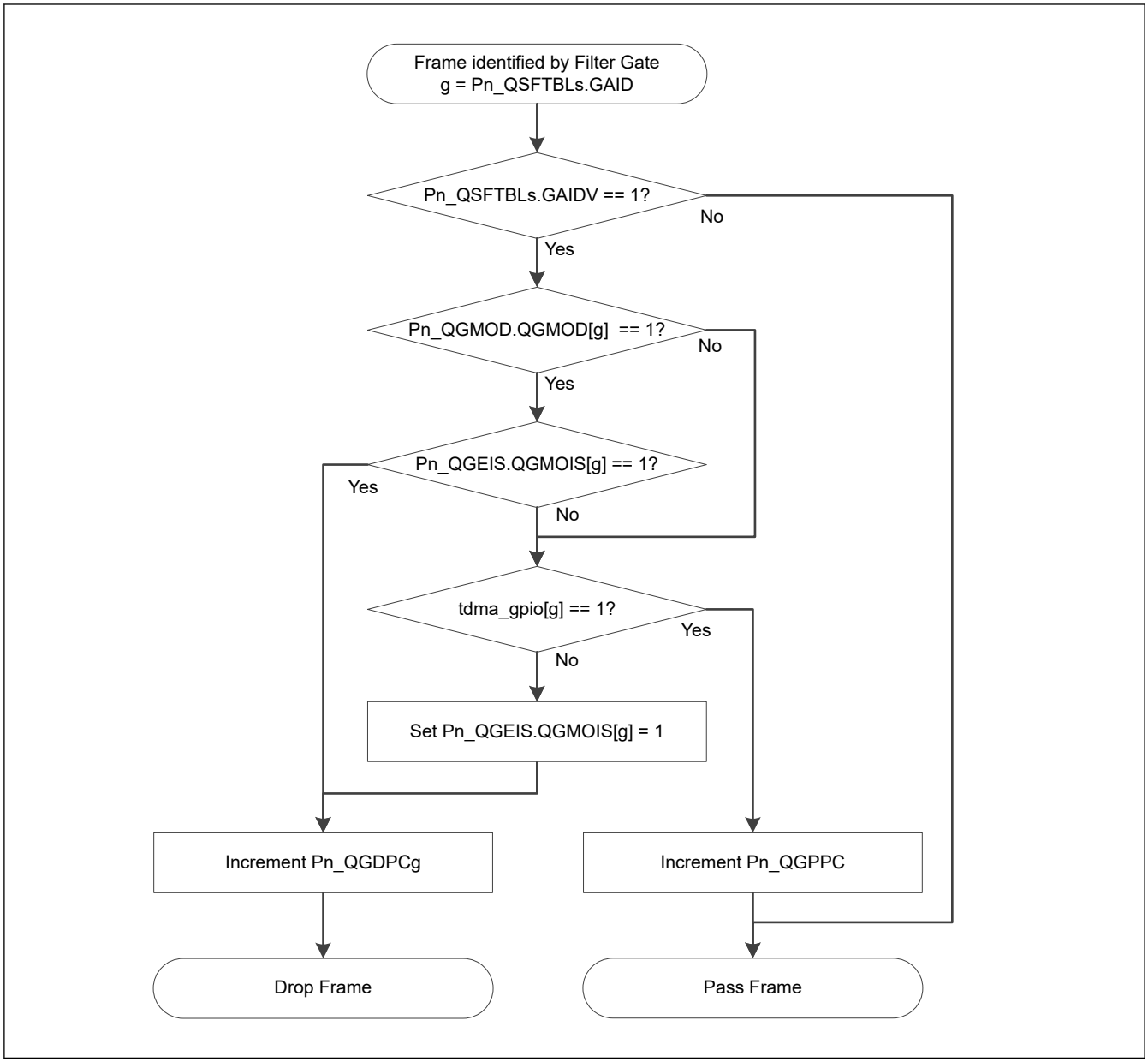


Figure 29.62 Gating check process flow

Set gating check mode by register Pn_QGMOD.

Set gating signals tdma_gpio[7:0] by the TDMA scheduler. See section 29.4.15. TDMA Operation for setting up the TDMA scheduler.

29.4.19.9 SDUmax Verification

After receiving the full frame, the maximum frame length is checked if the Pn_QSFTBLs.MSDUE bit is set. In this case, all frames longer than the value Pn_QSFTBLs.MSDU trigger an error event. The violating frame is dropped and the interrupt event is set.

Depending on the setting of Pn_QSFTBLs.QSMSM, the consecutive frames are not forwarded if SDU size event interrupt is pending.

Note: For checking the SDUmax length, the full frame has to be received. It means SDUmax check can be performed at the end of the frame transmission while most other checks can be applied at the beginning of a frame.

Note: Oversized cut-through frames cannot be stopped. But if the Pn_QSFTBLs.QSMSM bit is set, all consecutive frames mapped to the same filter will be dropped.

SDUmax verification parameters are set along with filter parameters. Refer to Pn_QSFTBLs register function and Figure 29.61 for details.

29.4.19.10 Flow Metering

The Flow meters use the algorithms described in MEF 10.3. Each channel uses the same metering submodule. The meters support two main functions:

1. Replenish tokens into the buckets
   Tokens are replenished into the buckets at a constant rate. The buckets can overflow into other buckets according to the algorithms defined in MEF 10.3.
2. Making acceptance check for frames
   A bucket can be checked for holding sufficient tokens corresponding to the length of a frame to be assessed. If the bucket contains enough tokens, an acceptance judgement is returned, and the corresponding number of tokens is removed from the bucket (frame is green). If the number of token is insufficient, an reject judgement is returned and the number of token in the bucket remains unchanged (frame is red). See Figure 29.65 for details.

The metering structure consists of 8 buckets, which can be cascaded to support the envelope mode.

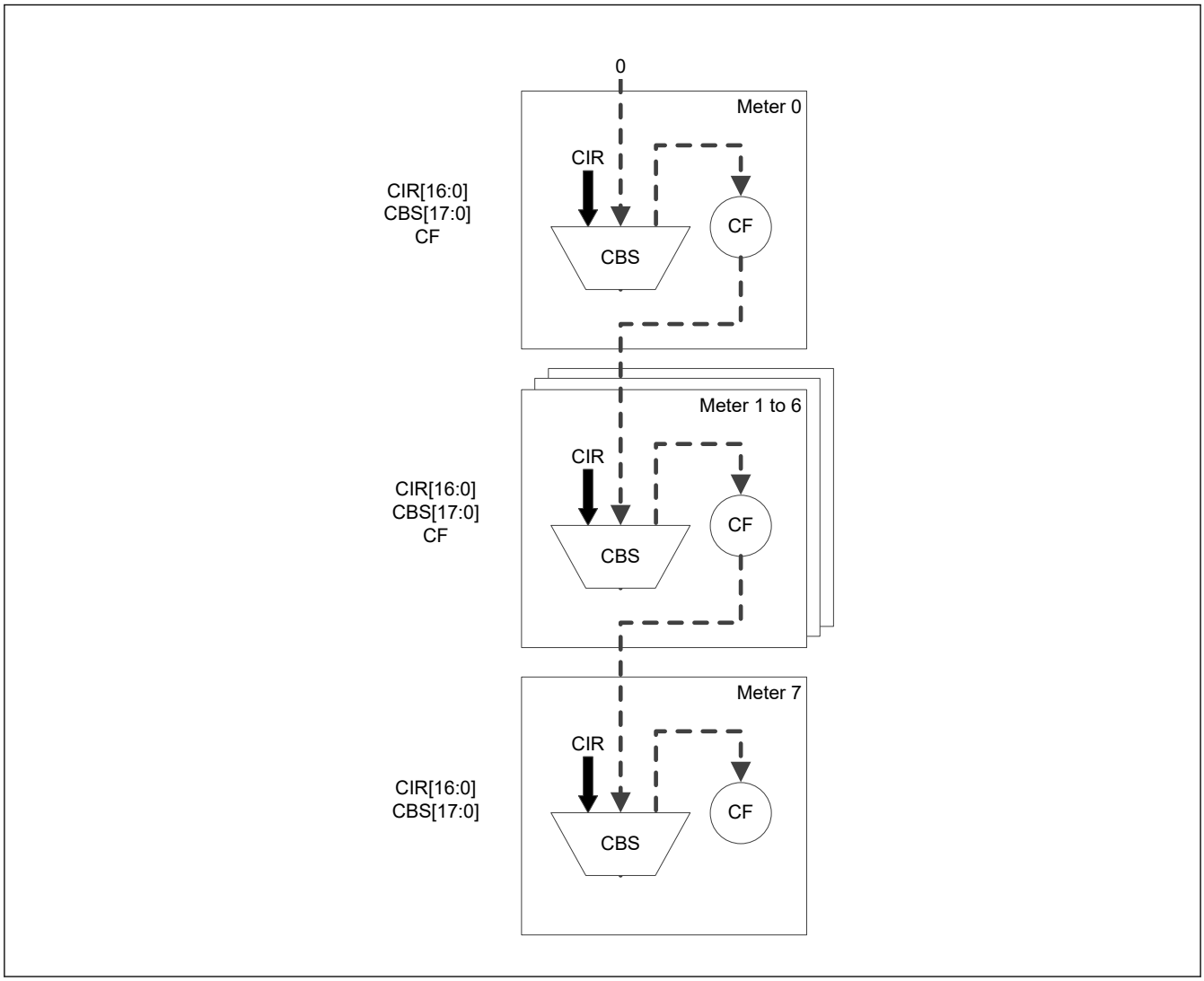


Figure 29.63 Configuration of flow meters

This function provides flow meters according to MEF10.3. Each flow meter provides a green token bucket, that can be configured to provide an envelope mode. Coupling is controlled by the Coupling Flags (CF).

Note: For checking bandwidth compliance by the assigned meter, the full frame has to be received. It means the meter check can be performed at the end of the frame transmission while most other checks can be applied at the beginning of a frame.

Note: Cut-through frames exceeding the bandwidth limit cannot be stopped. But if the Pn_QMDESCm.MM bit is set, all consecutive frames mapped to the same filter will be dropped until the software clears the error event.

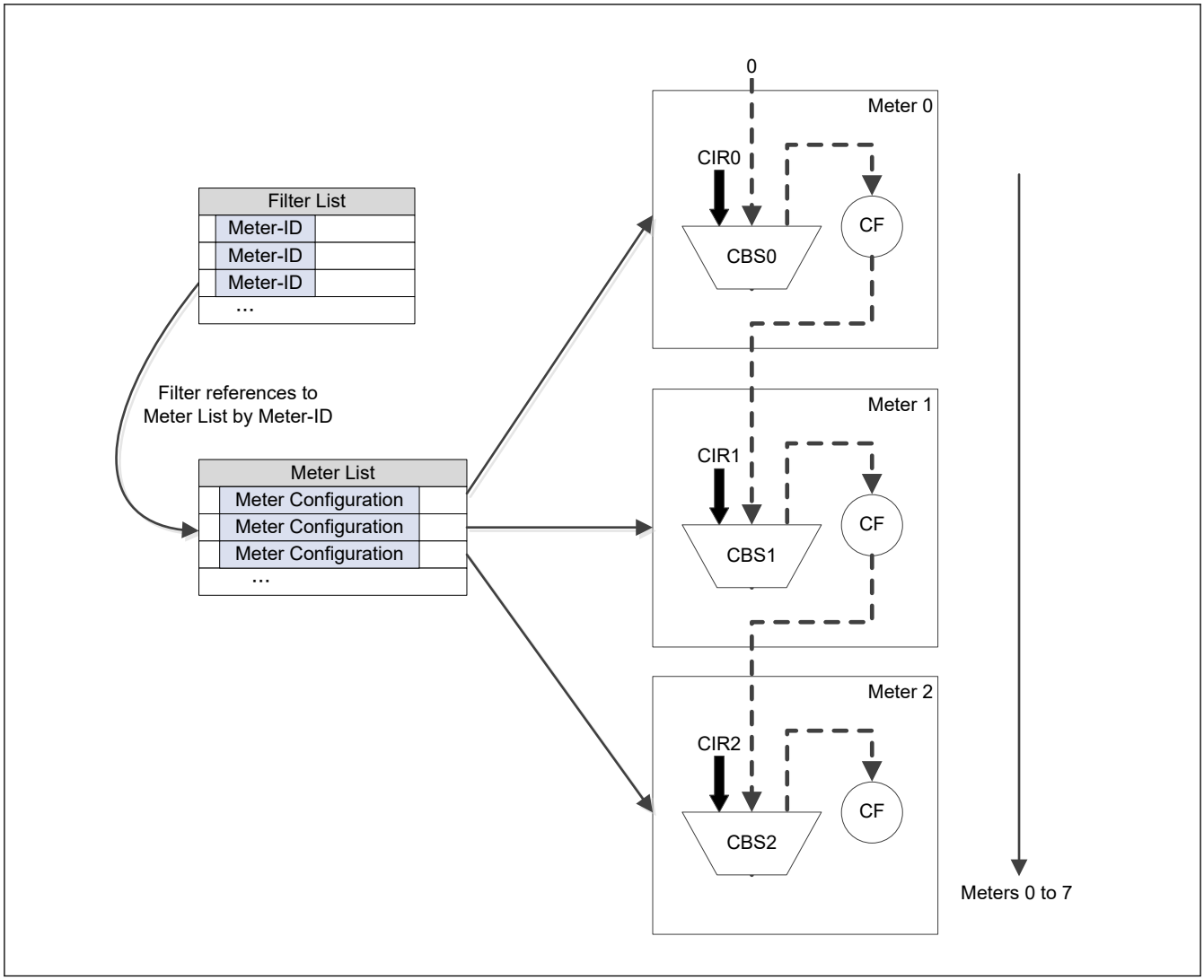


Figure 29.64 Flow meter architecture in EFP

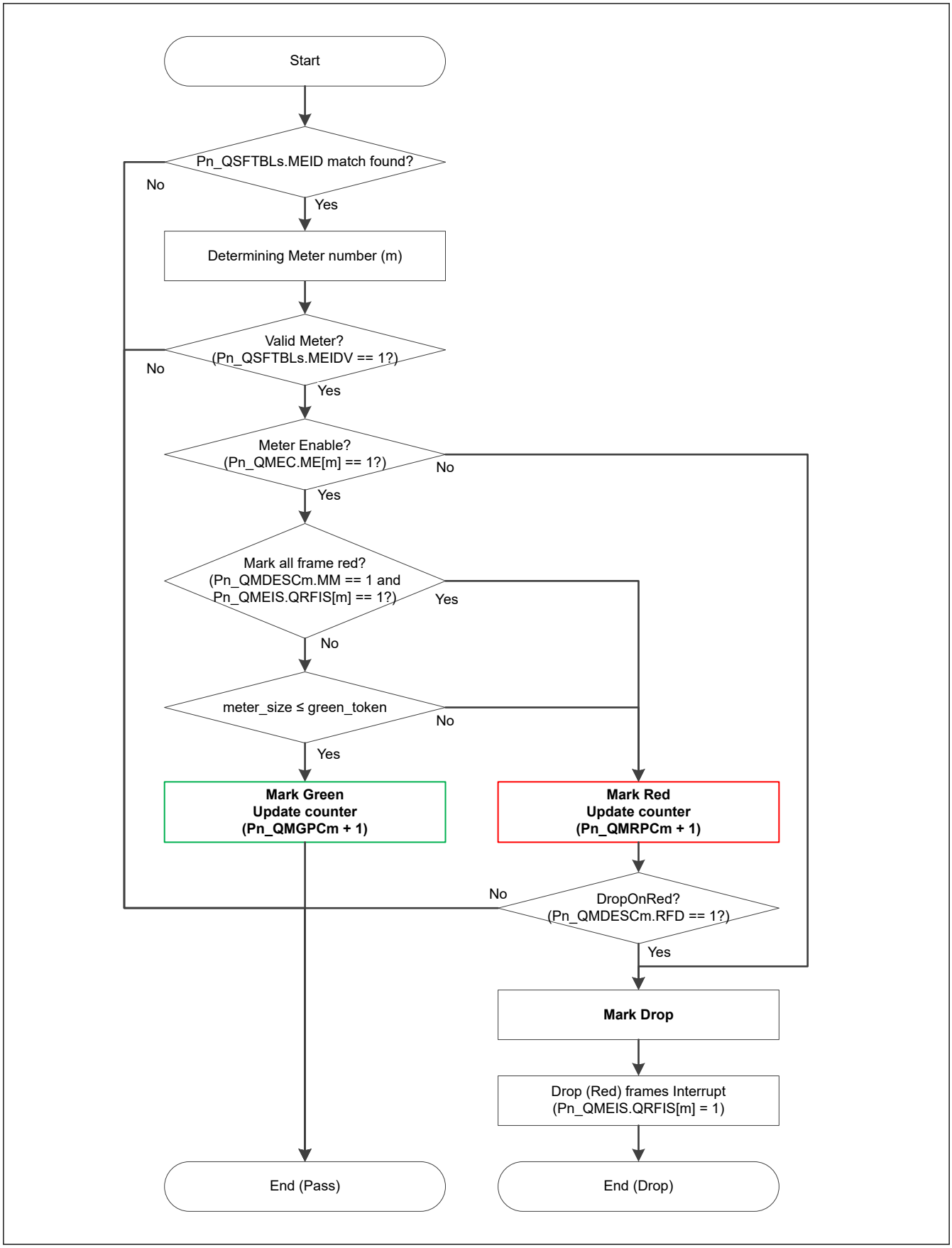


Figure 29.65 Flow meter flow in EFP

Flow Meter setting flow is shown in the figure below.

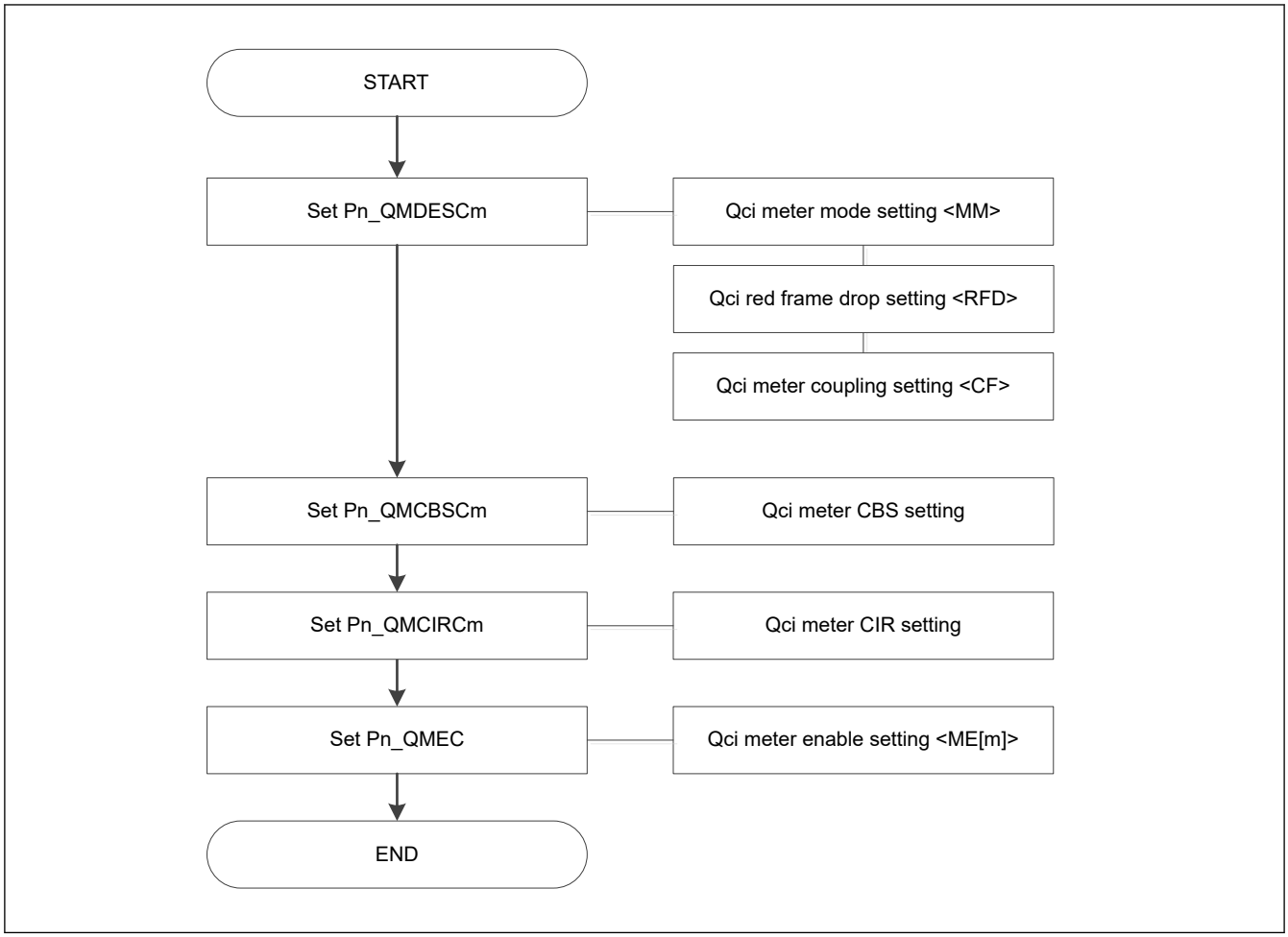


Figure 29.66 Flow Meter setting flow

- Bucket parameters
  - CIR
    CIR bits in Pn_QMCIRCm register is the information rate for replenishing the buckets. The maximum value is determined by the maximum line rate of the port. The minimum value is determined by the application requirement for minimum information rate.
    Max: 1 Gbps = 125 MBps = CIRmax
    Since the replenisher is running at 200 MHz → CIRmax = 0.625; only fractional part needed
  - CBS
    CBS bits in Pn_QMCBSCm register is the maximum token count in the buckets. The minimum value is determined by the maximum Ethernet frame size to be received. The maximum value is determined by the maximum burst length tolerated by the application.
    Min: Maximum frame size of an Ethernet Frame (2 kB).

29.4.19.11 Interrupt Source

Following table shows interrupt sources.

When any of the following interrupts occur, it becomes a global interrupt (INT_CONFIG.EFP_INT).

Table 29.45 Interrupt source for Extended Frame Parser (1 of 2)

Description	Individual Config register	Individual Status/Ack register
Qci stream filter error interrupt (max SDU over drop) (n = 0 to 2)	Pn_QSEIE/Pn_QSEID	Pn_QSEIS
Qci stream gate error interrupt (gating error drop) (n = 0 to 2)	Pn_QGEIE/Pn_QGEID	Pn_QGEIS
Qci meter error interrupt (flow meter drop) (n = 0 to 2)	Pn_QMEIE/Pn_QMEID	Pn_QMEIS

Table 29.45 Interrupt source for Extended Frame Parser (2 of 2)

Description	Individual Config register	Individual Status/Ack register
Frame parser error interrupt (runtime error) (n = 0 to 2)	Pn_ERROR_MASK	Pn_ERROR_STATUS

29.4.19.12 Restrictions

- Restrictions of metering function:
  - Metering function is not available for cut-through frames.
  - It introduces additional forwarding latency into the ingress path.
  - IEEE802.1Q-2018, 8.6.5 Flow classification and metering
    The DEI flag of forwarded frames cannot be modified according to the resulting color of the frame. The final color is determined after frame length is known, i.e. after reception of the full frame. Support of this function would need frame modification at egress process.
  - Supported color mode is color-blind only. EIR and EBS are not supported.
- Additional VLAN tag insert by VITM = 3 is not supported for 1 Gbps operation.
- For Bandwidth metering
  IEC61158-6-10 specifies support of envelope mode (bandwidth sharing) and has no requirement to support of color-awareness. These are the differences to IEEE802.1Q-2018, which is referring to MEF 10.3 without support of envelope mode.
- Forwarding frame dropped
  If the drop is given after the forwarding information has been passed to the memory controller, it will corrupt the frame's egress CRC to ensure that the receiving port will discard the frame. This has the side-effect of causing the port statistics of the receiving device to indicate CRC errors, which can lead to falsely interpret them as errors in the link.

29.4.20 PTP Pulse Generator

The 4 Pulse generators are implemented in ETHSW. These 4 Pulse generators can generate the pulses which have any of period and any of pulse width by utilizing Adjustable timers of the EtherSwitch (see section 29.4.4.7. Adjustable Timer). Selection of the Adjustable timer for these 4 Pulse generators is described in the register PTPMCTRL (see section 27.3.3. PTPMCTRL : PTP Mode Control Register).

This Pulse generator can be configured as following Table and following Figure by the register (see section 29.3.16. PTP Timer Pulse Control Registers).

Table 29.46 Parameter list of pulse signal generation (1 of 2)

Parameter	Registers	Description
Pulse generation enable	SWTMENn	Enables or disables to output pulse signal
Pulse output start time	SWTMSTSECn, SWTMSTNSn	Set the start time of the output signal in seconds and nanoseconds. After set the start time and set 1 to the SWTMEN register in order to enable to output the pulse signal, the pulse output starts when current time is over the start time set. If pulse output is enabled after the set start time, no pulse is output.
Pulse period	SWTMPSECn, SWTMPNSn	Set the pulse period of the output signal in seconds and nanoseconds. The value set in the SWTMPNS register must be larger than 16 × ATIME_INC[6:0] value (128 ns). The value must be set before pulse output is enabled.
Pulse width	SWTMWTHn	Set the pulse high level width in the cycle number of ts_clk (8 ns). Output pulse width is the set value × 8 (ns). If the pulse width set in the SWTMWTH register is greater than pulse period, output is fixed to high level. If the pulse width is set to 0, no pulse is generated and output is fixed to low level. The value must be set before pulse output is enabled.

Table 29.46 Parameter list of pulse signal generation (2 of 2)

Parameter	Registers	Description
Max nanosecond counter	SWTMMAXPn	This register sets the boundary value in nanoseconds to carry from the nanosecond field to the second field. When the nanosecond field reaches the set value of this register, the second field is carried up. This register is normally used with the initial value of 0x3B9ACA00 (= 10^9). In this case, the nanosecond field counts from 0 to 0x3B9AC9FF (= 10^9 - 1) and then carries to the second field. If 0 is set in this register, the boundary value for carrying to the second field is interpreted as 0x100000000 (= 2^{32}). In this case, the nanosecond field is fully counted from 0 to 0xFFFFFFFF (= 2^{32} - 1) and then carries to the second field. This register must always be set to the same value set in the ATIME_EVT_PERIOD register.
Pulse rising time latch	SWTMLATSECn, SWTMLATNSn	The pulse output rising time is latched into the registers. The value is updated whenever pulse output rises.

Note: n = 0 to 3

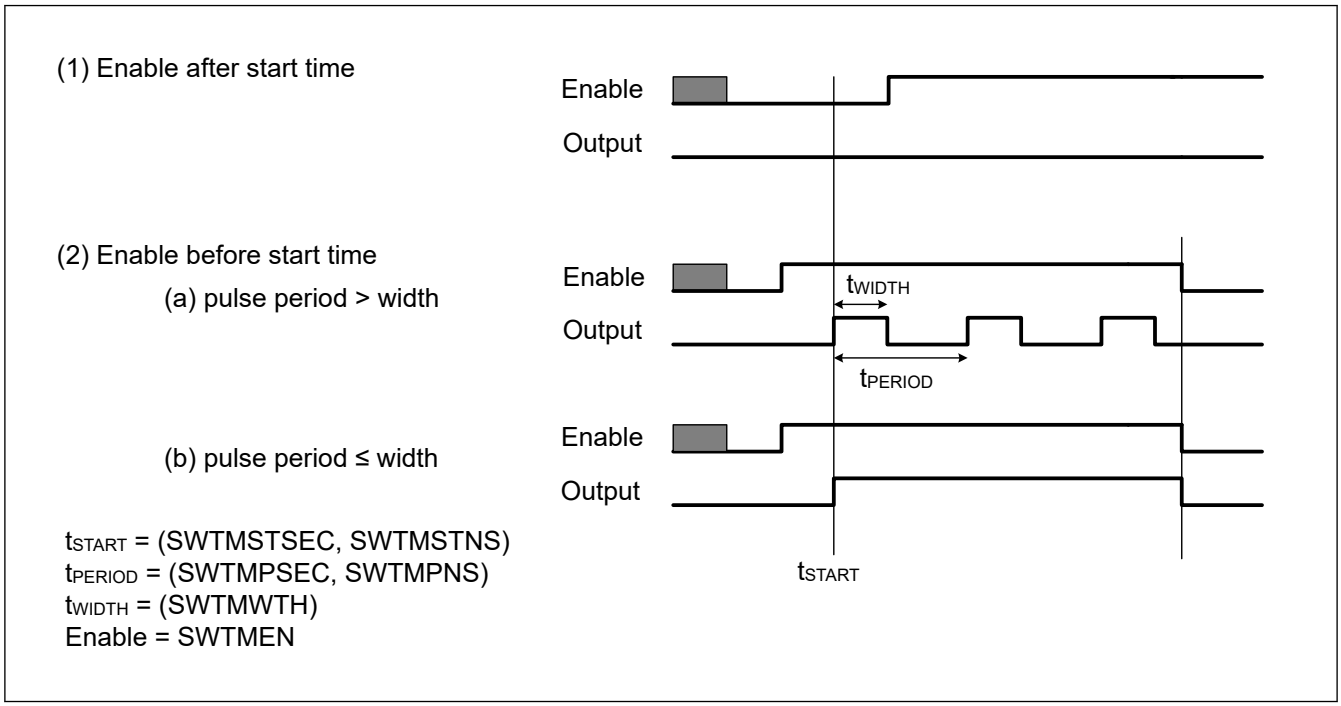


Figure 29.67 Timing chart of pulse generator function

29.4.21 Initializing ETHSW

The initializing sequence in this section is an example used for preparation of system environments for using ETHSW under configuration below.

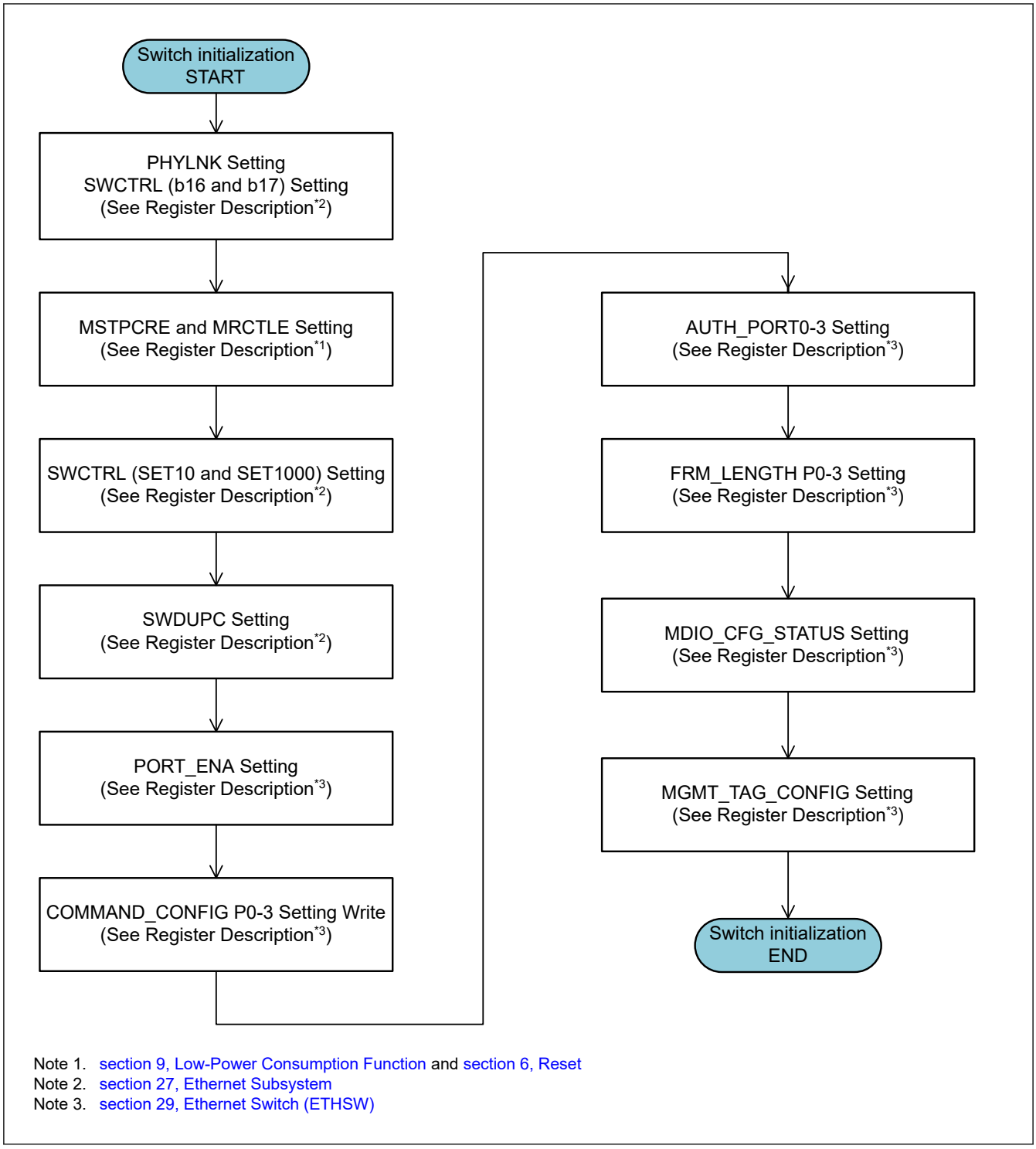


Figure 29.68 Initializing of ETHSW Flowchart

29.4.22 Timing Considerations

29.4.22.1 Cut-Through Frame Forwarding Delay

The forwarding delay is measured from the time of the first byte of the frame received at the phy receive interface in the switch (MII receive interface) to the same byte being transmitted on the phy transmission interface in the switch (MII transmit interface) for a single frame with no other activity in other ports. The delay shown is the expected minimum delay and it can be higher due to congestion in the forwarding tables and frame buffer.

The cut-through forwarding delay does not apply when the destination port is the management port as this port only receive frames in store-and-forward.

Table 29.47 Cut-through frame forwarding delay

Source port speed	Destination port speed	Delay (minimum)
10 Mbps	10 Mbps	44.4 µs
100 Mbps	10 Mbps	12.9 µs
100 Mbps	100 Mbps	3.96 µs
1000 Mbps	10 Mbps	10.8 µs
1000 Mbps	100 Mbps	1.50 µs
1000 Mbps	1000 Mbps	0.59 µs

Note: The measurements were made with the values for CT_DELAY register.

29.4.22.2 Store-and-Forward Frame Forwarding Delay

The forwarding delay is measured from the time of the first byte of the frame received at the phy receive interface in the switch (MII receive interface) to the same byte being transmitted on the phy transmission interface in the switch (MII transmit interface) for a single frame with no other activity in other ports. The delay shown is the expected minimum delay and it can be higher due to congestion in the forwarding tables and frame buffer.

Table 29.48 Store-and-forward frame forwarding delay

Source port speed	Destination port speed	Delay (minimum)
10 Mbps	10 Mbps	63.6 µs
10 Mbps	100 Mbps	54.5 µs
10 Mbps	1000 Mbps	53.6 µs
100 Mbps	10 Mbps	15.3 µs
100 Mbps	100 Mbps	6.60 µs
100 Mbps	1000 Mbps	5.70 µs
1000 Mbps	10 Mbps	10.8 µs
1000 Mbps	100 Mbps	1.81 µs
1000 Mbps	1000 Mbps	0.92 µs

29.4.22.3 Frame Timestamp Interface Delay

The frame timestamp interface delay is a value that needs to be added or subtracted from MAC captured timestamp in both receive and transmit directions. This value represents the delay between the internal frame timestamp point and the switch's MII interface. The timestamp is captured conceptually at the first bit of the frame after the SFD (i.e. the first bit of the destination address of the frame).

The correction is dependent on the timestamp clock frequency and the port rate. The values in the table below are given for a timer's clock frequency of 125 MHz. The port's receive and transmit frequencies depend on the rate: 2.5 MHz for 10 Mbps, 25 MHz for 100 Mbps, and 125 MHz for 1000 Mbps. The calculation column describes how to obtain the correction value for different clock frequencies.

The correction value has an uncertainty that is due to the clock-crossing of from/to the port's receive/transmit clock domain to/from the timer's clock domain, which can be reduced by using a faster timer's clock as well.

Table 29.49 Frame timestamp interface delay (correction offset) (1 of 2)

Correction value	Minimum	Typical	Maximum	Calculation
10 Mbps Receive Timestamp	-1216 ns	-1212 ns	-1208 ns	- (3 × rx_cyc + (1 to 2) × ts_cyc)
100 Mbps Receive Timestamp	-136 ns	-132 ns	-128 ns	- (3 × rx_cyc + (1 to 2) × ts_cyc)
1000 Mbps Receive Timestamp	-32 ns	-28 ns	-24 ns	- (2 × rx_cyc + (1 to 2) × ts_cyc)
Transmit Timestamp (all speeds, HUB disabled)	-16 ns	-12 ns	-8 ns	- ((1 to 2) × ts_cyc)

Table 29.49 Frame timestamp interface delay (correction offset) (2 of 2)

Correction value	Minimum	Typical	Maximum	Calculation
10 Mbps Transmit Timestamp (HUB enabled)	+2384 ns	+2388 ns	+2392 ns	+ (6 × tx_cyc – (1 to 2) × ts_cyc)
100 Mbps Transmit Timestamp (HUB enabled)	+224 ns	+228 ns	+232 ns	+ (6 × tx_cyc – (1 to 2) × ts_cyc)

Note: The correction value is given with a sign indicating that the value needs to be added (+) or subtracted (-) from the timestamp value. The ts_cyc is the cycle time for the timer's clock (fixed 8 ns); The rx_cyc and tx_cyc are the cycle times for the receive and transmit port's clocks respectively.

29.4.22.4 Timer Configuration Delay

The switch's internal timers are programmed via the register interface which has an inherent delay from the time when the register write operation is presented to the CPU host interface until the timer action is executed.

The table below describes the delay from the point of the CPU operation presented to the register interface (input pin hsel transitioning from 0 to1) to the action on the timer being executed. Two actions are relevant:

- Timer load: loading the timer with a new seconds and nanoseconds value via ATIME_SECn and ATIMEn. The delay indicated in the table should be added to the value in {ATIME_SECn, ATIMEn} to accurately program the timer to the desired time.
- Time capture: capturing the timer's current value using ATIME_CTRLn.CAPTURE or ATIME_CTRLn.CAPTURE_ALL. The captured time is read back in {ATIME_SECn, ATIMEn} and to this value the delay below should be subtracted to obtain the time at the point when hsel transitioned from 0 to 1.

Table 29.50 Timer configuration delay

Correction value	Minimum	Typical	Maximum	Calculation
Timer Load (using ATIMEn)	87 ns	93.5 ns	100 ns	3 × PCLKM_cyc + (5 to 6) × sx_cyc + (4 to 5) × ts_cyc
Timer Capture (using ATIME_CTRLn.CAPTURE)	68 ns	74.5 ns	81 ns	3 × PCLKM_cyc + (6 to 7) × sx_cyc + (1 to 2) × ts_cyc

Note: Condition is PCLKM_cyc is 8 ns, sx_cyc (switch clock) is 5 ns (fixed), and ts_cyc (timer clock) is 8 ns (fixed).

29.4.22.5 TDMA Action Delays

The following table lists the delays related to the TDMA actions. When the TDMA is operating there is a delay from the matching of the time in the TCV to the action indicated by the TCV being executed. There is also a minimum action time between events which limits the time between back to back actions.

For programming the TDMA we recommend to use a distance between vectors of at least the maximum value in the "minimum time between vectors" in the table below.

The delay values for state changes and gate/queue events must be subtracted to the time value programmed in the TCV vector to ensure a more precise execution. For instance, if the GPIO pin is to be toggled at 1000 ns after the period starts, the time offset to be programmed in the TCV should be (1000 - 43.5) = 956.5 ns.

Table 29.51 TDMA action delays

TDMA action	Minimum	Typical	Maximum	Calculation
GPIO change RED_PERIOD change OUT_CT_ENA change IN_CT_ENA change HOLD_REQ change	41 ns	43.5 ns	46 ns	2 × ts_cyc + (5 to 6) × sx_cyc
GATE close/open QUEUE trigger	51 ns	53.5 ns	56 ns	2 × ts_cyc + (7 to 8) × sx_cyc
Minimum time between vectors	115 ns	121.5 ns	128 ns	(5 to 6) × ts_cyc + (15 to 16) × sx_cyc

Note: Condition is that sx_cyc (switch clock) is 5 ns (fixed) and ts_cyc (timer clock) is 8 ns (fixed).

29.4.22.6 Pattern Matcher Delayed Frame Trigger Delay

The pattern matcher has a delayed frame trigger function (IMC_TRIGGER_DLY) which allows triggering a frame at a specified time after the trigger event occurs. Due to the logic doing the time comparison and generating the

event, there is a delay from the actual time programmed and the trigger event executing which could need to be accounted for. The table below describes this delay, and such delay should be subtracted to the value programmed in MMCTL_DLY_QTRIGGER_CTRL.

Table 29.52 Pattern matcher action delays

Pattern matcher action	Minimum	Typical	Maximum	Calculation
IMC_TRIGGER_DLY	51 ns	53.5 ns	56 ns	2 × ts_cyc + (7 to 8) × sx_cyc

Note: Condition is sx_cyc (switch clock) is 5 ns (fixed) and ts_cyc (timer clock) is 8 ns (fixed).

29.4.22.7 Transmit Frame Selection to Frame's FB Delay

The following table describes the delay from a queue becoming ready for transmission to the first byte being transmitted by switch at the MII/GMII interface. A queue becoming ready to transmit frames depends on various factors including:

- The queue being enabled by the firmware
- A frame queued-up for transmission (i.e. the output queue buffer going non-empty)
- A queue transitioning from closed to open
- A frame triggering action on a closed queue
- The hold request transitioning from 1 to 0 for a preemptable queue

The values in the table below are the minimum time, and this time can be slightly higher due to arbiters in the memory controller and clock crossing into the transmission port. The values also assume that the port is idle and all IPG has been transmitted for the previous frame when the queue becomes ready for transmission.

Table 29.53 Transmit frame selection to frame's FB delay

Port speed	Delay	Calculation
10 Mbps	10.1 µs	18 × sx_cyc + 25 × tx_cyc
100 Mbps	1.09 µs	18 × sx_cyc + 25 × tx_cyc
1000 Mbps	0.210 µs	18 × sx_cyc + 15 × tx_cyc

Note: The sx_cyc is the cycle time for the switch's clock; The tx_cyc is the cycle time for the transmit port's clock.

29.4.23 Usage notes

29.4.23.1 Restriction

1. If Ethernet ports are used as Hub mode, the ports must be configured to "RMII mode 10/100 Mpbs REF_CLK output" mode by CONV_MODE bits of "RGMII/RMII Converter n Control register (CONVCTRLn) (n = 0 to 2)" (see section 27, Ethernet Subsystem). Because PHYs connected with the Ethernet ports in Hub mode requires common clock supplied as REF_CLK.
2. To configure the port of ETHSW as half-duplex mode, it is necessary to clear PHY_DUPLEX[2:0] bits of "Switch Core Duplex Mode register (SWDUPC)" (see section 27, Ethernet Subsystem) and set HD_ENA bit of "Port n Command Configuration register (COMMAND_CONFIG_Pn)" (See section 29.3.6.2. COMMAND_CONFIG_Pn : Port n Command Configuration Register (n = 0 to 3)). Note that 1 Gbps is not supported at half-duplex mode.
3. For initialization of the Learning Table, wait until CLRTBL bit of LK_CTRL register is cleared.
4. IP Classify table is not initialized automatically, therefore it is necessary to initialize all areas by software. Refer to IP_PRIORITYn register (see section 29.3.1.48. IP_PRIORITYn : IP Priority Register n (n = 0 to 3)) description for an access procedure to IP Classify table.
5. The PHY, when operating in half-duplex, must not return transmit data to its receive interface.
6. The Parallel Redundancy Protocol (PRP) and 802.3br frame preemption should not use at the same time.
7. The following shows how to disable the MAC transmit function:
   (a) Disable the port via the control register PORT_ENA. This prevents frames to be forwarded to the port. The egress frames queued up can be flushed with the queue flush function via the control register MMCTL_QFLUSH, otherwise, the port becomes inactive once the queues are empty.

(b) Block the queues via the control register MMCTL_QGATE. The TDMA must be disabled for the port as well using TDMA_ENA_CTRL.PORT_ENA to prevent the TDMA from opening the queues again. The egress queues can be flushed via the control register MMCTL_QFLUSH if needed.

8. When using adjustment function other than immediate offset adjustment, the nanosecond time of the timer can glitch dependent on the difference between the normal increment value (ATIME_INCn[6:0]) and the adjustment value (ATIME_INCn[22:16] or ATIME_INCn[14:8]). ATIMEn and ATIME_SECn may output invalid value, for example, a transition from 24.4294967276 to 25.4294967292 and 24.999999980 to 25.1000000004. To mitigate these errors, consider two workarounds as below:

   (a) The firmware must monitor the timer with a frequency faster than twice every second using a time source that is different from the timer itself. This can detect an error in the seconds increment when a frame's timestamp needs to be processed. Subtract 1 from the seconds if occurred.

   (b) When using a progressive (offset) correction or continuous (drift) adjustment, program the values in ATIME_INCn[22:16] and ATIME_INCn[14:8] as similar as ATIME_INCn[6:0] (within 5 ns).

9. Setting ATIME_CTRLn[12] to 1 causes not only time capture from all the timers but also immediate timer correction from ATIME_INCn[14:8]. To avoid altering the timer value, consider below workaround:

   (a) The value in ATIME_INCn[14:8] must be programmed the same as ATIME_INCn[6:0] if continuous (drift) adjustments are not used. This ensures that a timer capture does not alter the timer value.

   (b) If continuous (drift) adjustments are needed, then prior to a capture ATIME_INCn[14:8] must be programmed the same as ATIME_INCn[6:0]. After the capture ATIME_INCn[14:8] can be returned to its original value. This can result in a small error in the timer if the period for the adjustments is short (ATIME_CORRn).

   (c) When a one-time small immediate adjustment is needed the value in ATIME_INCn[14:8] must be returned to be the same ATIME_INCn[6:0] after the adjustment is executed.

29.4.23.2 Known Issue and Workaround

There are known issues in ETHSW. Please consider workaround when using related function.

Table 29.54 Known issue and workaround

No.	Issues	Workaround
1	TX_RESPONSE_OK_CNT and TX_VERIFY_OK_CNT registers may be incremented count incorrectly.	No workaround. Do not use TX_RESPONSE_OK_CNT and TX_VERIFY_OK_CNT registers. Count the frame by software if needed.
2	When FRM_LENGTH register in port 3 (Management port) is set to very small value, the frame may be discarded since the frame length may be incorrectly detected as oversize.	Set FRM_LENGTH register in port 3 to more than or equal to the initial value. When you want to limit the frame length of the received frame, use FRM_LENGTH registers in port 0 to port2.
3	When setting 0 to the INT_VALUE bit in TDMA_CTR1 register, the interrupt is not generated.	When using the interrupt by TDMA_CTR1, set the value other than 0 to the INT_VALUE bit.
4	When Transmit Timestamp Capture Overflow occurs, Transmit Timestamp Capture Overflow interrupt may not be generated.	Do not use Transmit Timestamp Capture Overflow interrupt. Instead, use Transmit Timestamp Capture interrupt and obtain the captured timestamp before overflow occurs.
5	The IRQ_STAT bit in TSM_IRQ_STAT_ACK register does not show the TSM interrupt pending status correctly.	When checking TSM interrupt pending status, do not use the IRQ_STAT bit in TSM_IRQ_STAT_ACK register but the TSM_INT bit in the INT_STAT_ACK register.
6	When reading MDIO related registers (MDIO_CFG_STATUS, MDIO_COMMAND and MDIO_DATA), the read value from MDIO related registers may be incorrect.	When accessing statistic counter register between 0x8012_0300 to 0x8012_033C, perform dummy read to the 0x8012_0310 after the access.

29.4.23.3 References

1. RFC 2309; Recommendations on Queue Management and Congestion Avoidance in the Internet; IETF, 1998
2. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems; IEEE Std 1588-2008.
3. IEEE 802.1Qav; Virtual Bridged Local Area Networks, Amendment 12: Forwarding and Queuing Enhancements for Time Sensitive Streams; 2009.
4. IEEE 802.1X; Port Based Network Access Control; 2004.

5. The CIP Networks Library, Volume 2, Ethernet/IP Adaptation of CIP. Edition 1.9, April 2009.
6. IEC 62439-3; Industrial communication networks — High availability automation networks — Part 3: Parallel Redundancy Protocol (PRP) and High availability Seamless Redundancy (HSR); Edition 2.0; 2012.

30. EtherCAT Slave Controller (ESC)

30.1 Overview

The EtherCAT slave controller (ESC) uses an EtherCAT Slave Controller IP Core made by Beckhoff Automation GmbH, Germany. The ESC handles EtherCAT communications as an interface between the EtherCAT field bus and slave applications.

Table 30.1 Specification of the EtherCAT slave controller

Item		Description
Number of ports		3
FMMU		8
SyncManager		8
Process data RAM [KB]		8
Distributed clocks		64 bits
EBus		No
Process data interfaces (PDI)	Digital I/O	No
	SPI slave	No
	Host MPU interface	On-chip bus

Figure 30.1 shows a block diagram of the EtherCAT slave controller.

This section describes the function of the registers and modules in white. The registers and modules in gray are described in section 27, Ethernet Subsystem, section 28, Ethernet MAC (GMAC), and section 29, Ethernet Switch (ETHSW). Refer to the related sections regarding the registers and modules which are needed to operate the EtherCAT slave controller.

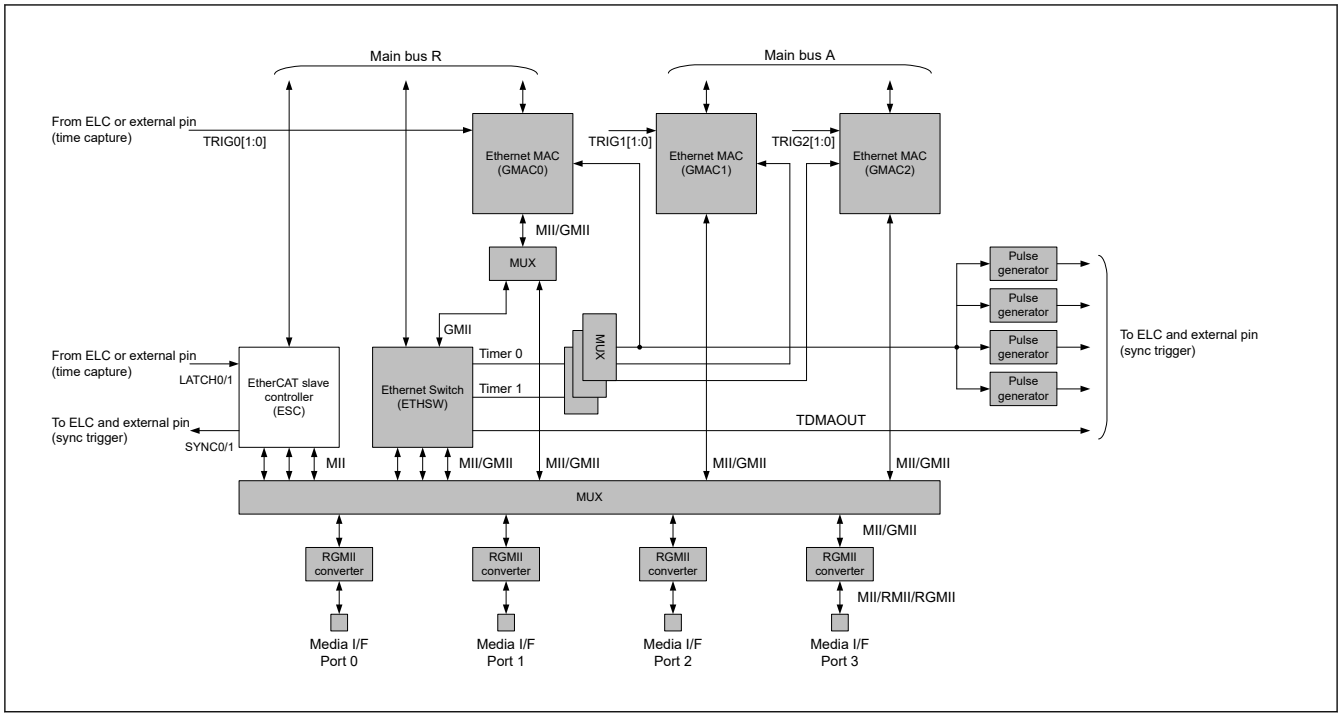


Figure 30.1 Block diagram of the EtherCAT slave controller

Table 30.2 lists the input/output pins of the EtherCAT controller.

Table 30.2 Input/Output pins of the EtherCAT slave controller (excluding PHY MII pins) (1 of 2)

Pin name	I/O	Function	Active
ESC_LEDRUN	O	EtherCAT RUN LED signal output pin	High

Table 30.2 Input/Output pins of the EtherCAT slave controller (excluding PHY MII pins) (2 of 2)

Pin name	I/O	Function	Active
ESC_IRQ	O	EtherCAT IRQ signal output pin	High
ESC_LEDSTER	O	EtherCAT Dual-color State LED signal output pin	High
ESC_LEDERR	O	EtherCAT Error LED signal output pin	High
ESC_LINKACT0	O	EtherCAT link/Activity LED signal output pin (port 0)	High
ESC_LINKACT1	O	EtherCAT link/Activity LED signal output pin (port 1)	High
ESC_LINKACT2	O	EtherCAT link/Activity LED signal output pin (port 2)	High
ESC_SYNC0	O	EtherCAT SYNC0 signal output pin	High
ESC_SYNC1	O	EtherCAT SYNC1 signal output pin	High
ESC_LATCH0	I	EtherCAT LATCH0 signal input pin	Rise/Fall (Both Edge)
ESC_LATCH1	I	EtherCAT LATCH1 signal input pin	Rise/Fall (Both Edge)
ESC_RESETOUT#	O	EtherCAT reset output pin	Low
ESC_I2CCLK	O	EtherCAT EEPROM I2C clock output pin	—
ESC_I2CDATA	I/O	EtherCAT EEPROM I2C data signal I/O pin	—
ESC_PHYLINK0	I	EtherCAT PHY0 link status signal input pin	High*1
ESC_PHYLINK1	I	EtherCAT PHY1 link status signal input pin	High*1
ESC_PHYLINK2	I	EtherCAT PHY2 link status signal input pin	High*1
ESC_MDC	O	Management data clock output pin	—
ESC_MDIO	I/O	Management data I/O pin	—

Note 1. ESC_PHYLINKn (n = 0 to 2) active level is controlled by PHYLNK register.

30.2 Functional Overview

Typical functions of the EtherCAT slave controller and supported functions by this LSI are shown in Table 30.3. Regarding the detailed specification of EtherCAT and ESC, refer to the documentation (ETG.1000 EtherCAT Specification) provided by EtherCAT Technology Group (ETG) and the EtherCAT Slave Controller IP Core (v2.04) data sheet provided by Beckhoff Automation.

Table 30.3 Typical functions of EtherCAT slave controller and supported functions by this LSI (1 of 3)

Features	Functions	Support
EtherCAT protocol	Handling the following frames: ● Ethernet frames with Ether type 0x88A4 ● EtherCAT frames encapsulated in UDP/IP ● EtherCAT frames with VLAN tag ● Normal Ethernet frames	✓
Addressing modes	Device addressing ● Auto increment address ● Configured station address ● Broadcast address	✓
	Logical addressing	✓
Working counter	Counting the number of read/write from/to the device	✓
EtherCAT command type	Processing the command that master requests slaves to address each addressing mode	✓
Loop control	Loop control and loop state in ESC	✓
Shadow buffer	Shadow buffers function when register is read/written	✓
Circulating frames	Processing of circulating frames during the failure	✓

Table 30.3 Typical functions of EtherCAT slave controller and supported functions by this LSI (2 of 3)

Features	Functions	Support
Link detection	Link MII signal (PHY link signal) (monitoring the PHY register through the management interface)	✓
	MI Link detection and configuration	—
	Enhanced link detection (monitoring the state of transfer by MII RX error monitor)	✓
FIFO size reduction	RX FIFO size reduction because of reduction of propagation delay	✓
Ethernet physical layer	MII	✓
	EBUS	—
	Back-to-Back MII connection	✓
	MII management interface	✓
	Read/write of the PHY register through MII management interface	✓
	PHY address offset	✓
	Manual TX shift compensation	✓
	Automatic TX shift compensation See Table 30.4 for enabling condition.	✓
FMMU	Mapping between logical address and physical address	✓
SyncManager	Buffer mode	✓
	Mailbox mode	✓
	Interrupt and latch event generation when a buffer was completely and successfully written or read.	✓
	Repeating mailbox communication	✓
	SyncManager deactivation by the PDI	✓
Distributed clocks	Clock synchronization considering propagation delay and drift compensation	✓
	Generation of synchronous output signals (SYNC0 and 1 signals) ● Cyclic mode ● Single shot mode ● Cyclic acknowledge mode ● Single shot acknowledge mode	✓
	Precise time stamping of input events (LATCH0 and 1 signals) ● Single event mode ● Continuous mode ● SyncManager event mode (for debugging)	✓
	Generation of synchronous interrupts	✓
	Synchronous digital output updates / Synchronous digital input sampling	—
	Exclusive control for the SYNC and LATCH signals of the ECAT and PDI	✓
	System time control by the PDI	—
	Communication Timing ● Free run ● Synchronized to output event ● Synchronized to SYNC signal	✓
EtherCAT state machine	Control of state machine/indication of the status and error code	✓
	Device emulation	—
SII EEPROM	SII EEPROM commands	✓
	SII EEPROM error indication	✓
	SII EEPROM access interface	✓
	EEPROM size selection	✓
	EEPROM emulation	—

Table 30.3 Typical functions of EtherCAT slave controller and supported functions by this LSI (3 of 3)

Features	Functions	Support
Interrupt	AL event request (PDI interrupt)	✓
	ECAT event request (ECAT interrupt)	✓
Watchdog	Process data watchdog	✓
	PDI watchdog	✓
Error counters	Port error counters	✓
	Forwarded RX error counter	✓
	ECAT processing unit error counter	✓
	PDI error counter	✓
	Lost link counter	✓
	Watchdog counter process data	✓
	Watchdog counter PDI	✓
LED signals	RUN LED signal	✓
	ERR LED signal	✓
	STATE LED and STATE_RUN LED signals	✓
	LINK/ACT LED signals	✓
	Port error LED signal	—
	RUN/ERR LED override	✓
Process data interface (PDI)	Digital I/O	—
	SPI slave	—
	8-bit/16-bit synchronous/asynchronous microcontroller interface	—
	On-chip bus	✓
	General purpose I/O	—
Write protection	Write protection for the register area (0x0000 to 0x0FFF)	✓
	Write protection for the whole area including the user RAM and process data RAM (0x0000 to 0x2FFF)	✓
ESC reset	ESC reset from the master or PDI	✓

Table 30.4 Configuration of TX shift function

PHYSEL bit in SCKCR register	ETHn_TXCLK	
	Connected	Not connected
0 (ETHn_REFCLK is based on PLL)	Valid (Automatic TX shift is enabled)	Valid (Manual TX shift may be needed)
1 (ETHn_REFCLK is based on Main OSC or external clock input)	Valid (Automatic TX shift is enabled)	Invalid (This combination is prohibited)

30.3 Register Map

Table 30.5 ESC register map (1 of 5)

Address	Register symbol	Register name	Write protect
0x8011_0200	ECATOFFADR	EtherCAT PHY Offset Address Setting Register	PRCMD
0x8011_0204	ECATOPMOD	EtherCAT Operation Mode Register	PRCMD
0x8011_0208	ECATDBGC	EtherCAT Debug Control Register	PRCMD

Table 30.5 ESC register map (2 of 5)

Address	Register symbol	Register name	Write protect
0x8011_020C	ECATTRGSEL	EtherCAT DC Latch Trigger Select Register	PRCMD
0x8011_0210	ECATRESOUT	EtherCAT Reset Out Control Register	PRCMD
0x8013_0000	TYPE	Type Register	—
0x8013_0001	REVISION	Revision Register	—
0x8013_0002	BUILD	Build Register	—
0x8013_0004	FMMU_NUM	FMMU Supported Register	—
0x8013_0005	SYNC_MANAGER	SyncManager Supported Register	—
0x8013_0006	RAM_SIZE	RAM Size Register	—
0x8013_0007	PORT_DESC	Port Descriptor Register	—
0x8013_0008	FEATURE	ESC Features Supported Register	—
0x8013_0010	STATION_ADR	Configured Station Address Register	—
0x8013_0012	STATION_ALIAS	Configured Station Alias Register	—
0x8013_0020	WR_REG_ENABLE	Write Register Enable Register	—
0x8013_0021	WR_REG_PROTECT	Write Register Protection Register	—
0x8013_0030	ESC_WR_ENABLE	ESC Write Enable Register	—
0x8013_0031	ESC_WR_PROTECT	ESC Write Protection Register	—
0x8013_0040	ESC_RESET_ECAT_R	ESC Reset ECAT Register for read	—
0x8013_0040	ESC_RESET_ECAT_W	ESC Reset ECAT Register for write	—
0x8013_0041	ESC_RESET_PDI_R	ESC Reset PDI Register for read	—
0x8013_0041	ESC_RESET_PDI_W	ESC Reset PDI Register for write	—
0x8013_0100	ESC_DL_CONTROL	ESC DL Control Register	—
0x8013_0108	PHYSICAL_RW_OFFSET	Physical Read/Write Offset Register	—
0x8013_0110	ESC_DL_STATUS	ESC DL Status Register	—
0x8013_0120	AL_CONTROL	AL Control Register	—
0x8013_0130	AL_STATUS	AL Status Register	—
0x8013_0134	AL_STATUS_CODE	AL Status Code Register	—
0x8013_0138	RUN_LED_OVERRIDE	RUN LED Override Register	—
0x8013_0139	ERR_LED_OVERRIDE	ERR LED Override Register	—
0x8013_0140	PDI_CONTROL	PDI Control Register	—
0x8013_0141	ESC_CONFIG	ESC Configuration Register	—
0x8013_0150	PDI_CONFIG	PDI Configuration Register	—
0x8013_0151	SYNC_LATCH_CONFIG	SYNC/LATCH PDI Configuration Register	—
0x8013_0152	EXT_PDI_CONFIG	Extended PDI Configuration Register	—
0x8013_0200	ECAT_EVENT_MASK	ECAT Event Mask Register	—
0x8013_0204	AL_EVENT_MASK	AL Event Mask Register	—
0x8013_0210	ECAT_EVENT_REQ	ECAT Event Request Register	—
0x8013_0220	AL_EVENT_REQ	AL Event Request Register	—
0x8013_0300 + 0x2 × n	RX_ERR_COUNTn	RX Error Counter n Register (n = 0 to 2)	—
0x8013_0308 + 0x1 × n	FWD_RX_ERR_COUNTn	Forwarded RX Error Counter n Register (n = 0 to 2)	—

Table 30.5 ESC register map (3 of 5)

Address	Register symbol	Register name	Write protect
0x8013_030C	ECAT_PROC_ERR_COUNT	ECAT Processing Unit Error Counter Register	—
0x8013_030D	PDI_ERR_COUNT	PDI Error Counter Register	—
0x8013_0310 + 0x1 × n	LOST_LINK_COUNTn	Lost Link Counter n Register (n = 0 to 2)	—
0x8013_0400	WD_DIVIDE	Watchdog Divider Register	—
0x8013_0410	WDT_PDI	Watchdog Time PDI Register	—
0x8013_0420	WDT_DATA	Watchdog Time Process Data Register	—
0x8013_0440	WDS_DATA	Watchdog Status Process Data Register	—
0x8013_0442	WDC_DATA	Watchdog Counter Process Data Register	—
0x8013_0443	WDC_PDI	Watchdog Counter PDI Register	—
0x8013_0500	EEP_CONF	EEPROM Configuration Register	—
0x8013_0501	EEP_STATE	EEPROM PDI Access State Register	—
0x8013_0502	EEP_CONT_STAT	EEPROM Control/Status Register	—
0x8013_0504	EEP_ADR	EEPROM Address Register	—
0x8013_0508	EEP_DATA	EEPROM Data Register	—
0x8013_0510	MII_CONT_STAT	MII Management Control/Status Register	—
0x8013_0512	PHY_ADR	PHY Address Register	—
0x8013_0513	PHY_REG_ADR	PHY Register Address Register	—
0x8013_0514	PHY_DATA	PHY Data Register	—
0x8013_0516	MII_ECAT_ACS_STAT	MII Management ECAT Access State Register	—
0x8013_0517	MII_PDI_ACS_STAT	MII Management PDI Access State Register	—
0x8013_0600 + 0x10 × n	FMMUn_L_START_ADR	FMMU Logical Start Address n Register (n = 0 to 7)	—
0x8013_0604 + 0x10 × n	FMMUn_LEN	FMMU Length n Register (n = 0 to 7)	—
0x8013_0606 + 0x10 × n	FMMUn_L_START_BIT	FMMU Logical Start Bit n Register (n = 0 to 7)	—
0x8013_0607 + 0x10 × n	FMMUn_L_STOP_BIT	FMMU Logical Stop Bit n Register (n = 0 to 7)	—
0x8013_0608 + 0x10 × n	FMMUn_P_START_ADR	FMMU Physical Start Address n Register (n = 0 to 7)	—
0x8013_060A + 0x10 × n	FMMUn_P_START_BIT	FMMU Physical Start Bit n Register (n = 0 to 7)	—
0x8013_060B + 0x10 × n	FMMUn_TYPE	FMMU Type n Register (n = 0 to 7)	—
0x8013_060C + 0x10 × n	FMMUn_ACT	FMMU Activate n Register (n = 0 to 7)	—
0x8013_0800 + 0x08 × n	SMn_P_START_ADR	SyncManager Physical Start Address n Register (n = 0 to 7)	—
0x8013_0802 + 0x08 × n	SMn_LEN	SyncManager Length n Register (n = 0 to 7)	—
0x8013_0804 + 0x08 × n	SMn_CONTROL	SyncManager Control n Register (n = 0 to 7)	—
0x8013_0805 + 0x08 × n	SMn_STATUS	SyncManager Status n Register (n = 0 to 7)	—
0x8013_0806 + 0x08 × n	SMn_ACT	SyncManager Activate n Register (n = 0 to 7)	—
0x8013_0807 + 0x08 × n	SMn_PDI_CONT	SyncManager PDI Control n Register (n = 0 to 7)	—
0x8013_0900	DC_RCV_TIME_PORT0	Receive Time Port 0 Register	—

Table 30.5 ESC register map (4 of 5)

Address	Register symbol	Register name	Write protect
0x8013_0904	DC_RCV_TIME_PORT1	Receive Time Port 1 Register	—
0x8013_0908	DC_RCV_TIME_PORT2	Receive Time Port 2 Register	—
0x8013_0910	DC_SYS_TIME_L	System Time Register L	—
0x8013_0914	DC_SYS_TIME_H	System Time Register H	—
0x8013_0918	DC_RCV_TIME_UNIT_L	Receive Time ECAT Processing Unit Register L	—
0x8013_091C	DC_RCV_TIME_UNIT_H	Receive Time ECAT Processing Unit Register H	—
0x8013_0920	DC_SYS_TIME_OFFSET_L	System Time Offset Register L	—
0x8013_0924	DC_SYS_TIME_OFFSET_H	System Time Offset Register H	—
0x8013_0928	DC_SYS_TIME_DELAY	System Time Delay Register	—
0x8013_092C	DC_SYS_TIME_DIFF	System Time Difference Register	—
0x8013_0930	DC_SPEED_COUNT_START	Speed Counter Start Register	—
0x8013_0932	DC_SPEED_COUNT_DIFF	Speed Counter Difference Register	—
0x8013_0934	DC_SYS_TIME_DIFF_FIL_DEPTH	System Time Difference Filter Depth Register	—
0x8013_0935	DC_SPEED_COUNT_FIL_DEPTH	Speed Counter Filter Depth Register	—
0x8013_0980	DC_CYC_CONT	Cyclic Unit Control Register	—
0x8013_0981	DC_ACT	Activation Register	—
0x8013_0982	DC_PULSE_LEN	SYNC Signal Pulse Length Register	—
0x8013_0984	DC_ACT_STAT	Activation Status Register	—
0x8013_098E	DC_SYNC0_STAT	SYNC0 Status Register	—
0x8013_098F	DC_SYNC1_STAT	SYNC1 Status Register	—
0x8013_0990	DC_CYC_START_TIME_L	Start Time Cyclic Operation/Next SYNC0 Pulse Register L	—
0x8013_0994	DC_CYC_START_TIME_H	Start Time Cyclic Operation/Next SYNC0 Pulse Register H	—
0x8013_0998	DC_NEXT_SYNC1_PULSE_L	Next SYNC1 Pulse Register L	—
0x8013_099C	DC_NEXT_SYNC1_PULSE_H	Next SYNC1 Pulse Register H	—
0x8013_09A0	DC_SYNC0_CYC_TIME	SYNC0 Cycle Time Register	—
0x8013_09A4	DC_SYNC1_CYC_TIME	SYNC1 Cycle Time Register	—
0x8013_09A8	DC_LATCH0_CONT	Latch 0 Control Register	—
0x8013_09A9	DC_LATCH1_CONT	Latch 1 Control Register	—
0x8013_09AE	DC_LATCH0_STAT	Latch 0 Status Register	—
0x8013_09AF	DC_LATCH1_STAT	Latch 1 Status Register	—
0x8013_09B0	DC_LATCH0_TIME_POS_L	Latch 0 Time Positive Edge Register L	—
0x8013_09B4	DC_LATCH0_TIME_POS_H	Latch 0 Time Positive Edge Register H	—
0x8013_09B8	DC_LATCH0_TIME_NEG_L	Latch 0 Time Negative Edge Register L	—
0x8013_09BC	DC_LATCH0_TIME_NEG_H	Latch 0 Time Negative Edge Register H	—
0x8013_09C0	DC_LATCH1_TIME_POS_L	Latch 1 Time Positive Edge Register L	—
0x8013_09C4	DC_LATCH1_TIME_POS_H	Latch 1 Time Positive Edge Register H	—
0x8013_09C8	DC_LATCH1_TIME_NEG_L	Latch 1 Time Negative Edge Register L	—
0x8013_09CC	DC_LATCH1_TIME_NEG_H	Latch 1 Time Negative Edge Register H	—

Table 30.5 ESC register map (5 of 5)

Address	Register symbol	Register name	Write protect
0x8013_09F0	DC_ECAT_CNG_EV_TIME	Buffer Change Event Time Register	—
0x8013_09F8	DC_PDI_START_EV_TIME	PDI Buffer Start Event Time Register	—
0x8013_09FC	DC_PDI_CNG_EV_TIME	PDI Buffer Change Event Time Register	—
0x8013_0E00	PRODUCT_ID_L	Product ID Register L	—
0x8013_0E04	PRODUCT_ID_H	Product ID Register H	—
0x8013_0E08	VENDOR_ID_L	Vendor ID Register L	—

Table 30.6 ESC related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
ECATOFFADR, ECATOPMOD, ECATDBGC, ECATTRGSEL, ECATRESOUT	MRCTLE.MRTCLE05	MSTPCRE.MSTPCRE03	SLVACCCTL6.ETHSS_SL
Other than the above	MRCTLE.MRTCLE03 MRCTLE.MRTCLE04	MSTPCRE.MSTPCRE02	SLVACCCTL6.ESC_SL

30.4 ESC Auxiliary Registers

30.4.1 ECATOFFADR : EtherCAT PHY Offset Address Setting Register

Base address: ESC_INI = 0x8011_0200

Offset address: 0x0_0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	OADD[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	OADD[4:0]	PHY Offset Address Setting Set the offset address of PHY of using the EtherCAT.	R/W
31:5	—	The read values are undefined. The write value should be 0.	R/W

Note: This register can only be written when protection is unlocked by the specific sequence using the Ethernet Protect register (PRCMD). For the protection unlock procedure, see section 27.3.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register.

The ECATOFFADR register sets the offset address of the Ethernet PHY in case of using the EtherCAT. This register can be read/written in 32-bit units.

30.4.2 ECATOPMOD : EtherCAT Operation Mode Register

Base address: ESC_INI = 0x8011_0200

Offset address: 0x0_0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EEPROMSIZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EEPROMSIZE	EEPROM Memory Size Specification Sets the size of the EEPROM memory of the EtherCAT. 0: 16 Kbits or less 1: 32 Kbits to 4 Mbits	R/W
31:1	—	The read values are undefined. The write value should be 0.	R/W

Note: This register can only be written when protection is unlocked by the specific sequence using the Ethernet Protect register (PRCMD). For the protection unlock procedure, see section 27.3.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register

The ECATOPMOD register sets the EEPROM memory size in case of using the EtherCAT. This register can be read/written in 32-bit units.

30.4.3 ECATDBGC : EtherCAT Debug Control Register

Base address: ESC_INI = 0x8011_0200

Offset address: 0x0_0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TXSFT2[1:0]		TXSFT1[1:0]		TXSFT0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TXSFT0[1:0]	Set the delay time for ETH0_TXEN and ETH0_TXDn of the EtherCAT 0 0: 0 ns 0 1: 10 ns 1 0: 20 ns 1 1: 30 ns	R/W
3:2	TXSFT1[1:0]	Set the delay time for ETH1_TXEN and ETH1_TXDn of the EtherCAT 0 0: 0 ns 0 1: 10 ns 1 0: 20 ns 1 1: 30 ns	R/W

Bit	Symbol	Function	R/W
5:4	TXSFT2[1:0]	Set the delay time for ETH2_TXEN and ETH2_TXDn of the EtherCAT 0 0: 0 ns 0 1: 10 ns 1 0: 20 ns 1 1: 30 ns	R/W
31:6	—	The read values are undefined. The write value should be 0.	R/W

Note: This register can only be written when protection is unlocked by the specific sequence using the Ethernet Protect register (PRCMD). For the protection unlock procedure, see section 27.3.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register.

The ECATDBGC register controls the delay time of TXEN and TXDn. This register can be read/written in 32-bit units.

30.4.4 ECATTRGSEL : EtherCAT DC Latch Trigger Select Register

Base address: ESC_INI = 0x8011_0200

Offset address: 0x0_000C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRGSEL1	TRGSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRGSEL0	Select DC Latch Trigger 0 for ESC 0: Select LATCH0 input from ELC 1: Select LATCH0 input from external	R/W
1	TRGSEL1	Select DC Latch Trigger 1 for ESC 0: Select LATCH1 input from ELC 1: Select LATCH1 input from external	R/W
31:2	—	The read values are undefined. The write value should be 0.	R/W

Note: This register can only be written when protection is unlocked by the specific sequence using the Ethernet Protect register (PRCMD). For the protection unlock procedure, see section 27.3.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register.

The ECATTRGSEL register selects the source to LATCHn (n = 0, 1) inputs to ESC. This register can be read/written in 32-bit units.

30.4.5 ECATRESOUT : EtherCAT Reset Out Control register

Base address: ESC_INI = 0x8011_0200

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RESOUT_EN	FORCE_RESET
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	FORCE_RESET	Force ESC_RESETOUT# pin state 0: Do not force reset state 1: Force reset state (Low level output)	R/W
1	RESOUT_EN	Specify whether Reset out from ESC is output to ESC_RESETOUT# pin or not. 0: Reset out from ESC is not output to ESC_RESETOUT# pin 1: Reset out from ESC is output to ESC_RESETOUT# pin	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be written when protection is unlocked by the specific sequence using the Ethernet Protect register (PRCMD). For the protection unlock procedure, see section 27.3.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register.

The ECATRESOUT register controls ESC_RESETOUT# pin state.

30.5 ESC Information Register

In the tables, ECAT indicates access by the EtherCAT master controller and PDI indicates access by the local CPU.

30.5.1 TYPE : Type Register

Base address: ESC = 0x8013_0000

Offset address: 0x0000

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TYPE[7:0]							
Value after reset:	1	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TYPE[7:0]	Type of the EtherCAT slave controller	R

The TYPE register indicates the type of the EtherCAT slave controller.

30.5.2 REVISION : Revision Register

Base address: ESC = 0x8013_0000

Offset address: 0x0001

Bit position:	7	6	5	4	3	2	1	0
Bit field:	REV[7:0]							
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
7:0	REV[7:0]	Revision of the EtherCAT slave controller	R

The REVISION register indicates the revision of the EtherCAT slave controller.

30.5.3 BUILD : Build Register

Base address: ESC = 0x8013_0000

Offset address: 0x0002

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BUILD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	BUILD[15:0]	Build number of the EtherCAT slave controller	R

The BUILD register indicates the build number of the EtherCAT slave controller.

30.5.4 FMMU_NUM : FMMU Supported Register

Base address: ESC = 0x8013_0000

Offset address: 0x0004

Bit position:	7	6	5	4	3	2	1	0
Bit field:	NUMFMMU[7:0]							
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
7:0	NUMFMMU[7:0]	Number of FMMU channels supported in the EtherCAT slave controller	R

The FMMU_NUM register indicates the number of FMMU channels supported in the EtherCAT slave controller.

30.5.5 SYNC_MANAGER : SyncManager Supported Register

Base address: ESC = 0x8013_0000

Offset address: 0x0005

Bit position:	7	6	5	4	3	2	1	0
Bit field:	NUMSYNC[7:0]							
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
7:0	NUMSYNC[7:0]	Number of SyncManager channels supported in the EtherCAT slave controller	R

The SYNC_MANAGER register indicates the number of SyncManager channels supported in the EtherCAT slave controller.

30.5.6 RAM_SIZE : RAM Size Register

Base address: ESC = 0x8013_0000

Offset address: 0x0006

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RAMSIZE[7:0]							
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
7:0	RAMSIZE[7:0]	Process data RAM size supported in the EtherCAT slave controller (unit: KB)	R

The RAM_SIZE register indicates the process data RAM size supported in the EtherCAT slave controller in KB.

30.5.7 PORT_DESC : Port Descriptor Register

Base address: ESC = 0x8013_0000

Offset address: 0x0007

Bit position:	7	6	5	4	3	2	1	0
Bit field:	P3[1:0]		P2[1:0]		P1[1:0]		P0[1:0]	
Value after reset:	0	0	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	P0[1:0]	Port 0 configuration Fixed to the setting for MII connection (11b) in this LSI. 0 0: Not implemented 0 1: Not configured (SII EEPROM) 1 0: EBUS 1 1: MII	R
3:2	P1[1:0]	Port 1 configuration Fixed to the setting for MII connection (11b) in this LSI. 0 0: Not implemented 0 1: Not configured (SII EEPROM) 1 0: EBUS 1 1: MII	R
5:4	P2[1:0]	Port 2 configuration Fixed to the setting for MII connection (11b) in this LSI. 0 0: Not implemented 0 1: Not configured (SII EEPROM) 1 0: EBUS 1 1: MII	R
7:6	P3[1:0]	Port 3 configuration This LSI does not implement port 3. Fixed to 00b. 0 0: Not implemented 0 1: Not configured (SII EEPROM) 1 0: EBUS 1 1: MII	R

The PORT_DESC register indicates the port configuration.

30.5.8 FEATURE : ESC Features Supported Register

Base address: ESC = 0x8013_0000

Offset address: 0x0008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	FSCONFIG	RWSUPP	LRW	DCSYNC	FCS	LINKDECMII	—	—	DCWID	DC	—	FMMU
Value after reset:	x	x	x	x	0	0	0	1	1	1	x	x	1	1	x	0

Bit	Symbol	Function	R/W
0	FMMU	FMMU Operation 0: Bit oriented 1: Byte oriented	R
1	—	The read value is undefined.	R
2	DC	Distributed Clock 0: Not available 1: Available	R

Bit	Symbol	Function	R/W
3	DCWID	Distributed Clock Width 0: 32 bits 1: 64 bits	R
5:4	—	The read values are undefined.	R
6	LINKDECMII	Enhanced Link Detection in MII 0: Not available 1: Available	R
7	FCS	Separate handling of FCS errors 0: Not supported 1: Supported. Frames with wrong FCS and additional nibble will be counted separately in forwarded RX error counter.	R
8	DCSYNC	Enhanced DC SYNC activation 0: Not available 1: Available	R
9	LRW	EtherCAT LRW command support 0: Supported 1: Not supported	R
10	RWSUPP	EtherCAT read/write command support (BRW, APRW, FPRW) 0: Supported 1: Not supported	R
11	FSCONFIG	Fixed FMMU/SyncManager configuration 0: Variable configuration 1: Fixed configuration	R
15:12	—	The read values are undefined.	R

The FEATURE register indicates the features supported in the EtherCAT slave controller.

30.6 Station Address Registers

30.6.1 STATION_ADR : Configured Station Address Register

Base address: ESC = 0x8013_0000

Offset address: 0x0010

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NODADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	NODADDR[15:0]	Node Addressing Address Indication Address used for node addressing (FPxx commands) The access type from the EtherCAT master is R/W.	R

The STATION_ADR register indicates the address used for node addressing.

30.6.2 STATION_ALIAS : Configured Station Alias Register

Base address: ESC = 0x8013_0000

Offset address: 0x0012

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NODALIADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	NODALIADDR[15:0]	Alias Address Indication Alias address used for node addressing (FPxx commands). The use of this alias is activated by setting bit 24 of the ESC DL control register (ESC_DL_CONTROL at 0x0100) to 1. The access type from the EtherCAT master is R.	R/W

Note: The initial value, 0, is retained until the EEPROM is loaded. After that, the value becomes the value at address 0x0004 in the EEPROM. This value is only taken over from the EEPROM the first time the EEPROM is loaded after a power-on or reset.

The STATION_ALIAS register indicates the alias address used for node addressing (FPxx commands).

30.7 Write Protection Registers

30.7.1 WR_REG_ENABLE : Write Register Enable Register

Base address: ESC = 0x8013_0000

Offset address: 0x0020

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ENABLE
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	ENABLE	Register Write Protection Unlock When registers are currently being protected against writing (bit 0 is 1 in the write register protection register, WR_REG_PROTECT, at 0x0021) and freely writing to registers of the given node is to be permitted, the operation to do so by writing to this register has to proceed in the same Ethernet frame and preceding the other desired writing to registers. Write protection will be reactivated once the frame period elapses (unless the value in the write register protection register is changed). The access type from the EtherCAT master is R/W.	R
7:1	—	The read values are undefined.	R

The WR_REG_ENABLE register is used to unlock the write protection temporarily while registers are write-protected.

30.7.2 WR_REG_PROTECT : Write Register Protection Register

Base address: ESC = 0x8013_0000

Offset address: 0x0021

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PROTECT
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	PROTECT	Register Write Protection Specification The access type from the EtherCAT master is R/W. 0: Protection disabled 1: Protection enabled	R
7:1	—	The read values are undefined.	R

The WR_REG_PROTECT register is used to protect registers against writing. The registers in the area 0x8013_0000 to 0x8013_0FFF are write-protected (except for the WR_REG_ENABLE register (0x0020) and ESC_WR_ENABLE register (0x0030)).

30.7.3 ESC_WR_ENABLE : ESC Write Enable Register

Base address: ESC = 0x8013_0000

Offset address: 0x0030

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ENABLE
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	ENABLE	Register/Memory Write Protection Unlock When registers are currently being protected against writing by ESC write protection (bit 0 is 1 in the ESC write protection register, ESC_WR_PROTECT, at 0x0031) and freely writing to registers of the given node is to be permitted, the operation to do so by writing to this register has to proceed in the same Ethernet frame and preceding the other desired writing to registers. Write protection will be reactivated once the frame period elapses (unless the value in the ESC write protection register is changed). The access type from the EtherCAT master is R/W.	R
7:1	—	The read values are undefined.	R

The ESC_WR_ENABLE register is used to unlock the write protection temporarily while registers and memories are write-protected by ESC write protection.

30.7.4 ESC_WR_PROTECT : ESC Write Protection Register

Base address: ESC = 0x8013_0000

Offset address: 0x0031

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PROTECT
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	PROTECT	Register/Memory Write Protection Specification The access type from the EtherCAT master is R/W. 0: Protection disabled 1: Protection enabled	R
7:1	—	The read values are undefined.	R

The ESC_WR_PROTECT register is used to protect registers against writing. Registers and memories in the area 0x8013_0000 to 0x8013_2FFF including the process data RAM are write-protected (except for the WR_REG_ENABLE register (0x0020) and ESC_WR_ENABLE register (0x0030)).

30.8 Data Link Layer Registers

30.8.1 ESC_RESET_ECAT_W : ESC Reset ECAT Register for write

Base address: ESC = 0x8013_0000

Offset address: 0x0040

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RESET_ECAT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RESET_ECAT[7:0]	Software Reset Setting A reset is enabled after writing 0x52 ('R'), 0x45 ('E'), and 0x53 ('S') consecutively to this register. The access type from the EtherCAT master is R/W.	R

This register is used to reset the EtherCAT slave controller from the ECAT (master) by software.

30.8.2 ESC_RESET_ECAT_R : ESC Reset ECAT Register for read

Base address: ESC = 0x8013_0000

Offset address: 0x0040

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RESET_ECAT[1:0]	
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
1:0	RESET_ECAT[1:0]	Reset Progress Status The access type from the EtherCAT master is R/W. 0 1: After writing 0x52 1 0: After writing 0x45 (if 0x52 was written before) 0 0: Others	R
7:2	—	The read values are undefined. The access type from the EtherCAT master is R/W. In this case, the write value should be 0.	R

This register is used to reset the EtherCAT slave controller from the ECAT (master) by software.

30.8.3 ESC_RESET_PDI_W : ESC Reset PDI Register for write

Base address: ESC = 0x8013_0000

Offset address: 0x0041

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RESET_PDI[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RESET_PDI[7:0]	Software Reset Setting A reset is enabled after writing 0x52 ('R'), 0x45 ('E'), and 0x53 ('S') consecutively to this register. The access type from the EtherCAT master is R.	R/W

This register is used to reset the EtherCAT slave controller from the PDI (slave) by software.

30.8.4 ESC_RESET_PDI_R : ESC Reset PDI Register for read

Base address: ESC = 0x8013_0000

Offset address: 0x0041

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RESET_PDI[1:0]	
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
1:0	RESET_PDI[1:0]	Reset Progress Status The access type from the EtherCAT master is R. 0 1: After writing 0x52 1 0: After writing 0x45 (if 0x52 was written before) 0 0: Others	R/W
7:2	—	The read values are undefined. The write value should be 0.	R/W

This register is used to reset the EtherCAT slave controller from the PDI (slave) by software.

30.8.5 ESC_DL_CONTROL : ESC DL Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x0100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	STAAL IAS	—	—	—	—	—	RXFIFO[2:0]		
Value after reset:	x	x	x	x	x	x	x	0	x	x	x	x	x	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LP3[1:0]		LP2[1:0]		LP1[1:0]		LP0[1:0]		—	—	—	—	—	—	TEMP USE	FWDR ULE
Value after reset:	1	1	0	0	0	0	0	0	x	x	x	x	0	0	0	1

Bit	Symbol	Function	R/W
0	FWDRULE	Forwarding Rule The source MAC address is changed for every frame (SOURCE_MAC[1] is set to 1 (locally administered address)) regardless of the forwarding rule. The access type from the EtherCAT master is R/W. 0: EtherCAT frames are processed. Non-EtherCAT frames are forwarded without processing. 1: EtherCAT frames are processed. Non-EtherCAT frames are destroyed.	R
1	TEMPUSE	Temporary Use of Bits 15 to 8 Settings The access type from the EtherCAT master is R/W. 0: Permanent use 1: Use for about 1 second, then revert to previous settings	R
3:2	—	These bits are read as 0.	R
7:4	—	The read values are undefined.	R
9:8	LP0[1:0]	Loop Port 0 Configuration The access type from the EtherCAT master is R/W. 0 0: Auto 0 1: Auto close 1 0: Open 1 1: Closed	R
11:10	LP1[1:0]	Loop Port 1 Configuration The access type from the EtherCAT master is R/W. 0 0: Auto 0 1: Auto close 1 0: Open 1 1: Closed	R
13:12	LP2[1:0]	Loop Port 2 Configuration Port 2 is not available on this LSI. The access type from the EtherCAT master is R/W. 0 0: Auto 0 1: Auto close 1 0: Open 1 1: Closed	R

Bit	Symbol	Function	R/W
15:14	LP3[1:0]	Loop Port 3 Configuration Port 3 is not available on this LSI. The access type from the EtherCAT master is R/W. 0 0: Auto 0 1: Auto close 1 0: Open 1 1: Closed	R
18:16	RXFIFO[2:0]	RX FIFO Size Set the RX FIFO size. The transfer time can be reduced by reducing the FIFO size. The access type from the EtherCAT master is R/W. 0 x x: Shortened by 40 ns 1 1 1: Default Others: No change	R
23:19	—	The read values are undefined.	R
24	STAALIAS	Station Alias Status The access type from the EtherCAT master is R/W. 0: Ignore station alias 1: Alias can be used for all configured address command types such as FPRD, FPWR.	R
31:25	—	The read values are undefined.	R

Note:
- Changes to loop configurations are delayed until any current reception or transmission of a frame through the port is completed.
- Reducing the size of the RX FIFO depends on all masters and slaves connected to the same network as the EtherCAT having very precise clock sources. An RX FIFO size of 7 (default) is sufficient if the precision of all clocks is 100 ppm or better and RX FIFO sizes of 0 to 3 are possible if the precision is 25 ppm or better.

The ESC_DL_CONTROL register is used to control loop in the EtherCAT slave controller and configure the RX FIFO size and station alias.

30.8.6 PHYSICAL_RW_OFFSET : Physical Read/Write Offset Register

Base address: ESC = 0x8013_0000

Offset address: 0x0108

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RWOFFSET[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RWOFFSET[15:0]	Offset between Read and Write Addresses Offset of R/W commands (FPRW, APRW) between read address and write address. When read, RD_ADR = ADR (the given address is read) When write, WR_ADR = ADR + R/W offset (writing is to the address obtained by adding the offset set in this register to the given address) The access type from the EtherCAT master is R/W.	R

The PHYSICAL_RW_OFFSET register is used to set the offset between read address and write address in the R/W commands.

30.8.7 ESC_DL_STATUS : ESC DL Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x0110

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	COMP 3	LP3	COMP 2	LP2	COMP 1	LP1	COMP 0	LP0	PHYP 3	PHYP 2	PHYP 1	PHYP 0	—	ENHLI NKD	PDIW DST	PDIOP E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	x	1	0	0

Bit	Symbol	Function	R/W
0	PDIOPE	PDI/EEPROM Load State Indication 0: EEPROM not loaded, the PDI not operational (process data RAM is not accessible) 1: EEPROM loaded correctly, the PDI operational (process data RAM is accessible)	R
1	PDIWDST	PDI Watchdog Timer Status 0: Timeout of the watchdog timer 1: Watchdog timer reloaded	R
2	ENHLINKD	Enhanced Link Detection Indication This bit is set to the value of bit 9 at address 0x0000 in the EEPROM the first time the EEPROM is loaded after power is initially supplied or after a reset. 0: Deactivated for all ports 1: Activated for at least one port	R
3	—	The read value is undefined.	R
4	PHYP0	Port 0 Link State Indication Physical link on port 0. 0: No link 1: Link detected	R
5	PHYP1	Port 1 Link State Indication Physical link on port 1. 0: No link 1: Link detected	R
6	PHYP2	Port 2 Link State Indication Port 2 is not available on this LSI. 0: No link 1: Link detected	R
7	PHYP3	Port 3 Link State Indication Port 3 is not available on this LSI. 0: No link 1: Link detected	R
8	LP0	Loop Port 0 State Indication 0: Open 1: Closed	R
9	COMP0	Port 0 Communication State Indication 0: No stable communication 1: Communication established	R
10	LP1	Loop Port 1 State Indication 0: Open 1: Closed	R
11	COMP1	Port 1 Communication State Indication 0: No stable communication 1: Communication established	R
12	LP2	Loop Port 2 State Indication Port 2 is not available on this LSI. 0: Open 1: Closed	R
13	COMP2	Port 2 Communication State Indication Port 2 is not available on this LSI. 0: No stable communication 1: Communication established	R
14	LP3	Loop Port 3 State Indication Port 3 is not available on this LSI. 0: Open 1: Closed	R
15	COMP3	Port 3 Communication State Indication Port 3 is not available on this LSI. 0: No stable communication 1: Communication established	R

Note: Reading this DL status register from the ECAT clears bit 2 of the ECAT event request register (ECAT_EVENT_REQ at 0x0210). Avoid reading DL Status register from PDI.

The ESC_DL_STATUS register indicates the state of the EtherCAT slave controller.

30.9 Application Layer Registers

30.9.1 AL_CONTROL : AL Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x0120

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	DEVICEID	ERRINDACK	INISTATE[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	INISTATE[3:0]	Change the state transition of the device state machine. The access type from the EtherCAT master is R/W. 0x1: Initial state request 0x2: Pre-operational state request 0x3: Bootstrap state request 0x4: Safe-operational state request 0x8: Operational state request Others: Reserved	R
4	ERRINDACK	Error Indication Acknowledge (Response) The access type from the EtherCAT master is R/W. 0: Error Indication in AL status register is not acknowledged 1: Error Indication in AL status register is acknowledged	R
5	DEVICEID	Device ID Request The access type from the EtherCAT master is R/W. 0: No request is present 1: A request is present	R
15:6	—	The read values are undefined. The access type from the EtherCAT master is R/W. In this case, the write value should be 0.	R

Note: The PDI has to read the AL control register after the ECAT has written it. Otherwise, the ECAT cannot write again to the AL control register. Reading the AL control register from the PDI clears bit 0 of the AL event request register (AL_EVENT_REQ at 0x0220).

The AL_CONTROL register is used to change the state transition of the device state machine and to acknowledge error indication.

30.9.2 AL_STATUS : AL Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x0130

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	DEVICEID	ERR	ACTSTATE[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	ACTSTATE[3:0]	State Machine State Indication Actual state of the device state machine The access type from the EtherCAT master is R. 0x1: Initial state 0x2: Pre-operational state 0x3: Request bootstrap state 0x4: Safe-operational state 0x8: Operational state	R/W
4	ERR	Error State Indication The access type from the EtherCAT master is R. 0: The device is in the state as requested or flag was cleared by command 1: The device has not entered the requested state or the state was changed as a result of local action	R/W
5	DEVICEID	Device ID Load State Indication The access type from the EtherCAT master is R. 0: Loading failed 1: Loading was successful	R/W
15:6	—	The read values are undefined. The write value should be 0.	R/W

Note: Reading this AL status register from the ECAT clears bit 3 of the ECAT event request register (ECAT_EVENT_REQ at 0x0210). Avoid reading DL Status register from PDI.

The AL_STATUS register indicates the state of slave application.

30.9.3 AL_STATUS_CODE : AL Status Code Register

Base address: ESC = 0x8013_0000

Offset address: 0x0134

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	STATUSCODE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	STATUSCODE[15:0]	AL status code The access type from the EtherCAT master is R.	R/W

The AL_STATUS_CODE register indicates an error code from slave application.

30.9.4 RUN_LED_OVERRIDE : RUN LED Override Register

Base address: ESC = 0x8013_0000

Offset address: 0x0138

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OVERRIDEEN	LEDCODE[3:0]			
Value after reset:	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	LEDCODE[3:0]	LED Code Indication (FSM state: Bits [3:0] of the AL Status register, AL_STATUS) 0x0: Off (FSM: 1-Init) 0xD: Blinking (FSM: 2-PreOp) 0xE: Flickering (FSM: 3-Bootstrap) 0xF: On (FSM: 8-Op) Others: Flash 1x - 12x (FSM: 4-SafeOp 1x)	R/W

Bit	Symbol	Function	R/W
4	OVERRIDEEN	Override Setting 0: Override disabled 1: Override enabled	R/W
7:5	—	The read values are undefined. The write value should be 0.	R/W

Note: Changing the value of the AL status register to an appropriate value will clear bit 4 (override enable).
Normally RUN LED is controlled by the AL status register (AL_STATUS at 0x0130) automatically. It is not necessary to override RUN LED in order to indicate the state of a general state machine.
For example, this register can be used to run special lighting patterns that indicate the positions of specific slaves.

The RUN_LED_OVERRIDE register is used to override control of the RUN LED pin.

30.9.5 ERR_LED_OVERRIDE : ERR LED Override Register

Base address: ESC = 0x8013_0000

Offset address: 0x0139

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OVERRIDEEN	LEDCODE[3:0]			
Value after reset:	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	LEDCODE[3:0]	LED Code Indication 0x0: Off 0xD: Blinking 0xE: Flickering 0xF: On Others: Flash 1x to 12x	R/W
4	OVERRIDEEN	Override Setting 0: Override disabled 1: Override enabled	R/W
7:5	—	The read values are undefined. The write value should be 0.	R/W

Note: Bit 4 (override enable) will be cleared if a new error occurs.
The ESC automatically controls an error LED under the conditions below. Regarding other errors, the error LED should be controlled by application using this register.

- SII EEPROM load error
- PDI watchdog timeout

The ERR_LED_OVERRIDE register is used to override control of the error LED pin.

30.10 PDI Registers

30.10.1 PDI_CONTROL : PDI Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x0140

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PDI[7:0]							
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PDI[7:0]	Process Data Interface. In this LSI, the following value is indicated. 0x80: On-chip bus	R

This register indicates the type of PDI.

30.10.2 ESC_CONFIG : ESC Configuration Register

Base address: ESC = 0x8013_0000

Offset address: 0x0141

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENLP3	ENLP2	ENLP1	ENLP0	DCLATCH	DCSYNC	ENLALLP	DEVEMU
Value after reset:	0	0	0	0	1	1	0	0

Bit	Symbol	Function	R/W
0	DEVEMU	Device emulation (control of AL status) 0: The AL status register must be set by the PDI 1: The AL status register is set to a value written to the AL control register	R
1	ENLALLP	Sets enhanced link detection for all ports 0: Disabled (if bits 15 to 12 of address 0 in the EEPROM = 0) 1: Enabled at all ports	R
2	DCSYNC	Sets the SYNC output unit for distributed clocks (fixed to 1 in this LSI) 0: Disabled (power saving) 1: Enabled	R
3	DCLATCH	Sets the latch input unit for distributed clocks Fixed to 1 in this LSI. 0: Disabled (power saving) 1: Enabled	R
4	ENLP0	Port 0 Enhanced Link Detection Setting 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled	R
5	ENLP1	Port 1 Enhanced Link Detection Setting 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled	R
6	ENLP2	Port 2 Enhanced Link Detection Setting Port 2 is not available in this LSI. 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled	R
7	ENLP3	Port 3 Enhanced Link Detection Setting Port 3 is not available in this LSI. 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled	R

The ESC_CONFIG register indicates configuration of the EtherCAT slave controller.

30.10.3 PDI_CONFIG : PDI Configuration Register

Base address: ESC = 0x8013_0000

Offset address: 0x0150

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ONCHIPBUS[2:0]			ONCHIPBUSCLK[4:0]				
Value after reset:	0	1	0	0	0	1	0	0

Bit	Symbol	Function	R/W
4:0	ONCHIPBUSCLK[4:0]	On-Chip Bus Clock Indication Indicate the frequency of the on-chip bus clock. In this LSI, the value is always 0x04 (corresponding to 100 MHz).	R
7:5	ONCHIPBUS[2:0]	On-Chip Bus Type Indication Indicate the type of on-chip bus. In this LSI, the value is always 010b.	R

The PDI_CONFIG register indicates configuration of the PDI.

30.10.4 SYNC_LATCH_CONFIG : SYNC/LATCH PDI Configuration Register

Base address: ESC = 0x8013_0000

Offset address: 0x0151

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SYNC 1MAP	SYNC LAT1	SYNC1OUT[1:0]		SYNC 0MAP	SYNC LAT0	SYNC0OUT[1:0]	
Value after reset:	1	1	1	0	1	1	1	0

Bit	Symbol	Function	R/W
1:0	SYNC0OUT[1:0]	SYNC0 Output Driver and Polarity Indication Indicate the SYNC0 output driver/polarity. In this LSI, the value is always 10b (push-pull and active-high).	R
2	SYNCLAT0	SYNC0/LATCH0 Indication Indicates the SYNC0/LATCH0 configuration. In this LSI, the value is always 1.*1 0: LATCH0 input 1: SYNC0 output	R
3	SYNC0MAP	SYNC0 State Mapping Indication Indicates enabling or disabling of mapping of the SYNC0 state to bit 2 of the AL event request register (AL_EVENT_REQ at 0x0220). This is always enabled in this LSI, so the value indicated is always 1 (enabled). 0: Disabled 1: Enabled	R
5:4	SYNC1OUT[1:0]	SYNC1 Output Driver and Polarity Indication Indicate the SYNC1 output driver/polarity. In this LSI, the value is always 10b (push-pull and active-high).	R
6	SYNCLAT1	SYNC1/LATCH1 Indication Indicates the SYNC1/LATCH1 configuration. In this LSI, the value is always 1.*1 0: LATCH1 input 1: SYNC1 output	R
7	SYNC1MAP	SYNC1 State Mapping Indication Indicates enabling or disabling of mapping of the SYNC1 state to bit 3 of the AL event request register (AL_EVENT_REQ at 0x0220). This is always enabled in this LSI, so the value indicated is always 1 (enabled). 0: Disabled 1: Enabled	R

Note 1. Latch input is available though the value indicates SYNC output. Use the MPC function in order to switch SYNC output to LATCH input and vice versa.

The SYNC_LATCH_CONFIG register indicates the configuration of SYNC output and LATCH input.

30.10.5 EXT_PDI_CONFIG : Extended PDI Configuration Register

Base address: ESC = 0x8013_0000

Offset address: 0x0152

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATABUSWID[1:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
1:0	DATABUSWID[1:0]	PDI Data Bus Width Indication Indicate the data bus width of the PDI. In this LSI, the value is always 00b (4 bytes). 0 0: 4 bytes 0 1: 1 byte 1 0: 2 bytes 1 1: Reserved	R
15:2	—	The read values are undefined.	R

The EXT_PDI_CONFIG register indicates configuration of the PDI.

30.11 Interrupt Registers

30.11.1 ECAT_EVENT_MASK : ECAT Event Mask Register

Base address: ESC = 0x8013_0000

Offset address: 0x0200

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ECATEVMASK[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ECATEVMASK[15:0]	Event Request Mask Setting The access type from the EtherCAT master is R/W. 0: The corresponding bit of the ECAT Event Request register (ECAT_EVENT_REQ at 0x0210) is not mapped 1: The corresponding bit of the ECAT Event Request register is mapped	R

The ECAT Event Request (ECAT interrupt) is used to transmit the slave event to the EtherCAT master. This register is used to set mask to each event of the ECAT Event Request register (ECAT_EVENT_REQ at 0x0210). The logical AND of each effective bit in the ECAT Event Request register and the corresponding bit of this register is taken and the result produces the interrupt signal.

30.11.2 AL_EVENT_MASK : AL Event Mask Register

Base address: ESC = 0x8013_0000

Offset address: 0x0204

Bit position:	31																													0
Bit field:	ALEVMASK[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
31:0	ALEVMASK[31:0]	Event Request Mask Setting The access type from the EtherCAT master is R. 0: The corresponding bit of the AL Event Request register (AL_EVENT_REQ at 0x0220) is not mapped 1: The corresponding bit of the AL Event Request register is mapped	R/W

The AL event request (PDI interrupt) is used to transmit the ESC interrupt to the slave application. This register is used to set mask to each event of the AL Event Request register (AL_EVENT_REQ at 0x0220). The logical AND of each effective bit in the AL Event Request register and the corresponding bit of this register is taken and the result produces the interrupt signal.

30.11.3 ECAT_EVENT_REQ : ECAT Event Request Register

Base address: ESC = 0x8013_0000

Offset address: 0x0210

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SMSTA7	SMSTA6	SMSTA5	SMSTA4	SMSTA3	SMSTA2	SMSTA1	SMSTA0	ALSTA	DLSTA	—	DCLATCH
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	0	0	x	0

Bit	Symbol	Function	R/W
0	DCLATCH	DC Latch Event State Indication This bit is cleared by reading DC latch event times from the ECAT for ECAT controlled latch units, so the latch 0 and 1 status registers (DC_LATCH_STAT0 and DC_LATCH_STAT1 at 0x09AE and 0x09AF) indicate no event. 0: No change on DC latch inputs 1: At least one change on DC latch inputs	R
1	—	The read value is undefined.	R
2	DLSTA	DL Status Event State Indication This bit is cleared by reading out the DL status register (ESC_DL_STATUS at 0x0110 or 0x0111) from the ECAT. 0: No change in DL status 1: DL status change	R
3	ALSTA	AL Status Event State Indication This bit is cleared by reading out the AL status register (AL_STATUS at 0x0130 or 0x0131) from the ECAT. 0: No change in AL status 1: AL status change	R
4	SMSTA0	Mirror value of SyncManager 0 Status Indication 0: No Sync channel 0 event 1: Sync channel 0 event pending	R
5	SMSTA1	Mirror value of SyncManager 1 Status Indication 0: No Sync channel 1 event 1: Sync channel 1 event pending	R
6	SMSTA2	Mirror value of SyncManager 2 Status Indication 0: No Sync channel 2 event 1: Sync channel 2 event pending	R
7	SMSTA3	Mirror value of SyncManager 3 Status Indication 0: No Sync channel 3 event 1: Sync channel 3 event pending	R
8	SMSTA4	Mirror value of SyncManager 4 Status Indication 0: No Sync channel 4 event 1: Sync channel 4 event pending	R
9	SMSTA5	Mirror value of SyncManager 5 Status Indication 0: No Sync channel 5 event 1: Sync channel 5 event pending	R
10	SMSTA6	Mirror value of SyncManager 6 Status Indication 0: No Sync channel 6 event 1: Sync channel 6 event pending	R
11	SMSTA7	Mirror value of SyncManager 7 Status Indication 0: No Sync channel 7 event 1: Sync channel 7 event pending	R
15:12	—	The read values are undefined.	R

The ECAT_EVENT_REQ register indicates the source of ECAT event requests (ECAT interrupts).

30.11.4 AL_EVENT_REQ : AL Event Request Register

Base address: ESC = 0x8013_0000

Offset address: 0x0220

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SMINT7	SMINT6	SMINT5	SMINT4	SMINT3	SMINT2	SMINT1	SMINT0	—	WDPD	—	SYNCACT	DCSYNC1STA	DCSYNC0STA	DCLATCH	ALCTRL
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ALCTRL	AL Control Event State Indication This bit is cleared by reading the AL control register (AL_CONTROL at 0x0120 or 0x0121) from the PDI. 0: No change in the AL control register 1: The AL control register has been written	R
1	DCLATCH	DC Latch Event State Indication This bit is cleared by reading DC latch event times from the PDI for PDI controlled latch units, so the latch 0 and 1 status registers (DC_LATCH_STAT0 and DC_LATCH_STAT1 at 0x09AE and 0x09AF) indicate no event. 0: No change on DC latch inputs 1: At least one change on DC latch inputs	R
2	DCSYNC0STA	DC SYNC0 State Indication This bit is cleared by reading the SYNC0 status register (DC_SYNC0_STAT at 0x098E) from the PDI.	R
3	DCSYNC1STA	DC SYNC1 State Indication This bit is cleared by reading the SYNC1 status register (DC_SYNC1_STAT at 0x098F) from the PDI.	R
4	SYNCACT	SyncManager Activation Indication Change of the SyncManager activation register (SMn.ACT at 0x0806 + 0x8 × n) This bit is cleared by reading SyncManager activation registers from the PDI. 0: No change in any SyncManager 1: At least one SyncManager changed	R
5	—	The read value is undefined.	R
6	WDPD	Watchdog Process Data Indication This bit is cleared by reading the watchdog status process data register (WDS_DATA at 0x0440) from the PDI. 0: Valid 1: Timeout	R
7	—	The read value is undefined.	R
8	SMINT0	SyncManager 0 interrupt (bit 0 or 1 of the SyncManager status register (0x0805)) 0: No SyncManager 0 interrupt 1: SyncManager 0 interrupt pending	R
9	SMINT1	SyncManager 1 interrupt (bit 0 or 1 of the SyncManager status register (0x080D)) 0: No SyncManager 1 interrupt 1: SyncManager 1 interrupt pending	R
10	SMINT2	SyncManager 2 interrupt (bit 0 or 1 of the SyncManager status register (0x0815)) 0: No SyncManager 2 interrupt 1: SyncManager 2 interrupt pending	R
11	SMINT3	SyncManager 3 interrupt (bit 0 or 1 of the SyncManager status register (0x081D)) 0: No SyncManager 3 interrupt 1: SyncManager 3 interrupt pending	R

Bit	Symbol	Function	R/W
12	SMINT4	SyncManager 4 interrupt (bit 0 or 1 of the SyncManager status register (0x0825)) 0: No SyncManager 4 interrupt 1: SyncManager 4 interrupt pending	R
13	SMINT5	SyncManager 5 interrupt (bit 0 or 1 of the SyncManager status register (0x082D)) 0: No SyncManager 5 interrupt 1: SyncManager 5 interrupt pending	R
14	SMINT6	SyncManager 6 interrupt (bit 0 or 1 of the SyncManager status register (0x0835)) 0: No SyncManager 6 interrupt 1: SyncManager 6 interrupt pending	R
15	SMINT7	SyncManager 7 interrupt (bit 0 or 1 of the SyncManager status register (0x083D)) 0: No SyncManager 7 interrupt 1: SyncManager 7 interrupt pending	R
31:16	—	The read values are undefined.	R

The AL_EVENT_REQ register indicates the source of AL event requests (PDI interrupts).

30.12 Error Counter Registers

30.12.1 RX_ERR_COUNTn : RX Error Counter n Register (n = 0 to 2)

Base address: ESC = 0x8013_0000

Offset address: 0x0300 + 0x2 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXERRCNT[7:0]								INVFRMCNT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	INVFRMCNT[7:0]	Invalid Frame Counter Value Indication Counter value of invalid frames for port n counting is stopped when 0xFF is reached. These bits are cleared if one of the RX error counters (RX_ERR_COUNTn, FWD_RX_ERR_COUNTn) is written. The access type from the EtherCAT master is R/W.	R
15:8	RXERRCNT[7:0]	RX Frame Error Counter Value Indication Counter value of RX errors for port n counting is stopped when 0xFF is reached. The number of RX errors of MII interface is counted. These bits are cleared if one of the RX error counters (RX_ERR_COUNTn, FWD_RX_ERR_COUNTn) is written. The access type from the EtherCAT master is R/W.	R

The RX_ERR_COUNTn register counts RX frame errors.

30.12.2 FWD_RX_ERR_COUNTn : Forwarded RX Error Counter n Register (n = 0 to 2)

Base address: ESC = 0x8013_0000

Offset address: 0x0308 + 0x1 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FWDERRCNT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	FWDERRCNT[7:0]	Forwarded Error Counter Value Indication Counter value of forwarded RX error frames for port n counting is stopped when 0xFF is reached. These bits are cleared if one of the RX error counters (RX_ERR_COUNTn, FWD_RX_ERR_COUNTn) is written. The access type from the EtherCAT master is R/W.	R

The FWD_RX_ERR_COUNTn register counts forwarded RX frame errors.

30.12.3 ECAT_PROC_ERR_COUNT : ECAT Processing Unit Error Counter Register

Base address: ESC = 0x8013_0000

Offset address: 0x030C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	EPUERRCNT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	EPUERRCNT[7:0]	Processing Unit Error Counter Value Indication ECAT processing unit error counter value counting is stopped when 0xFF is reached. This register counts errors of frames passing the processing unit. Writing to this register clears it. The access type from the EtherCAT master is R/W.	R

The ECAT_PROC_ERR_COUNT register counts frame errors passing the ECAT processing unit.

30.12.4 PDI_ERR_COUNT : PDI Error Counter Register

Base address: ESC = 0x8013_0000

Offset address: 0x030D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PDIERRCNT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PDIERRCNT[7:0]	PDI Error Counter Value Indication Counting is stopped when 0xFF is reached. Counting starts when an interface error occurs due to access to the PDI. Writing to this register clears it. The access type from the EtherCAT master is R/W.	R

The PDI_ERR_COUNT register counts PDI access errors.

30.12.5 LOST_LINK_COUNTn : Lost Link Counter n Register (n = 0 to 2)

Base address: ESC = 0x8013_0000

Offset address: 0x0310 + 0x1 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LOSTLINKCNT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	LOSTLINKCNT[7:0]	Lost Link Counter Value Indication Lost link counter value for port n counting is stopped when 0xFF is reached. Counting starts only when port loop is Auto or Auto-Close. Only lost links at open ports are counted. Writing to one of the lost link counter registers clears it. The access type from the EtherCAT master is R/W.	R

The LOST_LINK_COUNTn register counts lost links at port n.

30.13 Watchdog Registers

30.13.1 WD_DIVIDE : Watchdog Divider Register

Base address: ESC = 0x8013_0000

Offset address: 0x0400

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WDDIV[15:0]															
Value after reset:	0	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0

Bit	Symbol	Function	R/W
15:0	WDDIV[15:0]	Watchdog Clock Frequency Divisor Setting Set the frequency divisor of the clock to drive counting by the watchdog timer in units of ticks at 25 MHz. The clock that drives counting by the watchdog timer is obtained by dividing 25 MHz by the value in this register plus 2. The default value is 2498, which corresponds to a period of 100 µs. The access type from the EtherCAT master is R/W.	R

The WD_DIVIDE register is used to set the ratio for dividing 25 MHz to obtain the basic period for incrementing the watchdog timer.

30.13.2 WDT_PDI : Watchdog Time PDI Register

Base address: ESC = 0x8013_0000

Offset address: 0x0410

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WDTIMPDI[15:0]															
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0

Bit	Symbol	Function	R/W
15:0	WDTIMPDI[15:0]	Watchdog Overflow Time Setting Set the time until the PDI watchdog timer overflows as a number of times the watchdog is incremented. With the default values for these bits and the setting of the watchdog divider, the time for a single incrementation is 100 µs, so the watchdog timer overflows when 100 µs ×1000 = 100 ms elapsed. Setting these bits to 0 disables the watchdog timer. Access to the PDI restarts the watchdog timer. The access type from the EtherCAT master is R/W.	R

The WDT_PDI register is used to set the time until the PDI watchdog timer overflows.

30.13.3 WDT_DATA : Watchdog Time Process Data Register

Base address: ESC = 0x8013_0000

Offset address: 0x0420

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WDTIMPD[15:0]															
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0

Bit	Symbol	Function	R/W
15:0	WDTIMPD[15:0]	Watchdog Overflow Time Setting Set the time until the process data watchdog timer overflows as a number of times the watchdog is incremented. With the default values for these bits and the setting of the watchdog divider, the time for a single incrementation is 100 µs, so the watchdog timer overflows when 100 µs × 1000 = 100 ms elapsed. There is one watchdog for all SyncManagers. Setting these bits to 0 disables the watchdog timer. Access to the Watchdog Trigger Enable bit of SyncManager restarts the watchdog timer. The access type from the EtherCAT master is R/W.	R

The WDT_DATA register is used to set the time until the process data watchdog timer overflows.

30.13.4 WDS_DATA : Watchdog Status Process Data Register

Base address: ESC = 0x8013_0000

Offset address: 0x0440

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDSTAPD
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	WDSTAPD	Watchdog State Indication Indicates the state of the process data watchdog timer triggered by SyncManagers. Reading this register clears bit 6 of the AL Event Request register (AL_EVENT_REQ at 0x0220). 0: The timeout period of the process data watchdog timer elapses 1: The process data watchdog timer is active or disabled	R
15:1	—	The read values are undefined.	R

The WDS_DATA register indicates the state of the process data watchdog timer.

30.13.5 WDC_DATA : Watchdog Counter Process Data Register

Base address: ESC = 0x8013_0000

Offset address: 0x0442

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WDCNTPD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	WDCNTPD[7:0]	Watchdog Counter Value Indication Counter value of the process data watchdog timer counting stops when 0xFF is reached. Counting starts on a timeout of the process data watchdog timer. Writing to one of the watchdog counter registers (WDC_DATA, WDC_PDI at 0x0442 and 0x0443) clears the counter. The access type from the EtherCAT master is R/W.	R

The WDC_DATA register indicates the timeout counter value of the process data watchdog timer.

30.13.6 WDC_PDI : Watchdog Counter PDI Register

Base address: ESC = 0x8013_0000

Offset address: 0x0443

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WDCNTPDI[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	WDCNTPDI[7:0]	Watchdog Counter Value Indication Counter value of the PDI watchdog timer counting stops when 0xFF is reached. Counting starts on a timeout of the PDI watchdog timer. Writing to one of the watchdog counter registers (WDC_DATA, WDC_PDI at 0x0442 and 0x0443) clears the counter. The access type from the EtherCAT master is R/W.	R

The WDC_PDI register indicates the timeout counter value of the PDI watchdog timer.

30.14 SII EEPROM Interface Registers

The EtherCAT controls the SII EEPROM interface if bit 0 is 0 in the EEPROM Configuration register (EEP_CONF at 0x0500) and in the EEPROM PDI Access State register (EEP_STATE at 0x0501). Otherwise, the PDI controls the EEPROM interface.

30.14.1 EEP_CONF : EEPROM Configuration Register

Base address: ESC = 0x8013_0000

Offset address: 0x0500

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	FORCEECAT	CTRLPDI
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	CTRLPDI	PDI EEPROM Control Specifies whether EEPROM control is offered to the PDI. The access type from the EtherCAT master is R/W. 0: The PDI has no EEPROM control 1: The PDI has EEPROM control	R
1	FORCEECAT	EEPROM Access Right Change Forcibly changes the right of access to the EEPROM by the ECAT. The access type from the EtherCAT master is R/W. 0: No change 1: Reset bit 0 of the EEPROM PDI Access State register (EEP_STATE at 0x0501) to 0. That is, prohibit access to the EEPROM by the PDI.	R

Bit	Symbol	Function	R/W
7:2	—	The read values are undefined.	R

The EEP_CONF register is used to configure EEPROM access.

30.14.2 EEP_STATE : EEPROM PDI Access State Register

Base address: ESC = 0x8013_0000

Offset address: 0x0501

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PDIACCESS
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	PDIACCESS	EEPROM Access Right Setting Write access from the PDI is only possible when bit 0 is 1 and bit 1 is 0 in the EEPROM Configuration register (EEP_CONF at 0x0500). The access type from the EtherCAT master is R. 0: Prohibits the PDI from access to the EEPROM 1: The PDI has access to the EEPROM	R/W
7:1	—	The read values are undefined. The write value should be 0. The access type from the EtherCAT master is R.	R/W

The EEP_STATE register is used to set the right of access to the EEPROM by the PDI.

30.14.3 EEP_CONT_STAT : EEPROM Control/Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x0502

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BUSY	WRENERR	ACKCMDERR	LOADSTA	CKSUMERR	COMMAND[2:0]			PROMSIZE	READBYTE	—	—	—	—	—	ECATWREN
Value after reset:	0	0	0	0	0	0	0	0	0/1*1	0	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	ECATWREN	ECAT Write Enable*3 This bit is always 1 if the PDI has EEPROM control. The access type from the EtherCAT master is R/W. 0: Write requests are disabled 1: Write requests are enabled	R
5:1	—	The read values are undefined. The write value should be 0.	R/W
6	READBYTE	EEPROM Read Byte Indication Indicates supported EEPROM read bytes. 0: 4 bytes 1: 8 bytes	R
7	PROMSIZE	EEPROM Algorithm Indication Indicates the selected EEPROM algorithm. 0: 1 address byte (1-Kbit to 16-Kbit EEPROMs) 1: 2 address bytes (32-Kbit to 4-Mbit EEPROMs)	R

Bit	Symbol	Function	R/W
10:8	COMMAND[2:0]	Command*3 Write: Initiates the commands below. Read: Indicates the currently executed command. Commands: 0 0 0: No command/EEPROM idle (clear error bits) 0 0 1: Read 0 1 0: Write 1 0 0: Reload Others: Reserved/invalid commands (must not be issued)	R/W
11	CKSUMERR	Checksum Error Indication Indicates checksum error in the ESC configuration area. 0: No error in the checksum 1: Error in the checksum	R
12	LOADSTA	EEPROM Loading Status Indication Indicates EEPROM loading status. 0: EEPROM has been loaded and device information has no problem 1: EEPROM has not been loaded and device information is not available (EEPROM loading in progress or finished with a failure)	R
13	ACKCMDERR	Acknowledge/Command Error Indication Indicates acknowledge/command error.*2 0: No error 1: Missing EEPROM acknowledge or invalid command	R
14	WRENERR	Write Enable Error Indication Indicates write enable error.*2 0: No error 1: Write command without write enable	R
15	BUSY	EEPROM Interface State Indication Indicates a busy state of the EEPROM interface. 0: The EEPROM interface is idle 1: The EEPROM interface is busy	R

Note: Write access depends on the assignment of the EEPROM interface (ECAT/PDI). Write access is generally blocked if the EEPROM interface is busy (bit 15 = 1).
Note 1. This initial value will be changed by setting of EEPROMSIZE bit in EtherCAT Operation Mode Register (ECATOPMOD).
Note 2. Error bits are cleared by writing 000b (or any valid command) to command bits [10:8].
Note 3. ECAT write enable bit 0 is self-cleared at the SOF of the next frame. Command bits [10:8] are also self-cleared after the command is executed (EEPROM busy ends).
Writing 000b to command bits [10:8] also clears the error bits 14 and 13. Command bits [10:8] are ignored if the acknowledge/command error bit 13 is 1.

The EEP_CONT_STAT register is used to control access to the EEPROM and indicate the status.

30.14.4 EEP_ADR : EEPROM Address Register

Base address: ESC = 0x8013_0000

Offset address: 0x0504

Bit position:	31																												0
Bit field:	ADDRESS[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ADDRESS[31:0]	EEPROM Address Setting Actually used EEPROM address bits: [9:0]: EEPROM size of up to 16 Kbits [17:0]: EEPROM size of 32 Kbits to 4 Mbits 0: First word (= 16 bits) 1: Second word ⋮	R/W

Note: Write access depends on the assignment of the EEPROM interface (ECAT/PDI). Write access is generally blocked if the EEPROM interface is busy (bit 15 is 1 in the EEPROM Control/Status register, EEP_CONT_STAT, at 0x0502).

The EEP_ADR register is used to set the EEPROM address to be accessed.

30.14.5 EEP_DATA : EEPROM Data Register

Base address: ESC = 0x8013_0000

Offset address: 0x0508

Bit position:	31																16	15														0
Bit field:	HIDATA[15:0]																LODATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	LODATA[15:0]	Data to be written to the EEPROM or data read from the EEPROM (lower 2 bytes)	R/W
31:16	HIDATA[15:0]	Data read from the EEPROM (upper 2 bytes)	R

Note: Write access depends on the assignment of the EEPROM interface (ECAT/PDI). Write access is generally blocked if the EEPROM interface is busy (bit 15 is 1 in the EEPROM Control/Status register, EEP_CONT_STAT, at 0x0502).

The EEP_DATA register is used to set write data to the EEPROM or indicates read data from the EEPROM. It can be written in 1-word units and read in 2-word units.

30.15 MII Management Interface Registers

30.15.1 MII_CONT_STAT : MII Management Control/Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x0510

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BUSY	CMDERR	READERR	—	—	—	COMMAND[1:0]		PHYOFFSET[4:0]					MILINK	PDICTRL	WREN
Value after reset:	0	0	0	x	x	x	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	WREN	Write Enable This bit is always 1 if the PDI controls the MII management interface. The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
1	PDICTRL	PDI Control Indication Indicates whether the MII management interface can be controlled by the PDI. The interface is controlled by the MII management ECAT Access State register (MII_ECAT_ACS_STAT at 0x0516) and the MII management PDI Access State register (MII_PDI_ACS_STAT at 0x0517). 0: Only ECAT control 1: PDI control possible	R
2	MILINK	MI Link Detection 0: Not available 1: Available	R
7:3	PHYOFFSET[4:0]	PHY Address Offset Indication Indicate the PHY address offset.	R

Bit	Symbol	Function	R/W
9:8	COMMAND[1:0]	Command Write: Initiates the commands below. Read: Indicates the currently executed command. Commands: 0 0: No command/MI idle (clear error bits) 0 1: Read 1 0: Write Others: Setting prohibited	R/W
12:10	—	The read values are undefined. The write value should be 0.	R/W
13	READERR	Read Error Indication Indicates whether a read error occurred. This bit is cleared by writing to this register. 0: No read error 1: Read error occurred (PHY or register not available)	R/W
14	CMDERR	Command Error Indication Indicates whether a command error occurred. This bit is cleared by executing a valid command or writing 00b to COMMAND[1:0]. 0: Last command was successful 1: Invalid command or write command without write enable	R
15	BUSY	MII Management State Indication Indicates that the MII management interface is busy. 0: MII management interface is idle 1: MII management interface is busy	R

The MII_CONT_STAT register is used to control the MII management interface and to indicate the status.

Write enable bit 0 is self-cleared at the SOF of the next frame. Command bits [9:8] are also self-cleared after the command is executed (busy ends). Writing 00b to command bits also clears the error bits 14 and 13. Command bits are cleared after the command is executed.

Write access depends on the assignment of the management interface (ECAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 = 1 in this register).

30.15.2 PHY_ADR : PHY Address Register

Base address: ESC = 0x8013_0000

Offset address: 0x0512

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PHYADDR[4:0]				
Value after reset:	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	PHYADDR[4:0]	PHY Address Setting	R/W
7:5	—	The read values are undefined.	R

The PHY_ADR register is used to set the PHY address.

Write access depends on the assignment of the management interface (ECAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 is 1 in the MII management control/status register, MII_CONT_STAT, at 0x0510).

30.15.3 PHY_REG_ADR : PHY Register Address Register

Base address: ESC = 0x8013_0000

Offset address: 0x0513

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PHYREGADDR[4:0]				
Value after reset:	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	PHYREGADDR[4:0]	Address of PHY register	R/W
7:5	—	The read values are undefined.	R/W

The PHY_REG_ADR register is used to set the PHY register address.

Write access depends on the assignment of the management interface (ECAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 is 1 in the MII management Control/Status register, MII_CONT_STAT, at 0x0510).

30.15.4 PHY_DATA : PHY Data Register

Base address: ESC = 0x8013_0000

Offset address: 0x0514

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PHYREGDATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PHYREGDATA[15:0]	PHY Register Data Indication/Setting	R/W

The PHY_DATA register is used to set data to write to PHY registers or to indicate data read from PHY registers.

Write access depends on the assignment of the management interface (ECAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 is 1 in the MII management Control/Status register, MII_CONT_STAT, at 0x0510).

30.15.5 MII_ECAT_ACS_STAT : MII Management ECAT Access State Register

Base address: ESC = 0x8013_0000

Offset address: 0x0516

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ACSMII
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	ACSMII	MII Management Interface Access Right Setting Right of access to the MII management interface The access type from the EtherCAT master is R/W. 0: Enables access to the MII management interface by the PDI. 1: Exclusive access to the MII management interface by the ECAT	R
7:1	—	The read values are undefined.	R

Note: Write access is only possible when bit 0 is 1 in the MII management PDI Access State register (MII_PDI_ACS_STAT at 0x0517).

The MII_ECAT_ACS_STAT register is used to set the right of access to the MII management interface.

30.15.6 MII_PDI_ACS_STAT : MII Management PDI Access State Register

Base address: ESC = 0x8013_0000

Offset address: 0x0517

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	FORPDI	ACSMII
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	ACSMII	Right of access to the MII management interface The access type from the EtherCAT master is R. 0: Access to the MII management interface by the ECAT 1: Access to the MII management interface by the PDI	R/W
1	FORPDI	Forced change of access by the PDI (forced change of bit 0) The access type from the EtherCAT master is R/W. 0: The value of bit 0 of this register is not changed 1: The value of bit 0 of this register is reset to 0 (the right of access is changed to the ECAT)	R
7:2	—	The read values are undefined. The access type from the EtherCAT master is R.	R/W

Note: Write access to bit 0 from the PDI is only possible if the following two conditions are satisfied.
- Bit 0 is 0 in the MII management ECAT Access State register (MII_ECAT_ACS_STAT at 0x0516).
- Bit 1 is 0 in the MII management PDI Access State register (MII_PDI_ACS_STAT at 0x0517).

The MII_PDI_ACS_STAT register is used to set the right of access to the MII management interface.

30.16 FMMU Registers

30.16.1 FMMUn_L_START_ADR : FMMU Logical Start Address n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0600 + 0x10 × n

Bit position:	31	0
Bit field:	LSTAADR[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	LSTAADR[31:0]	Logical Start Address Setting Set the start of the logical address within the EtherCAT address space. The access type from the EtherCAT master is R/W.	R

The FMMUn_L_START_ADR register is used to set the logical start address within the EtherCAT address space for FMMU.

30.16.2 FMMUn_LEN : FMMU Length n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0604 + 0x10 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FMMULEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	FMMULEN[15:0]	Area Size Specification Set the area size in byte units. Address at the end of the logical address range set by the FMMU minus the address at the start of the logical address range set by the FMMU plus 1. The access type from the EtherCAT master is R/W.	R

The FMMUn_LEN register is used to set the length for FMMU area in byte units.

30.16.3 FMMUn_L_START_BIT : FMMU Logical Start Bit n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0606 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	LSTABIT[2:0]		
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	LSTABIT[2:0]	Start Bit Setting Set the start bits of the logical start address for FMMU. The access type from the EtherCAT master is R/W.	R
7:3	—	The read values are undefined.	R

The FMMUn_L_START_BIT register is used to set the start bits of the logical start address for FMMU.

30.16.4 FMMUn_L_STOP_BIT : FMMU Logical Stop Bit n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0607 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	LSTPBIT[2:0]		
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	LSTPBIT[2:0]	Last Bit Setting Set the last bits of the logical end address for FMMU. The access type from the EtherCAT master is R/W.	R
7:3	—	The read values are undefined.	R

The FMMUn_L_STOP_BIT register is used to set the last bits of the logical end address for FMMU.

30.16.5 FMMUn_P_START_ADR : FMMU Physical Start Address n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0608 + 0x10 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PHYSTAADR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PHYSTAADR[15:0]	Physical Start Address Setting Set the physical start address to which the logical start address will be mapped. The address is set as an offset from the base address (0x8013_0000). The access type from the EtherCAT master is R/W.	R

The FMMUn_P_START_ADR register is used to set the physical start address of the ESC to which the logical start address will be mapped by the FMMU.

30.16.6 FMMUn_P_START_BIT : FMMU Physical Start Bit n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x060A + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PHYSTABIT[2:0]		
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	PHYSTABIT[2:0]	Physical Start Bit Setting Set the start bits of the physical start address to which the start bits of the logical start address will be mapped. The access type from the EtherCAT master is R/W.	R
7:3	—	The read values are undefined.	R

The FMMUn_P_START_BIT register is used to set the start bits of the physical start address of the ESC to which the start bits of the logical start address will be mapped by the FMMU.

30.16.7 FMMUn_TYPE : FMMU Type n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x060B + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	WRITE	READ
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	READ	Read Access Mapping Setting The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
1	WRITE	Write Access Mapping Setting The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
7:2	—	The read values are undefined.	R

The FMMUn_TYPE register is used to set the type of FMMU access.

30.16.8 FMMUn_ACT : FMMU Activate n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x060C + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ACTIVATE
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	ACTIVATE	FMMU Enable/Disable The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
7:1	—	The read values are undefined.	R

The FMMUn_ACT register is used to enable or disable FMMU.

30.17 SyncManager Registers

30.17.1 SMn_P_START_ADR : SyncManager Physical Start Address n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0800 + 0x8 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SMSTAADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SMSTAADDR[15:0]	Physical Start Address Setting Specify the physical start address for the area assigned to SyncManager. The access type from the EtherCAT master is R/W.	R

Note: The SMn_P_START_ADR register can only be written when SyncManager is disabled (bit 0 is 0 in the SyncManager Activate n register (SMn.ACT at 0x0806 + 0x8 × n).

This register is used to set the physical start address for the area assigned to SyncManager.

30.17.2 SMn_LEN : SyncManager Length n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0802 + 0x8 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SMLEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SMLEN[15:0]	Area Size Setting Set the number of bytes assigned to SyncManager. Set a value greater than 1. Otherwise, SyncManager is not activated. The access type from the EtherCAT master is R/W.	R

Note: This register can only be written when SyncManager is disabled (bit 0 is 0 in the SyncManager Activate n register (SMn.ACT at 0x0806 + 0x8 × n)).

The SMn_LEN register is used to set the size of the area assigned to SyncManager.

30.17.3 SMn_CONTROL : SyncManager Control n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0804 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	WDTRGEN	IRQPDI	IRQECAT	DIR[1:0]		OPEMODE[1:0]	
Value after reset:	x	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OPEMODE[1:0]	Operating Mode Setting The access type from the EtherCAT master is R/W. 0 0: Buffer mode (3-buffer mode) 1 0: Mailbox mode (single buffer mode) Others: Reserved	R
3:2	DIR[1:0]	Transfer Direction Setting The access type from the EtherCAT master is R/W. 0 0: Read (ECAT: read access; PDI: write access) 0 1: Write (ECAT: write access; PDI: read access) Others: Reserved	R
4	IRQECAT	ECAT Event Interrupt Setting Enables or disables interrupts (ECAT interrupts) by the ECAT Event Request register (ECAT_EVENT_REQ at 0x0210). The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
5	IRQPDI	AL Event Interrupt Setting Enables or disables interrupts (PDI interrupts) by the AL Event Request register (AL_EVENT_REQ at 0x0220). The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
6	WDTRGEN	Watchdog Trigger Setting The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
7	—	The read value is undefined.	R

Note: This register can only be written when SyncManager is disabled (bit 0 is 0 in the SyncManager Activate n register (SMn.ACT at 0x0806 + 0x8 × n)).

The SMn_CONTROL register is used to set operation of SyncManager.

30.17.4 SMn_STATUS : SyncManager Status n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0805 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WRBUF	RDBUF	BUFFERED[1:0]		MAILBOX	—	INTRD	INTWR
Value after reset:	0	0	1	1	0	x	0	0

Bit	Symbol	Function	R/W
0	INTWR	Write Complete Interrupt State Indication 0: Indicates that the buffer was successfully written 1: Indicates that the first byte of the buffer was read (interrupt cleared)	R
1	INTRD	Read Complete Interrupt State Indication 0: Indicates that the buffer was successfully read 1: Indicates that the first byte of the buffer was written (interrupt cleared)	R
2	—	The read value is undefined.	R
3	MAILBOX	Mailbox Status Indication This bit is not used in buffer mode. 0: Mailbox empty 1: Mailbox full	R
5:4	BUFFERED[1:0]	Buffer Status Indication Indicates the buffer status in buffer mode (last written buffer). This bit is not used in mailbox mode. 0 0: 1st buffer 0 1: 2nd buffer 1 0: 3rd buffer 1 1: No buffer written	R
6	RDBUF	Read State Indication Indicates that the buffer is being read.	R
7	WRBUF	Write State Indication Indicates that the buffer is being written.	R

The SMn_STATUS register indicates the state of SyncManager.

30.17.5 SMn_ACT : SyncManager Activate n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0806 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LATCHPDI	LATCHECAT	—	—	—	—	REPEATREQ	SMEN
Value after reset:	0	0	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	SMEN	SyncManager Enable/Disable The access type from the EtherCAT master is R/W. 0: Disabled. Memory is accessed without SyncManager control. 1: Enabled. SyncManager is active and controls memory area set in configuration.	R (ack)
1	REPEATREQ	Repeat Request Toggling of the repeat request signal means that retrying the mailbox is required (primarily used in conjunction with reading of the ECAT mailbox). The access type from the EtherCAT master is R/W.	R (ack)
5:2	—	The read values are undefined.	R
6	LATCHECAT	ECAT Latch Event Specification The access type from the EtherCAT master is R/W. 0: No latch events 1: Generates latch events if the EtherCAT master switches the buffers	R (ack)
7	LATCHPDI	PDI Latch Event Specification The access type from the EtherCAT master is R/W. 0: No latch events 1: Generates latch events if the PDI switches the buffers or accesses the buffer start address	R (ack)

Note: Reading this register from the PDI in all SyncManagers which have changed activation clears bit 4 of the AL Event Request register (AL_EVENT_REQ at 0x0220).

The SMn_ACT register is used to set operation of SyncManager.

30.17.6 SMn_PDI_CONT : SyncManager PDI Control n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0807 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	REPEATACK	DEACTIVE
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	DEACTIVE	SyncManager Operation Indication/Setting Deactivates SyncManager. Writing 1 is delayed until the end of a frame which is currently processed. The access type from the EtherCAT master is R. 0: Read: Normal operation. SyncManager is activated. Write: Activates SyncManager. 1: Read: SyncManager is deactivated and reset. SyncManager locks access to memory area. Write: Deactivates SyncManager.	R/W
1	REPEATACK	Repeat Acknowledge If this bit is set to the same value as set by bit 0 (repeat request) of the SyncManager activate register (SMn.ACT at 0x0806 + 0x8 × n), the PDI acknowledges the execution of a previous set repeat request. The access type from the EtherCAT master is R.	R/W
7:2	—	The read values are undefined. The write value should be 0.	R/W

The SMn_PDI_CONT register is used to set operation of SyncManager from the PDI.

30.18 Distributed Clock Registers

30.18.1 DC Receive Time Registers

30.18.1.1 DC_RCV_TIME_PORT0 : Receive Time Port 0 Register

Base address: ESC = 0x8013_0000

Offset address: 0x0900

Bit position:	31	0
Bit field:	RCVTIME0[31:0]	
Value after reset:	x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x	

Bit	Symbol	Function	R/W
31:0	RCVTIME0[31:0]	Receive Time Indication/Latch The time stamps cannot be read in the same frame in which this register was written. The access type from the EtherCAT master is R/W. Write: A write access to this register with the BWR or FPWR command latches the local time when each port starts to receive a frame (first start bit of preamble). Read: Indicate the local time at the beginning of reception of the last frame containing write access to this register.	R

Writing to the DC_RCV_TIME_PORT0 register latches the received time of frames at all ports and reading this register indicates the received time of a frame latched at port 0.

30.18.1.2 DC_RCV_TIME_PORT1 : Receive Time Port 1 Register

Base address: ESC = 0x8013_0000

Offset address: 0x0904

Bit position:	31																												0
Bit field:	RCVTIME1[31:0]																												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	RCVTIME1[31:0]	Receive Time Indication Indicate the local time when port 1 starts to receive a frame (first start bit of preamble) containing the BWR or FPWR command to the receive time port 0 register (DC_RCV_TIME_PORT0 at 0x0900).	R

The DC_RCV_TIME_PORT1 register indicates the received time of the frame latched at port 1.

30.18.1.3 DC_RCV_TIME_PORT2 : Receive Time Port 2 Register

Base address: ESC = 0x8013_0000

Offset address: 0x0908

Bit position:	31																												0
Bit field:	RCVTIME2[31:0]																												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	RCVTIME2[31:0]	Receive Time Indication Indicate the local time when port 2 starts to receive a frame (first start bit of preamble) containing the BWR or FPWR command to the receive time port 0 register (DC_RCV_TIME_PORT0 at 0x0900).	R

The DC_RCV_TIME_PORT2 register indicates the received time of the frame latched at port 2.

30.18.2 DC Time Loop Control Unit Registers

30.18.2.1 DC_SYS_TIME : System Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x0910

Bit position:	63																												32
Bit field:	SYSTIME[63:32]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31																												0
Bit field:	SYSTIME[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	SYSTIME[63:0]	System Time Indication Access from the ECAT (read-write) Read: Indicate the local copy of the system time when the frame passed the reference clock (such as including a system time delay). The time latched at the start of frame (SOF) is indicated. Write: A written value is compared with the local copy of the system time. The result is input to the time control loop unit. Access from the PDI (read-only) Indicate the local copy of the system time. The time latched when the first byte of this register was read is indicated.	R

The DC_SYS_TIME register indicates the local copy of the system time.

30.18.2.2 DC_RCV_TIME_UNIT : Receive Time ECAT Processing Unit Register

Base address: ESC = 0x8013_0000

Offset address: 0x0918

Bit position:	63																												32
Bit field:	RCVTIMEEPU[63:32]																												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	31																												0
Bit field:	RCVTIMEEPU[31:0]																												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
63:0	RCVTIMEEPU[63:0]	Receive Time Indication This register indicates the local time at the beginning of reception by the EtherCAT processing unit of a frame (such as the first start bit of the preamble), including write access to the Receive Time Port 0 register (DC_RCV_TIME_PORT0 at 0x0900).	R

Note: If port 0 is open, the value in this register reflects the same time as the value in the Receive Time Port 0 register (DC_RCV_TIME_PORT0 at 0x0900), but as 64 bits.

The DC_RCV_TIME_UNIT register indicates the received time of a frame latched at EtherCAT processing unit.

30.18.2.3 DC_SYS_TIME_OFFSET : System Time Offset Register

Base address: ESC = 0x8013_0000

Offset address: 0x0920

Bit position:	63																												32
Bit field:	SYSTIMOFST[63:32]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31																												0
Bit field:	SYSTIMOFST[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

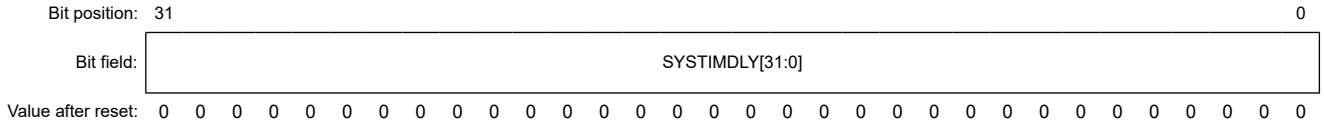
Bit	Symbol	Function	R/W
63:0	SYSTIMOFST[63:0]	System Time and Local Time Difference Indication Indicate a difference between the local time and system time. This offset is added to the local time to obtain the local system time. The access type from the EtherCAT master is R/W.	R

The DC_SYS_TIME_OFFSET register is used to indicate a difference (offset) between the local time and system time.

30.18.2.4 DC_SYS_TIME_DELAY : System Time Delay Register

Base address: ESC = 0x8013_0000

Offset address: 0x0928



Bit	Symbol	Function	R/W
31:0	SYSTIMDLY[31:0]	Propagation Delay Indication Indicate a delay between the reference clock and the ESC. The access type from the EtherCAT master is R/W.	R

The DC_SYS_TIME_DELAY register indicates a propagation delay between the reference clock and slave (ESC).

30.18.2.5 DC_SYS_TIME_DIFF : System Time Difference Register

Base address: ESC = 0x8013_0000

Offset address: 0x092C

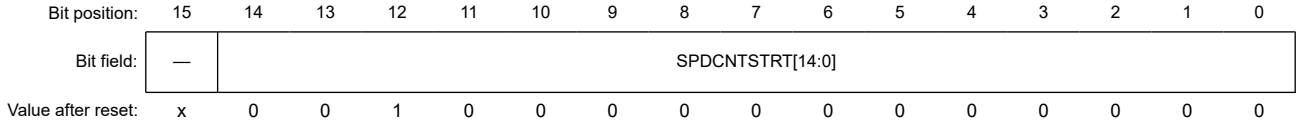
Bit	Symbol	Function	R/W
30:0	DIFF[30:0]	System Time Mean Difference Indication Indicates a mean difference between the local copy of the system time and received system time.	R
31	LCP	System Time Greater/Less Indication Indicates whether the local copy of the system time is greater than or equal to, or is less than, the latest received copy of the system time. 0: Local copy of the system time greater than or equal to the received system time 1: Local copy of the system time less than the received system time	R

The DC_SYS_TIME_DIFF register indicates a mean difference between the local copy of the system time and received system time.

30.18.2.6 DC_SPEED_COUNT_START : Speed Counter Start Register

Base address: ESC = 0x8013_0000

Offset address: 0x0930



Bit	Symbol	Function	R/W
14:0	SPDCNTSTRT[14:0]	Drift Correction Bandwidth Setting Indicate the bandwidth for adjustment of the local copy of the system time (larger values → smaller bandwidth and smoother adjustment). A write access resets the System Time Difference register (DC_SYS_TIME_DIFF at 0x092C) and the Speed Counter Difference register (DC_SPEED_COUNT_DIFF at 0x0932). Valid range: 0x0080 to 0x3FFF The access type from the EtherCAT master is R/W.	R
15	—	The read value is undefined. The access type from the EtherCAT master is R/W. In this case, the write value should be 0.	R

The DC_SPEED_COUNT_START register is used to set the bandwidth for drift correction of the local copy of the system time.

30.18.2.7 DC_SPEED_COUNT_DIFF : Speed Counter Difference Register

Base address: ESC = 0x8013_0000

Offset address: 0x0932

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPDCNTDIFF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SPDCNTDIFF[15:0]	Clock Period Deviation Indication Indicate the deviation between the local clock period and the clock period of the reference clock (represented by two's complements). Range: ± (speed counter start value - 0x7F)	R

The DC_SPEED_COUNT_DIFF register indicates the deviation between the local clock period and the clock period of the reference clock.

30.18.2.8 DC_SYS_TIME_DIFF_FIL_DEPTH : System Time Difference Filter Depth Register

Base address: ESC = 0x8013_0000

Offset address: 0x0934

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SYSTIMDEP[3:0]			
Value after reset:	x	x	x	x	0	1	0	0

Bit	Symbol	Function	R/W
3:0	SYSTIMDEP[3:0]	Filter Depth Setting Set the filter depth for averaging the received system time deviation. A write access resets the System Time Difference register (DC_SYS_TIME_DIFF at 0x092C). The access type from the EtherCAT master is R/W.	R
7:4	—	The read values are undefined. The access type from the EtherCAT master is R/W. In this case, the write value should be 0.	R

The DC_SYS_TIME_DIFF_FIL_DEPTH register is used to set the filter depth for averaging the received system time deviation.

30.18.2.9 DC_SPEED_COUNT_FIL_DEPTH : Speed Counter Filter Depth Register

Base address: ESC = 0x8013_0000

Offset address: 0x0935

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CLKPERDEP[3:0]			
Value after reset:	x	x	x	x	1	1	0	0

Bit	Symbol	Function	R/W
3:0	CLKPERDEP[3:0]	Filter Depth Setting Set the filter depth for averaging the clock period deviation. A write access resets the internal speed counter filter. The access type from the EtherCAT master is R/W.	R
7:4	—	The read values are undefined. The access type from the EtherCAT master is R/W. In this case, the write value should be 0.	R

The DC_SPEED_COUNT_FIL_DEPTH register is used to set the filter depth for averaging the clock period deviation.

30.18.3 Cyclic Unit Control Registers

30.18.3.1 DC_CYC_CONT : Cyclic Unit Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x0980

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	LATCH1	LATCH0	—	—	—	SYNCOUT
Value after reset:	x	x	0	0	x	x	x	0

Bit	Symbol	Function	R/W
0	SYNCOUT	SYNC Output Unit Control Setting The access type from the EtherCAT master is R/W. 0: ECAT control 1: PDI control	R
3:1	—	The read values are undefined.	R
4	LATCH0	Latch Input Unit 0 Control Setting Latch interrupt is routed to the ECAT or PDI in accordance with this setting. The access type from the EtherCAT master is R/W. 0: ECAT control 1: PDI control	R
5	LATCH1	Latch Input Unit 1 Control Setting Latch interrupt is routed to the ECAT or PDI in accordance with this setting. The access type from the EtherCAT master is R/W. 0: ECAT control 1: PDI control	R
7:6	—	The read values are undefined.	R

The DC_CYC_CONT register sets whether to control SYNC and latch units by the ECAT or PDI.

30.18.4 SYNC Output Unit Registers

30.18.4.1 DC_ACT : Activation Register

Base address: ESC = 0x8013_0000

Offset address: 0x0981

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DBGPULSE	NEARFUTURE	STARTTIME	EXTSTARTTIME	AUTOACT	SYNC1	SYNC0	SYNCACT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SYNCACT	Sync Output Unit Activation Activates the Sync output unit. Write 1 after the start time is written. 0: Deactivated 1: Activated	R/W
1	SYNC0	SYNC0 Output Setting Sets SYNC0 output. 0: Deactivated 1: SYNC0 pulse output is generated	R/W
2	SYNC1	SYNC1 Output Setting Sets SYNC1 output. 0: Deactivated 1: SYNC1 pulse output is generated	R/W
3	AUTOACT	SYNC Output Unit Activation Sets whether to activate the Sync output unit automatically by writing to the Start Time Cyclic Operation register (DC_CYC_START_TIME at 0x0990): 0: Deactivated 1: Activated. Bit 0 is automatically set to 1 in this register after the start time is written.	R/W
4	EXTSTARTTIME	Start Time Cyclic Operation Extension Extends start time cyclic operation. 0: No extension 1: Extends the start time written with 32 bits to 64 bits	R/W
5	STARTTIME	Start Time Plausibility Selects whether checking the plausibility of the start time and response to implausible start times is to proceed. 0: Disabled. Sync signal is generated if the start time is reached. 1: Sync signal is generated immediately if the start time is outside the range of the near future.	R/W
6	NEARFUTURE	Near Future Range Setting Sets the range to be considered the near future. 0: Up to 2^{63} ns from now (1/2 of the DC width) 1: Up to 2^{31} ns from now (about 2.1 sec.)	R/W
7	DBGPULSE	Debug Pulse Setting Sets Sync signal debug pulse. This bit is self-cleared and always read as 0. 0: Deactivated 1: Immediately generates a single debug ping on the SYNC0 and SYNC1 pins in accord with the setting of bits 2 and 1 of this register.	R/W

Note: Writing to this register depends on the setting of bit 0 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_ACT register is used to activate the Sync output unit.

30.18.4.2 DC_PULSE_LEN : SYNC Signal Pulse Length Register

Base address: ESC = 0x8013_0000

Offset address: 0x0982

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PULSELEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PULSELEN[15:0]	SYNC Signal Pulse Length Indication Indicate the pulse length of SYNC signals (in units of 10 ns) 0: Acknowledge mode. In this mode, SYNC signal is cleared by reading the SYNC0 or SYNC1 status register (DC_SYNC0/1_STAT at 0x098E, 0x098F).	R

The DC_PULSE_LEN register indicates the pulse length of SYNC signals.

30.18.4.3 DC_ACT_STAT : Activation Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x0984

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	START TIME	SYNC 1ACT	SYNC 0ACT
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
0	SYNC0ACT	SYNC0 Status Indication Indicates the activation state of SYNC0. 0: First SYNC0 pulse is not pending 1: First SYNC0 pulse is pending	R
1	SYNC1ACT	SYNC1 Status Indication Indicates the activation state of SYNC1. 0: First SYNC1 pulse is not pending 1: First SYNC1 pulse is pending	R
2	STARTTIME	Plausibility Result Indication Indicates the plausibility check result of the Start Time Cyclic Operation register (DC_CYC_START_TIME at 0x0990) while the Sync output unit is activated. 0: The start time is within the near future 1: The start time is out of the near future	R
7:3	—	The read values are undefined.	R

The DC_ACT_STAT register indicates the activation status of SYNC output signals.

30.18.4.4 DC_SYNC0_STAT : SYNC0 Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x098E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SYNC 0STA
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	SYNC0STA	SYNC0 State Indication Indicates the SYNC0 state for acknowledge mode. SYNC0 in acknowledge mode is cleared by reading this register from the PDI. This bit is only used in acknowledge mode.	R
7:1	—	The read values are undefined.	R

The DC_SYNC0_STAT register indicates the state of SYNC0 output. It is only used in acknowledge mode.

30.18.4.5 DC_SYNC1_STAT : SYNC1 Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x098F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SYNC 1STA
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	SYNC1STA	SYNC1 State Indication Indicates the SYNC1 state for acknowledge mode. SYNC1 in acknowledge mode is cleared by reading this register from the PDI. This bit is only used in acknowledge mode.	R
7:1	—	The read values are undefined.	R

The DC_SYNC1_STAT register indicates the state of SYNC1 output. It is only used in acknowledge mode.

30.18.4.6 DC_CYC_START_TIME : Start Time Cyclic Operation/Next SYNC0 Pulse Register

Base address: ESC = 0x8013_0000

Offset address: 0x0990

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	STATIM[63:32]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	STATIM[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	STATIM[63:0]	Start Time Setting/System Time Indication Write: Set the start time (in the system time) of cyclic operation in ns units. Read: Indicate the system time of the next SYNC0 pulse in ns units.	R/W

Note:
- Writing to this register depends on the setting of bit 0 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980). Only writable when bit 0 is 0 in the SYNC Activation register (DC_ACT at 0x0981).
- When auto-activation is enabled, the upper 32 bits are automatically extended if only the lower 32 bits are written within one frame.

Writing to this register sets the start time of cyclic operation. Reading this register indicates the system time of the next SYNC0 pulse.

30.18.4.7 DC_NEXT_SYNC1_PULSE : Next SYNC1 Pulse Register

Base address: ESC = 0x8013_0000

Offset address: 0x0998

Bit position:	63																												32
Bit field:	SYNC1PULSE[63:32]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	31																												0
Bit field:	SYNC1PULSE[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	SYNC1PULSE[63:0]	SYNC1 Pulse System Time Indication Indicate the system time of the next SYNC1 pulse in ns units.	R

The DC_NEXT_SYNC1_PULSE register indicates the system time of the next SYNC1 pulse.

30.18.4.8 DC_SYNC0_CYC_TIME : SYNC0 Cycle Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x09A0

Bit position:	31																												0
Bit field:	SYNC0CYC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SYNC0CYC[31:0]	Time Between Consecutive SYNC0 Pulses Set the time between two consecutive SYNC0 pulses in ns units. 0: Single-shot mode. Only one SYNC0 pulse is generated in single-shot mode.	R/W

Note: Writing to this register depends on the setting of bit 0 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_SYNC0_CYC_TIME register is used to set the time between consecutive SYNC0 pulses.

30.18.4.9 DC_SYNC1_CYC_TIME : SYNC1 Cycle Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x09A4

Bit position:	31																												0
Bit field:	SYNC1CYC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SYNC1CYC[31:0]	Time between SYNC1 and SYNC0 Pulses Set the time between SYNC1 and SYNC0 pulses in ns units.	R/W

Note: Writing to this register depends on the setting of bit 0 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_SYNC1_CYC_TIME register is used to set the time between SYNC1 and SYNC0 pulses.

30.18.5 Latch Input Unit Registers

30.18.5.1 DC_LATCH0_CONT : Latch 0 Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x09A8

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	NEGEDGE	POSEDGE
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	POSEDGE	Latch 0 Positive Edge Function Setting Sets the function of the rising edge of the latch 0 input signal. 0: Continuous latch active 1: Single event (only first event active)	R/W
1	NEGEDGE	Latch 0 Negative Edge Function Setting Sets the function of the falling edge of the latch 0 input signal. 0: Continuous latch active 1: Single event (only first event active)	R/W
7:2	—	The read values are undefined. The write value should be 0. The access type from the EtherCAT master is R.	R/W

Note: Writing to this register depends on the setting of bit 4 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_LATCH0_CONT register is used to control the edge function of the latch 0 input signal.

30.18.5.2 DC_LATCH1_CONT : Latch 1 Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x09A9

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	NEGEDGE	POSEDGE
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	POSEDGE	Latch 1 Positive Edge Function Setting Sets the function of the rising edge of the latch 1 input signal. 0: Continuous latch active 1: Single event (only first event active)	R/W
1	NEGEDGE	Latch 1 Negative Edge Function Setting Sets the function of the falling edge of the latch 1 input signal. 0: Continuous Latch active 1: Single event (only first event active)	R/W
7:2	—	The read values are undefined. The write value should be 0. The access type from the EtherCAT master is R.	R/W

Note: Writing to this register depends on the setting of bit 5 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_LATCH1_CONT register is used to control the edge function of the latch 1 input signal.

30.18.5.3 DC_LATCH0_STAT : Latch 0 Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x09AE

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PINSTATE	EVENTNEG	EVENTPOS
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
0	EVENTPOS	Latch 0 Positive Edge Event Indication Indicates detection of rising edges of the event latch 0 signal. This flag is cleared by reading the Latch 0 Time Positive Edge register (DC_LATCH0_TIME_POS at 0x09B0). 0: Rising edge not detected or continuous mode 1: Rising edge detected and mode is single-event	R
1	EVENTNEG	Latch 0 Negative Edge Event Indication Indicates detection of falling edges of the event latch 0 signal. This flag is cleared by reading the Latch 0 Time Negative Edge register (DC_LATCH0_TIME_NEG at 0x09B8). 0: Falling edge not detected or continuous mode 1: Falling edge detected and mode is single-event	R
2	PINSTATE	Latch 0 Input Pin State Indication Indicates the state of the latch 0 input pin.	R
7:3	—	The read values are undefined.	R

The DC_LATCH0_STAT register indicates the state of the latch 0 input signal.

30.18.5.4 DC_LATCH1_STAT : Latch 1 Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x09AF

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PINSTATE	EVENTNEG	EVENTPOS
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
0	EVENTPOS	Latch 1 Positive Edge Event Indication Indicates detection of rising edges of the event latch 1 signal. This flag is cleared by reading the Latch 1 Time Positive Edge register (DC_LATCH1_TIME_POS at 0x09C0). 0: Rising edge not detected or continuous mode 1: Rising edge detected and mode is single-event	R
1	EVENTNEG	Latch 1 Negative Edge Event Indication Indicates detection of falling edges of the event latch 1 signal. This flag is cleared by reading the Latch 1 Time Negative Edge register (DC_LATCH1_TIME_NEG at 0x09C8). 0: Falling edge not detected or continuous mode 1: Falling edge detected and mode is single-event	R
2	PINSTATE	Latch 1 Input Pin State Indication Indicates the state of the latch 1 input pin.	R
7:3	—	The read values are undefined.	R

The DC_LATCH1_STAT register indicates the state of the latch 1 input signal.

30.18.5.5 DC_LATCH0_TIME_POS : Latch 0 Time Positive Edge Register

Base address: ESC = 0x8013_0000

Offset address: 0x09B0

Bit position:	63																												32
Bit field:	SYSTIME[63:32]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31																												0
Bit field:	SYSTIME[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	SYSTIME[63:0]	System Time Indication Indicate the system time captured at the rising edge of the latch 0 input signal. Reading this register clears bit 0 of the Latch 0 Status register (DC_LATCH0_STAT at 0x09AE).	R

Note:
- Bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value
- Clearing the latch 0 status flag function depends on the setting of bit 4 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_LATCH0_TIME_POS register indicates the system time at the rising edge of the latch 0 input signal.

30.18.5.6 DC_LATCH0_TIME_NEG : Latch 0 Time Negative Edge Register

Base address: ESC = 0x8013_0000

Offset address: 0x09B8

Bit position:	63																												32
Bit field:	SYSTIME[63:32]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31																												0
Bit field:	SYSTIME[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	SYSTIME[63:0]	System Time Indication Indicate the system time captured at the falling edge of the latch 0 input signal. Reading this register clears bit 1 of the Latch 0 Status register (DC_LATCH0_STAT at 0x09AE).	R

Note:
- Bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value
- Clearing the latch 0 status flag function depends on the setting of bit 4 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_LATCH0_TIME_NEG register indicates the system time at the falling edge of the latch 0 input signal.

30.18.5.7 DC_LATCH1_TIME_POS : Latch 1 Time Positive Edge Register

Base address: ESC = 0x8013_0000

Offset address: 0x09C0

Bit position:	63																												32
Bit field:	SYSTIME[63:32]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	31																												0
Bit field:	SYSTIME[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	SYSTIME[63:0]	System Time Indication Indicate the system time captured at the rising edge of the latch 1 input signal. Reading this register clears bit 0 of the Latch 1 Status register (DC_LATCH1_STAT at 0x09AF).	R

Note:
- Bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value
- Clearing the latch 1 status flag function depends on the setting of bit 5 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_LATCH1_TIME_POS register indicates the system time at the rising edge of the latch 1 input signal.

30.18.5.8 DC_LATCH1_TIME_NEG : Latch 1 Time Negative Edge Register

Base address: ESC = 0x8013_0000

Offset address: 0x09C8

Bit position:	63																												32
Bit field:	SYSTIME[63:32]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	31																												0
Bit field:	SYSTIME[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	SYSTIME[63:0]	System Time Indication Indicate the system time captured at the falling edge of the latch 1 input signal. Reading this register clears bit 1 of the Latch 1 Status register (DC_LATCH1_STAT at 0x09AF).	R

Note:
- Bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value
- Clearing the latch 1 status flag function depends on the setting of bit 5 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_LATCH1_TIME_NEG register indicates the system time at the falling edge of the latch 1 input signal.

30.18.6 SyncManager Event Time Registers

30.18.6.1 DC_ECAT_CNG_EV_TIME : Buffer Change Event Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x09F0

Bit position:	31																												0
Bit field:	ECATCHANGE[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ECATCHANGE[31:0]	Local Time Indication Indicate the local time at the beginning of a frame which causes at least one SyncManager to generate an ECAT event (switching the buffers).	R

Note: Bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

The DC_ECAT_CNG_EV_TIME register indicates the local time at the beginning of a frame which causes SyncManager to generate an ECAT event (switching the buffers).

30.18.6.2 DC_PDI_START_EV_TIME : PDI Buffer Start Event Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x09F8

Bit position:	31																												0
Bit field:	PDISTART[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PDISTART[31:0]	Local Time Indication Indicate the local time when at least one SyncManager has generated a PDI event (access to the address where a buffer starts).	R

Note: Bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

The DC_PDI_START_EV_TIME register indicates the local time when SyncManager has generated a PDI event (access to the address where a buffer starts).

30.18.6.3 DC_PDI_CNG_EV_TIME : PDI Buffer Change Event Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x09FC

Bit position:	31																												0
Bit field:	PDICHANGE[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PDICHANGE[31:0]	Local Time Indication Indicate the local time when at least one SyncManager has generated a PDI event (switching the buffers).	R

Note: Bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

The DC_PDI_CNG_EV_TIME register indicates the local time when SyncManager has generated a PDI event (switching the buffers).

30.19 Other Registers

30.19.1 PRODUCT_ID : Product ID Register

Base address: ESC = 0x8013_0000

Offset address: 0x0E00

Bit position:	63																													32
Bit field:	PROID[63:32]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31																													0
Bit field:	PROID[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	PROID[63:0]	Product ID Indication	R

The PRODUCT_ID register indicates the product ID.

30.19.2 VENDOR_ID : Vendor ID Register

Base address: ESC = 0x8013_0000

Offset address: 0x0E08

Bit position:	63																													32
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	31																													0
Bit field:	VENDORID[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	VENDORID[31:0]	Vendor ID Indication	R
63:32	—	The read values are undefined.	R

The VENDOR_ID register indicates the vendor ID.

30.19.3 USER_RAM : User RAM

This area of RAM indicates the supported features dependent on the IP core configuration and takes up the 128 bits from 0x8013_0F80 to 0x8013_0FFF. An initial value of 1 means that the implementation of this module supports the corresponding feature, except in the case of bits [7:0], which indicate the number of bits defined in the user RAM and the initial value is 0x33 for this module.

Bit position	Description	Initial value
7:0	Number of bits for defining extended functionality. The value is 51 (0x33) for this module.	0x33
8	Extended DL Control register (0x0102, 0x0103)	1

Bit position	Description	Initial value
9	AL Status Code register (0x0134, 0x0135)	1
10	ECAT event mask (0x0200, 0x0201)	1
11	Configured station alias (0x0012, 0x0013)	1
12	General input (0x0F18, 0x0F1F)	0
13	General output (0x0F10, 0x0F17)	0
14	AL event mask (0x0204, 0x0207)	1
15	Physical read/write offset (0x0108, 0x0109)	1
16	Watchdog divider writable (0x0400, 0x0401) and watchdog PDI (0x0410, 0x0F11)	1
17	Watchdog counter (0x0442, 0x0443)	1
18	Write protection (0x0020, 0x0031)	1
19	Reset (0x0040, 0x0041)	1
20	Reserved	0
21	DC SyncManager event time (0x09F0, 0x09FF)	1
22	ECAT processing unit/PDI error counter (0x030C, 0x030D)	1
23	EEPROM size configurable (bit 7 at 0x0502) 0: EEPROM size fixed up to 16 Kbits 1: EEPROM size configurable	1
26:24	Reserved	0
27	Lost link counter (0x0310, 0x0313)	1
28	MII management interface (0x0510, 0x0515)	1
29	Enhanced link detection MII	1
30	Enhanced link detection EBUS	0
31	RUN LED	1
32	Link/activity LED	1
33	Reserved	0
35:34	Reserved	1
36	Reserved	0
37	Reserved	1
38	DC Time loop control assigned to PDI	0
39	Link detection and configuration by MI	0
40	MI control by PDI	1
41	Automatic TX shift	1
42	EEPROM emulation	0
49:43	Reserved	0
50	ERR LED, RUN/ERR LED override	1
Others	Reserved	Reserved

30.19.4 DATA_RAM : Process Data RAM

The process data RAM is used for process data and mailbox and takes up 8 KB from 0x8013_1000 to 0x8013_2FFF.

This RAM is only accessible when the EEPROM is correctly loaded (for example, when bit 0 is 1 in the ESC DL Status register, ESC_DL_STATUS, at 0x0110).

30.20 Setting the Module-Stop Function

EtherCAT modules are stopped in the initial state. If the modules are to be used, set the MSTPCRE.MSTPCRE02 bit for release from the module-stop state. Resetting the module-stop state is prohibited following release from the module-stop state. Operation of the modules if they are stopped following release and then released again is not guaranteed. Release from the module-stop state can proceed again after the modules have been returned to their initial states (stopped) by a reset.

30.21 Initial Settings

To initialize the EtherCAT, follow the procedure below:

- Set the offset address of the PHY module in the ECATOFFADR register.
- Set the size of the EEPROM in the ECATOPMOD register.
- Set the delay time of the TXEN and TXDn in the ECATDBGC register.
  It is recommended to use automatic TX shift function by connecting TXCLK from PHY to this LSI. In that case, manual delay time adjustment of the TXEN and TXDn is not basically needed.
- Release the EtherCAT from the module-stop state by using the MSTPCRE02 bit in the MSTPCRE register.
- Release the ESC and PHY modules from the reset state by using the MRCTLE03 and MRCTLE04 bits in the MRCTLE register.

On release from the reset state, the ESC will automatically load the EEPROM data and be activated.

30.22 Configuration of the Reset Circuit

Figure 30.2 shows the configuration of the reset circuit of the EtherCAT Slave Controller (ESC). When reset request by ESC_RESET_ECAT (0x0040) or ESC_RESET_PDI (0x0041) is received, ESC stops and reset output from ESC becomes high level. At the same time, ESC_RST interrupt is generated and ESC_RESETOUT# pin outputs low level.

To release reset state of ESC, MRCTLE04 bit of MRCTLE register must be changed from 0 → 1 → 0 after ESC_RST interrupt is detected. When reset input to ESC is deasserted, reset output from ESC becomes 0 and simultaneously ESC is rebooted with loading EEPROM. It takes around 1ms to finish EEPROM loading. Adjust clear timing of MRCTLE04 bit of MRCTLE register so that Ethernet PHY have enough reset duration. Figure 30.3 shows the timing diagram.

ECATRESOUT register can also control ESC_RESETOUT# pin independent of reset of ESC. When setting RESOUT_EN bit in ECATRESOUT register to 0, reset output from ESC is not output to ESC_RESETOUT# pin. When setting FORCE_RESET bit in ECATRESOUT register to 1, ESC_RESETOUT# pin goes low regardless the status of reset output from ESC.

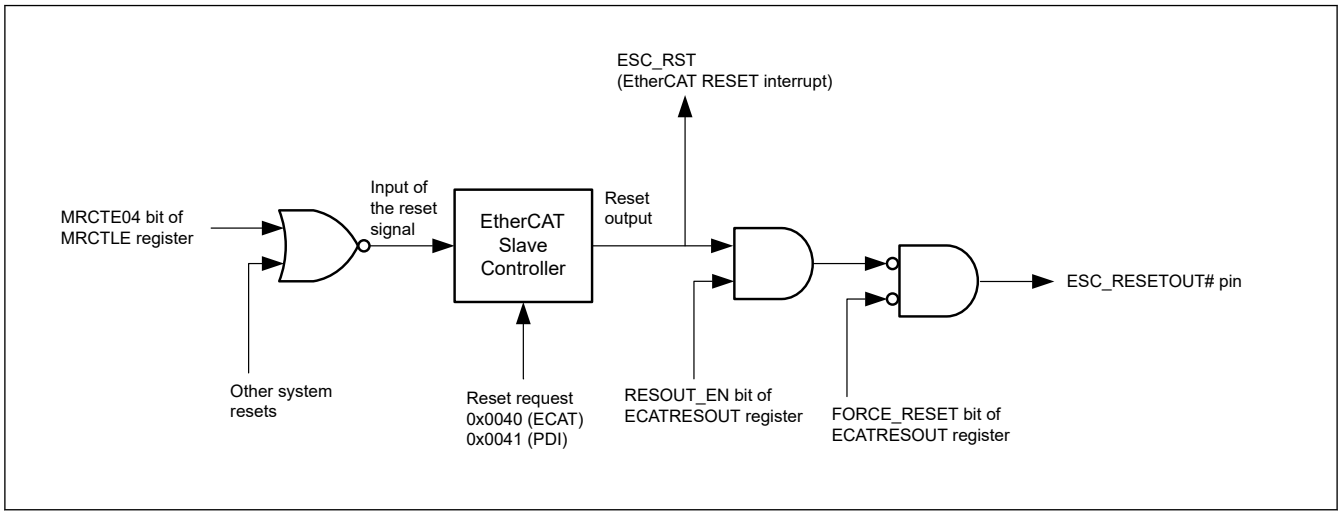
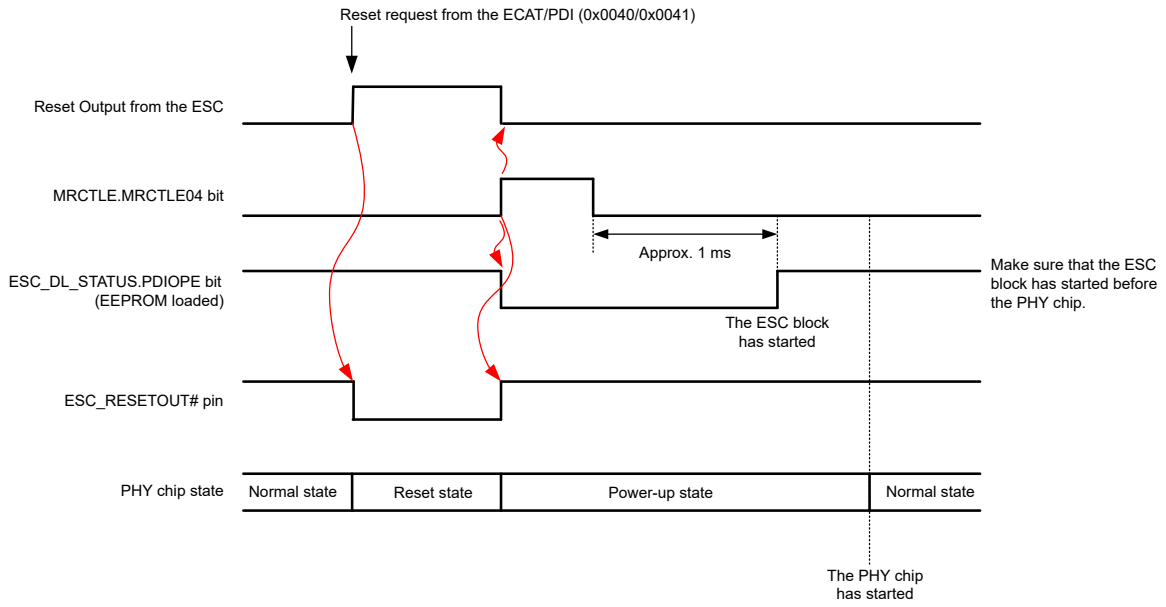


Figure 30.2 Configuration of the reset circuit of the EtherCAT slave controller



Note: ECASTRESOUT register is the initial value (0x0000_0002).

Figure 30.3 Reset Timing of EtherCAT Slave Controller

30.23 Usage Notes

1. Even if write protection is enabled, 0x0138 can be written.
2. When consecutive commands were issued from PDI to EEPROM, the latter command may not be accepted. Issue the commands with enough interval (e.g. 2 µs).
3. When VLAN tagged frame is used, ECAT reset register 0x0040 cannot be written.
4. ESC DL Control (0x0101) loop setting Auto-Close (01): if port waits for write access to be opened, the temporary loop bit (0x0100[1]) is not taken into account when a write command to ESC DL Control occurs.
5. Changing SyncSignal cycle times during activation may lead to extremely long cycle times. Additionally, changing cycle time from PDI may result in unwanted cycle times or extremely long delays. When cycle times are changed, disable SyncSignal.
6. DC Sync1 is missed once, when single shot is used, then DC SyncSignal generation deactivated, then cyclic generation activated.
7. Incorrect station alias address may be used when 0x0012 is changed and frame is received while change of 0x0013 is not completed yet.

31. USB 2.0 Host Module (USBHC)

31.1 Overview

The USB module of this LSI is a dual-role device that has USB 2.0 host and function modules.

The LSI handles a single USB port for either host controller or function controller operation. The Common Control register (COMMCTRL) is used to switch between host controller and function controller operation.

Since operation as a host controller or a function controller is exclusive of each other, dynamic switching between the types of operation is not possible. This section describes host controller operation.

The USB 2.0 host controller:

- Conforms to Universal Serial Bus Specification Revision 2.0
- Conforms to Open Host Controller Interface (OHCI) Specification for USB Rev. 1.0a
- Conforms to Enhanced Host Controller Interface (EHCI) Specification for USB Rev.1.1*1
- Supports USB 2.0 high-speed (480 Mbps), full-speed (12 Mbps), and low-speed (1.5 Mbps) transfer
- Supports the USB 2.0 compliance test function.

Note 1. Hardware prefetching is not supported.

Figure 31.1 shows a block diagram of the USB module.

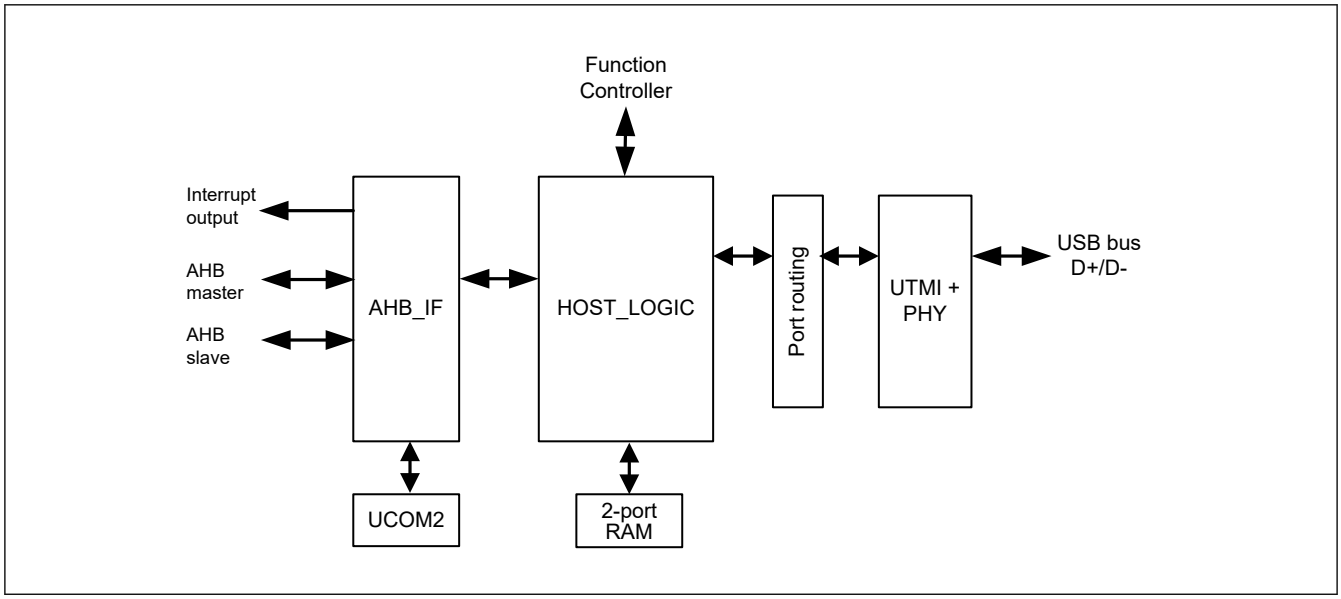


Figure 31.1 Block diagram of the USB module

(1) Host Logic

A USB 2.0 logic conforms to the EHCI and OHCI standards. It has control circuits such as a list processing circuit, a serial-parallel circuit, and a USB buffer. It executes high-speed, low-speed, and full-speed USB transfer.

(2) UTMI+ Transceiver

A USB 2.0 UTMI transceiver with an interface conforms to the UTMI+ standard.

31.1.1 Precautions on the Use of USB Host Controller

31.1.1.1 General Precautions

This section describes precautions on the use of the USB host controller.

1. Dynamic changes in PCLKAM and USB_CLK are not supported except for the clock stop state.

2. When using isochronous transfer, evaluate performance at the application level to confirm that the expected performance is obtained.
3. The USB host controller outputs the logical OR of multiple internal interrupt signals. See section 31.5. Interrupts.
4. It may take a long time to actually clear an interrupt after a register is accessed to clear the interrupt. For the action necessary during this wait time, see section 31.5.3. Time Required to Clear Interrupt Signals.

31.1.1.2 AHB Interface

The read or written data is not guaranteed for register access while the clock is stopped.

31.1.1.3 Operating Procedure

See the operating procedure described in section 31.7.1. Sample of Initial Settings.

31.2 Register Map

Table 31.1 USBHC register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x9204_0000	HCREVISION	HcRevision Register	—
0x9204_0004	HCCONTROL	HcControl Register	—
0x9204_0008	HCCOMMANDSTATUS	HcCommandStatus Register	—
0x9204_000C	HCINTERRUPTSTATUS	HcInterruptStatus Register	—
0x9204_0010	HCINTERRUPTENABLE	HcInterruptEnable Register	—
0x9204_0014	HCINTERRUPTDISABLE	HcInterruptDisable Register	—
0x9204_0018	HCHCCA	HcHCCA Register	—
0x9204_001C	HCPERIODCCURRENTIED	HcPeriodicCurrentED Register	—
0x9204_0020	HCCONTROLHEADED	HcControlHeadED Register	—
0x9204_0024	HCCONTROLCURRENTED	HcControlCurrentED Register	—
0x9204_0028	HCBULKHEADED	HcBulkHeadED Register	—
0x9204_002C	HCBULKCURRENTED	HcBulkCurrentED Register	—
0x9204_0030	HCDONEHEAD	HcDoneHead Register	—
0x9204_0034	HCFMINTERVAL	HcFmInterval Register	—
0x9204_0038	HCFNREMAINING	HcFmRemaining Register	—
0x9204_003C	HCFMNUMBER	HcFmNumber Register	—
0x9204_0040	HCPERIODSTART	HcPeriodicStart Register	—
0x9204_0044	HCLSTHRESHOLD	HcLSThreshold Register	—
0x9204_0048	HCRHDESCRIPTORA	HcRhDescriptorA Register	—
0x9204_004C	HCRHDESCRIPTORB	HcRhDescriptorB Register	—
0x9204_0050	HCRHSTATUS	HcRhStatus Register	—
0x9204_0054	HCRHPORTSTATUS1	HcRhPortStatus1 Register	—
0x9204_0100	CAPL_VERSION	Capability Registers Length and EHCI Version Number Register	—
0x9204_0104	HCSPARAMS	Structural Parameters Register	—
0x9204_0108	HCCPARAMS	Capability Parameters Register	—
0x9204_010C	HCSP_PORTROUTE	Companion Port Route Description Register	—
0x9204_0120	USBCMD	USB Command Register	—
0x9204_0124	USBSTS	USB Status Register	—
0x9204_0128	USBINTR	USB Interrupt Enable Register	—

Table 31.1 USBHC register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x9204_012C	FRINDEX	USB Frame Index Register	—
0x9204_0130	CTRLDSSEGMENT	Control Data Structure Segment Register	—
0x9204_0134	PERIODICLISTBASE	Periodic Frame List Base Address Register	—
0x9204_0138	ASYNCLISTADDR	Next Asynchronous List Address Register	—
0x9204_0160	CONFIGFLAG	Configure Flag Register	—
0x9204_0164	PORTSC1	Port 1 Status and Control Register	—
0x9204_0200	INTENABLE	INT_ENABLE Register	—
0x9204_0204	INTSTATUS	INT_STATUS Register	—
0x9204_0208	AHBBUSCTR	AHB_BUS_CTR Register	—
0x9204_020C	USBCTR	USBCTR Register	—
0x9204_0300	REVID	Revision and Core ID Register	—
0x9204_0310	OCSLPTIMSET	Overcurrent Detection/Sleep Timer Setting Register	—
0x9204_0800	COMMCTRL	Common Control Register	—
0x9204_0804	OBINTSTA	OTG-BC Interrupt Status Register	—
0x9204_0808	OBINTEN	OTG-BC Interrupt Enable Register	—
0x9204_080C	VBCTRL	VBUS Control Register	—
0x9204_0810	LINECTRL1	Line Control Port 1 Register	—
0x9204_0830	PHYCTRL	PHY Control Register	—

Table 31.2 USBHC related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
All the registers	—	MSTPCRE.MSTPCRE08	SLVACCCTL4.USB_SL

31.3 Register Descriptions

This section describes details of the register functions.

31.3.1 OHCI Operational Registers

Access the OHCI operational registers after starting up the PHY internal PLL. For details, see Figure 31.7.

31.3.1.1 HCREVISION : HcRevision Register

Base address: USBHC = 0x9204_0000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	REV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
7:0	REV[7:0]	HCI revision These bits indicate the version of the HCI specifications implemented in the host logic. As the host logic conforms to OHCI standard 1.0a, this value is set to 0x10.	R
31:8	—	These bits are read as 0.	R

31.3.1.2 HCCONTROL : HcControl Register

Base address: USBHC = 0x9204_0000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	RWE	RWC	IR	HCFS[1:0]		BLE	CLE	IE	PLE	CBSR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CBSR[1:0]	Control/bulk transfer service ratio (ControlBulkServiceRatio) **CBSR** / **Service ratio of bulk ED : Control ED** 00b / 1:1 01b / 2:1 10b / 3:1 11b / 4:1	R/W
2	PLE	Periodic list setting (PeriodicListEnable) This bit specifies whether to perform periodic list processing. The setting of this bit takes effect from the next frame. 0: Periodic list processing is not done 1: Period list processing is done	R/W
3	IE	Isochronous ED processing setting (IsochronousEnable) This bit specifies whether to perform isochronous ED processing. The setting of this bit takes effect from the next frame. When an isochronous ED is found during list processing, the host logic checks this bit to determine whether to perform isochronous ED processing. 0: Isochronous ED processing is enabled 1: Isochronous ED processing is disabled	R/W
4	CLE	Control list processing setting (ControlListEnable) This bit specifies whether to perform control list processing. The setting of this bit takes effect from the next frame. The control list can be modified only when this value is 0. 0: Control list processing is not done 1: Control list processing is done	R/W
5	BLE	Bulk list processing setting (BulkListEnable) This bit specifies whether to perform bulk list processing. The setting of this bit takes effect from the next frame. The bulk list can be modified only when this value is 0. 0: Bulk list processing is not done 1: Bulk list processing is done	R/W
7:6	HCFS[1:0]	Host logic operation status (Host Controller FunctionalState) These bits indicate the operating state of the host logic. On entering the USB operational state, the host logic starts management of frames delimited at 1-ms intervals. The operating state is always controlled by software except for transition to the USB resume state due to remote wakeup in the USB suspend state. These bits indicate the USB reset state after a hardware reset. After a software reset, the state shifts to the USB suspend state. 0 0: USB Reset 0 1: USB Resume 1 0: USB Operational 1 1: USB Suspend	R/W

Bit	Symbol	Function	R/W
8	IR	HcInterruptStatus interrupt path setting (InterruptRouting) This bit sets the output path for the interrupts indicated in the HcInterruptStatus register. The value set for this bit has no meaning because there is only one OHCI interrupt path for this host logic.	R/W
9	RWC	Remote Wakeup support setting (RemoteWakeUpConnect) This bit indicates whether the host logic supports remote wakeup. To support remote wakeup by the system, set this bit during initialization. 0: Remote Wakeup is not supported 1: Remote Wakeup is supported	R/W
10	RWE	PME assertion control (RemoteWakeUpEnable) This host logic does not use the PME assertion control bit. The value set for this bit has no meaning in this host logic.	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

The HCCONTROL register sets the operation mode for the OHCI host logic.

31.3.1.3 HCCOMMANDSTATUS : HcCommandStatus Register

Base address: USBHC = 0x9204_0000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	OCR	BLF	CLF	HCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HCR	Host logic software reset start (HostController Reset) This bit starts software reset of the host logic. When this bit is set, the USB suspend state is entered regardless of the functional state of the host logic.	W
1	CLF	Control list TD (ControlList Filled) This bit indicates whether TD exists in the control list. The host logic checks this bit when starting the processing of the start ED in the control list. When this bit is 0, the host logic does not begin list processing. When this bit is 1, the host logic starts control list processing and clears this bit to 0. On detecting TD, the host logic restores this bit to 1 and continues control list processing. When the host logic completes list processing, it clears this bit to 0. If TD is not found in the list or software does not set this bit to 1, this bit remains at 0 and list processing stops. When creating a list again and executing processing of the list, set this bit before starting list processing by setting bit 4 (CLE) in the HcControl register. 0: TD exists in the control list 1: TD does not exist in the control list	R/W
2	BLF	Bulk list TD (BulkListFilled) This bit indicates whether TD exists in the bulk list. The host logic checks this bit when starting the processing of the start ED in the bulk list. When this bit is 0, the host logic does not begin list processing. When this bit is 1, the host logic starts bulk list processing and clears this bit to 0. Upon detecting TD, the host logic restores this bit to 1 and continues bulk list processing. When the host logic completes list processing, it clears this bit to 0. If TD is not found in the list or software does not set this bit to 1, this bit remains at 0 and list processing stops. When creating a list again and executing processing of the list, set this bit before starting list processing by setting bit 5 (BLE) in the HcControl register. 0: TD exists in the bulk list 1: TD does not exist in the bulk list	R/W

Bit	Symbol	Function	R/W
3	OCR	Host logic control right change (OwnershipChangeRequest) This host logic does not use this bit, which changes the control right of the host logic. The value set for this bit has no meaning in this host logic.	W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
17:16	SOC[1:0]	Schedule overrun count (Scheduling OverrunCount) These bits count the number of times a schedule overrun occurs. This value is incremented every time a schedule overrun occurs. When the value reaches 11b, it returns to 00b. Counting continues even while bit 0 (SO) in the HcInterruptStatus register is set.	R
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The HCCOMMANDSTATUS register is used when the host logic receives commands from software or checks the current host logic status.

31.3.1.4 HCINTERRUPTSTATUS : HcInterruptStatus Register

Base address: USBHC = 0x9204_0000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	OC	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RHSC	FNO	UE	RD	SF	WDH	SO
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SO	USB schedule overrun (Scheduling Overrun) This is an interrupt bit to indicate that the USB schedule has overrun in a frame. When the USB schedule overruns, this bit is set after HccaFrameNumber for the next frame is updated. When this bit is set, bits [17:16] (SOC) in the HcCommandStatus register are also incremented. Writing 1 to this bit clears the interrupt. 0: No SO interrupt has occurred 1: SO interrupt has occurred	R/W
1	WDH	Host logic HccaDoneHead update (Writeback Done Head) This is an interrupt bit to indicate that the host logic has updated the contents of HccaDoneHead. The host logic sets this bit immediately after updating HccaDoneHead, and does not update HccaDoneHead until this bit is cleared. Writing 1 to this bit clears the interrupt. 0: No WDH interrupt has occurred 1: WDH interrupt has occurred	R/W
2	SF	HccaFrameNumber update (StartOfFrame) This is an interrupt bit to indicate that HccaFrameNumber has been updated at the beginning of a frame. The host logic updates HccaFrameNumber and sets this bit when transmitting an SOF packet. Writing 1 to this bit clears the interrupt. 0: No SF interrupt has occurred 1: SF interrupt has occurred	R/W

Bit	Symbol	Function	R/W
3	RD	Resume detection (Resume Detected) This is an interrupt bit to indicate that Resume has been detected. This bit is set when assertion of the Resume signal by a device on the USB is detected. This bit is not set when software issues USB Resume. Writing 1 to this bit clears the interrupt. 0: No RD interrupt has occurred 1: RD interrupt has occurred	R/W
4	UE	USB non-related system error detection (Unrecoverable Error) This is an interrupt bit to indicate that a system error on the PCI bus that is not related to the USB has been detected. Writing 1 to this bit clears the interrupt. 0: No UE interrupt has occurred 1: UE interrupt has occurred	R/W
5	FNO	FrameNumber bit MSB change (Frame Number Overflow) This is an interrupt bit to indicate that the MSB of bits [15:0] (FrameNumber) in the HcFmNumber register has changed. This bit is set after HccaFrameNumber is updated in the frame where the MSB of the FrameNumber bits changes from 0 to 1 or from 1 to 0. Writing 1 to this bit clears the interrupt. 0: No FNO interrupt has occurred 1: FNO interrupt has occurred	R/W
6	RHSC	HcRhStatus/HcRhPortStatus register status (RootHubStatus Change) This is an interrupt bit to indicate that the state of the HcRhStatus register or the HcRhPortStatus register has changed. This bit is set when the HcRhStatus or HcRhPortStatus register state is changed by a hardware source. Writing 1 to this bit clears the interrupt. 0: No RHSC interrupt has occurred 1: RHSC interrupt has occurred	R/W
29:7	—	These bits are read as 0. The write value should be 0.	R/W
30	OC	Host logic control right change (OwnershipChange) This is an interrupt bit to indicate that bit 3 (OCR) in the HcCommandStatus register is set (the OCR bit is not used in this host logic). The value set for this bit has no meaning in this host logic.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The HCINTERRUPTSTATUS register indicates the states of events that trigger interrupts.

If an event occurs, the host logic sets 1 to the corresponding bit. When 1 is set to a bit in this register while 1 is set to the corresponding interrupt enable bit and the MasterInterruptEnable bit in the HcInterruptEnable register, an interrupt is generated.

31.3.1.5 HCINTERRUPTENABLE : HcInterruptEnable Register

Base address: USBHC = 0x9204_0000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MIE	OCE	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RHSCE	FNOE	UEE	RDE	SFE	WDHE	SOE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SOE	SO interrupt source enable (Scheduling OverrunEnable) This bit enables SO as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: SO interrupt is enabled	R/W
1	WDHE	WDH interrupt source enable (WritebackDone HeadEnable) This bit enables WDH as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done. 1: WDH interrupt is enabled	R/W
2	SFE	SF interrupt source enable (StartOfFrame) This bit enables SF as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: SF interrupt is enabled	R/W
3	RDE	RD interrupt source enable (Resume DetectedEnable) This bit enables RD as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: RD interrupt is enabled	R/W
4	UEE	UE interrupt source enable (Unrecoverable ErrorEnable) This bit enables UE as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: UE interrupt is enabled	R/W
5	FNOE	FNO interrupt source enable (FrameNumber OverflowEnable) This bit specifies FNO as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: FNO interrupt is enabled	R/W
6	RHSCE	RHSC interrupt source enable (RootHubStatus ChangeEnable) This bit enables RHSC as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: RHSC interrupt is enabled	R/W
29:7	—	These bits are read as 0. The write value should be 0.	R/W
30	OCE	OC interrupt source enable (OwnershipChangeEnable) This bit enables OC as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: OC interrupt is enabled	R/W
31	MIE	Interrupt 8 source enable (MasterInterrupt Enable) This bit enables interrupt source settings in bits [30:0] in this register. When this bit is 0, all interrupts are masked. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: The specified interrupt is enabled	R/W

The HCINTERRUPTENABLE register controls whether to generate an interrupt for each event. The position of each enable bit in this register is the same as the corresponding interrupt bit in the HcInterruptStatus register.

An interrupt is generated when all the following conditions are satisfied:

1. The corresponding bit in the HcInterruptStatus register is set to 1.

2. The corresponding bit in the HcInterruptEnable register is set to 1.
3. The MasterInterruptEnable (MIE) bit in the HcInterruptEnable register is set to 1.

Note: Writing 0 to any of the bits in this register does not change the read value.

31.3.1.6 HCINTERRUPTDISABLE : HcInterruptDisable Register

Base address: USBHC = 0x9204_0000

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MID	OCD	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RHSCD	FNOD	UED	RDD	SFD	WDHD	SOD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SOD	SO interrupt source disable (Scheduling Overrun Disable) This bit disables the interrupt source SO. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source SO is disabled	R/W
1	WDHD	WDH interrupt source disable (Writeback DoneHead Disable) This bit disables the interrupt source WDH. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source WDH is disabled	R/W
2	SFD	SF interrupt source disable (StartOfFrame Disable) This bit disables the interrupt source SF. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source SF is disabled	R/W
3	RDD	RD interrupt source disable (Resume Detected Disable) This bit disables the interrupt source RD. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source RD is disabled.	R/W
4	UED	UE interrupt source disable (Unrecoverable ErrorDisable) This bit disables the interrupt source UE. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source UE is disabled	R/W
5	FNOD	FNO interrupt source disable (FrameNumberOverflow Disable) This bit disables the interrupt source FNO. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source FNO is disabled	R/W

Bit	Symbol	Function	R/W
6	RHSCD	RHSC interrupt source disable (RootHub StatusChange Disable) This bit disables the interrupt source RHSC. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source RHSC is disabled	R/W
29:7	—	These bits are read as 0. The write value should be 0.	R/W
30	OCD	OC interrupt source disable (OwnershipChangeDisable) 0: Nothing is to be done 1: OC is deleted from the interrupt sources	R/W
31	MID	Interrupt 8 source disable (Master Interrupt Disable) This bit disables interrupt source settings in bits [30:0] in HcInterruptEnable. When this bit is 0, all interrupts are masked. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt set is disabled	R/W

The position of each disable bit in this register is the same as the corresponding interrupt bit in the HcInterruptStatus register.

The HCINTERRUPTDISABLE register is connected to the HcInterruptEnable register. When 1 is set to a bit in this register, the corresponding bit in the HcInterruptEnable register is cleared.

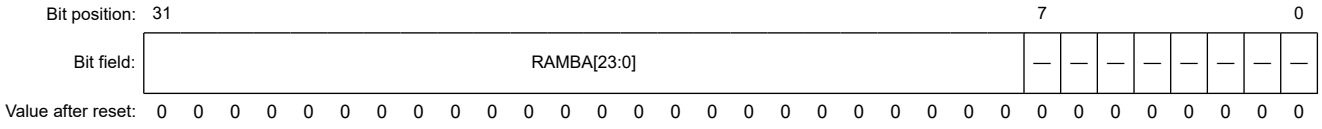
The read values in this register are the current values in the HcInterruptEnable register.

Note: Writing 0 to any of the bits in this register does not change the value of the corresponding bit in the HcInterruptEnable register.

31.3.1.7 HCHCCA : HcHCCA Register

Base address: USBHC = 0x9204_0000

Offset address: 0x018



Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
31:8	RAMBA[23:0]	RAM base address setting These bits specify the base address of the RAM allocated to the host controller communication area. Specify a value in these bits during initialization. The host logic requests a 256-byte area starting from the base address specified in these bits as HCCA.	R/W

The HCHCCA register indicates the physical address of the Host Controller Communication Area (referred to as HCCA hereafter).

Bits [7:0] of this register are fixed to 0x00 because the HCCA is allocated in units of 256 bytes.

31.3.1.8 HCPERIODCCURRENTIED : HcPeriodicCurrentED Register

Base address: USBHC = 0x9204_0000

Offset address: 0x01C

Bit position:	31																								4			0
Bit field:	PCED[27:0]																								—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
31:4	PCED[27:0]	ED physical address (PeriodicCurrentED) These bits indicate the physical address of the current ED being processed in the periodic list. The host logic updates these bits when completing the ED list processing.	R

The HCPERIODCCURRENTIED register indicates the physical address of the current isochronous or interrupt endpoint descriptor being processed.

31.3.1.9 HCCONTROLHEADED : HcControlHeadED Register

Base address: USBHC = 0x9204_0000

Offset address: 0x020

Bit position:	31																								4			0
Bit field:	CHED[27:0]																								—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
31:4	CHED[27:0]	Start ED physical address (ControlHeadED) These bits specify the physical address of the start ED in the control list. To execute control transfer, specify these bits before setting the CLE bit in the HcControl register.	R/W

The HCCONTROLHEADED register indicates the physical address of the start endpoint descriptor in the control list.

31.3.1.10 HCCONTROLCURRENTED : HcControlCurrentED Register

Base address: USBHC = 0x9204_0000

Offset address: 0x024

Bit position:	31																								4			0
Bit field:	CCED[27:0]																								—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31:4	CCED[27:0]	ED physical address (ControlCurrentED) These bits indicate the physical address of the current ED being processed in the control list. When processing of the current ED is completed, the host logic moves on to the next ED. The host logic continues list processing until the end of the frame. When the host logic reaches the end of the list, it checks bit 1 (CLF) in the HcCommandStatus register. When the CLF bit is 1, the host logic copies the content of the CHED bits in the HcControlHeadED register to the CCED bits and clears the CLF bit. When the CLF bit is 0, the host logic does nothing. Software is permitted to update the HcControlCurrentED register only when bit 4 (CLE) in the HcControl register is cleared. When the CLE bit is 1, software is only allowed to read the value of the HcControlCurrentED register. To indicate the end of the list, 0x0 is initially set to this register.	R/W

The HCCONTROLCURRENTED register indicates the physical address of the current endpoint descriptor being processed in the control list.

31.3.1.11 HCBULKHEADED : HcBulkHeadED Register

Base address: USBHC = 0x9204_0000

Offset address: 0x028

Bit position:	31																									4			0
Bit field:	BHED[27:0]																									—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
31:4	BHED[27:0]	Start ED physical address (BulkHeadED) These bits specify the physical address of the start ED in the bulk list. To execute bulk transfer, specify these bits before setting bit 5 (BLE) in the HcControl register.	R/W

The HCBULKHEADED register indicates the physical address of the start endpoint descriptor in the bulk list.

31.3.1.12 HCBULKCURRENTED : HcBulkCurrentED Register

Base address: USBHC = 0x9204_0000

Offset address: 0x02C

Bit position:	31																									4			0
Bit field:	BCED[27:0]																									—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31:4	BCED[27:0]	ED physical address (BulkCurrentED) These bits indicate the physical address of the current ED being processed in the bulk list. When processing of the current ED is completed, the host logic moves on to the next ED. The host logic continues list processing until the end of the frame. When the host logic reaches the end of the list, it checks bit 1 (CLF) in the HcCommandStatus register. When the CLF bit is 1, the host logic copies the content of the BHED bits in the HcBulkHeadED register to the BCED bits and clears the CLF bit. When the CLF bit is 0, the host logic does nothing. Software is permitted to update the HcBulkCurrentED register only when bit 4 (CLE) in the HcControl register is cleared. When the CLE bit is 1, software is only allowed to read the value of the HcBulkCurrentED register. To indicate the end of the list, 0 is initially set to this register.	R/W

The HCBULKCURRENTED register indicates the physical address of the current endpoint descriptor being processed in the bulk list.

31.3.1.13 HCDONEHEAD : HcDoneHead Register

Base address: USBHC = 0x9204_0000

Offset address: 0x030

Bit position:	31																									4			0
Bit field:	DH[27:0]																									—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
31:4	DH[27:0]	HcDoneHead physical address (DoneHead) These bits indicate the physical address of HcDoneHead in the host logic. This is the physical address of the last completed TD to be added to the Done queue. When processing of the TD is completed, the host logic writes the content of the HcDoneHead register to the NextTD bit of the TD. At that time, the host logic overwrites the content of the HcDoneHead register with the address of the TD. After the host logic writes the content of this register to the HCCA, it sets 0 to this register. In addition, the host logic sets bit 1 (WDH) in the HcInterruptStatus register to 1.	R

The HCDONEHEAD register indicates the physical address of the last completed transfer descriptor (referred to as TD hereafter) to be added to the Done queue.

In normal operation, software is not required to read this register because the content of this register is periodically written to the HCCA.

31.3.1.14 HCFMINTERVAL : HcFmInterval Register

Base address: USBHC = 0x9204_0000

Offset address: 0x034

Bit position:	31	30														16		13												0
Bit field:	FIT	FSMPS[14:0]															—	—	FI[13:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1

Bit	Symbol	Function	R/W
13:0	FI[13:0]	Frame interval setting (FrameInterval) These bits specify the length (bit time) of a frame used for FS transfer. To conform to the USB standard (one frame = 1 ms), set these bits to 0x2EDF.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
30:16	FSMPS[14:0]	FS transfer packet maximum size setting (FSLagest DataPacket) These bits specify the maximum amount of data that can be transferred without generating a schedule overrun. This setting is compared with the current location in the frame to determine up to which data in the frame can be transferred. The maximum amount depends on the system bus performance, and it should be specified by the driver. The maximum value allowed for these bits is 0x2778. Do not set a value greater than 0x2778.	R/W
31	FIT	Frame synchronization (FrameInterval Toggle) This bit is used to match the frame settings between software and the host logic. When updating the FI bits, software should toggle this bit. When loading the FI bit value, the host logic copies the FIT value to bit 31 (FRT) in the HcFmRemaining register. Software can check if the new FI bit setting has been reflected by comparing the value specified in this bit when the FI bits are written to, and the value read from the FRT bit.	R/W

The HCFMINTERVAL register contains a 14-bit value that indicates the length of a frame (bit time) and a 15-bit value that indicates the maximum size of a packet that the host logic can send and receive at full speed without generating a schedule overrun.

By using the FI bit, adjustments are possible in units of frames.

31.3.1.15 HCFNREMAINING : HcFmRemaining Register

Base address: USBHC = 0x9204_0000

Offset address: 0x038

Bit position:	31														13														0
Bit field:	FRT	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FR[13:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1

Bit	Symbol	Function	R/W
13:0	FR[13:0]	Down counter frame (FrameRemaining) These are the register bits of a 14-bit down counter which counts down the current remaining time value during transmission of a frame. The value of these bits is decremented over time. When the value reaches 0x0000, these bits are reloaded with the frame interval value by copying this value from bits [13:0] (FI) in the HcFmInterval register, after which counting down is restarted.	R
30:14	—	These bits are read as 0.	R
31	FRT	Frame synchronization (FrameRemainingToggle) This bit is used to match the frame settings between software and the host logic. When the FR bits reach 0x0000, the host logic copies the value of bits [13:0] (FI) in the HcFmInterval register to the FR bits and also copies the value of bit 31 (FIT) in the HcFmInterval register to this bit. Software can check if the FI bit setting has been copied to the FR bits by comparing the values of the FIT bit and this bit.	R

The HCFNREMAINING register is a 14-bit down counter which counts down the current remaining time during transfer of a frame.

31.3.1.16 HCFMNUMBER : HcFmNumber Register

Base address: USBHC = 0x9204_0000

Offset address: 0x03C

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	FN[15:0]	Elapsed frame number (FrameNumber) These bits indicate the elapsed frame count. The count in these bits is incremented when the HcFmRemaining register is reloaded. When the value reaches 0xFFFF, it is rolled over to 0x0000. When the host logic enters the USB operational state, the value in these bits is automatically incremented. After the host logic increments the value in these bits at the frame boundary and sends an SOF packet, the value in these bits is written to the HCCA. However, that happens before the host logic reads the first ED of that frame. After writing to the HCCA, the host logic sets bit 2 (SF) in the HcInterruptStatus register.	R
31:16	—	These bits are read as 0.	R

The HCFMNUMBER register indicates a 16-bit counter value.

31.3.1.17 HCPERIODSTART : HcPeriodicStart Register

Base address: USBHC = 0x9204_0000

Offset address: 0x040

Bit position:	31																		13												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PS[13:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	PS[13:0]	Periodic list processing start time (PeriodicStart) These bits indicate the time when the host logic starts periodic list processing in a frame. Specify a value in these bits by software during initialization of the host logic. When the value of bits [13:0] (FR) in the HcFmRemaining register is greater than the value specified in these bits, non-periodic lists take priority over periodic lists. The OHCI standard recommends that this value be set to about 90% of the value of bits [13:0] (FI) in the HcFmInterval register. Therefore, the recommended value is 0x2A2F.	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The HCPERIODSTART register indicates the time (using 14 bits) when the host logic starts periodic list processing in a frame.

31.3.1.18 HCLSTHRESHOLD : HcLSThreshold Register

Base address: USBHC = 0x9204_0000

Offset address: 0x044

Bit position:	31																				11										0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LS[11:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0

Bit	Symbol	Function	R/W
11:0	LS[11:0]	Transferrable threshold (LSThreshold) These bits specify the threshold value to determine whether LS transfer can be done in the remaining frame time. When the FmRemaining value is greater than this threshold value, LS transfer can be started.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The HCLSTHRESHOLD register contains the value of 12 bits that are used to determine whether the host logic can perform LS transfer in the remaining frame time.

31.3.1.19 HCRHDESCRIPTORA : HcRhDescriptorA Register

Base address: USBHC = 0x9204_0000

Offset address: 0x048

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	POTPGT[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	NOCP	OCPM	DT	NPS	PSM	NDP[7:0]							
Value after reset:	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
7:0	NDP[7:0]	Downstream port number (NumberDownstreamPorts) These bits specify the number of downstream ports supported by the root hub in the host logic.	R
8	PSM	Power switch control (PowerSwitchingMode) This bit specifies how to control the power switch to each port of the root hub. While bits [31:16] (PPCM) in the HcRhDescriptorB register are set, each port responds only to Set/ClearPortPower. While the bits are cleared, each port is controlled by Set/ClearGlobalPower. This bit setting is valid only when the NPS bit is cleared. 0: Power to all ports is controlled together 1: Power to each port is separately controlled	R/W
9	NPS	Power control (NoPower Switching) This bit specifies how to control the port power. 0: Port power can be switched on and off 1: Port power is always turned on while the host logic is operating	R/W
10	DT	Device type (DeviceType) This bit indicates that the root hub is not a compound device. As the root hub is not allowed to be a compound device, this bit is always read as 0.	R
11	OCPM	Overcurrent state reporting (OverCurrentProtection Mode) This bit specifies how to report the overcurrent state of the root hub. This bit should be set to the same mode as that specified in the PSM bit. This setting is valid only when the NOCP bit is cleared. 0: Overcurrent state of all ports is collectively reported 1: Overcurrent state of each port is separately reported	R/W
12	NOCP	Overcurrent function support (NoOver Current Protection) This bit specifies whether to support the overcurrent protection function for the root hub. 0: Overcurrent protection function is supported 1: Overcurrent protection function is not supported	R/W
23:13	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31:24	POTPGT[7:0]	Wait time (PowerOnToPowerGood Time) These bits indicate the time that software should wait before accessing a root hub port after power supply to the port is started. The unit of time is 2 ms. Therefore, the wait time is obtained by POTPGT × 2 ms.	R/W

The HCRHDESCRIPTORA register is one of the registers that indicates the characteristics of the root hub.

31.3.1.20 HCRHDESCRIPTORB : HcRhDescriptorB Register

Base address: USBHC = 0x9204_0000

Offset address: 0x04C

Bit position:	31															16	15													0
Bit field:	PPCM[15:0]																DR[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DR[15:0]	Device Removable These bits indicate whether the device connected to each port in the root hub is removable. Each bit is dedicated to a single port. The USB host controller has port 1 only.	R/W
31:16	PPCM[15:0]	Port Power Control Mask These bits specify power control commands for each port. This setting is valid when bit 8 (PSM) in the HcRhDescriptorA register is set. Each bit is dedicated to a single port. The USB host controller has port 1 only.	R/W

Bit	Value	Description
DR[0]	—	Reserved
DR[1]	0	Device connected to port 1 is not removable.
	1	Device connected to port 1 is removable.
DR[15:2]	—	Reserved

Bit	Value	Description
PPCM[0]	—	Reserved
PPCM[1]	0	All ports are collectively controlled (Set/ClearGlobalPower).
	1	Port 1 is separately controlled (Set/ClearPortPower).
PPCM[15:2]	—	Reserved

The HCRHDESCRIPTORB register is one of the registers that indicate the characteristics of the root hub.

31.3.1.21 HCRHSTATUS : HcRhStatus Register

Base address: USBHC = 0x9204_0000

Offset address: 0x050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CRWE	—	—	—	—	—	—	—	—	—	—	—	—	—	OCIC	LPSC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DRWE	—	—	—	—	—	—	—	—	—	—	—	—	—	OCI	LPS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>0</td><td>LPS</td><td>Local power status (LocalPowerStatus)
The meaning of this bit differs when it is read and written.
When read: LocalPowerStatus
This product does not support the local power status function. This bit is always read as 0.
When written: ClearGlobalPower
Setting this bit to 1 turns off the power to a port.
The port to be turned off is determined according to the settings of bit 8 (PSM) in the HcRhDescriptorA register and bits [31:16] (PPCM) in the HcRhDescriptorB register.
<table>
<tr><th>Value</th><th>PSM</th><th>PPCM[1]</th><th>Description</th></tr>
<tr><td>0</td><td>—</td><td>—</td><td>No change</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>—</td><td>The PPS bit in the HcRhPortStatus1 register is cleared</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>The PPS bit in the HcRhPortStatus1 register is cleared</td></tr>
<tr><td>1</td><td>No change</td></tr>
</table>
Writing 0 has no effect.</td><td>R/W</td></tr>
<tr><td>1</td><td>OCI</td><td>Overcurrent indicator (OverCurrent Indicator)
This bit reports the overcurrent state in global overcurrent detection mode.
When overcurrent is reported separately for each port, this value is always 0.
0: Port is in normal state
1: Port is in overcurrent state</td><td>R</td></tr>
<tr><td>14:2</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>15</td><td>DRWE</td><td>Device remote start enable (DeviceRemoteWakeupEnable)
When written: SetRemoteWakeupEnable
This bit is used to set the DRWE bit.
Setting this bit sets the DRWE bit.
Writing 0 has no effect.
When read: DeviceRemoteWakeupEnable
This bit specifies whether to include bit 16 (CSC) in the HcRhPortStatus1 register as a remote wakeup event.
When a connect status change event occurs while this bit is set, transition from the USB suspend state to the USB resume state occurs and the resume detect interrupt is generated.
0: ConnectStatusChange is not a RemoteWakeup source
1: ConnectStatusChange is a RemoteWakeup source</td><td>R/W</td></tr>
<tr><td>16</td><td>LPSC</td><td>Local power status change (LocalPowerStatusChange)
The meaning of this bit differs when it is read and written.
When read: LocalPowerStatusChange
This product does not support the local power status function and this bit is always read as 0.
When written: SetGlobalPower
Setting this bit to 1 turns on the power to a port.
The port to be turned on is determined according to the settings of bit 8 (PSM) in the HcRhDescriptorA register and bits [31:16] (PPCM) in the HcRhDescriptorB register.
<table>
<tr><th>Value</th><th>PSM</th><th>PPCM[1]</th><th>Description</th></tr>
<tr><td>0</td><td>—</td><td>—</td><td>No change</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>—</td><td>The PPS bit in the HcRhPortStatus1 register is set</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>The PPS bit in the HcRhPortStatus1 register is set</td></tr>
<tr><td>1</td><td>No change</td></tr>
</table>
Writing 0 has no effect.</td><td>R/W</td></tr>
<tr><td>17</td><td>OCIC</td><td>OCI bit change report (OverCurrent Indicate Change)
This bit notifies that the OCI bit value has changed.
It is set when the OCI bit changes.
Writing 1 to this bit while it is set to 1 clears it.
Writing 0 has no effect.
0: Nothing is to be done in case of an overcurrent
1: OverCurrent state has changed</td><td>R/W</td></tr>
<tr><td>30:18</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
</table>

Bit	Symbol	Function	R/W
31	CRWE	DRWE bit clear (Clear Remote Wakeup Enable) 0: No effect 1: Clears the DRWE bit	W

31.3.1.22 HCRHPORTSTATUS1 : HcRhPortStatus1 Register

Base address: USBHC = 0x9204_0000

Offset address: 0x054

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	PRSC	OCIC	PSSC	PESC	CSC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LSDA	PPS	—	—	—	PRS	POCI	PSS	PES	CCS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CCS	Connection status indication (CurrentConnectStatus) The meaning of this bit differs when it is read and written. When written: ClearPortEnable This bit is used to clear the PES bit. Writing 1 causes the port to enter the disabled state. Writing 0 has no effect. When read: CurrentConnectStatus This bit indicates the current connection state of the downstream port. 0: No device is connected 1: Device is connected	R/W
1	PES	Port enable status (PortEnableStatus) The meaning of this bit differs when it is read and written. When written: SetPortEnable This bit is used to set the PES bit. Writing 1 causes the port to enter the enabled state. Writing 0 has no effect. If this bit is written to while the CCS bit is cleared, this bit is not set but the CSC bit is set to notify the driver of an attempt to enable a disconnected port. When read: PortEnableStatus This bit indicates whether the port is enabled or disabled. The host logic automatically clears this bit on detecting the overcurrent state, disconnection, power-off, or a babble error. At this time, the PESC bit is set. This bit cannot be set while the CCS bit is cleared (no device is connected). The host logic sets this bit when a port reset is completed and the PRSC bit is set, or when a port enters the suspend state and the PRSC bit is set. 0: Port is disabled 1: Port is enabled	R/W
2	PSS	Suspend/Resume status (PortSuspendStatus) The meaning of this bit differs when it is read and written. When written: SetPortSuspend This bit is used to clear the PES bit. Writing 1 makes the port enter the disabled state. Writing 0 has no effect. If this bit is written to while the CCS bit is cleared, this bit is not set but the CSC bit is set to notify the driver of an attempt to suspend a disconnected port. When read: PortSuspendStatus This bit indicates that the port is in the suspend state or the resume sequence. This bit is not set while the CCS bit is cleared (no device is connected). This bit is cleared with the following timing: ● When the resume sequence is completed and the PSSC bit is set ● When a port reset is completed and the PRSC bit is set ● In the USB RESUME state 0: Port is in normal transfer state 1: Port is in suspend state	R/W

Bit	Symbol	Function	R/W
3	POCI	Downstream port overcurrent detection (PortOverCurrentIndicator) The meaning of this bit differs when it is read and written. When written: ClearSuspendStatus This bit terminates the suspend state and starts the resume sequence. Writing 1 starts the resume sequence. Writing 0 has no effect. The resume sequence starts only while the PSS bit is set. When read: PortOverCurrentIndicator This bit indicates that a downstream port has entered the overcurrent state. It is valid only while the overcurrent state is reported separately for each port (OCPM = 1 in the HcRhDescriptorA register). While the overcurrent state in all ports is collectively reported, this bit is 0b. 0: Port is in normal state 1: Port is in overcurrent state	R
4	PRS	Port reset status (PortResetStatus) The meaning of this bit differs when it is read and written. When written: SetPortReset This bit applies a port reset to a downstream port. Writing 1 to this bit starts a 10-ms port reset. Writing 0 has no effect. If this bit is written to while the CCS bit is cleared, this bit is not set but the CSC bit is set to notify software of an attempt to reset a port where no device is connected. When read: PortResetStatus This bit indicates the reset state of the port. It is cleared at the same time as when the PRSC bit is set after a 10-ms port reset is completed. This bit cannot be set while the CCS bit is cleared (no device is connected). 0: Port reset is not in progress 1: Port reset is in progress	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	PPS	Power status (PortPowerStatus) The meaning of this bit differs when it is read and written. When written: SetPortPower This bit turns the port power on when power is separately controlled for each port. Writing 1 turns the port on. Writing 0 has no effect. When read: PortPowerStatus This bit indicates the power state of the port. It is cleared when an overcurrent is detected. 0: Port power is off 1: Port power is on	R/W
9	LSDA	Device speed (LowSpeedDeviceAttached) The meaning of this bit differs when it is read and written. When written: ClearPortPower This bit turns off the port power when power is separately controlled for each port. Writing 1 turns the port off. Writing 0 has no effect. When read: LowSpeedDeviceAttached This bit indicates the speed of the device connected to the port. This status bit is valid only while the CCS bit is set. 0: FS device is connected 1: LS device is connected	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	CSC	CCS bit status (ConnectStatus Change) This bit indicates that the CCS bit value has changed. The host logic sets this bit when the CCS bit changes due to device connection or disconnection. In addition, when a request for port reset, port suspension, or port enable is issued in the disconnection state, this bit is set to make software re-evaluate the device connection state. It is cleared when 1 is written by software. 0: Nothing is to be done regarding CCS 1: CCS has changed	R/W
17	PESC	PES bit status (PortEnable StatusChange) This bit indicates that the PES bit value has changed. It is set when the port state has changed due to a hardware event, such as the overcurrent state, disconnection, power-off, or a babble error. It is cleared when 1 is written by software. 0: Nothing is to be done regarding PES 1: PES state has changed	R/W

Bit	Symbol	Function	R/W
18	PSSC	RESUME sequence complete (PortSuspend StatusChange) This bit indicates that the RESUME sequence has been completed. It is set when all RESUME processing by hardware has been completed. It is cleared when 1 is written by software. In addition, it is cleared when the PRSC bit is set. 0: RESUME has not been completed 1: RESUME has been completed	R/W
19	OCIC	Overcurrent state detection (OverCurrent IndicateChange) This bit is set when the overcurrent state of the port has been detected. It is cleared when 1 is written by software. It is valid only while the overcurrent state is reported separately for each port (OCPM = 1 in the HcRhDescriptorA register). 0: Nothing is to be done regarding overcurrent state 1: Overcurrent state has changed	R/W
20	PRSC	Port reset complete (PortReset StatusChange) This bit indicates that a port reset has been completed. It is set when the host logic has completed a 10-ms hardware reset. It is cleared when 1 is written by software. 0: Port reset has not been completed, or nothing is to be done regarding the PRS bit 1: Port reset has been completed	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The HCRHPORTSTATUS1 register is used to control and report port events.

31.3.2 EHCI Operational Registers

Access the EHCI operational registers after starting up the PHY internal PLL. For details, see Figure 31.7.

31.3.2.1 CAPL_VERSION : Capability Registers Length and EHCI Version Number Register

Base address: USBHC = 0x9204_0000

Offset address: 0x100

Bit position:	31																16								7								0
Bit field:	HCIVN[15:0]																—	—	—	—	—	—	—	—	CRL[7:0]								
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	

Bit	Symbol	Function	R/W
7:0	CRL[7:0]	Capability Registers Length These bits indicate the start address of the host logic operational registers. As the host logic operational registers are allocated from address 0x20, this value is set to 0x20.	R
15:8	—	These bits are read as 0.	R
31:16	HCIVN[15:0]	EHCI Version Number These bits indicate the version of the EHCI specifications supported by the host logic. As the host logic conforms to the EHCI Rev. 1.1, this value is set to 0x0110.	R

31.3.2.2 HCSPARAMS : Structural Parameters Register

Base address: USBHC = 0x9204_0000

Offset address: 0x104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DBGPTNUM[3:0]				—	—	—	P_INDICATOR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	N_CC[3:0]				N_PCC[3:0]				PTRR	—	—	PPC	N_PORTS[3:0]			
Value after reset:	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0	1

Bit	Symbol	Function	R/W
3:0	N_PORTS[3:0]	Number of downstream ports (Number of Ports) These bits indicate the number of physical downstream ports used by the host logic. These bits reflect the setting of bits [1:0] (Port_no) in the PCI Configuration EXT1 register. As the USB host controller has only one port, this value is set to 1. 0x1: 1 port Others: Reserved	R
4	PPC	Port power control (Port Power Control) This bit indicates how to control the port power in the host logic. 0: Port power is always on 1: Port power is controlled using bit 12 (PP) in the PORTSC1 register	R
6:5	—	These bits are read as 0.	R
7	PTRR	Port routing rules This bit indicates how ports are mapped in the OHCI host logic. As the value of the HCSP_PORTROUTE register shows the mapping in the host logic, this value is set to 1.	R
11:8	N_PCC[3:0]	Number of ports (Number of Ports per Companion Controller) These bits indicate the number of the ports supported by one OHCI host logic unit. These bits reflect the setting of bits [1:0] (Port_no) in the PCI Configuration EXT1 register.	R
15:12	N_CC[3:0]	Number of OHCI host logic (Number of Companion Controller) These bits indicate the number of OHCI host logic units related to the EHCI host logic. As the host logic has one OHCI host logic unit, this value is set to 1.	R
16	P_INDICATOR	Port indicator control support This bit indicates whether the host logic supports the port indicator control function. As the host logic does not support the port indicator control function, this value is set to 0.	R
19:17	—	These bits are read as 0.	R
23:20	DBGPTNUM[3:0]	Debug port number These bits indicate that the host logic ports are debug ports. As the host logic has no debug ports, this value is set to 0.	R
31:24	—	These bits are read as 0.	R

31.3.2.3 HCCPARAMS : Capability Parameters Register

Base address: USBHC = 0x9204_0000

Offset address: 0x108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PL32	PCEC	LPMC	HP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EECP[7:0]								IST[3:0]				—	ASPC	PFLF	AC64
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
0	AC64	Memory pointer selection This bit indicates whether data structures use 32-bit address memory pointers or 64-bit address memory pointers. As the host logic supports data structures using 32-bit address memory pointers, this value is set to 0. Address memory pointers of 64-bits are not supported.	R
1	PFLF	Programming frame list flag This bit indicates the setting regarding the frame list size that software can use. This value is set to 1 in the host logic. When this bit is set to 1, the available frame list size can be specified through bits [3:2] (Frame List Size) in the USBCMD register. A frame list of size smaller than 4 KB can be specified.	R
2	ASPC	Asynchronous schedule park support capability This bit indicates whether to support the park mode for the high-speed QH (queue head) in the asynchronous schedule. As the host logic supports this function, this value is set to 1.	R
3	—	This bit is read as 0.	R
7:4	IST[3:0]	Isochronous data structure threshold As the host logic in this product does not support caching of the isochronous data structure in the entire frame, these bits are set to 0.	R
15:8	EECP[7:0]	Offset address (EHCI Extend Capabilities Pointer) These bits indicate the offset address for the EHCI extend capabilities registers. As the host logic does not support the EHCI extend capabilities registers, these bits are set to 0.	R
16	HP	Hardware prefetch capability As the host logic does not support the hardware prefetch capability defined in EHCI V1.1, this bit is set to 0.	R
17	LPMC	Link power management capability As the host logic supports Link Power Management (LPM) defined in EHCI V1.1, this bit is set to 1. When LPMC = 1, it means that the host logic supports the LPM L1 state and controls LPM by using bit 9 (LPMCTL), bits [24:23] (SUSPSTS[1:0]), and bits [31:25] (DVADDR[6:0]) in the PORTSC1 register. When LPMC = 0, the applicable bits in the PORTSC1 register mentioned above are treated as reserved bits.	R
18	PCEC	Per-port change event capability As the host logic supports the per-port change event capability defined in EHCI V1.1, this bit is set to 1. When this bit is 1, it means that the host logic supports the per-port change event capability. It also means that this bit is associated with bit 15 (per-port change event enable) in the USBCMD register, bit 2 (Port Change Detect) in the USBSTS register, and bit 2 (Port Change Interrupt Enable) in the USBINTR register. When PCEC = 0, the applicable bits in the registers mentioned above are treated as reserved bits.	R

Bit	Symbol	Function	R/W
19	PL32	32-frame periodic list capability As the host logic supports 32-Frame Periodic List defined in EHCI V1.1, this bit is set to 1. When this bit is 1, it means that the host logic supports 32-Frame Periodic List when 11b is set in bits [3:2] (frame list size) in the USBCMD register.	R
31:20	—	These bits are read as 0.	R

31.3.2.4 HCSP_PORTROUTE : Companion Port Route Description Register

Base address: USBHC = 0x9204_0000

Offset address: 0x10C

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	OHCI Host Port Indication These bits indicate the ports that the OHCI host logic controls. As the host logic has one OHCI host logic unit, these bits are set to 0x00000000.	R

The HCSP_PORTROUTE register indicates how ports are mapped in the OHCI controller. The register is enabled only when bit 7 (port routing rules) in the HCSPARAMS register is 1.

31.3.2.5 USBCMD : USB Command Register

Base address: USBHC = 0x9204_0000

Offset address: 0x120

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	HIRD[3:0]				ITC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PPCE E	—	—	—	ASPM E	—	ASPMC[1:0]		LHCR	IAAD	ASYN SE	PSE	FLS[1:0]		HCRE SET	RS
Value after reset:	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RS	EHCI host logic run/stop (Run/Stop) This bit activates or stops the EHCI host logic. When this bit is set to 1, the host logic starts operation. As long as this value is 1, the host logic continues operation. Note that this bit should be set to 1 only while the host logic is in the halt state. Bit 12 (HCHalted) in the USBSTS register indicates that the host logic has completed transactions and entered the halt state. 0: Stopped (the host logic has completed transactions and halted) 1: Activated (the host logic executes the schedule)	R/W

Bit	Symbol	Function	R/W
1	HCRESET	Host logic initialization (Host Controller Reset) This bit initializes the host logic. When this bit is set to 1, the host logic initializes the internal pipelines and state machine. Communication through the USB stops immediately. A USB reset is not issued to downstream ports at this time. This reset does not initialize the PCI configuration registers, but it initializes the EHCI operational registers and the port ownership is returned to OHCI. This bit is automatically cleared to 0 by the host logic when the reset is completed. The reset cannot be aborted by software by writing 0 to this bit. This bit should be set only while bit 12 (HCHalted) in the USBSTS register is 1.	R/W
3:2	FLS[1:0]	Frame list size These bits specify the frame list size. The setting in these bits determines the size of the Frame List Current index in the FRINDEX register. 0 0: 1024 elements (4096 bytes) 0 1: 512 elements (2048 bytes) 1 0: 256 elements (1024 bytes) 1 1: 32 elements (128 bytes)	R/W
4	PSE	Periodic schedule enable This bit specifies whether to advance or skip periodic list processing. 0: Periodic list processing is not executed (skipped) 1: Periodic list processing is executed using the PERIODICLISTBASE register	R/W
5	ASYNSE	Asynchronous schedule enable This bit specifies whether to advance or skip asynchronous list processing. 0: Asynchronous list processing is not executed (skipped) 1: Asynchronous list processing is executed using the ASYNCLISTADDR register	R/W
6	IAAD	Interrupt on Async Advance Doorbell This bit is used as a doorbell by software. Software sets this bit to 1 to generate an interrupt when processing proceeds with the next QH (queue head). While bit 5 (Interrupt on Async Advance Enable) in the USBINTR register is set to 1, an interrupt is generated with the next interrupt timing after 1 is written to this bit. If this bit is set while bit 5 (Interrupt on Async Advance Enable) in the USBINTR register is 0, correct operation is not guaranteed. This bit is cleared by the host logic. On completing the processing of a QH normally, the host logic clears this bit to 0 and sets bit 5 (Interrupt on Async Advance) in the USBSTS register to 1.	R/W
7	LHCR	Light host controller reset execution status This bit indicates the state of Light Host Controller Reset execution. As the host logic does not support Light Host Controller Reset, this value is fixed to 0.	R
9:8	ASPMC[1:0]	Asynchronous schedule park mode count These bits specify the number of transactions that can be executed in succession from one QH (queue head). A value from 0x1 to 0x3 can be specified. This setting is valid when bit 11 (Asynchronous Schedule Park Mode Enable) is 1.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	ASPME	Asynchronous schedule park mode enable This bit enables or disables the park mode. 0: Disabled 1: Enabled	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	PPCEE	Per-port change event enable This bit is used by software to enable the per-port change event capability defined by bit 2 (Port Change Detect) in the USBSTS register and bit 2 (Port Change Interrupt Enable) bit in the USBINTR register.	R/W

Bit	Symbol	Function	R/W
23:16	ITC[7:0]	Host logic interrupt generation maximum rate (Interrupt Threshold Control) These bits indicate the maximum rate at which the host logic generates an interrupt. Correct operation is not guaranteed if a value not shown below is written to these bits. 0x00: Reserved 0x01: 1 micro-frame 0x02: 2 micro-frames 0x04: 4 micro-frames 0x08: 8 micro-frames (1 ms) 0x10: 16 micro-frames (2 ms) 0x20: 32 micro-frames (4 ms) 0x40: 64 micro-frames (8 ms)	R/W
27:24	HIRD[3:0]	Host-Initiated Resume Duration (Minimum K-state drive time) These bits are used by software. The bits indicate the minimum amount of time the host controller drives the K-state during a resume from an LPM state (L1). The value of these bits is sent to the connected devices with the LPM capability by the HIRD field contained in the bmAttributes field in an LPM token. Encoding for these bits is defined in the bits named HIRD in an LPM token. Specifically, 0x0 equals 50 µs and each additional increment adds 75 µs. For example, 0x1 equals 125 µs and 0x4 equals 1175 µs.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The USBCMD register indicates the commands that are executed by the host logic.

31.3.2.6 USBSTS : USB Status Register

Base address: USBHC = 0x9204_0000

Offset address: 0x124

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PTCGDETC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASS	PSCHSTS	RECLAM	EHCSTS	—	—	—	—	—	—	IAAIS	HSYSE	FLROV	PTCGDET	USBERRINT	USBINT
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USBINT	USB transfer complete (USB Interrupt) This bit indicates that USB transfer has been completed. The host logic sets this bit to 1 when either of the following conditions is satisfied: ● USB transfer is completed ● A short packet is received Even if USB transfer has ended with an error, this bit is set to 1 when the IOC (Interrupt On Complete) bit in the TD is 1. Writing 1 can clear this bit. Writing 0 has no effect. 0: USB transfer has not been completed 1: USB transfer has been completed	R/W
1	USBERRINT	USB transaction status (USB Error Interrupt) This bit indicates that a USB transaction has ended with an error. The host logic sets this bit to 1 when a USB transaction has ended with an error. If 1 is set to the IOC bit in the TD that caused an error interrupt, 1 is set to both the USBERRINT bit and the USBINT bit. Writing 1 can clear this bit. Writing 0 has no effect. 0: USB transaction has been completed normally 1: USB transaction has ended with an error	R/W

Bit	Symbol	Function	R/W
2	PTCGDET	Port state change detection This bit indicates that the port state has changed. The host logic sets this bit to 1 when any of the following conditions is satisfied in a port for which bit 13 (Port Owner) in the PORTSC[n] register is set to 0. [1 setting condition] ● Bit 1 (Connect Status Change) in the PORTSC[n] register has changed from 0 to 1 (device connection or disconnection has been detected) ● Bit 3 (Port Enable/Disable Change) in the PORTSC[n] register has changed from 0 to 1 (the port enabled state has changed) ● Bit 5 (Over-current Change) in the PORTSC[n] register has changed from 0 to 1 (the overcurrent state has been detected) ● Bit 6 (Force Port Resume) in the PORTSC[n] register has changed from 0 to 1 (indicating detection of a J-to-K transition on a port while suspended) Writing 1 by software can clear this bit. Writing 0 has no effect.	R/W
3	FLROV	Frame list rollover The host logic sets this bit to 1 when the value of the Frame Index bits in the FRINDEX register returns from the maximum value to 0x000 (rollover). The maximum value at which rollover occurs depends on bits [3:2] (Frame List Size) in the USBCMD register. Writing 1 can clear this bit. Writing 0 has no effect. 0: Frame list has not returned to 0x000 1: Frame list has returned to 0x000	R/W
4	HSYSE	Host system error This bit is set to 1 when a serious error occurs in the host logic. Such an error includes a parity error in the PCI system. If this error occurs, the host logic clears bit 0 (RS) in the USBCMD register to 0 to stop execution of the remaining TDs. Writing 1 can clear this bit. Writing 0 has no effect. 0: No system error has occurred 1: System error has occurred	R/W
5	IAAIS	Async advance interrupt status This bit indicates the Async Advance interrupt state. On fetching QH, the host logic checks bit 6 (Interrupt on Async Advance Doorbell (IAAD)) in the USBCMD register. When IAAD = 1, the host logic clears the IAAD bit and sets this bit after completing QH processing normally. When bit 5 (Interrupt on Async Advance Enable) in the USBINTR register is 1, an interrupt is generated from this source with the next interrupt timing after this bit is set to 1. Writing 1 can clear this bit. Writing 0 has no effect. 0: No Async Advance interrupt has occurred 1: Async Advance interrupt state has been detected	R/W
11:6	—	These bits are read as 0. The write value should be 0.	R/W
12	EHCSTS	EHCI host logic status (HCHalted) This bit is set to 0 when bit 0 (RS) in the USBCMD register is 1. When the RS bit is cleared to 0 by the host logic or software, the EHCI host logic stops execution and this bit is set to 1 by the host logic. 0: EHCI host logic is in execution 1: EHCI host logic is stopped	R
13	RECLAM	Empty asynchronous schedule detection (Reclamation) This bit is used to detect an empty asynchronous schedule. The host logic clears this bit to 0 after a reset or when it fetches QH with H = 1. When the host logic executes an asynchronous transaction or has detected a start event, it sets this bit to 1. When the host logic fetches QH with H = 1 while this bit is 0, the Async Sched Sleeping mode is entered.	R
14	PSCHSTS	Periodic schedule status This bit indicates the current state of the periodic schedule. The periodic schedule is enabled (1) or disabled (0) when the values of this bit and bit 4 (Periodic Schedule Enable) in the USBCMD register are the same. 0: Periodic schedule is disabled 1: Periodic schedule is enabled	R

Bit	Symbol	Function	R/W
15	ASS	Asynchronous schedule status This bit indicates the current state of the asynchronous schedule. The asynchronous schedule is enabled (1) or disabled (0) when the values of this bit and bit 5 (Asynchronous schedule enable) in the USBCMD register are the same. 0: Asynchronous schedule is disabled 1: Asynchronous schedule is enabled	R
31:16	PTCGDETC[15:0]	Port-n Change Detect Each of these bits is the Port Change Detect bit for a specific port. Bit 16 is the port change detect bit for port 1, bit 17 is the port change detect bit for port 2, and so on. For example, when bit 16 is set to 1, it means the status change of port 1 was detected. These bits are used by software only when bit 15 (Per-Port Change Events Enable) in the USBCMD register is set to 1.	R/W

The USBSTS register indicates interrupt sources and various states of the host logic.

The result of a transaction on the USB bus is not indicated in this register.

31.3.2.7 USBINTR : USB Interrupt Enable Register

Base address: USBHC = 0x9204_0000

Offset address: 0x128

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PCGIE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	INTAADVE	HSEE	FMLSTROE	PTCGIE	USBEIE	USBIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USBIE	USB interrupt enable This bit enables or disables the setting in bit 0 (USBINT) in the USBSTS register. When USBSTS.USBINT = 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the USBINT bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the USBINT bit)	R/W
1	USBEIE	USB error interrupt enable This bit enables or disables the setting in bit 1 (USBERRINT) in the USBSTS register. When USBSTS.USBERRINT = 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the USBERRINT bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the USBERRINT bit)	R/W
2	PTCGIE	Port change interrupt enable This bit enables or disables the setting in bit 2 (Port Change Detect) in the USBSTS register. When the Port Change Detect bit is set to 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the Port Change Detect bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the Port Change Detect bit)	R/W

Bit	Symbol	Function	R/W
3	FMLSTROE	Frame list rollover enable This bit enables or disables the setting in bit 3 (Frame List Rollover) in the USBSTS register. When the Frame List Rollover bit is set to 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the Frame List Rollover bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the Frame List Rollover bit)	R/W
4	HSEE	Host system error enable This bit enables or disables the setting in bit 4 (Host System Error) in the USBSTS register. When the Host System Error bit is set to 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the Host System Error bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the Host System Error bit)	R/W
5	INTAADVE	Interrupt on async advance enable This bit enables or disables the setting in bit 5 (Interrupt on Async Advance) in the USBSTS register. When the Interrupt on Async Advance bit is set to 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the Interrupt on Async Advance bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the Interrupt on Async Advance bit)	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
31:16	PCGIE[15:0]	Port-n Change Interrupt Enable Each of these bits is the Port Change Interrupt Enable bit for a specific port. Bit 16 is the Port Change Interrupt Enable bit for port 1, bit 17 is the Port Change Interrupt Enable bit for port 2, and so on. When one of these bits is set to 1 and the corresponding bit (one of PTCGDETC[15:0], Port-n Change Detect) in the USBSTS register is set to 1, the host logic generates an interrupt. Only bit 16 is enabled because the USB host controller only supports port 1. The interrupt is acknowledged by software when the Port-n Change Detect bit is cleared.	R/W

The USBINTR register enables or disables reporting of interrupts.

When a bit in this register is set to 1 and the corresponding interrupt bit is set to 1, an interrupt is generated.

The interrupt sources that are disabled in this register can be detected by allowing software to poll for events stored in the USBSTS register.

31.3.2.8 FRINDEX : USB Frame Index Register

Base address: USBHC = 0x9204_0000

Offset address: 0x12C

Bit position:	31														13																0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRAMEINDEX[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	FRAMEINDEX[13:0]	Frame index These bits are used by the host logic to add an index to the periodic frame list. This value is incremented at the end of each micro-frame. Bits [N:3] are used as the Frame List Current index. This means that the current frame list is accessed 8 times before proceeding with the next index number. The value of N is determined as follows according to the setting of bits [3:2] (Frame List Size) in the USBCMD register. **Frame List Size / Number Elements / N** 00b (1024) 12 01b (512) 11 10b (256) 10 11b (32) 12 This register should be accessed only while the host logic is stopped (bit 12 (HCHalted) = 1 in the USBSTS register). This setting is reflected in the SOF frame number for the SOF token.	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The FRINDEX register is used by the host logic to add an index to the periodic frame list.

31.3.2.9 CTRLDSSEGMENT : Control Data Structure Segment Register

Base address: USBHC = 0x9204_0000

Offset address: 0x130

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	The host logic does not support 64-bit addressing and does not use this register. Therefore, do not access this register.	R

31.3.2.10 PERIODICLISTBASE : Periodic Frame List Base Address Register

Base address: USBHC = 0x9204_0000

Offset address: 0x134

Bit position:	31																				12												0
Bit field:	PFLSA[19:0]																				—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0. The write value should be 0.	R/W
31:12	PFLSA[19:0]	Periodic frame list start address These bits indicate the start address of the periodic frame list stored in the system memory. Software should specify the start address in this register before the host logic starts list processing. The host logic uses the values in these bits and bits [13:0] (Frame Index) in the FRINDEX register to determine the frame list to be processed. The address of the periodic frame list should be aligned with a 4-KB boundary. Correct operation cannot be guaranteed if the value in these bits is modified during operation.	R/W

31.3.2.11 ASYNCLISTADDR : Next Asynchronous List Address Register

Base address: USBHC = 0x9204_0000

Offset address: 0x138

Bit position:	31																								5					0
Bit field:	LPL[26:0]																								—	—	—	—	—	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0. The write value should be 0.	R/W
31:5	LPL[26:0]	Asynchronous Queue Head link pointer address (Link Pointer Low) These bits indicate the address of the next asynchronous queue head in the system memory to be processed. The address of the asynchronous queue head should be aligned with a 32-byte boundary.	R/W

31.3.2.12 CONFIGFLAG : Configure Flag Register

Base address: USBHC = 0x9204_0000

Offset address: 0x160

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CF	Port routing control circuit configuration flag (Configure Flag) This bit controls the default setting of the port routing control circuit regarding whether the ports are routed to OHCI or EHCI. Software should set this bit to 1 at the end of host logic configuration process. 0: Port routing control circuit routes each port to the OHCI host logic by default 1: Port routing control circuit routes each port to the EHCI host logic by default	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

31.3.2.13 PORTSC1 : Port 1 Status and Control Register

Base address: USBHC = 0x9204_0000

Offset address: 0x164

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DVADDR[6:0]							SUSPSTS[1:0]		WKOC_E	WKDSCNNT_E	WKCNNT_E	PTTST[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PTINDCTL[1:0]		PTOWNR	PP	LINESTS[1:0]		LPMCTL	PTRST	SUSPEND	FRCPTRSM	OVCC	OVCACT	PTESC	PTE	CSC	CCSTS
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CCSTS	Port connection status This bit indicates the connection state of the port. On detecting device connection, the host logic sets this bit to 1. The host logic also sets this bit to 1 when PTTST[3:0] = 0101b (Test FORCE_ENABLE) even if no device is connected. On detecting device disconnection, the host logic clears this bit to 0. When PP = 0 or PTOWNR = 0, this bit becomes 0. 0: No device is connected to the port 1: Device is connected to the port	R
1	CSC	Connect status change This bit indicates that the value of bit 0 (Current Connect Status) has changed. Writing 1 can clear this bit. Writing 0 has no effect. When PP = 0, this bit becomes 0. 0: Value of bit 0 (Current Connect Status) has not changed 1: Value of bit 0 (Current Connect Status) has changed	R/W
2	PTE	Port enable/disable status This bit indicates the enabled/disabled state of the port. After resetting a port and detecting that the connected device is an HS device, the host logic enables the port and sets this bit to 1. Software cannot set this bit to 1. On detecting device disconnection or an error, the host logic disables the port and clears this bit to 0. Software can write 0 to disable the port. In this case, however, the bit value changes only after the port state actually changes. While the port is disabled, data transfer to downstream ports is blocked except for a port reset. When PP = 0, this bit becomes 0. Note that when PTTST[3:0] = 0101b (Test FORCE_ENABLE), the port is enabled regardless of the port state and this bit is set to 1. 0: Port is disabled 1: Port is enabled	R/W
3	PTESC	Port enable/disable status change This bit indicates that the enabled/disabled state of the port has changed. On detecting a frame babble error, the host logic disables the port and sets this bit to 1. Writing 1 can clear this bit. Writing 0 has no effect. When PP = 0, this bit becomes 0. 0: Frame babble error has not occurred 1: Frame babble error has been detected	R/W
4	OVCACT	Port overcurrent status This bit indicates the overcurrent state of the port. On detecting an overcurrent, the host logic clears bit 12 (PP) and its related bits and sets this bit to 1. This bit is automatically cleared from 1 to 0 when the overcurrent state is resolved. 0: Port is not in overcurrent state 1: Port is in overcurrent state	R
5	OVCC	Over-current Change This bit indicates that the value of bit 4 (OVCACT) has changed. Writing 1 can clear this bit. Writing 0 has no effect. 0: Value of bit 4 (OVCACT) has not changed 1: Value of bit 4 (OVCACT) has changed	R/W

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>6</td><td>FRCPTRSM</td><td>Force Port Resume (Port resume detection flag)
This bit indicates detection of the resume state at the port.
When a transition from the J to the K state (RemoteWakeup) is detected while the port is suspended, the host logic sets this bit and bit 2 (Port Change Detect) or bits [31:16] (Port-n Change Detect [15:0]) in the USBSTS register to 1. If software writes 1 to the Force Port Resume bit, the host logic does not set the Port Change Detect bit or the Port-n Change Detect bits to 1.
The resume signal (FS K State) is driven on the USB port while this bit is 1. Clear this bit to 0 after an appropriate period has passed.
Writing 0 to this bit while it is 1 returns the port to the HS Idle state. Until then, this bit remains at 1.
After software writes 0 to this bit, the host logic must complete the transition to the HS Idle state within 2 ms. If the port is in the L1 state, software does not need to write 0 to this bit because the host logic sends a resume signal when necessary and clears this bit after the port resumes. However, the length of the resume signal driven by the host logic must be set in bits [27:24] (Host-Initiated Resume Duration) in the USBCMD register.
When PP = 0, this bit becomes 0.
0: Resume (K-state) has not been detected or output
1: Resume (K-state) has been detected or output</td><td>R/W</td></tr>
<tr><td>7</td><td>SUSPEND</td><td>Port suspend
The combination of the settings of this bit and bit 2 (Port Enabled/Disabled) indicate the port state as follows.
<table>
<tr><th>PTE</th><th>SUSPEND</th><th>Port State</th></tr>
<tr><td>0</td><td>x</td><td>Disabled</td></tr>
<tr><td>1</td><td>0</td><td>Enabled</td></tr>
<tr><td>1</td><td>1</td><td>Suspend</td></tr>
</table>
Set this bit to 1 to transition the port to the L1 or L2 suspend state. The type of suspend state (L1 or L2) the host logic supports depends on the value of the Suspend using L1 bit.
In the suspend state, data transfer from this port to the downstream port is blocked unless the port is reset. When this bit is set to 1 during data transfer, reflection in the status bit and blocking of transfer follow completion of the current transfer.
Software should write 1 to this bit. It can write 1 only while PP = 1, PTOWNR = 0, and CCSTS = 1 in the register in the host logic.
This bit is always cleared to 0 under either of the following conditions:
● Software clears the Force Port Resume bit to 0
● Completion of resume is detected
● Software sets the Port Reset (PTRST) bit to 1
● Port Owner (PTOWNR) bit is set to 1 (OHCI)
● Port Power (PP) bit is set to 0
● Port Enabled/Disabled (PTE) bit is set to 0
This bit indicates the suspend state of the port.
0: Port is not in suspend state
1: Port is in suspend state</td><td>R/W</td></tr>
<tr><td>8</td><td>PTRST</td><td>Port reset status
When software writes 1 to this bit while it is 0, the bus reset sequence defined in the USB2.0 standard begins. Write 0 to this bit to terminate the bus reset sequence. Note that this bit must be retained at 1 for long enough time for the bus reset sequence to be completed according to the USB2.0 standard.
When bit 12 (HCHalted) in the USBSTS register is 1, the port should not be reset.
This bit becomes 0 when the PP, PTOWNR, or CCSTS bit satisfies the following conditions.
Writing 1 does not start the bus reset sequence.
● PP = 0
● PTOWNR = 1
● CCSTS = 0
This bit indicates the reset state of the port.
0: Port is not in reset state
1: Port is in reset state</td><td>R/W</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>9</td><td>LPMCTL</td><td>LPM control
This bit is used to select Suspend using L1 (LPM) or Suspend using L2.
When 1 is written to bit 7 (SUSPEND) while this bit is 1, the host logic transitions the attached device to the LPM state.
When this bit is 1 and the Device Address bits are other than 0x0000, setting the Suspend bit to 1 makes the host logic create an LPM token for transitioning the attached device to the L1 state. When this bit is 0, the host logic uses the L2 suspend mechanism.
Writing to this bit is permitted only when bit 7 (SUSPEND) is 0. When SUSPEND = 1, both 0 and 1 cannot be written to this bit.
0: Suspend using L2
1: Suspend using L1 (LPM)</td><td>R/W</td></tr>
<tr><td>11:10</td><td>LINESTS[1:0]</td><td>D+/D− logic level
These bits indicate the current logical levels on the D+ and D− lines of the USB port (bit 11: D+, bit 10: D−).
They are used to detect an LS device before a port reset or a sequence for enabling the port. Therefore, the values in these bits are valid only while bit 3 (Port Enable/Disable) = 0 and bit 0 (Current Connect Status) = 1.
<table>
<tr><th>Bit 11 (D+)</th><th>Bit 10 (D−)</th><th>USB State</th><th>Interpretation</th></tr>
<tr><td>0</td><td>0</td><td>SE0</td><td>Not an LS device.
Moves to the EHCI port reset execution process.</td></tr>
<tr><td>0</td><td>1</td><td>K State</td><td>An LS device has been connected.
Passes the port ownership from EHCI to OHCI.</td></tr>
<tr><td>1</td><td>0</td><td>J State</td><td>Not an LS device.
Moves to the EHCI port reset execution process.</td></tr>
<tr><td>1</td><td>1</td><td>Undefined</td><td>Not an LS device.
Moves to the EHCI port reset execution process.</td></tr>
</table></td><td>R</td></tr>
<tr><td>12</td><td>PP</td><td>Port Power Supply Control (Port Power)
This bit controls power supply to the port.
When this bit is 0, power is not supplied to the port. The port does not operate and connection or disconnection cannot be detected.
If an overcurrent is detected while this bit is 1, the host logic clears this bit to 0 and stops power supply to the port. However, as explained later, when the PPC bit is 0, this bit is fixed to 1 and power supply to the port does not stop.
The function of this bit depends on the setting of bit 4 (PPC) in the HCSPARAMS register.
<table>
<tr><th>PPC</th><th>PP</th><th>Description</th></tr>
<tr><td>0</td><td>1</td><td>This bit is fixed to 1, and power is always supplied to the port</td></tr>
<tr><td>1</td><td>0/1</td><td>Power supply to the port is determined by this bit</td></tr>
</table>
0: No power is supplied to the port
1: Power is supplied to the port</td><td>R/W</td></tr>
<tr><td>13</td><td>PTOWNR</td><td>Port ownership
This bit indicates whether OHCI or EHCI has the port ownership.
This bit is cleared to 0 when bit 0 (CF) in the CONFIGFLAG register changes from 0 to 1.
This bit is set to 1 when the CF bit is 0.
Software should set this bit to 1 to pass the port ownership to OHCI when the connected device is not a high-speed device.
0: EHCI has the port ownership
1: OHCI has the port ownership</td><td>R/W</td></tr>
<tr><td>15:14</td><td>PTINDCTL[1:0]</td><td>As the host logic does not support the port indicator control function, these bits are set to 00b.
Writing to these bits does not affect the operation.</td><td>R</td></tr>
</table>

Bit	Symbol	Function	R/W
19:16	PTTST[3:0]	Pin test control These bits control the test mode. For details of the test mode, see Chapter 7 in the USB Specification Revision 2.0. When the value is other than 0000b, the host logic is operating in test mode. **Port Test Control[3:0] / Mode** 0000b: Normal 0001b: Test J_STATE 0010b: Test K_STATE 0011b: Test SE0_NAK 0100b: Test Packet 0101b: Test FORCE_ENABLE Other than the above: Reserved	R/W
20	WKCNNT_E	Device connection detection enable (Wake on Connect Enable) Writing 1 to this bit enables detection of device connection as a wakeup event. When PP = 0, this bit becomes 0.	R/W
21	WKDSCNNT_E	Device disconnection detection enable (Wake on Disconnect Enable) Writing 1 to this bit enables detection of device disconnection as a wakeup event. When PP = 0, this bit becomes 0.	R/W
22	WKOC_E	Overcurrent state detection enable (Wake on Over-current Enable) Writing 1 to this bit enables detection of the overcurrent state as a wakeup event. When PP = 0, this bit becomes 0.	R/W
24:23	SUSPSTS[1:0]	Suspend status These bits indicate a response from a connected device to an LPM token (L1 transition request). The host logic updates these bits immediately when the L1 transition request processing is completed. To avoid race conditions with hardware, a value must be set to these bits when SUSPEND = 0. 0 0: Device successfully entered the L1 state (ACK is received from the device) 0 1: Device was unable to enter the L1 state at this time (NYET is received from the device) 1 0: Device does not support L1 state transitions (STALL is received from the device) 1 1: Other than above (such as a timeout error)	R/W
31:25	DVADDR[6:0]	USB device address These bits indicate the USB device address (7 bits) of the device attached to the downstream port and are used when an LPM token is sent. When the value of these bits is 0x00, no device that requires these bits is connected.	R/W

31.3.3 AHB Bridge Registers

31.3.3.1 INTENABLE : INT_ENABLE Register

Base address: USBHC = 0x9204_0000

Offset address: 0x200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	WAKEON_INTEN	UCOM_INTEN	USBH_INTBEN	USBH_INTAEN	AHB_INTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AHB_INTEN	AHB_INT bit control This bit enables or disables the interrupt source indicated in bit 0 (AHB_INT) in the INT_STATUS register. 0: Disabled 1: Enabled	R/W
1	USBH_INTAEN	USBH_INTA bit control This bit enables or disables the interrupt source indicated in bit 1 (USBH_INTA) in the INT_STATUS register. 0: Disabled 1: Enabled	R/W
2	USBH_INTBEN	USBH_INTB bit control This bit enables or disables the interrupt source indicated in bit 2 (USBH_INTB) in the INT_STATUS register. 0: Disabled 1: Enabled	R/W
3	UCOM_INTEN	UCOM_INT bit control This bit enables or disables the interrupt source indicated in bit 3 (UCOM_INT) in the INT_STATUS register. 0: Disabled 1: Enabled	R/W
4	WAKEON_INTEN	WAKEON_INT bit control This bit enables or disables the interrupt source indicated in bit 4 (WAKEON_INT) in the INT_STATUS register. 0: Disabled 1: Enabled	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

The INTENABLE register enables or disables each interrupt source indicated in the INT_STATUS register. When an interrupt source is disabled, the interrupt signal is not asserted even when the interrupt source is generated and the corresponding bit in the INT_STATUS register is set to 1.

31.3.3.2 INTSTATUS : INT_STATUS Register

Base address: USBHC = 0x9204_0000

Offset address: 0x204

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	WAKEON_INT	UCOM_INT	USBH_INTB	USBH_INTA	AHB_INT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AHB_INT	AHB bus error indication This bit indicates that an AHB bus error has occurred. Writing 1 clears this bit. 0: AHB bus error has not occurred. 1: AHB bus error has occurred	R/W
1	USBH_INTA	OHCI interrupt status This bit indicates the state of the OHCI interrupt. The interrupt can be cleared using the HcInterruptStatus register (one of OHCI operational registers). 0: INTA interrupt has not occurred 1: INTA interrupt has occurred	R

Bit	Symbol	Function	R/W
2	USBH_INTB	USBH_INTB EHCI interrupt status This bit indicates the state of the USBH_INTB EHCI interrupt. The interrupt can be cleared using the USBSTS register (one of EHCI operational registers). 0: INTB interrupt has not occurred 1: INTB interrupt has occurred	R
3	UCOM_INT	UCOM register interrupt status This bit indicates the state of the interrupt from the UCOM register. The interrupt can be cleared using the UCOM2 register. 0: UCOM register interrupt has not occurred 1: UCOM register interrupt has occurred	R
4	WAKEON_INT	WAKEON interrupt status This bit indicates the state of the WAKEON interrupt from the host logic. Writing 1 clears this bit. 0: WAKEON interrupt has not occurred 1: WAKEON interrupt has occurred	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

The INTSTATUS register indicates the state of interrupt sources in the AHB bridge and the state of interrupt signals from the OHCI, EHCI, and UCOM modules.

31.3.3.3 AHBBUSCTR : AHB_BUS_CTR Register

Base address: USBHC = 0x9204_0000

Offset address: 0x208

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PROT_TYPE[3:0]				—	—	—	PROT_MODE	—	—	ALIGN_ADDRESS[1:0]		—	—	MAX_BURST_LEN[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	MAX_BURST_LEN[1:0]	Maximum burst length These bits select the maximum burst length when the AHB master interface initiates a transfer. 0 0: INCR16 0 1: INCR8 1 0: INCR4 1 1: SINGLE	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	ALIGN_ADDRESS[1:0]	Address boundary setting These bits set the address boundary when the AHB master interface initiates a burst transfer. 0 0: 1-KB boundary is not exceeded when initiating a burst transfer 0 1: 64-byte boundary is not exceeded when initiating a burst transfer 1 0: 32-byte boundary is not exceeded when initiating a burst transfer (The maximum burst length is INCR8 in this case. The 32-byte boundary is exceeded when the maximum burst length is INCR16.) 1 1: 16-byte boundary is not exceeded when initiating a burst transfer. (The maximum burst length is INCR4 in this case. The 16-byte boundary is exceeded when the maximum burst length is INCR8 or greater.)	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	PROT_MODE	This bit selects the MHPROT[3:0] mode when the AHB master interface initiates a transfer. 0: Value of PROT_TYPE[3:0] is output as MHPROT[3:0] 1: 0000b is set as MHPROT[3:0] for DMA transfer when the last data of a burst is transferred. The value of PROT_TYPE[3:0] is used as MHPROT[3:0] for other data in a burst.	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
15:12	PROT_TYPE[3:0]	These bits set MHPROT[3:0] when the AHB master interface initiates a transfer.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Name	Value	Description
15	PROT_TYPE[3]	0	Cache is disabled
		1	Cache is enabled
14	PROT_TYPE[2]	0	Buffer is disabled
		1	Buffer is enabled
13	PROT_TYPE[1]	0	Access by a user
		1	Privileged access
12	PROT_TYPE[0]	0	Operation code
		1	Data

The AHBBUSCTR register specifies the AHB master and slave functions.

31.3.3.4 USBCTR : USBCTR Register

Base address: USBHC = 0x9204_0000

Offset address: 0x20C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	DIRPD	PLL_RST	USBH_RST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	USBH_RST	Software reset for the core When this bit is set to 1, the entire core is reset. Set this bit when the AHB master interface in the core is inactive. Access to the host logic is enabled in 10 PCLKAM clock cycles after this bit is written to. 0: Nothing occurs 1: Host logic is reset	W
1	PLL_RST	Reset of USB PHY PLL Reset to the PHY PLL is asserted. 0: Release PLL reset 1: Assert PLL reset	R/W
2	DIRPD	Direct transition to power-down state If this bit is set to 1, USB subsystem moves to power-down state. 0: Normal operation 1: Direct power-down state	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

31.3.4 Core Defined Registers

31.3.4.1 REVID : Revision and Core ID Register

Base address: USBHC = 0x9204_0000

Offset address: 0x300

Bit position:	31								24								15								8	7							0
Bit field:	COREID[7:0]								—	—	—	—	—	—	—	—	MAJV[7:0]								MINV[7:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
7:0	MINV[7:0]	Minor Version These bits indicate the minor version of the host logic.	R
15:8	MAJV[7:0]	Major Version These bits indicate the major version of the host logic.	R
23:16	—	This bit is read as 0.	R
31:24	COREID[7:0]	Core ID These bits indicate the ID of the host logic.	R

31.3.4.2 OCSLPTIMSET : Overcurrent Detection/Sleep Timer Setting Register

Base address: USBHC = 0x9204_0000

Offset address: 0x310

Bit position:	31			28				20	19																						0
Bit field:	—	—	—	TIMER_SLEEP[8:0]					TIMER_OC[19:0]																							
Value after reset:	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
19:0	TIMER_OC[19:0]	Overcurrent Timer setting When the overcurrent input, USB_OVRCUR, is asserted (driven to low) continuously longer than these bit setting time, the USB module determines the occurring overcurrent. This value should be set to more than 1 to 2 ms for the AHB clock and PCLKAM frequencies. One bit means 1 clock cycle. 1 ms at PCLKAM = 200 MHz: 0x3_0D04 (default setting)	R/W
28:20	TIMER_SLEEP[8:0]	Detection/Sleep Timer Setting These bits are timer setting for RemoteWakeup receive detection time during sleep and measuring Resume-K time. This value should be set to more than 1 µs for the AHB clock and PCLKAM frequencies. One bit means 1 clock cycle. PCLKAM = 200 MHz: 0x0C8 (default setting)	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

The OCSLPTIMSET register specifies the detection time of overcurent.

31.3.5 UCOM Registers

31.3.5.1 COMMCTRL : Common Control Register

Base address: USBHC = 0x9204_0000

Offset address: 0x800

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PERI	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
30:0	—	These bits are read as 0. The write value should be 0.	R/W
31	PERI	USB mode setting This bit specified whether to place the USB module in host mode or function mode. 0: Host mode 1: Function mode	R/W

The COMMCTRL register specifies whether to use USB module in host mode or function mode.

31.3.5.2 OBINTSTA : OTG-BC Interrupt Status Register

Base address: USBHC = 0x9204_0000

Offset address: 0x804

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DPMONCHG_STA	DMMONCHG_STA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SESSVLDCHG_STA	IDDIGCHG_STA	VVLDCHG_STA	—	—	—	—	—	—	VBSTAINT_STA	VBSTACHG_STA	OCINT_STA	IDCHG_STA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	IDCHG_STA	USB_OTG_ID change status This bit is set when the value that is input from the USB_OTG_ID pin has changed. The initial value is 1. Before using this bit, clear the status. 0: Value input from the USB_OTG_ID pin has not changed 1: Value input from the USB_OTG_ID pin has changed	R/W*1
1	OCINT_STA	USB_OVRCUR assertion status This bit is set when the USB_OVRCUR pin is asserted. 0: USB_OVRCUR pin is not asserted (remains 1) 1: USB_OVRCUR pin is asserted (change to 0)	R/W*1
2	VBSTACHG_STA	VBSTA[3:0] change status This bit is set when the value of bits [31:28] (VBSTA[3:0]) in the VBCTRL register has changed. 0: Value of the VBSTA[3:0] bits has not changed 1: Value of the VBSTA[3:0] bits has changed	R/W*1

Bit	Symbol	Function	R/W
3	VBSTAINT_STA	VBUS voltage status interrupt This bit is set when the value of bits [31:28] (VBSTA[3:0]) in the VBCTRL register equals to the value set for bits [23:20] (VBLVL[3:0]) in the VBCTRL register. This bit is set instantly when the value of the VBSTA[3:0] bits equals to the value of the VBLVL[3:0] bits. After that, if this bit is cleared and the value of the VBSTA[3:0] bits is unchanged, the bit is not set even if the values of the VBSTA[3:0] bits and the VBLVL[3:0] bits are the same. This bit is set again when the value of the VBSTA[3:0] bits changes and the values of the VBSTA[3:0] bits and the VBLVL[3:0] bits become the same again. 0: Value of VBSTA[3:0] bits is not the same as the value of the VBLVL[3:0] bits 1: Value of VBSTA[3:0] bits is the same as the value of the VBLVL[3:0] bits.	R/W*1
9:4	—	These bits are read as 0. The write value should be 0.	R/W
10	VVLDCHG_STA	VBUSVALID change status This bit is set if the VBUSVALID bit of the PHYCTRL register has changed. 0: The VBUSVALID bit of the PHYCTRL register has not changed 1: The VBUSVALID bit of the PHYCTRL register has changed	R/W*1
11	IDDIGCHG_STA	IDDIG0 change status This bit is set if the IDDIG0 bit of the PHYCTRL register has changed. 0: The IDDIG0 bit of the PHYCTRL register has not changed 1: The IDDIG0 bit of the PHYCTRL register has changed	R/W*1
12	SESSVLDCHG_STA	OTGSESSVLD change status This bit is set if the OTGSESSVLD bit of the PHYCTRL register has changed. 0: The OTGSESSVLD bit of the PHYCTRL register has not changed 1: The OTGSESSVLD bit of the PHYCTRL register has changed	R/W*1
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	DMMONCHG_STA	DMMON change status This bit is set if the DMMON bit of the LINECTRL1 register has changed. 0: The DMMON bit of the LINECTRL1 register has not changed 1: The DMMON bit of the LINECTRL1 register has changed	R/W*1
17	DPMONCHG_STA	DPMON change status This bit is set if the DPMON bit of the LINECTRL1 register has changed. 0: The DPMON bit of the LINECTRL1 register has not changed 1: The DPMON bit of the LINECTRL1 register has changed	R/W*1
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 clears this bit. Writing 0 has no effect.

The OBINTSTA register indicates whether interrupt sources related to USB OTG have occurred.

An OBINT interrupt is generated when a bit is set in this register and the corresponding bit is not masked in the OBINTEN register.

31.3.5.3 OBINTEN : OTG-BC Interrupt Enable Register

Base address: USBHC = 0x9204_0000

Offset address: 0x808

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DPMONCHG_EN	DMMONCHG_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SESSVLDCHG_EN	IDDIGCHG_EN	VVLDCHG_EN	—	—	—	—	—	—	VBSTAINT_EN	VBSTACHG_EN	OCINT_EN	IDCHG_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IDCHG_EN	IDCHG_STA Interrupt enable This bit enables or disables the interrupt flagged by the IDCHG_STA bit. 0: Interrupt flagged by the IDCHG_STA bit is disabled. 1: Interrupt flagged by the IDCHG_STA bit is enabled	R/W
1	OCINT_EN	OCINT_STA interrupt enable This bit enables or disables the interrupt flagged by the OCINT_STA bit. 0: Interrupt flagged by the OCINT_STA bit is disabled 1: Interrupt flagged by the OCINT_STA bit is enabled	R/W
2	VBSTACHG_EN	VBSTACHG_STA interrupt enable This bit enables or disables the interrupt flagged by the VBSTACHG_STA. 0: Interrupt flagged by the VBSTACHG_STA bit is disabled 1: Interrupt flagged by the VBSTACHG_STA bit is enabled	R/W
3	VBSTAINT_EN	VBSTAINT_STA interrupt enable This bit enables or disables the interrupt flagged by the VBSTAINT_STA. 0: Interrupt flagged by the VBSTAINT_STA bit is disabled 1: Interrupt flagged by the VBSTAINT_STA bit is enabled	R/W
9:4	—	These bits are read as 0. The write value should be 0.	R/W
10	VVLDCHG_EN	VVLDCHG_STA interrupt enable This bit enables or disables the interrupt flagged by the VVLDCHG_STA. 0: Interrupt flagged by the VVLDCHG_STA bit disabled 1: Interrupt flagged by the VVLDCHG_STA bit enabled	R/W
11	IDDIGCHG_EN	IDDIGCHG_STA interrupt enable This bit enables or disables the interrupt flagged by the IDDIGCHG_STA. 0: Interrupt flagged by the IDDIGCHG_STA bit disabled 1: Interrupt flagged by the IDDIGCHG_STA bit enabled	R/W
12	SESSVLDCHG_EN	SESSVLDCHG_STA interrupt enable This bit enables or disables the interrupt flagged by the SESSVLDCHG_STA. 0: Interrupt flagged by the SESSVLDCHG_STA bit disabled 1: Interrupt flagged by the SESSVLDCHG_STA bit enabled	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	DMMONCHG_EN	DMMONCHG_STA interrupt enable This bit enables or disables the interrupt flagged by the DMMONCHG_STA. 0: Interrupt flagged by the DMMONCHG_STA bit disabled 1: Interrupt flagged by the DMMONCHG_STA bit enabled	R/W
17	DPMONCHG_EN	DPMONCHG_STA interrupt enable This bit enables or disables the interrupt flagged by the DPMONCHG_STA. 0: Interrupt flagged by the DPMONCHG_STA bit disabled 1: Interrupt flagged by the DPMONCHG_STA bit enabled	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The OBINTEN register enables or disables the interrupts that are flagged by the OBINTSTA register.

An OBINT interrupt is generated when a bit is set the OBINTSTA register and the corresponding bit is set in the OBINTEN register.

31.3.5.4 VBCTRL : VBUS Control Register

Base address: USBHC = 0x9204_0000

Offset address: 0x80C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	VBSTA[3:0]				—	—	—	—	VBLVL[3:0]				—	—	OCISEL	OCCLRIEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	VGPUO	—	SIDDQREL	VBUSENSEL	VBOUT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBOUT	VBUS drive control (USB_VBUSEN pin) This bit is used to assert the VBUS power path by controlling external power supply IC. 0: VBUS output is disabled. (USB_VBUSEN is driven to low.) 1: VBUS output is enabled. (USB_VBUSEN is driven to high.)	R/W
1	VBUSENSEL	USB_VBUSEN pin control 0: USB_VBUSEN is controlled by PPON1 1: USB_VBUSEN is controlled by VBOUT bit in this register	R/W
2	SIDDQREL	SIDDQ mode control 0: SIDDQ mode is set 1: SIDDQ mode is released	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	VGPUO	USB_EXICEN pin control This bit is used to control external power supply IC. 0: USB_EXICEN is driven to low 1: USB_EXICEN is driven to high	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
16	OCCLRIEN	USB_VBUSEN pin control at occurrence of overcurrent This bit specifies whether to automatically clear the VBOUT bit if an overcurrent occurs. If an overcurrent occurs, bit 1 (OCINT_STA) in the OBINTSTA register is set. 0: VBOUT bit is not automatically cleared if an overcurrent occurs 1: VBOUT bit is automatically cleared if an overcurrent occurs	R/W
17	OCISEL	Overcurrent detection This bit selects the signal to be used for detecting an overcurrent. 0: OCI signal that is output by the host logic This signal is treated as an indication of an overcurrent if the USB_OVRCUR pin is asserted longer than the overcurrent detection time set in bits [19:0] (TIMER_OC[19:0]) in the OCSLPTIMSET register. 1: Inverted signal from the USB_OVRCUR pin When this signal is selected, the overcurrent detection time set in the TIMER_OC[19:0] bits is not used. The system assumes an overcurrent has occurred if the USB_OVRCUR pin is asserted for 5 clock cycles or more.	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
23:20	VBLVL[3:0]	VBUS level detection These bits are used to detect, using an interrupt, that VBUS has reached a certain level. When the value of the voltage level of VBUS equals to the value set for these bits, the interrupt flagged by the VBSTAINT_STA bit in the OBINTSTA register is generated.	R/W
27:24	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31:28	VBSTA[3:0]	VBUS indication These bits indicate the voltage level of VBUS. The bits indicate the input from the USB_VUBUSIN pin as is. VBSTA[3:2] and VBSTA[0] bits are fixed to 0. VBSTA[1] bit indicates USB_VUBUSIN input level if DRVVBUS bit in PHYCTRL register is 1 and VBSTA[1] is fixed to 0 if DRVVBUS bit in PHYCTRL register is 0.	R

The VBCTRL register controls VBUS when the USB module is used in OTG mode.

31.3.5.5 LINECTRL1 : Line Control Port 1 Register

Base address: USBHC = 0x9204_0000

Offset address: 0x810

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DPRPD_EN	DP_RPD	DMRPD_EN	DM_RPD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DPMON	DMMON	—	IDMON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1

Bit	Symbol	Function	R/W
0	IDMON	Indicates a value of USB_OTG_ID input pin.	R
1	—	This bit is read as 0.	R
2	DMMON	Indicates a value of USB bus DM.	R
3	DPMON	Indicates a value of USB bus DP.	R
15:4	—	These bits are read as 0. The write value should be 0.	R/W
16	DM_RPD	Controls USB bus (DM) 15 kΩ pulldown resistor when DMPPD_EN = 1. 0: DM-side 15 kΩ pulldown resistor is OFF 1: DM-side 15 kΩ pulldown resistor is ON	R/W
17	DMRPD_EN	Enables DM_RPD to control USB bus (DM) 15 kΩ pulldown resistor. 0: Control of DM-side 15 kΩ pulldown resistor by DM_RPD disabled 1: Control of DM-side 15 kΩ pulldown resistor by DM_RPD enabled	R/W
18	DP_RPD	Controls USB bus (DP) 15 kΩ pulldown resistor when DRPPD_EN = 1. 0: DP-side 15 kΩ pulldown resistor is OFF 1: DP-side 15 kΩ pulldown resistor is ON	R/W
19	DPRPD_EN	Enables DP_RPD to control USB bus (DP) 15 kΩ pulldown resistor. 0: Control of DP-side 15 kΩ pulldown resistor by DP_RPD disabled 1: Control of DP-side 15 kΩ pulldown resistor by DP_RPD enabled	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This value depends on the USB_OTG_ID input pin.

31.3.5.6 PHYCTRL : PHY Control Register

Base address: USBHC = 0x9204_0000

Offset address: 0x830

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	OTGSESSVLD	IDDIG0	VBUSVALID	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	IDPULLUP0	DRVVBUS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	DRVVBUS	Drive VBUS 0: The VBUS Valid comparator is disabled. 1: The VBUS Valid comparator is enabled.	R/W
5	IDPULLUP0	ID Input Sample Enable 0: ID pin sampling is disabled, and the IDDIG0 bit is not valid. 1: ID pin sampling is enabled, and the IDDIG0 bit is valid.	R/W
17:6	—	These bits are read as 0. The write value should be 0.	R/W
18	VBUSVALID	VBUS Valid Indicator 0: USB_VUBUSIN level is not valid. 1: USB_VUBUSIN level is valid.	R
19	IDDIG0	Mini A/B Plug Indicator 0: The connected plug is a mini-A plug. 1: The connected plug is a mini-B plug.	R
20	OTGSESSVLD	OTG Device Session Valid Indicator 0: OTG Device Session is no valid. 1: OTG Device Session is valid.	R
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The PHYCTRL register controls VBUS and OTG_ID.

31.4 Clock Lines

31.4.1 Externally Supplied Clocks

Table 31.3 describes the two clock signals that are supplied from within the LSI to the USB host controller.

Table 31.3 List of externally supplied clocks

Clock signal	Function	Frequency
PCLKAM	AHB clock	200 MHz
USB_CLK	Reference clock	50 MHz

31.5 Interrupts

31.5.1 Interrupt Control Registers

31.5.1.1 U2H_INT Control Registers

U2H_INT is an interrupt signal generated by the AHB bridge. Registers in the AHB bridge are used to check and clear the interrupt state and to enable interrupts.

Table 31.4 U2H_INT Control Registers

Operation	Control Register
Check and clear interrupt state	INT_STATUS register
Enable interrupts	INT_ENABLE register

31.5.1.2 U2H_OHCI_INT Control Registers

U2H_OHCI_INT is the INTA interrupt signal from the host logic. Each interrupt is basically controlled through registers in the host logic, but to assert the interrupt signal, the interrupt enable bit in the AHB bridge must be set.

Table 31.5 U2H_OHCI_INT Control Registers

Operation	Control Register
Check and clear interrupt state	HcInterruptStatus register
Enable interrupts	HcInterruptEnable register HcInterruptDisable register INT_ENABLE register (bit 1 (USBH_INTAEN))

31.5.1.3 U2H_EHCI_INT Control Registers

U2H_EHCI_INT is the INTB interrupt signal from the host logic. Each interrupt is basically controlled through registers in the host logic, but to assert the interrupt signal, the interrupt enable bit in the AHB bridge must be set.

Table 31.6 U2H_EHCI_INT Control Registers

Operation	Control Register
Check and clear interrupt state	USBSTS register
Enable interrupts	USBINTR register INT_ENABLE register (bit 2 (USBH_INTBEN))

31.5.1.4 U2H_OBINT Control Registers

U2H_OBINT is the UCOM interrupt signal from the host logic. Each interrupt is basically controlled through registers in the host logic, but to assert the interrupt signal, the interrupt enable bit in the AHB bridge must be set.

Table 31.7 U2H_OBINT Control Registers

Operation	Control Register
Check and clear interrupt state	OBINTSTA register
Enable interrupts	OBINTEN register INT_ENABLE register (bit 3 (UCOM_INTEN))

31.5.1.5 U2H_WAKEON_INT Control Registers

U2H_WAKEON_INT is the WAKEON interrupt signal from the host logic. Each interrupt is basically controlled through registers in the host logic, but to assert the interrupt signal, the interrupt enable bit in the AHB bridge must be set.

Table 31.8 U2H_WAKEON_INT Control Registers

Operation	Control Register
Check and clear interrupt state	INT_STATUS register (bit 4 (WAKEON_INT))
Enable interrupts	PORTSC1 register (bits 22 to 20) INT_ENABLE register (bit 4 (WAKEON_INTEN))

31.5.2 U2H_BIND_INT

The U2H_BIND_INT interrupt signal is generated as the logical OR of the following five interrupt source signals:

- U2H_INT
- U2H_OHCI_INT

- U2H_EHCI_INT
- U2H_OBINT
- U2H_WAKEON_INT.

The states of the U2H_OBINT, U2H_WAKEON_INT, U2H_OHCI_INT, and U2H_EHCI_INT signals are reflected in the INT_STATUS register in the AHB bridge. Read the register to check which source was responsible for a generated interrupt.

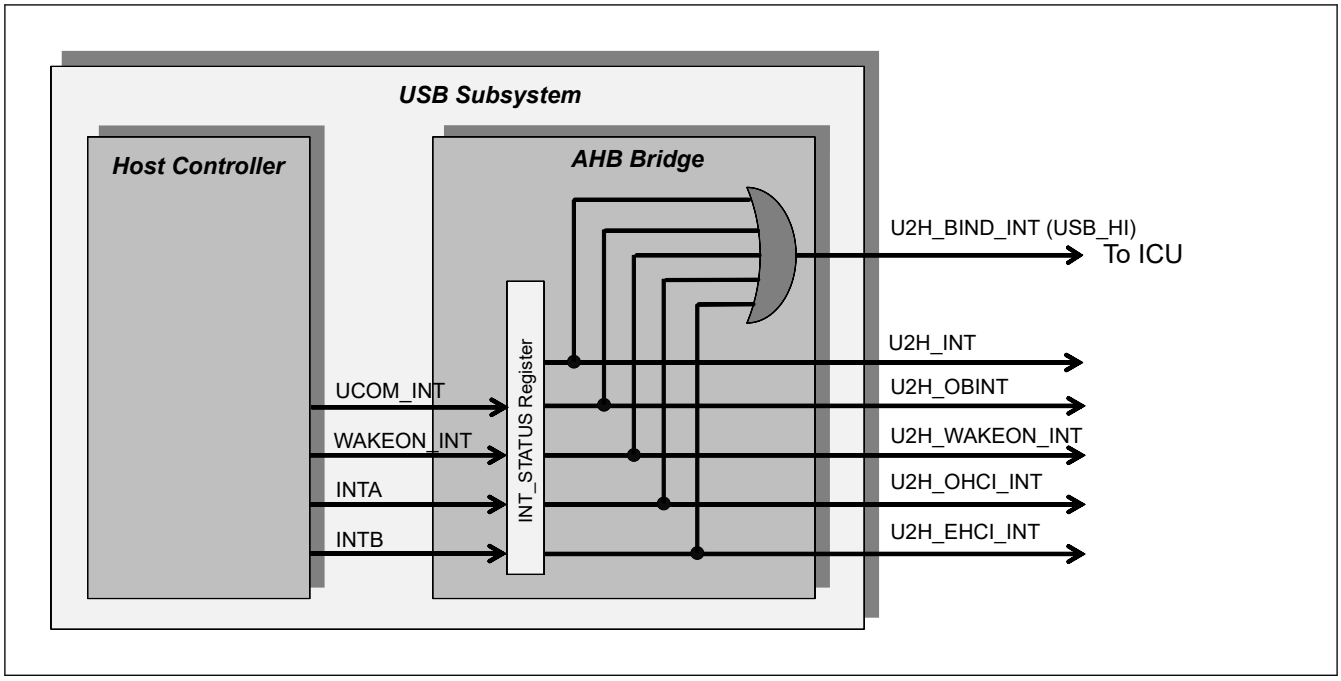


Figure 31.2 Image of interrupt signal integration

31.5.3 Time Required to Clear Interrupt Signals

It may take a long time to actually clear the interrupt after the register for clearing the interrupt source is written to due to the posted write operation of the AHB bridge.

Measure should be taken to prevent false detection before the next interrupt is detected after the register access for clearing the interrupt is completed.

31.6 Overcurrent Control and VBUS Control

31.6.1 Overcurrent Control

This section describes the operation of the USB_OVRCUR and USB_VBUSEN signals used for overcurrent detection at the USB port and VBUS control.

31.6.1.1 USB_OVRCUR and USB_VBUSEN Signal Functions

Table 31.9 shows the meaning of the USB_OVRCUR and USB_VBUSEN signals.

Table 31.9 USB_OVRCUR and USB_VBUSEN

Pin	I/O	Level	Meaning
USB_OVRCUR	Input	1	No overcurrent has been detected
		0	Overcurrent has been detected
USB_VBUSEN	Output	1	Power supply to VBUS is on
		0	Power supply to VBUS is off

31.6.1.2 Conditions for Asserting and Deasserting USB_VBUSEN Output Signal

Figure 31.3 shows a timing diagram for the assertion and deassertion of USB_OVRCUR and USB_VBUSEN.

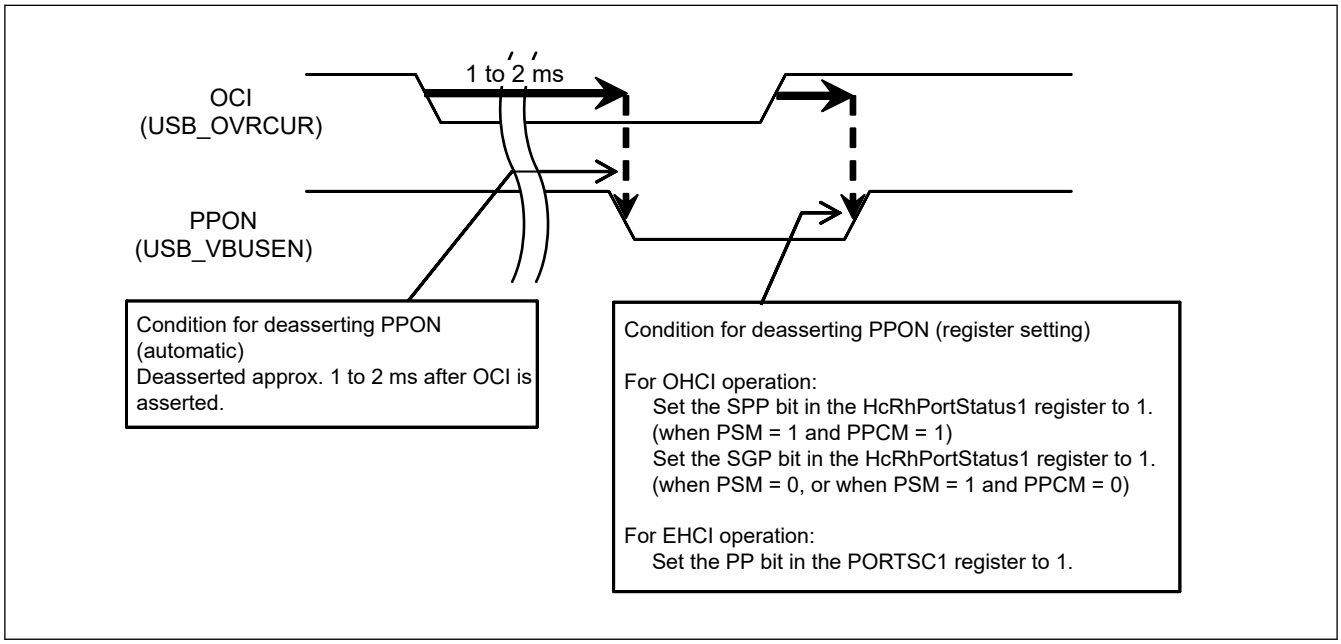


Figure 31.3 Timing diagram for the assertion and deassertion of USB_OVRCUR and USB_VBUSEN

USB_VBUSEN is not asserted automatically after USB_OVRCUR is deasserted. USB_VBUSEN is asserted when software sets the port power bit after USB_OVRCUR is deasserted.

31.6.2 VBUS Control

When the USB port is not used, VBUS can be stopped to reduce power consumption by connecting the USB_OVRCUR and USB_VBUSEN pins to the high-side switch (although this may not be possible depending on the connections with peripheral circuits).

Table 31.10 shows the relationship between USB_VBUSEN and VBUS.

Table 31.10 Relationship between USB_VBUSEN and VBUS

USB_VBUSEN	VBUS
0	Stopped
1	Operated

The USB_VBUSEN operation when USB_OVRCUR is asserted depends on the settings in the OHCI operational register as shown in Table 31.11.

Table 31.11 Register settings and USB_VBUSEN operation

OHCI operational registers				
HcRhDescriptorA Register			HcRhDescriptorB	
NOCP bit	NPS bit	PSM bit	PPCM[1] bit	USB_VBUSEN output pin operation
1	—	—	—	Fixed to 1
—	1	—	—	Fixed to 1
0	0	0	—	Deasserted (0) when USB_OVRCUR is asserted (0)
		1	0	
			1	

Note: When NPS = 1, the host logic detects an overcurrent but does not deassert USB_VBUSEN.

31.6.3 Initial Settings of PPON1

ON and OFF of PPON1 (PortPower) is controlled by the Port Control register of OHCI/EHCI operational registers listed in Table 31.12 for general usage after reset without setting the NOCP and NPS bits in the HcRhDescriptorA register. PPON1 is turned on when at least one of the following bits is ON:

- SPP
- SGP
- PP

In general, bits SPP, SGP, and PP are cleared to 0 when an overcurrent is detected.

Table 31.12 OHCI and EHCI operational registers

Register		Bit	Symbol
OHCI operation	HcRhPortStatus1 register	8	SPP
	HcRhPortStatus register	16	SGP*1
EHCI operation	PORTSC1 register	12	PP

Note 1. This is not generally used with the single-port version. For PPON control by this bit, the bit must be in the state of PSM = 0, or PSM = 1 and PPCM = 0.

PPON1 is turned on by setting the NOCP or NPS bit to 1 regardless of the Port Power bit of the specified ports.

The initial settings of the PPON1 control bits (SPP, SGP, PP, NOCP, and NPS) are all 0 (OFF). Setting these bits to 1 (ON) during initial setting required a procedure in the following flow.

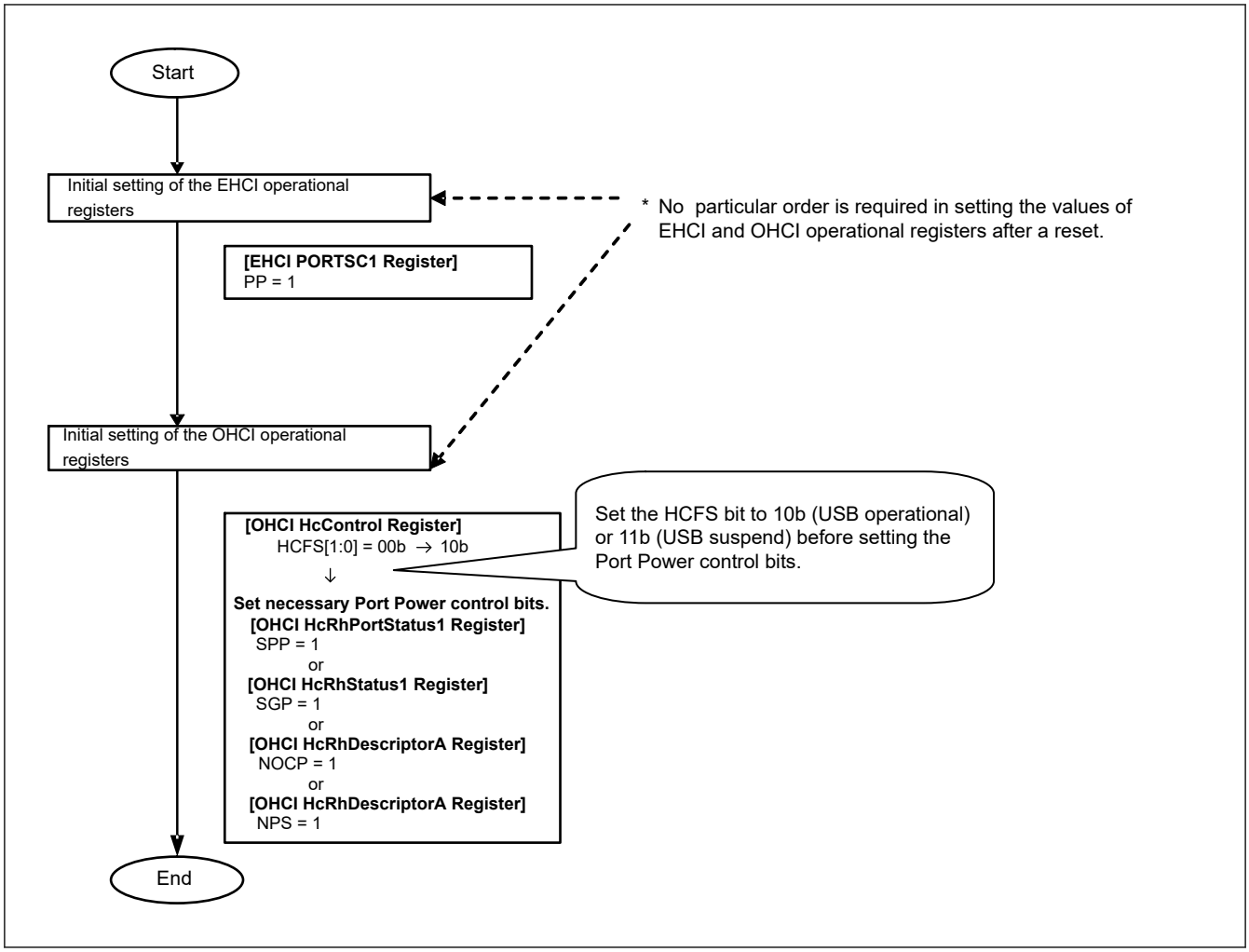


Figure 31.4 Initial settings flow of PPON1

31.6.4 Procedure for USB_VBUSEN Control for Overcurrent Detection

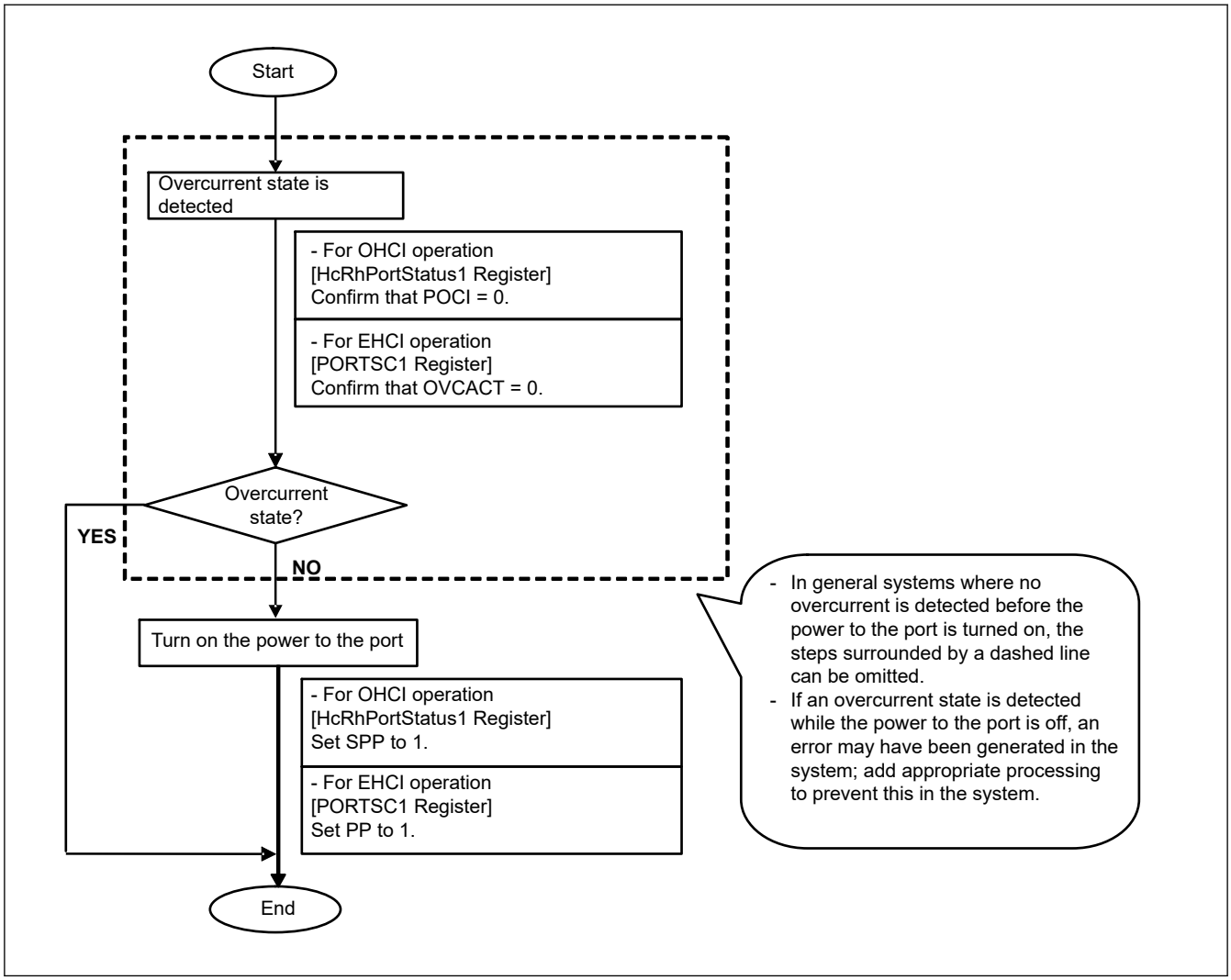


Figure 31.5 Flow of USB_VBUSEN control for overcurrent detection

31.6.5 Procedure for USB_VBUSEN Setting

Figure 31.6 shows the procedure for USB_VBUSEN setting in a system where USB_OVRCUR may be active (0) at system startup.

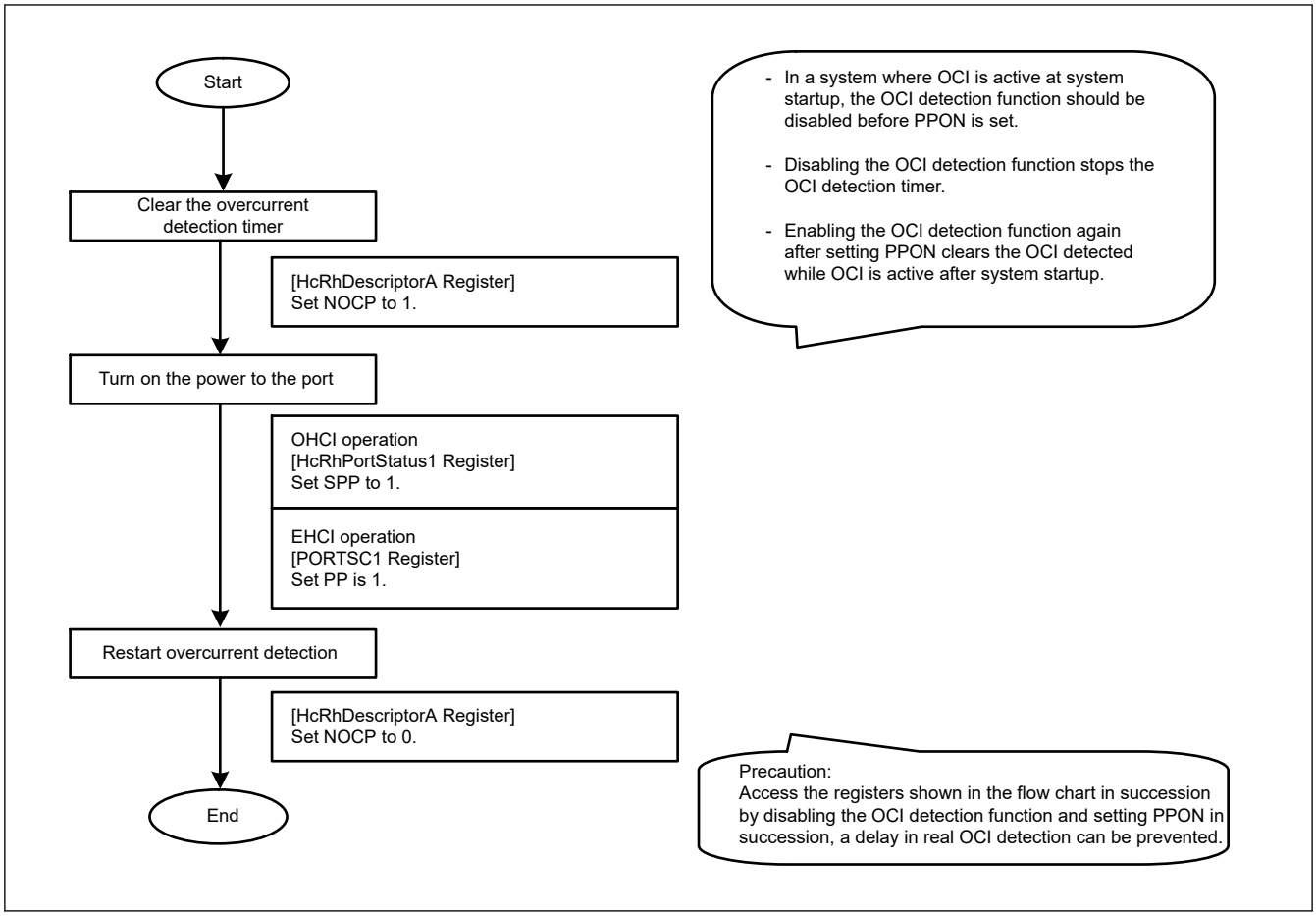


Figure 31.6 Flow for USB_VBUSEN setting

31.7 Initial Setting Sequence

31.7.1 Sample of Initial Settings

This section shows sample initial settings to implement the following functions:

- Access to OHCI/EHCI operational registers
- Data transfer from the host logic to the AHB bus.

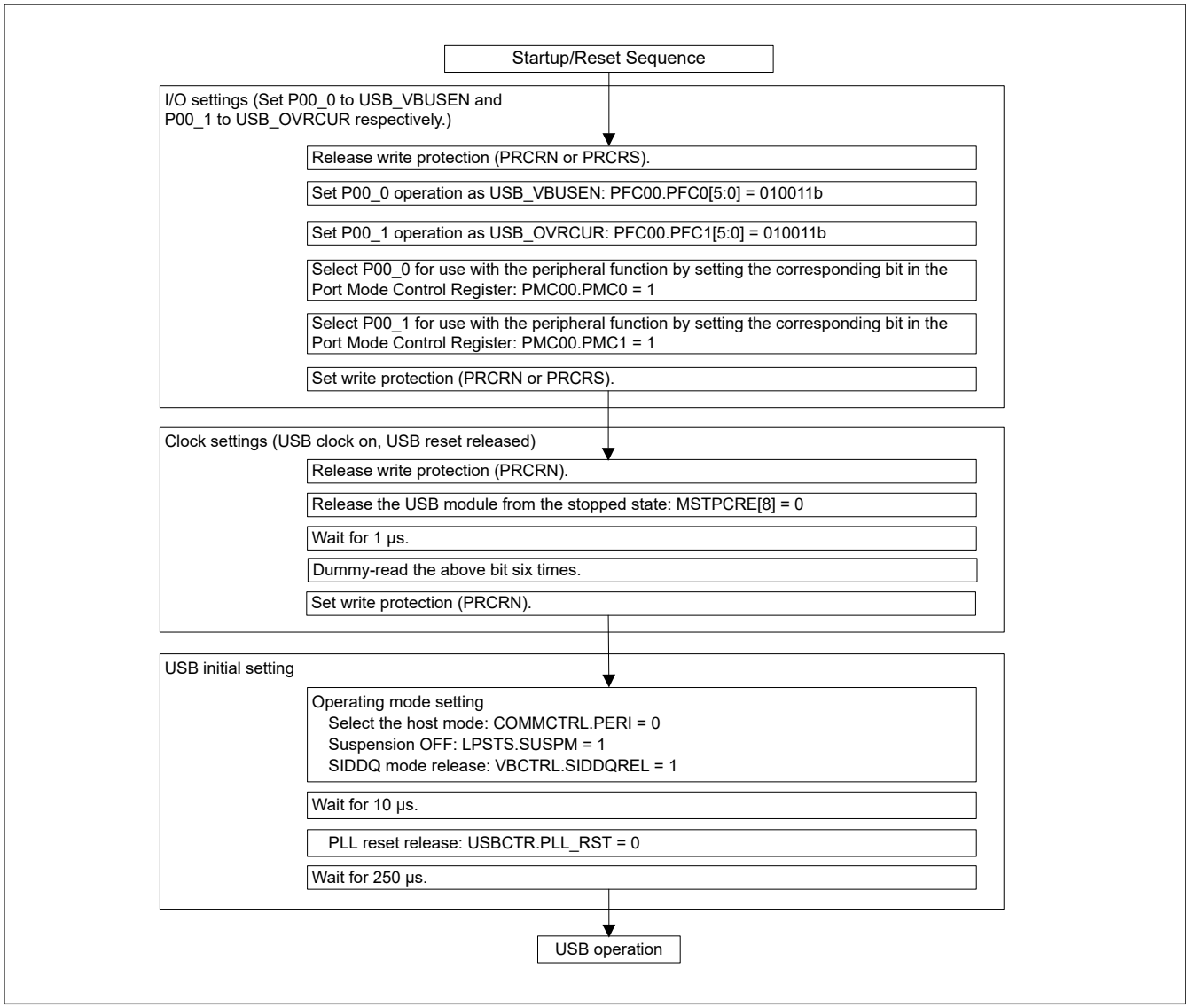


Figure 31.7 Initial setting sequence

31.8 OTG Control

31.8.1 Port Power (VBUS) Control and Connection with External Power-Supply IC

The following procedure describes how to control the port power (VBUS) when using this module for OTG:

1. Control the port power through the associated UCOM registers.
2. Connect the external power-supply IC and the corresponding pins of this module according to the following table.

LSI pin name	I/O	Connection
USB_OTG_ID	I	Connect this pin as input from the IC power supply
USB_VUBUSIN	I	Connect this pin as input from the IC power supply
USB_VBUSEN	O	Connect this pin as output to the IC power supply
USB_EXICEN	O	Connect this pin as output to the IC power supply
USB_OVRCUR	I	Connect this pin as input from the IC power supply If the power-supply IC does not support overcurrent detection capability, this input signal should be tied to high level.

31.8.2 OTG System Configuration Example

Figure 31.8 shows an example of connections to support the OTG function. The configuration includes an external power-supply IC.

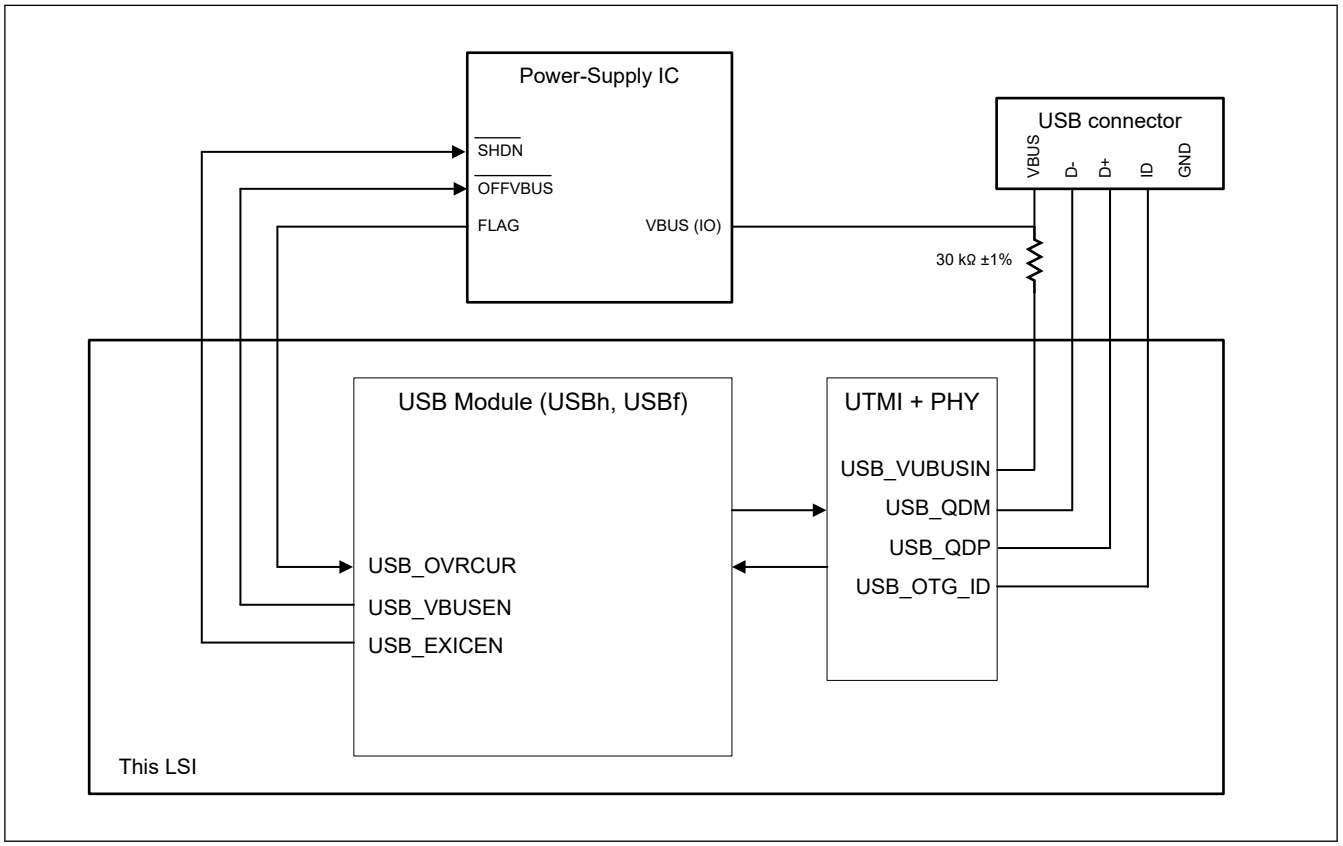


Figure 31.8 OTG system configuration example

31.8.3 OTG Initial Setting Procedure

Figure 31.9 shows the initial setting procedure when this module is used for OTG. The figure shows the steps before detection of A-plug or B-plug connection through the ID and VBUS pins.

Assuming the control of ID and VBUS:

VBUS control: VBCTRL.VBOUT

ID detection: UTMI+PHY

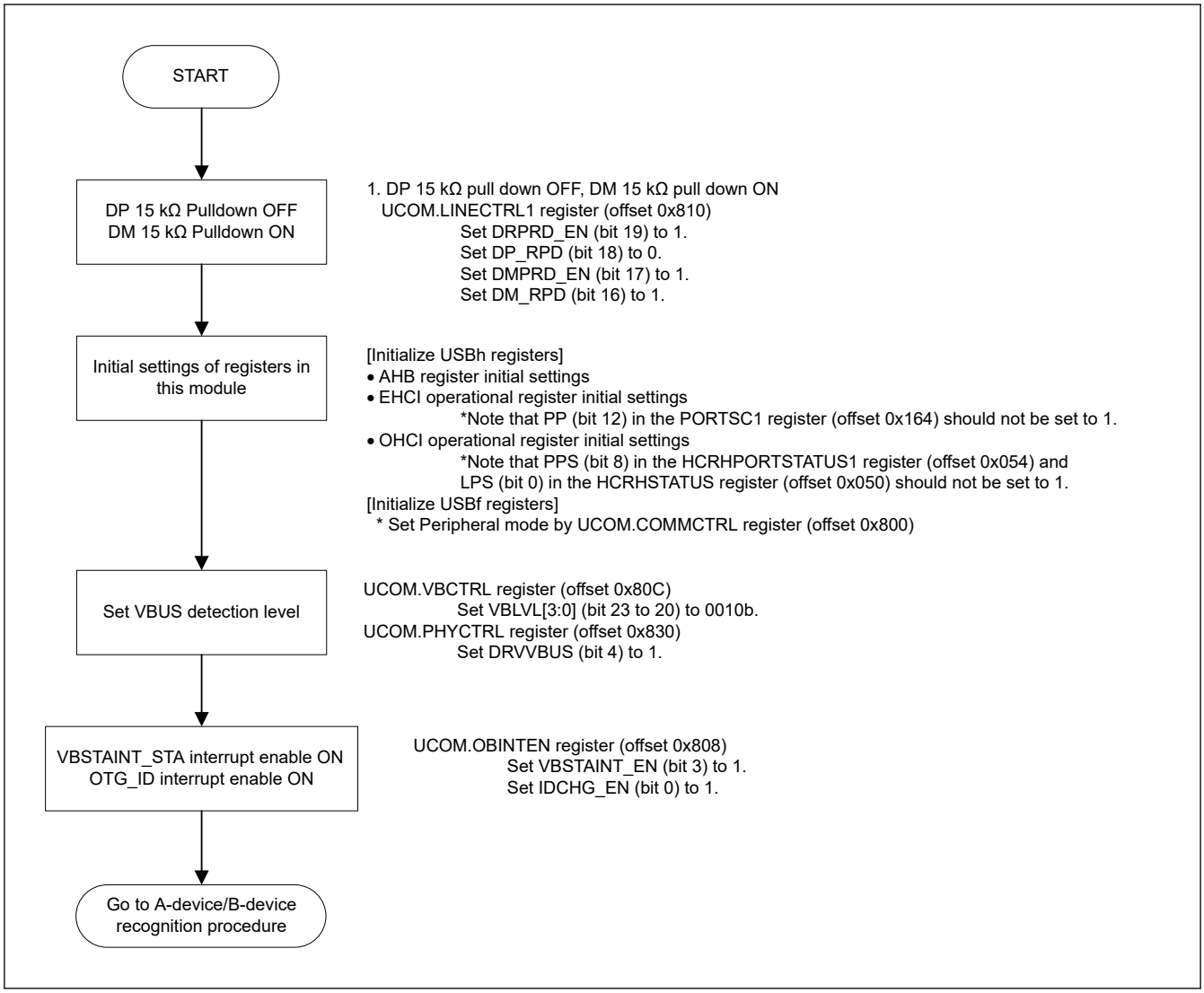


Figure 31.9 OTG initial setting procedure

31.8.4 A-Device/B-Device Recognition Procedure

Figure 31.10 shows the control procedure for operating this module as an A-device or a B-device through A-plug or B-plug connection following the steps described in section 31.8.3. OTG Initial Setting Procedure or section 31.8.5. Disconnect Detection Procedure.

Assuming the control of ID and VBUS:

VBUS control: VBCTRL.VBOUT

ID detection: UTMI+PHY

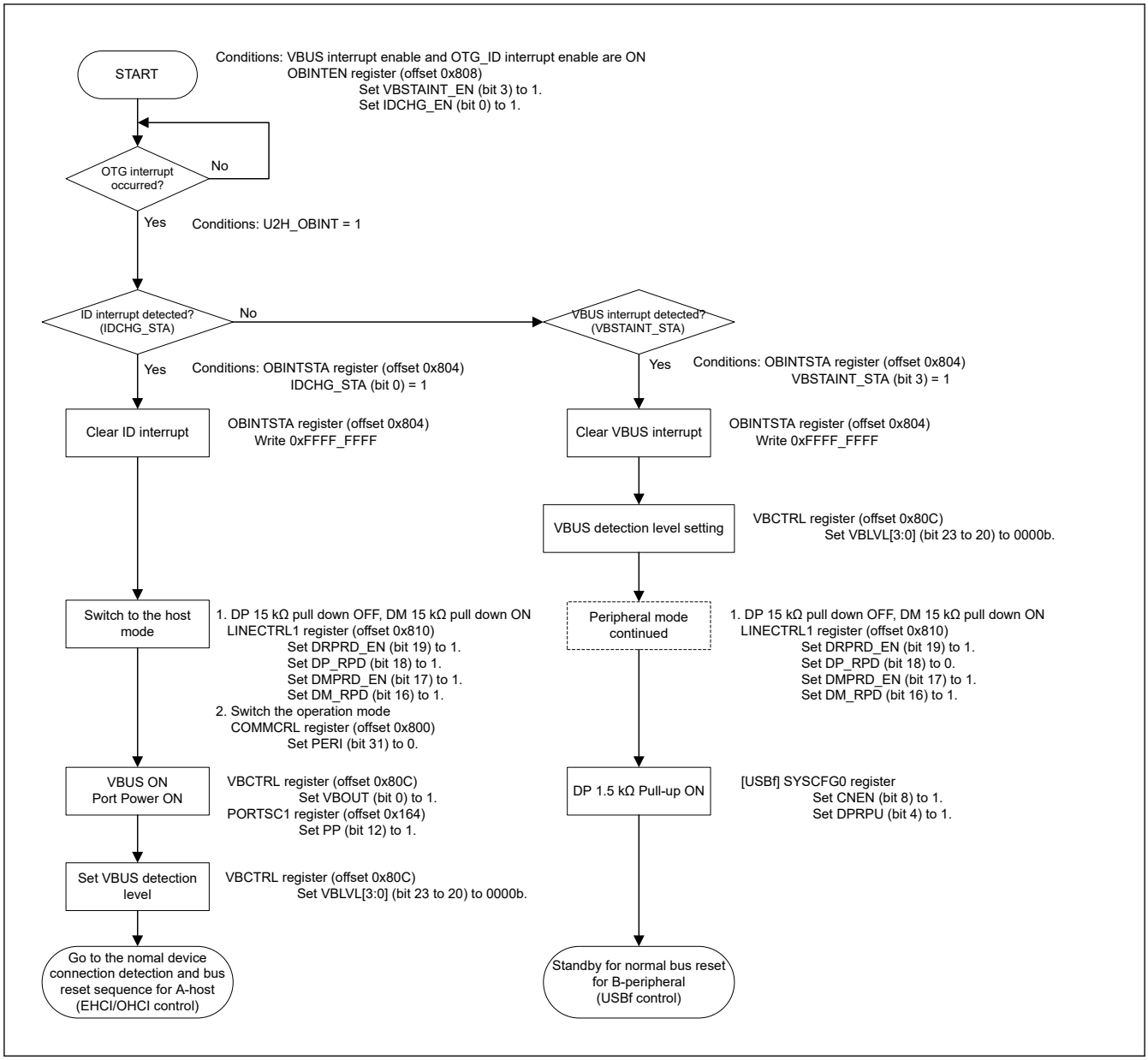


Figure 31.10 A-Device/B-Device recognition procedure

31.8.5 Disconnect Detection Procedure

Figure 31.11 shows the disconnect detection procedure. Disconnect detection means detection of a physical disconnection of an A-plug or B-plug from a USB port.

Assuming the control of ID and VBUS:

VBUS control: VBCTRL.VBOUT

ID detection: UTMI+PHY

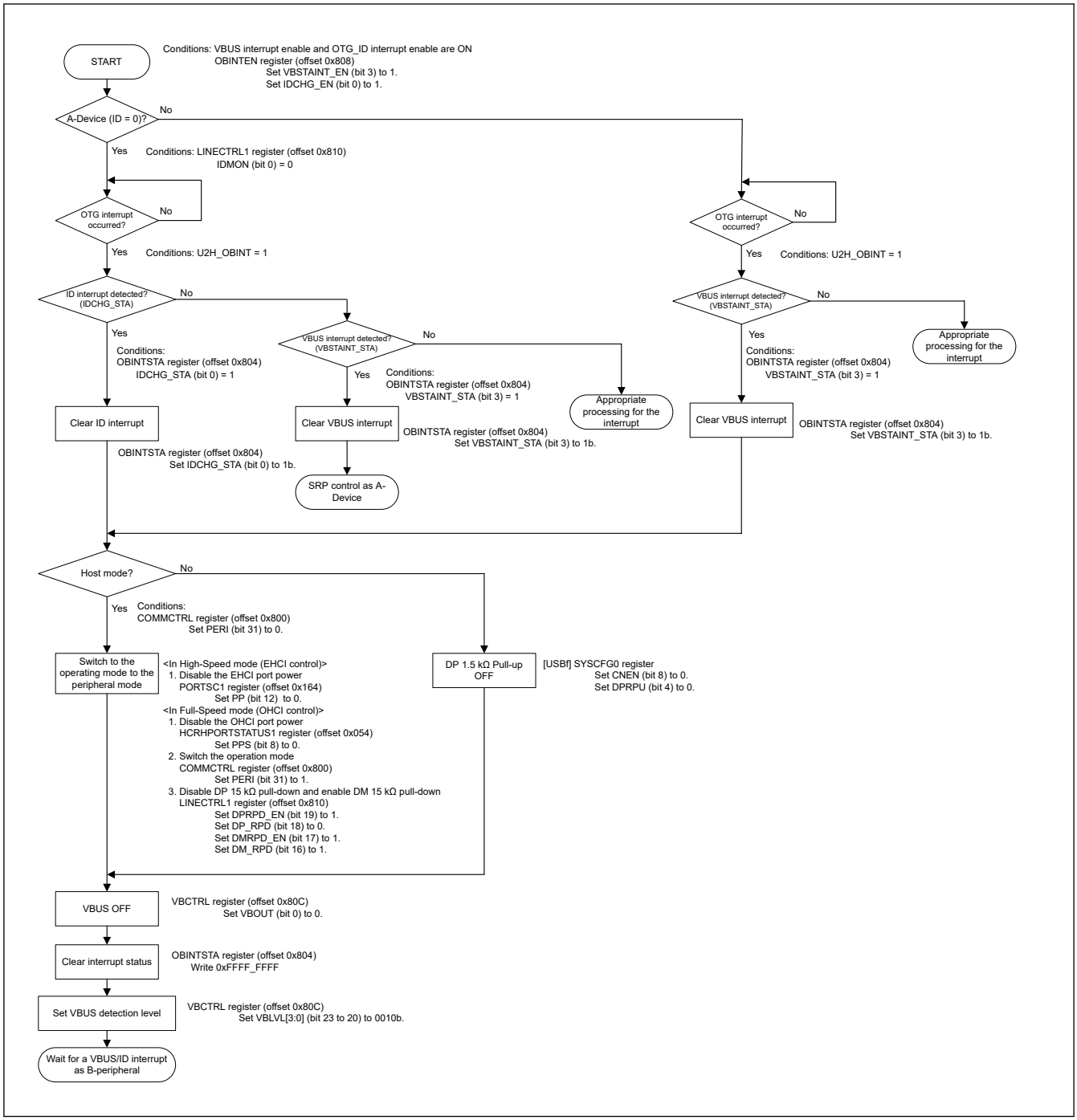


Figure 31.11 Disconnect detection procedure

31.8.6 HNP Control Procedure (Conforming to OTG Revision 2.0)

31.8.6.1 HNP Control Procedure for A-Device (Switching from A-Host to A-Peripheral)

Figure 31.12 shows the control procedure after a request from B-device for the host role is detected from the result of HNP polling for the B-device.

For HNP Control in High-Speed Mode (controlled by EHCI):

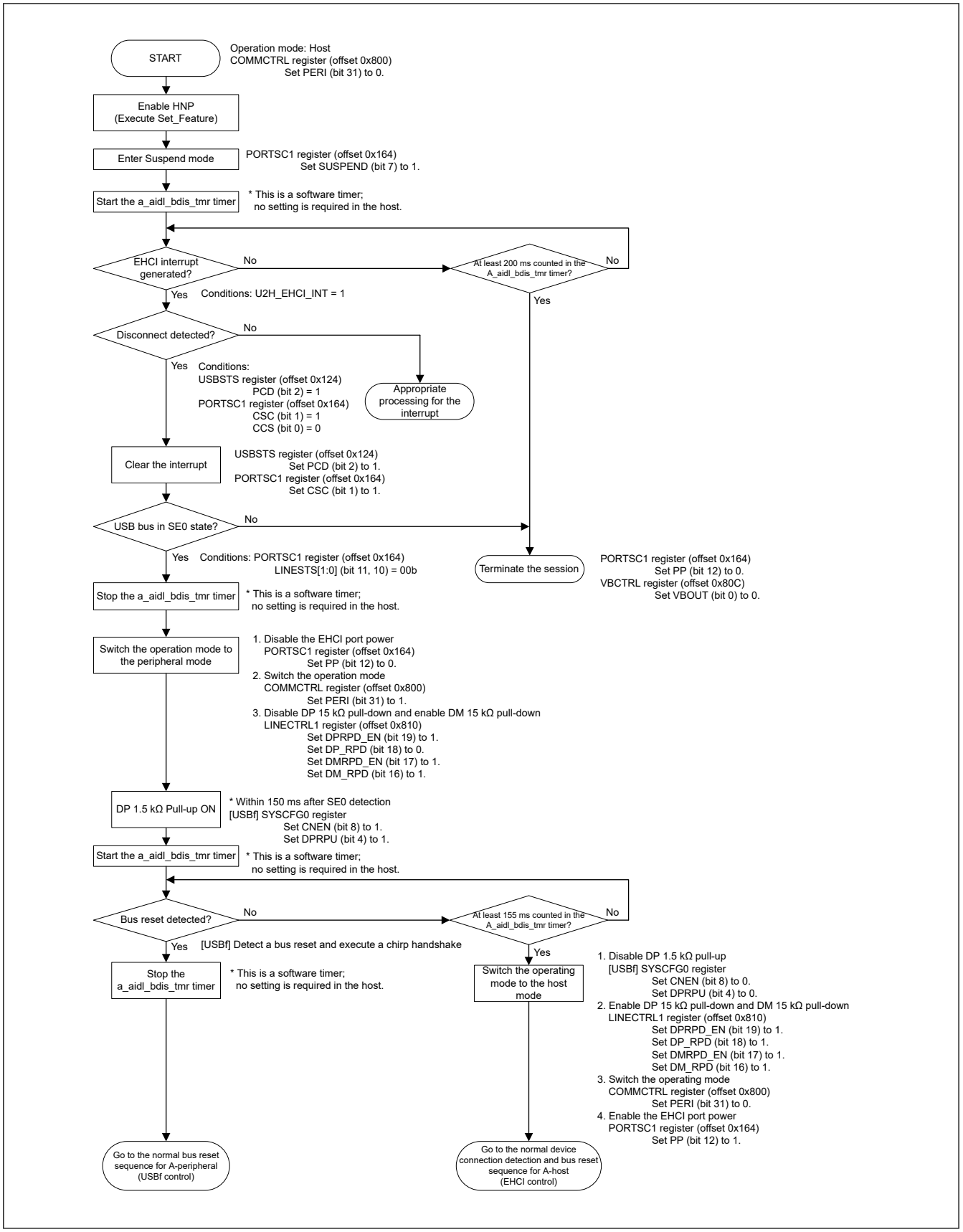


Figure 31.12 HNP control procedure for A-device (switching from A-host to A-peripheral in High-Speed mode)

For HNP control in Full-Speed mode (controlled by OHCI):

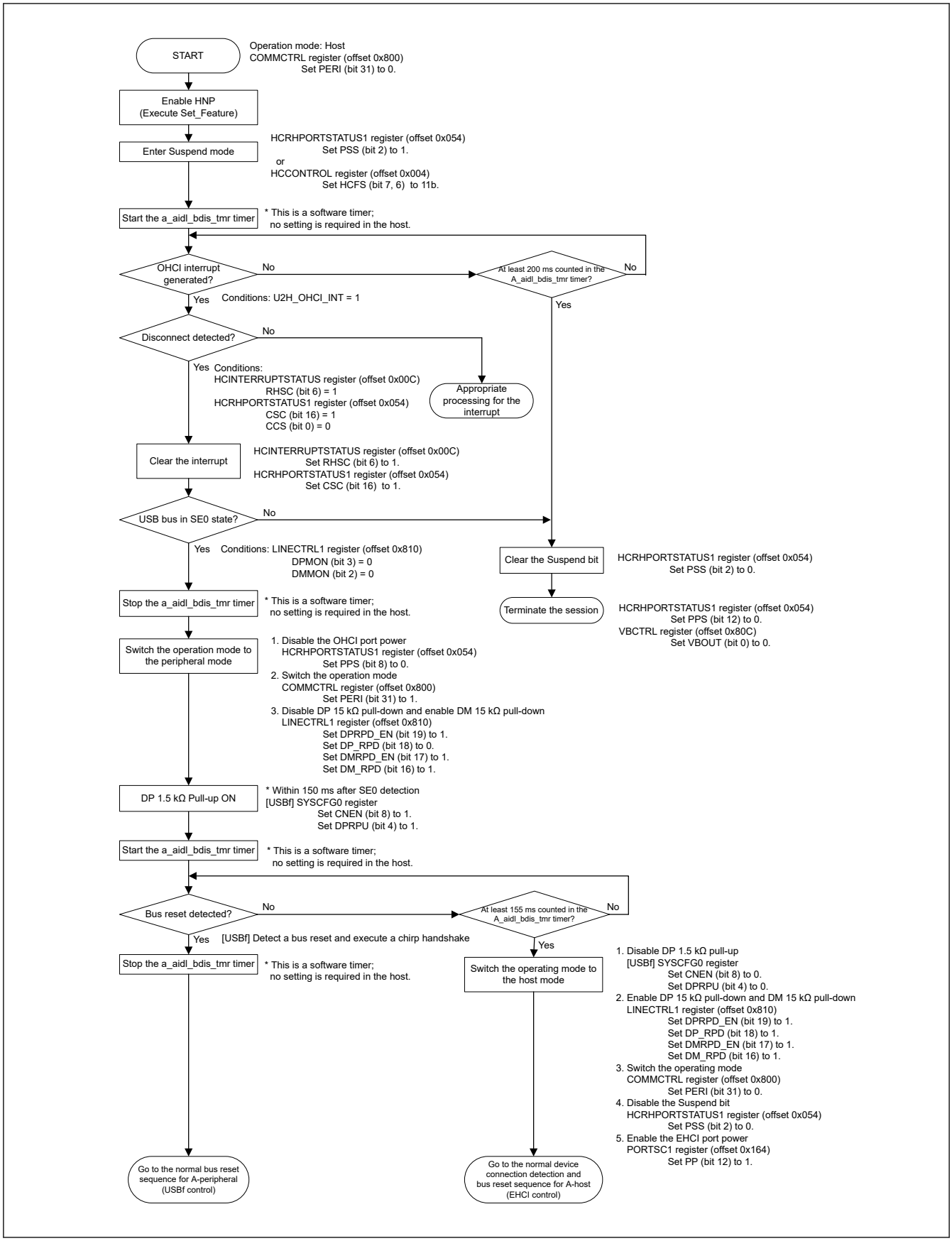


Figure 31.13 HNP control procedure for A-device (switching from A-host to A-peripheral in Full-Speed mode)

31.8.6.2 HNP Control Procedure for A-Device (Switching from A-Peripheral to A-Host)

Assumed conditions:

Even for FS OTG, EHCI is assumed to be the port owner after switching of the role from peripheral to host. Therefore, the procedure for switching from A-peripheral to A-host is not separately provided for each of OHCI and EHCI.

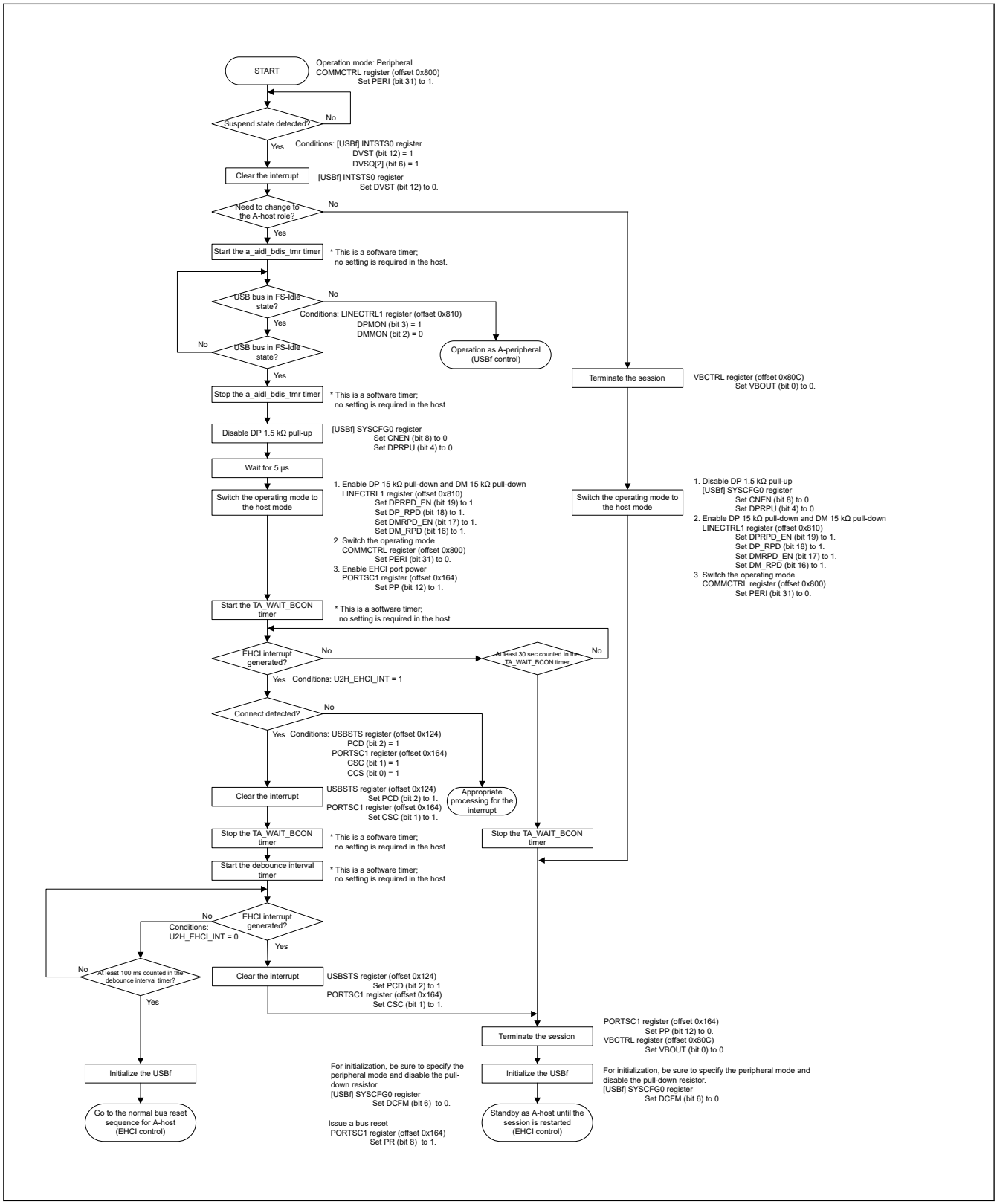


Figure 31.14 HNP control procedure for A-device (switching from A-periperal to A-host)

31.8.6.3 HNP Polling Procedure for A-Device

Figure 31.15 shows the procedure to check whether B-peripheral is requesting the host role when this LSI is an A-host.

For HNP polling in High-Speed mode (controlled by EHCI):

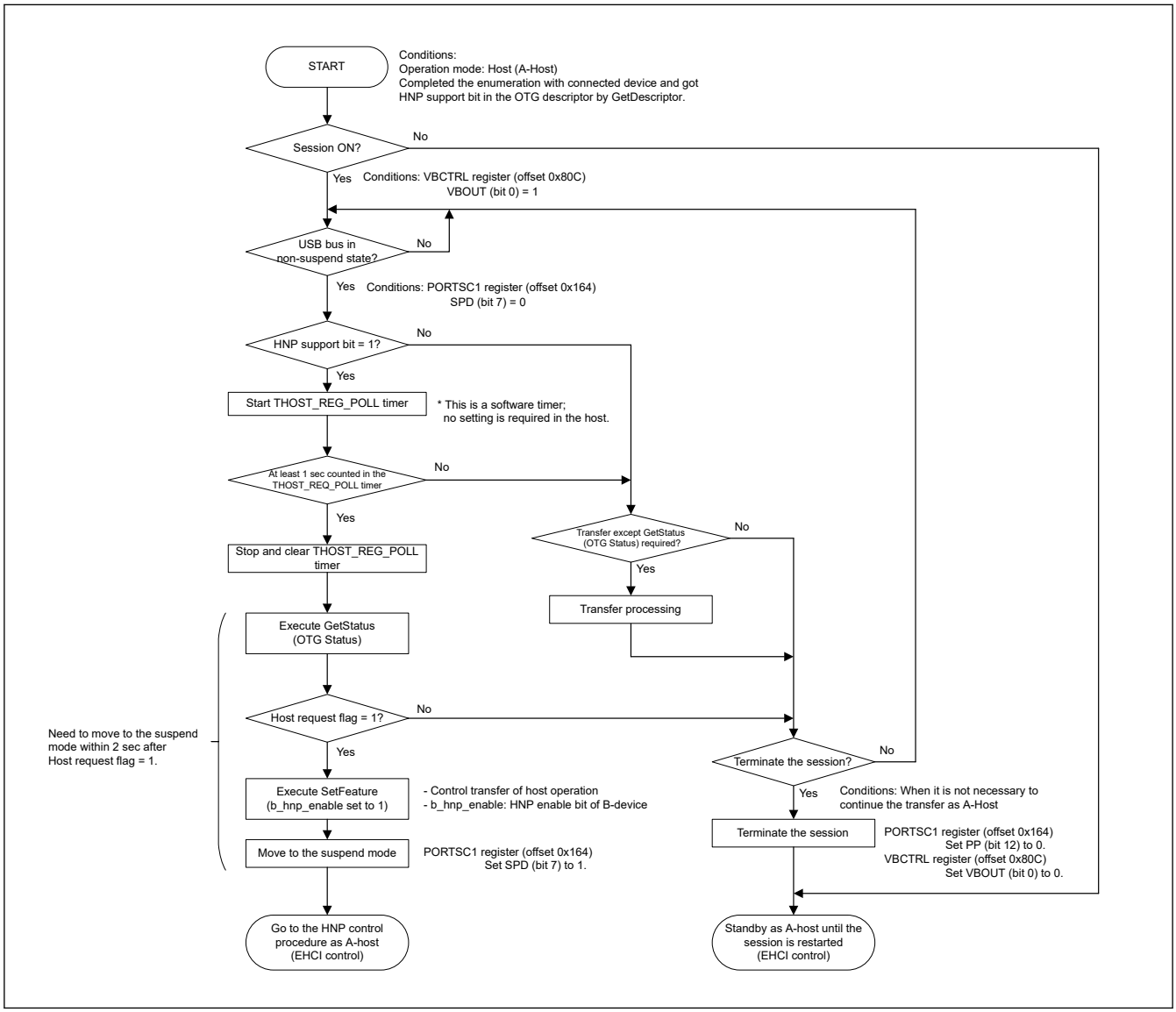


Figure 31.15 HNP polling procedure for A-device

For HNP polling in Full-Speed mode (controlled by OHCI):

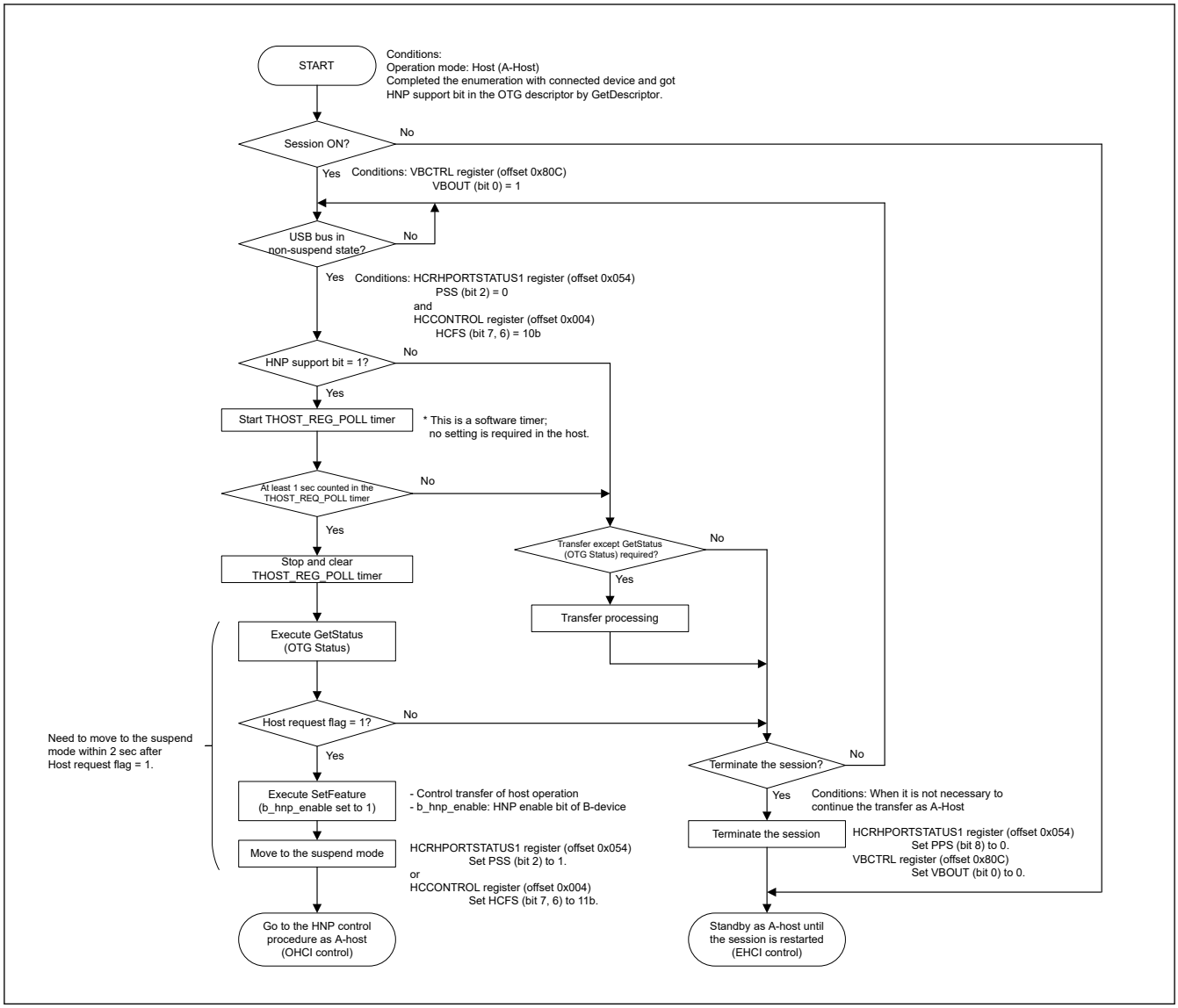


Figure 31.16 HNP polling procedure for A-device

31.8.6.4 HNP Control Procedure for B-Device (Switching from B-Peripheral to B-Host)

Figure 31.17 shows the procedure after the A-device issues set feature.

Assumed conditions:

Even for FS OTG, EHCI is assumed to be the port owner after switching the role from peripheral to host. Therefore, the procedure for switching from the B-peripheral to the B-host is not separately provided for each of OHCI and EHCI.

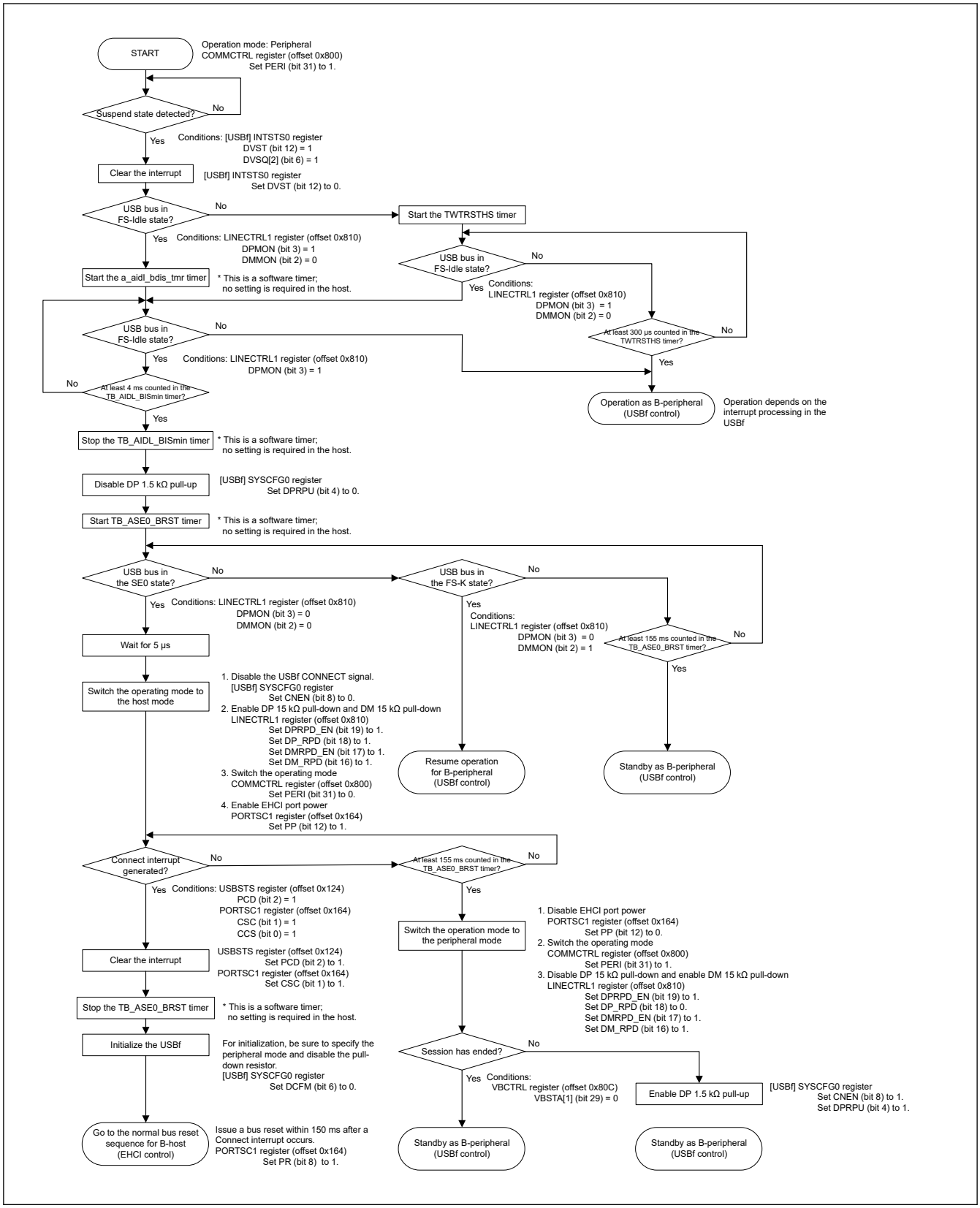


Figure 31.17 HNP control procedure for B-device (switching from B-periperal to B-host)

31.8.6.5 HNP Control Procedure for B-Device (Switching from B-Host to B-Peripheral)

Figure 31.18 shows the procedure after no transfer is performed by the B-host.

Assuming the control of ID and VBUS

VBUS control: VBCTRL.VBOUT

ID detection: Power-supply IC

For HNP control in High-Speed mode (controlled by EHCI):

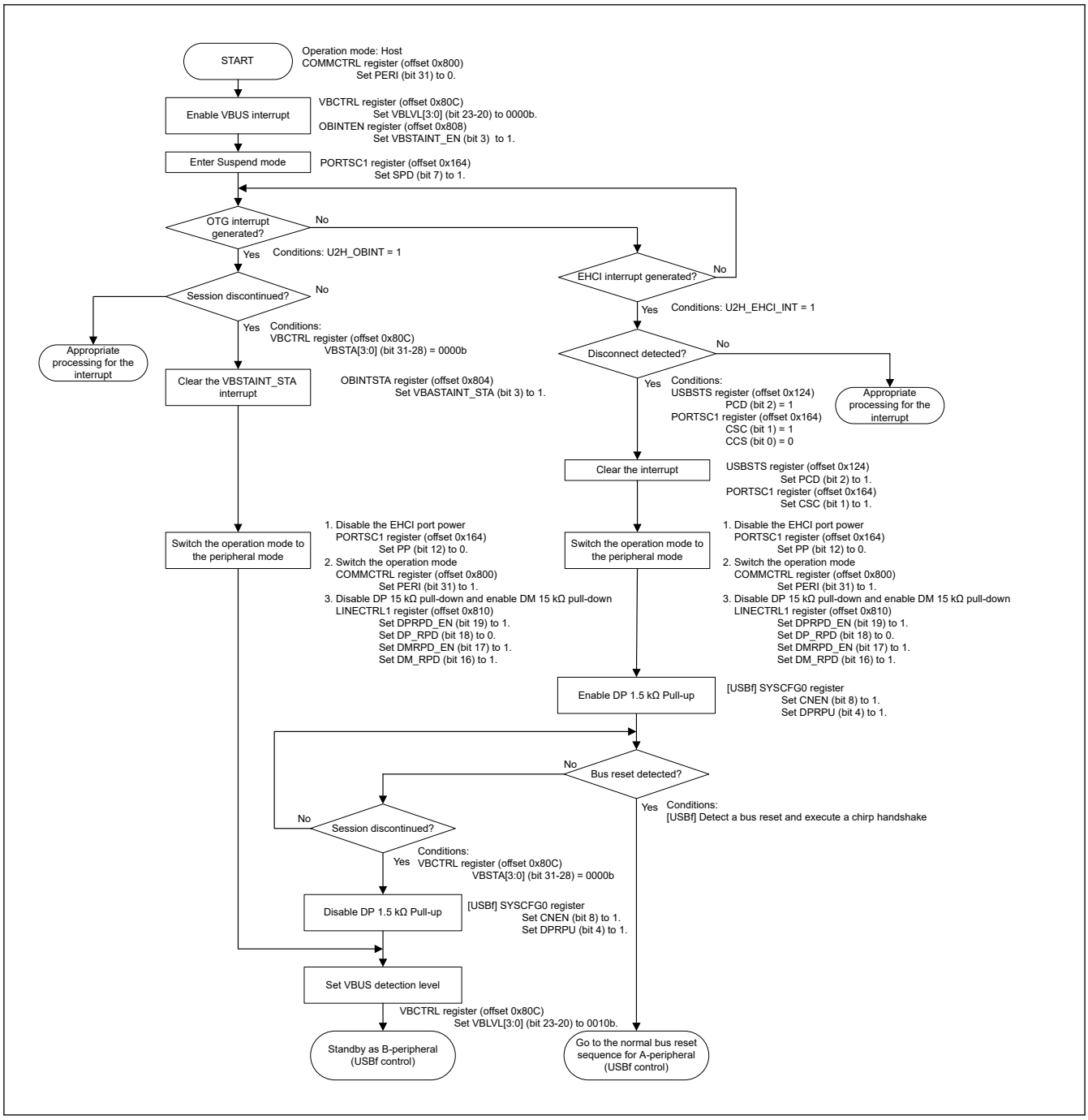


Figure 31.18 HNP control procedure for B-device (switching from B-host to B-peripheral in High-Speed mode)

For HNP control in Full-Speed mode (controlled by OHCI):

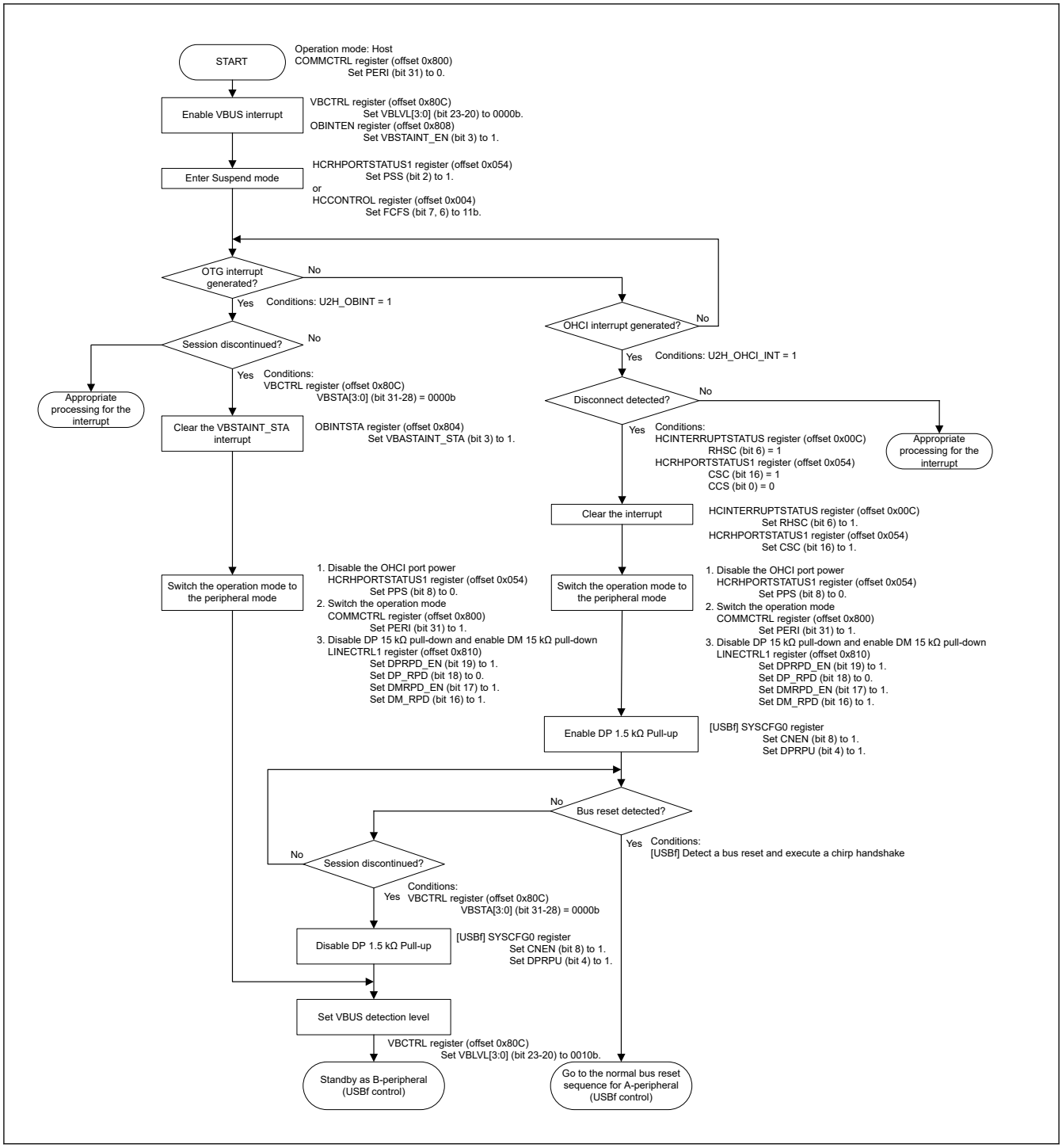


Figure 31.19 HNP control procedure for B-device (switching from B-host to B-peripheral in Full-Speed mode)

31.8.7 SRP Control Procedure (Conforming to OTG Revision 2.0)

31.8.7.1 SRP Control Procedure for A-Device

Figure 31.20 shows the procedure starting with the step for terminating a session because the B-device issues no request for the host role and the A-device does not need to transfer data.

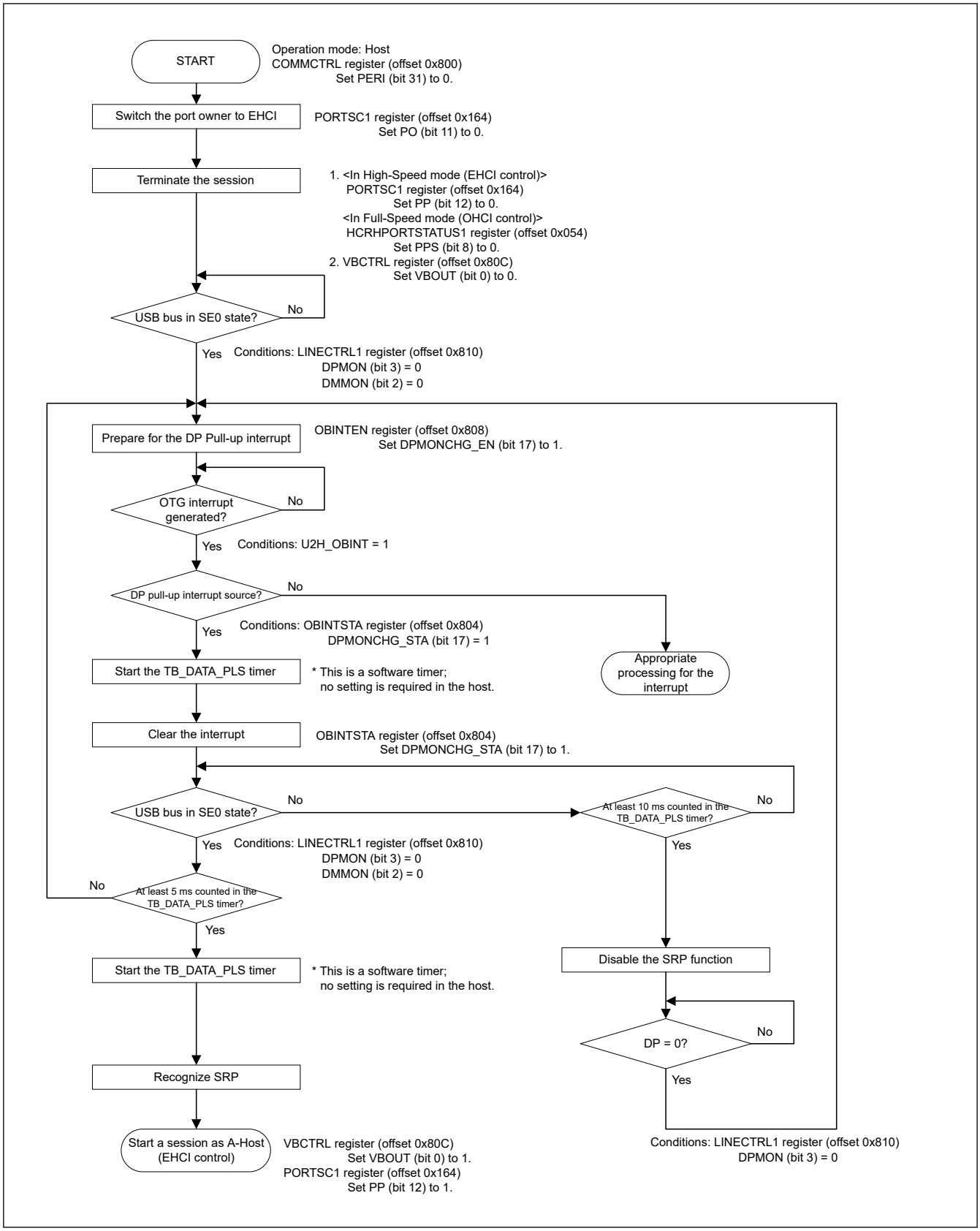


Figure 31.20 SRP control procedure for A-device

31.8.7.2 SRP Control Procedure for B-Device

Figure 31.21 shows the procedure from the standby state for a termination request of a session from the A-device.

Assuming the control of ID and VBUS

VBUS control: VBCTRL.VBOUT

ID detection: Power-supply IC

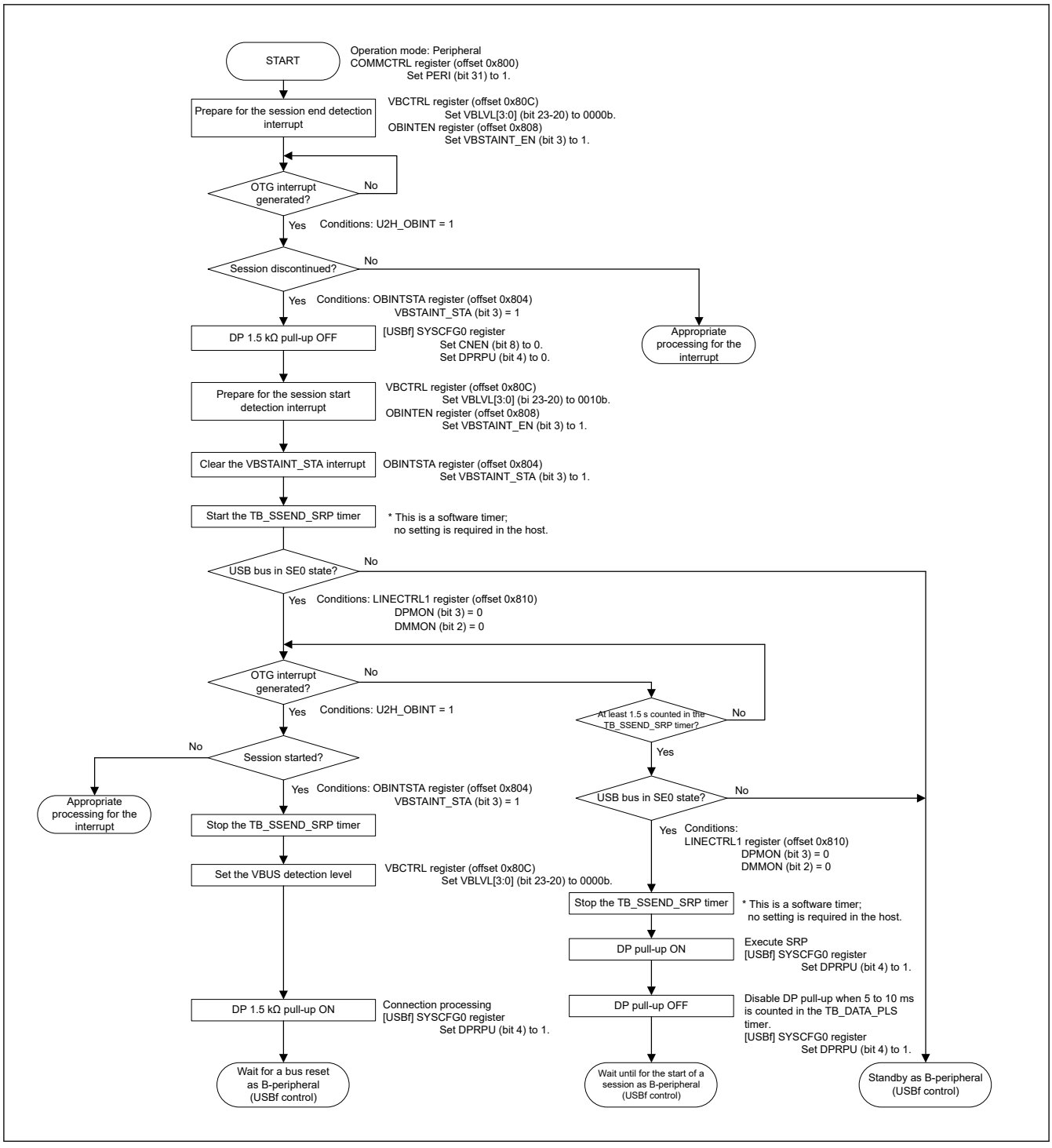


Figure 31.21 SRP control procedure for B-device

32. USB 2.0 HS Function Module (USBf)

32.1 Overview

The USB module of this LSI is a dual-role device that has USB 2.0 host and function modules.

This LSI has a single USB port for two functions for host and function. These functions can be switched by the common control register (COMMCTRL).

Since operation as a host controller or a function controller are exclusive of each other, dynamic switching between the types of operation is not possible.

This section describes function controller operation.

The USBf module supports high-speed transfer and full-speed transfer defined by the USB 2.0 specification.

This LSI supports all transfer types that are defined in the USB specification. It incorporates an 8 KB buffer memory for data transfer and can use up to 10 pipes. Pipes 1 to 9 can be assigned an arbitrary endpoint number according to the user system.

Table 32.1 Specifications of the USB Module

Item	Description
For High-Speed USB	• Built-in USB function controller
For all types of USB transfer	• Support for all types of USB transfer: – Control transfer – Bulk transfer – Interrupt transfer (high-bandwidth transfers not supported) – Isochronous transfer (high-bandwidth transfers not supported)
Bus interface	• Two DMA interfaces available – The dedicated AHB master interface for each DMA channel • High-speed data transfer for access to the internal FIFOs at 60 MB/second (when the bus width is 32 bits)
Pipe configuration	• 8 KB of buffer memory for USB communications • Up to 10 pipes can be selected (including the default control pipe) • Programmable pipe configuration • Any endpoint number can be assigned to pipes 1 to 9 • Transfer conditions that can be set for each pipe are listed below. Pipe 0: Control transfer, single buffer fixed at 64 bytes Pipes 1 and 2: Bulk transfer or isochronous transfer can be selected, continuous transfer mode, programmable buffer size (specifiable as up to 2 KB, double buffer is also specifiable) Pipes 3 to 5: Bulk transfer, continuous transfer mode, programmable buffer size (specifiable as up to 2 KB, double buffer is also specifiable) Pipes 6 to 9: Interrupt transfer, single buffer fixed at 64 bytes
Features of function controller operation	• High-speed (480 Mbps) and full-speed transfer (12 Mbps) are supported • Automatic recognition of high-speed operation or full-speed operation based on automatic response to the reset handshake • Control transfer stage monitoring • Monitoring of device state • Automatic response to SET_ADDRESS requests • NAK response interrupt function (NRDY) • SOF interpolation
Other functions	• Judging the end of transfer on the basis of a transaction count • Adjustable timing for BRDY interrupt event notification (BFRE) • Automatic clearing of the buffer memory after the data for the pipe specified in the DxFIFO port has been read (DCLRM) • NAK setting for response PID generated by end of transfer (SHTNAK)

Figure 32.1 is a block diagram of the USB module.

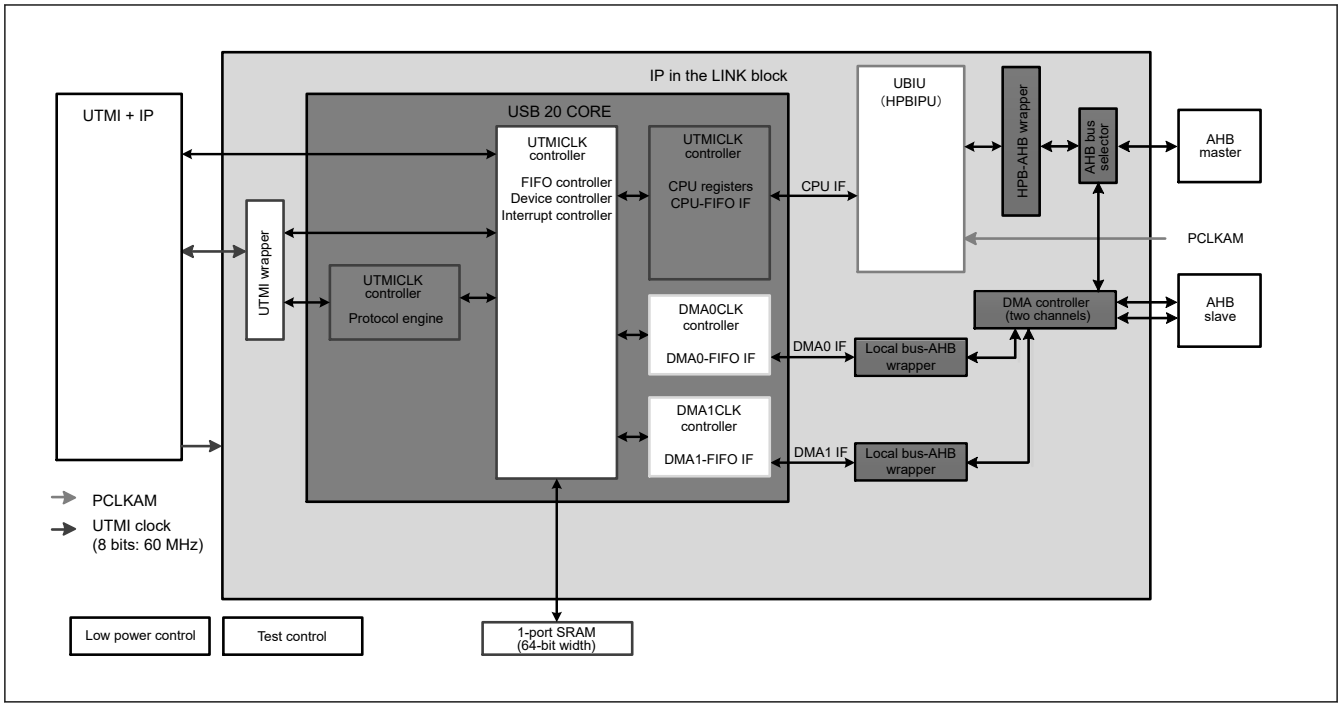


Figure 32.1 Block diagram of the USB module

Data transfer between this controller and the host controller connected to the USB bus uses buffer memory assigned to each pipe. For communication between this controller and the host controller, this controller converts the data stored in buffer memory into USB data packets and outputs them to the USB bus serially, and also inputs data packets from the USB bus and stores them in buffer memory.

32.1.1 Identification of the USB Transfer Speed

The hardware can automatically identify the USB transfer speed.

32.1.2 Bus Interface

(1) How to access the FIFO buffer memory

The following two types of access are available for the FIFO buffer memory for USB data transfers. To read from or write to the FIFO buffer memory, access (read or write) the FIFO ports from the CPU (DMAC).

Two channels of DMAC are provided.

1. CPU access
   Specify a FIFO port address and write data to the FIFO buffer memory or read data from the FIFO buffer memory.
2. DMA access
   Specify the dedicated DMAC and write data to the FIFO buffer memory or read data from the FIFO buffer memory.

USB data transfer is done in little endian. The byte endian swap function is available for FIFO port access; for 16 bit or 32 bit access, the endian can be switched through register settings.

32.1.3 USB Event

This controller sends an interrupt to the ICU to notify an event in USB operation.

Sending interrupts for notification can be enabled or disabled separately for each interrupt type and source through software settings.

32.1.4 USB Data Transfer

This controller performs all types of USB data transfer: control transfer, bulk transfer, interrupt transfer, and isochronous transfer. The following numbers of pipes are available for each transfer type.

- One pipe dedicated to control transfer
- Four pipes dedicated to interrupt transfer
- Three pipes dedicated to bulk transfer
- Two pipes selectively used for bulk transfer or isochronous transfer

Each pipe should be set up as necessary for the desired USB transfer in accordance with the user system; for example, transfer type, endpoint number, or maximum packet size.

This controller has an 8 KB buffer memory. For pipes dedicated for bulk transfer or selective pipes for bulk transfer or isochronous transfer, buffer memory assignment, buffer operating mode setting, or other necessary settings should be made in accordance with the user system. By setting the buffer operating mode, such as double buffer structure or continuous data packet transfer mode, fast data transfer is achieved with fewer interrupts.

CPU and DMA controller access to the buffer memory is made through three FIFO port registers.

32.1.5 DMA Transfer

This controller provides two channels of the DMA interface having the following functions.

- DMA transaction mode: Fetching in both register and link modes are supported
- Transfer size: A transfer size from 1 to 128 bytes can be selected separately for the transfer source and transfer destination
- Skip (scatter/gather) function: The access size and skip size can be specified separately for the transfer source and destination
- Suspend function: A running DMA transaction can be suspended temporarily
- Interval function: The interval of DMA transfers can be specified to control the bus occupancy

32.2 Register Map

Table 32.2 USBf register map (1 of 3)

Address	Register symbol	Register name	Write protection
0x9204_1000	SYSCFG0	System Configuration Control Register 0	—
0x9204_1002	SYSCFG1	System Configuration Control Register 1	—
0x9204_1004	SYSSTS0	System Configuration Status Register	—
0x9204_1008	DVSTCTR0	Device State Control Register 0	—
0x9204_100C	TESTMODE	USB Test Mode Register	—
0x9204_1014	CFIFO	CFIFO Port Register	—
0x9204_1018 + 0x4 × n	DnFIFO	DnFIFO Port Register (n = 0, 1)	—
0x9204_1020	CFIFOSEL	CFIFO Port Select Register	—
0x9204_1022	CFIFOCTR	CFIFO Port Control Register	—
0x9204_1028 + 0x4 × n	DnFIFOSEL	DnFIFO Port Select Register (n = 0, 1)	—
0x9204_102A + 0x4 × n	DnFIFOCTR	DnFIFO Port Control Register (n = 0, 1)	—
0x9204_1030	INTENB0	Interrupt Enable Register 0	—
0x9204_1032	INTENB1	Interrupt Enable Register 1	—
0x9204_1036	BRDYENB	BRDY Interrupt Enable Register	—
0x9204_1038	NRDYENB	NRDY Interrupt Enable Register	—
0x9204_103A	BEMPENB	BEMP Interrupt Enable Register	—
0x9204_103C	SOFCFG	SOF Pin Configuration Register	—
0x9204_1040	INTSTS0	Interrupt Status Register 0	—
0x9204_1042	INTSTS1	Interrupt Status Register 1	—
0x9204_1046	BRDYSTS	BRDY Interrupt Status Register	—
0x9204_1048	NRDYSTS	NRDY Interrupt Status Register	—

Table 32.2 USBf register map (2 of 3)

Address	Register symbol	Register name	Write protection
0x9204_104A	BEMPSTS	BEMP Interrupt Status Register	—
0x9204_104C	FRMNUM	Frame Number Register	—
0x9204_104E	UFRMNUM	Frame Number Register	—
0x9204_1050	USBADDR	USB Address Register	—
0x9204_1054	USBREQ	USB Request Type Register	—
0x9204_1056	USBVAL	USB Request Value Register	—
0x9204_1058	USBINDX	USB Request Index Register	—
0x9204_105A	USBLENG	USB Request Length Register	—
0x9204_105C	DCPCFG	DCP Configuration Register	—
0x9204_105E	DCPMAXP	DCP Maximum Packet Size Register	—
0x9204_1060	DCPCTR	DCP Control Register	—
0x9204_1064	PIPESEL	Pipe Window Select Register	—
0x9204_1068	PIPECFG	Pipe Configuration Register	—
0x9204_106A	PIPEBUF	Pipe Buffer Specification Register	—
0x9204_106C	PIPEMAXP	Pipe Maximum Packet Size Register	—
0x9204_106E	PIPEPERI	Pipe Timing Control Register	—
0x9204_1070 + 0x2 × (n - 1)	PIPEnCTR	PIPEn Control Register (n = 1 to 5)	—
0x9204_107A + 0x2 × (n - 6)	PIPEnCTR	PIPEn Control Register (n = 6 to 9)	—
0x9204_1090 + 0x4 × (n - 1)	PIPEnTRE	PIPEn Transaction Counter Enable Register (n = 1 to 5)	—
0x9204_1092 + 0x4 × (n - 1)	PIPEnTRN	PIPEn Transaction Counter Register (n = 1 to 5)	—
0x9204_1102	LPSTS	Low Power Status Register	—
0x9204_1400 + 0x40 × n 0x9204_140C + 0x40 × n	N0SA_n N1SA_n	Next Source Address Register n (n = 0, 1)	—
0x9204_1404 + 0x40 × n 0x9204_1410 + 0x40 × n	N0DA_n N1DA_n	Next Destination Address Register n (n = 0, 1)	—
0x9204_1408 + 0x40 × n 0x9204_1414 + 0x40 × n	N0TB_n N1TB_n	Next Transaction Byte Register n (n = 0, 1)	—
0x9204_1418 + 0x40 × n	CRSA_n	Current Source Address Register n (n = 0, 1)	—
0x9204_141C + 0x40 × n	CRDA_n	Current Destination Address Register n (n = 0, 1)	—
0x9204_1420 + 0x40 × n	CRTB_n	Current Transaction Byte Register n (n = 0, 1)	—
0x9204_1424 + 0x40 × n	CHSTAT_n	Channel Status Register n (n = 0, 1)	—
0x9204_1428 + 0x40 × n	CHCTRL_n	Channel Control Register n (n = 0, 1)	—
0x9204_142C + 0x40 × n	CHCFG_n	Channel Configuration Register n (n = 0, 1)	—
0x9204_1430 + 0x40 × n	CHITVL_n	Channel Interval Register n (n = 0, 1)	—
0x9204_1434 + 0x40 × n	CHEXT_n	Channel Extension Register n (n = 0, 1)	—
0x9204_1438 + 0x40 × n	NXLA_n	Next Link Address Register n (n = 0, 1)	—
0x9204_143C + 0x40 × n	CRLA_n	Current Link Address Register n (n = 0, 1)	—
0x9204_1600 + 0x20 × n	SCNT_n	Source Continuous Register n (n = 0, 1)	—
0x9204_1604 + 0x20 × n	SSKP_n	Source Skip Register n (n = 0, 1)	—
0x9204_1608 + 0x20 × n	DCNT_n	Destination Continuous Register n (n = 0, 1)	—
0x9204_160C + 0x20 × n	DSKP_n	Destination Skip Register n (n = 0, 1)	—
0x9204_1700	DCTRL	DMA Control Register	—

Table 32.2 USBf register map (3 of 3)

Address	Register symbol	Register name	Write protection
0x9204_1704	DSCITVL	Descriptor Interval Register	—
0x9204_1710	DSTAT_EN	DMA Status EN Register	—
0x9204_1714	DSTAT_ER	DMA Status ER Register	—
0x9204_1718	DSTAT_END	DMA Status END Register	—
0x9204_171C	DSTAT_TC	DMA Status TC Register	—
0x9204_1720	DSTAT_SUS	DMA Status SUS Register	—

Table 32.3 USBf related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
All the registers	—	MSTPCRE.MSTPCRE08	SLVACCCTL4.USB_SL

32.3 Register Descriptions

32.3.1 System Configuration Control

32.3.1.1 SYSCFG0 : System Configuration Control Register 0

Base address: USBF = 0x9204_1000

Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CNEN	HSE	—	DRPD	DPRPU	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USBE	USB Block Operation Enable Enables or disables operation of the USB block. 0: USB block operation is disabled. 1: USB block operation is enabled.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	DPRPU	D+ Line Resistor Control Enables or disables pulling up the D+ line in function controller operation. 0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
5	DRPD	D+/D− Line Resistor Control Always set 0 at initialization.	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HSE	High-Speed Operation Enable Enables or disables high-speed operation. 0: High-speed operation is disabled (full-speed). 1: High-speed operation is enabled (the controller detects the communication speed)	R/W
8	CNEN	Single-End Receiver Operation Enable Enables or disables the single-end receiver to operate. 0: Single-end receiver operation is disabled. 1: Single-end receiver operation is enabled.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

Writing to this register is possible even while the UTMI clock (output of the clock signal from the PLL in the USB) is stopped. Note however that any values that are set while the UTMI clock is stopped are reflected after oscillation of the UTMI clock is restarted.

USBE bit (USB Block Operation Enable)

The setting of the USBE bit enables or disables operation of the USB block.

When the USBE bit is modified from 1 to 0, this controller initializes the bits listed in Table 32.4.

Table 32.4 Registers initialized by writing USBE = 0

Register Name	Bit Name
SYSSTS0	LNST
DVSTCTR0	RHST
INTSTS0	DVSQ
USBADDR	USBADDR
USBREQ	bRequest bmRequestType
USBVAL	wValue
USBINDX	wIndex
USBLENG	wLength

The value of this bit must be changed when SUSPM = 1 and after the UTMI clock oscillation is started.

DRPD and DPRPU bit (D+/D− Line Resistor Control)

Settings related to USB data bus resistance are given in Table 32.5, USB Data Bus Resistance Control. Use the DRPD and DPRPU bits to select USB data bus resistance.

Table 32.5 USB data bus resistance control

Settings		USB data bus resistance control		
DRPD	DPRPU	D− Line	D+ Line	Remarks
0	0	Open	Open	
0	1	Open	Pull-up	Make these settings when the module is used as a function controller.
1	0	Pull-down	Pull-down	Setting prohibited
1	1	Pull-down	Pull-up	Setting prohibited

D+ pull-up resistance control for function controller operation (DPRPU)

When this bit set to 1, this controller pulls up the D+ line to 3.3V and can notify the USB host that the function module is attached.

The controller cancels pulling-up of the D+ line by modifying this bit from 1 to 0, and the state for the USB host can be shown as detached.

HSE bit (High-Speed Operation Enable)

Setting this bit to 1 enables high-speed operation. Setting the HSE bit to 1 lets this controller perform high-speed or fullspeed operation based on the results of reset handshake.

Setting the HSE bit to 0 lets this controller perform full-speed operation.

On the other hand, setting the HSE bit to 1 lets this controller perform the reset handshake protocol and then perform high-speed or full-speed operation automatically according to the results.

This bit can be modified when DPRPU = 0.

CNEN bit (Single-End Receiver Operation Enable)

Setting this bit to 1 enables the single-end receiver to operate. This bit is used to prevent inrush current damage due to the floating state of the single-end receiver when it is detached. Furthermore, this bit makes it possible to see the LNST bits.

Set this bit to 1 when VBUS is detected by a VBUS interrupt, and clear this bit to 0 when VBUS is removed.

32.3.1.2 SYSCFG1 : System Configuration Control Register 1

Base address: USBF = 0x9204_1000

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	AWAIT[5:0]						—	—	BWAIT[5:0]					
Value after reset:	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
5:0	BWAIT[5:0]	CPU Bus Access Wait Specification Specify the number of wait cycles for access to this controller. 0x00: 0 wait cycles (2 access cycles) ⋮ 0x02: 2 wait cycles (4 access cycles) ⋮ 0x04: 4 wait cycles (6 access cycles) ⋮ 0x0F: 15 wait cycles (17 access cycles) (value after reset) ⋮ 0x3F: 63 wait cycles (65 access cycles)	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	AWAIT[5:0]	AHB-DMA Bridge Bus Access Wait Specification Specify the number of wait cycles for access to this controller. 0x00: 0 wait cycles (2 access cycles) ⋮ 0x02: 2 wait cycles (4 access cycles) ⋮ 0x04: 4 wait cycles (6 access cycles) ⋮ 0x0F: 15 wait cycles (17 access cycles) (value after reset) ⋮ 0x3F: 63 wait cycles (65 access cycles)	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

BWAIT[5:0] bits (CPU Bus Access Wait Specification)

For continuous access to the registers at addresses of this controller beginning at 0x80201004, at least 67 ns must be secured.

To satisfy this restriction, control wait cycles using the frequency of CPU clock (PCLKAM). The value after a reset is the maximum value (17 clock cycles), so select an appropriate value no greater than this. For this LSI, more than 9 wait cycles (11 access cycles) is needed.

This setting is the same as that for waiting in access to a FIFO port register. The maximum speeds of access to the FIFO ports are as follows:

MBW = 10b (32 bit width): Max 60 MB/sec

MBW = 01b (16 bit width): Max 30 MB/sec

MBW = 00b (8 bit width): Max 15 MB/sec

AWAIT[5:0] bits (AHB-DMA Bridge Bus Access Wait Specification)

These bits specify the number of wait cycles for access to the AHB-DFIFO0/AHB-DFIFO1 bus.

Restrictions for access cycles are the same as those for the CPU bus access wait specification bits (BWAIT).

32.3.1.3 SYSSTS0 : System Configuration Status Register

Base address: USBF = 0x9204_1000

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LNST[1:0]	
Value after reset:	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	LNST[1:0]	USB Data Line Status Monitor Indicate the state of the USB line.	R
15:2	—	These bits are read as 0.	R

Note: When a USB bus reset is detected, this register value is undefined.

LNST[1:0] bits (USB Data Line Status Monitor)

The states of USB data bus line of this controller is shown in Table 32.6. The controller monitors the states of USB data bus lines (D+ and D− lines) in the LNST bits of the SYSSTS0 register.

Reference to the LNST bits must be made following attach processing (DPRPU = 1) after USBE has been set to 1.

Table 32.6 States of USB data bus line

LNST[1]	LNST[0]	Full-speed operations	High-speed operations	Chirp operations*1
0	0	SE0	Squelch*2	Squelch*2
0	1	J state	Unsquelch*3	Chirp J*4
1	0	K state	Invalid	Chirp K*5
1	1	SE1	Invalid	Invalid

Note 1. Chirp: Reset handshake protocol being executed in high-speed operations enabled state (HSE = 1)
Note 2. Squelch: SE0 or idle state
Note 3. Unsquelch: High-speed J state or high-speed K state
Note 4. Chirp J: Chirp J-state
Note 5. Chirp K: Chirp K-state

32.3.2 USB Signal Control

32.3.2.1 DVSTCTR0 : Device State Control Register 0

Base address: USBF = 0x9204_1000

Offset address: 0x08

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	WKUP	—	—	—	—	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RHST[2:0]	Reset Handshake Indicate the state of reset handshake.	R
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	WKUP	Wakeup Output Enables or disables the remote wakeup (resume signal output). Only 1 can be written to this bit. 0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W

Bit	Symbol	Function	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, The WKUP is set to 0 and other bits are undefined.

RHST[2:0] bits (Reset Handshake)

These bits indicate the result of reset handshake. Table 32.7 lists the results of reset handshake.

Table 32.7 State of Reset Handshake

Bus state	RHST bit value
Powered or no connection	000b
Reset handshake in progress	100b
Full-speed connection	010b
High-speed connection	011b

When the HSE bit is set to 1, the RHST bits indicate 100b when this controller detects a USB bus reset. Then, these bits indicate 011b when this controller outputs Chirp-K and detects Chirp-JK from the USB host three times. If the connection speed is not fixed to high speed within 2.5 ms after Chirp-K output, these bits indicate 010b.

When the HSE bit is set to 0, these bits indicate 010b when the controller detects a USB bus reset.

A DVST interrupt is generated when the value of the RHST bits is 010b or 011b after the USB bus reset is detected by the controller.

WKUP bit (Wakeup Output)

With the WKUP bit set to 1, this controller outputs the remote wakeup signal to the USB bus.

This controller controls the output time of a remote wakeup signal. When this bit is set to 1 by software, the controller clears this bit to 0 after outputting the 10 ms K-state.

According to the USB Specifications, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If this controller writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.

Do not write 1 to this bit unless the device is in the suspended state (bits DVSQ = 1xxb) and the USB host enables the remote wakeup signal.

When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while the SUSPM bit = 1).

32.3.3 Test Mode

32.3.3.1 TESTMODE : USB Test Mode Register

Base address: USBF = 0x9204_1000

Offset address: 0x0C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	UTST[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	UTST[3:0]	Test Mode *See the detailed description below.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

UTST[3:0] bits (Test Mode)

Writing values to these bits allows this controller to output a USB test signal during high-speed operations. Table 32.8 shows the test mode operations of the controller.

Table 32.8 Test mode operations

Test mode	UTST settings
Normal operation	0000b
Test_J	0001b
Test_K	0010b
Test_SE0_NAK	0011b
Test_Packet	0100b
Test_Force_Enable	—
Reserved	0101b to 0111b

Write to these bits according to the SetFeature request from the USB host during high-speed communication.

This controller does not enter the suspended state while these bits are 0001b to 0100b.

To perform normal USB communications after the test mode is set, apply a hardware reset.

32.3.4 FIFO Ports

32.3.4.1 CFIFO, DnFIFO : FIFO Port Register (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x014 (CFIFO/CFIFOL/CFIFOLL)
0x016 (CFIFOH)
0x017 (CFIFOHH)
0x018 + 0x4 × n (DnFIFO/DnFIFOL/DnFIFOLL)
0x01A + 0x4 × n (DnFIFOH)
0x01B + 0x4 × n (DnFIFOHH)

Bit position:	31																												0
Bit field:	FIFOPORT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FIFOPORT[31:0]	FIFO Port Accessing these bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

The transmission/reception buffer of this controller is made up of a FIFO structure (FIFO buffer). Use the FIFO port register for access to the FIFO buffer. There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO ports. Also, there is a DxFIFO continuous transfer port for continuous transfer. Each FIFO port consists of a port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).

The FIFO ports have the following restrictions.

1. Access to an FIFO buffer for DCP should be through the CFIFO port.
2. Access to an FIFO buffer by DMA transfer when DFACC = 00b (cycle stealing mode) should be through the DxFIFO port.
3. Access to an FIFO buffer by DMA transfer when DFACC = 01b or 10b (16 byte or 32 byte continuous access mode) should be through the DxFIFO continuous transfer port.
4. The DxFIFO ports can also be accessed by the CPU.

5. When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed (when the DMA transfer function is used, etc.).
6. Registers containing the FIFO port do not affect the other FIFO ports.
7. Do not assign the same pipe to separate FIFO ports.
8. In the FIFO buffer state, there are two types of access rights: one assigned to the CPU, and the other to SIE. Access from the CPU is not possible when SIE has the rights to access the FIFO buffer.

FIFOPORT[31:0] bits (FIFO Port)

This controller accesses the FIFO buffer assigned to the pipe number written to the CURPIPE bits of the corresponding select registers (CFIFOSEL, D0FIFOSEL, and D1FIFOSEL) through access to any of these registers by software.

This register can only be accessed when the FRDY bit of each control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) indicates 1 (or while DnFIFOSEL.DREQE = 1).

The valid bits of this register depend on the settings of the corresponding MBW and BIGEND bits.

Table 32.9 to Table 32.11 list the valid bits.

Table 32.9 Endian operation in 32 bit access (MBW = 10)

BIGEND	b31 to b24	b23 to b16	b15 to b8	b7 to b0
0	N + 3 address	N + 2 address	N + 1 address	N + 0 address
1	N + 0 address	N + 1 address	N + 2 address	N + 3 address

Table 32.10 Endian operation in 16 bit access (MBW = 01)

BIGEND	b31 to b24	b23 to b16	b15 to b8	b7 to b0
0	Writing: Invalid Reading: Prohibited*1		N + 1 address	N + 0 address
1	N + 0 address	N + 1 address	Writing: Invalid Reading: Prohibited*1	

Note 1. Reading data from the invalid bits in word or byte units is prohibited.

Table 32.11 Endian operation in 8 bit Access (MBW = 00)

BIGEND	b31 to b24	b23 to b16	b15 to b8	b7 to b0
0	Writing: Invalid Reading: Invalid*1			N + 0 address
1	N + 0 address	Writing: Invalid Reading: Prohibited*1		

Note 1. Reading data from the invalid bits in word or byte units is prohibited.

32.3.4.2 CFIFOSEL : CFIFO Port Select Register

Base address: USBF = 0x9204_1000

Offset address: 0x20

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	—	—	MBW[1:0]		—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	FIFO Port Access Pipe Specification Specify the pipe number to access the CFIFO port. 0x0: DCP 0x1: Pipe 1 0x2: Pipe 2 ⋮ 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	ISEL	FIFO Port Access Direction when DCP is Selected Specifies access direction of the FIFO port when DCP is selected in CURPIPE bits. 0: Reading from the buffer memory 1: Writing to the buffer memory	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	BIGEND	FIFO Port Endian Control Specifies the CFIFO port byte endian. 0: Little endian 1: Big endian	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
11:10	MBW[1:0]	CFIFO Port Access Bit Width Specify the bit width for accessing the CFIFO port. 0 0: 8 bit width 0 1: 16 bit width 1 0: 32 bit width 1 1: Setting prohibited	R/W
13:12	—	These bits are read as 0. The write value should be 0.	R/W
14	REW	Buffer Pointer Rewind Specifies 1 when rewinding the buffer pointer. This bit is read as 0. 0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W
15	RCNT	Read Count Mode Specifies the read mode for the value in the DTLN bits in CFIFOCTR. 0: The DTLN bits are cleared when all of the received data has been read from the CFIFO. 1: The DTLN bits are decremented each time the received data is read from the CFIFO.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

Write the pipe number for the data to be read or written through the CFIFO port.

When modifying this bit field, write this bit first and then read this bit. Only when the written value matches the read one, proceed to the next process.

Do not specify the same pipe in the CURPIPE bits of the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

Even if the setting of these bits is modified during access to the FIFO buffer, the state of the FIFO buffer is retained, with continued access proceeding after these bits are re-set to the value before the modification.

ISEL bit (FIFO Port Access Direction when DCP is Selected)

To change this bit when the specified pipe is DCP, first write the data to this bit and then read it. Proceed to the next process after checking if the written value matches with the read value.

When the settings of this bit are modified during access to the FIFO buffer, access up to then is saved. Access to the buffer can be continued after rewriting the settings.

Settings in this bit should be made at the same as that of the CURPIPE bits.

BIGEND bit (FIFO Port Endian Control)

In this bit, set the byte endian for the CFIFO port.

For details, see the description of FIFO Port Bits (CFIFO, D0FIFO, and D1FIFO).

MBW[1:0] bits (CFIFO Port Access Bit Width)

In this bit field, set the bit width for accessing the CFIFO port.

When the pipe specified in the CURPIPE bits is in the receiving direction, if reading is started after setting this bit field, do not modify the value of the MBW bits until all data is read.

When the specified pipe is in the receiving direction, modify the value of the CURPIPE bit once and then set the CURPIPE and MBW bits at the same time.

For the procedure for changing the CURPIPE bit, see the description for the same.

When the specified pipe is in the transmitting direction, to switch the bit width from 8 bits to 16 or 32 bits or from 16 bits to 32 bits is not allowed while writing to the buffer memory is in progress.

With the 16 bit or 32 bit width setting, writing to odd bytes is possible by exercising byte access control.

REW bit (Buffer Pointer Rewind)

When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double-buffer operation, re-reading the currently-read FIFO buffer plane from the first data is allowed).

Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, be sure to check that FRDY is 1.

To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT bit (Read Count Mode)

When 0 is written to this bit, if all received data of the FIFO buffer assigned to the pipe specified in the CURPIPE bits (selected pipe) is read (when the data is read on one side of a double buffer), this controller clears the DTLN bits in the CFIFOCTR register to 0.

When 1 is written to this bit, the controller decrements the value of the DTLN bits in the CFIFOCTR register whenever the received data is read from the FIFO buffer assigned to the specified pipe.

32.3.4.3 DnFIFOSEL : DnFIFO Port Select Register (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x028 + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	DCLRM	DREQE	MBW[1:0]		—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	FIFO Port Access Pipe Specification 0x0: Not specified 0x1: Pipe 1 0x2: Pipe 2 ⋮ 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BIGEND	FIFO Port Endian Control Specifies the DnFIFO port byte endian. 0: Little endian 1: Big endian	R/W

Bit	Symbol	Function	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
11:10	MBW[1:0]	FIFO Port Access Bit Width Specify the bit width for accessing the FIFO port. 0 0: 8 bit width 0 1: 16 bit width 1 0: 32 bit width 1 1: Setting prohibited	R/W
12	DREQE	DMA Request Enable This bit is used to notify for DMAC. 0: DMA request is disabled 1: DMA request is enabled	R/W
13	DCLRM	Auto Buffer Memory Clear Mode after Specified Pipe Data is Read Enables or disables automatic clearing of the buffer memory after data has been read out using the selected pipe. 0: Auto buffer clear mode is disabled 1: Auto buffer clear mode is enabled	R/W
14	REW	Buffer Pointer Rewind Specifies 1 when rewinding the buffer pointer. This bit is read as 0. 0: The buffer pointer is not rewound 1: The buffer pointer is rewound	R/W
15	RCNT	Read Count Mode Specifies the read mode for the value in the DTLN bits in Dx_FIFOCTR. 0: The DTLN bits are cleared when all of the received data has been read. 1: The DTLN bits are decremented each time the received data is read.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

Write the pipe number for the data to be read or written through the DxFIFO port.

When modifying this bit field, first write a value to this bit field and then read it. Check if the write value matches the read value and then proceed to the next process.

Do not specify the same pipe in the CURPIPE bits of the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

Even if the setting of these bits is modified during access to the FIFO buffer, the state of the FIFO buffer is retained, with continued access proceeding after these bits are re-set to the value before the modification.

BIGEND bit (FIFO Port Endian Control)

Write the byte endian for the DxFIFO port in this bit.

For details, see the description of FIFO Port Bits (CFIFO, D0FIFO, and D1FIFO).

MBW[1:0] bits (FIFO Port Access Bit Width)

Write the bit width for accessing the DxFIFO port in this bit.

When the pipe specified in the CURPIPE bits is in the receiving direction, if reading is started after setting this bit field, do not modify the value of the MBW bits until all data is read.

When the specified pipe is in the receiving direction, set the CURPIPE and MBW bits at the same time.

For the procedure for changing the CURPIPE bit, see the description for the same.

For details, see the description of CURPIPE[3:0] Bits (FIFO Port Access Pipe Specification Bits).

DREQE bit (DMA Request Enable)

Use this bit to enable or disable DMA request.

When enabling DMA request, set this bit to 1 after setting the CURPIPE bits.

When modifying the setting of the CURPIPE bits, do so after setting this bit to 0.

DCLRM bit (Auto Buffer Memory Clear Mode after Specified Pipe Data is Read)

After reading data in the specified pipe, disable or enable automatic clearing of the FIFO buffer. When 1 is written to this bit, this controller sets the BCLR bit to 1 for a FIFO buffer if a zero-length packet is received when the FIFO buffer assigned to the specified pipe is empty, or when a short packet is received and the data is completely read while BFRE = 1.

When using this controller with the BRDYM bit set to 1, be sure to set this bit to 0.

REW bit (Buffer Pointer Rewind)

When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double-buffer operation, re-reading the currently-read FIFO buffer plane from the first data is allowed).

Do not set the REW bit to 1 simultaneously with modifying the CURPIPE bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.

To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT bit (Read Count Mode)

When 0 is written to this bit, if all received data of the FIFO buffer assigned to the pipe specified in the CURPIPE bits (selected pipe) is read (when the data is read on one side of a double buffer), this controller clears the DTLN bits in the DxFIFOCTR register to 0.

When 1 is written to this bit, the controller decrements the value of the DTLN bits in the DxFIFOCTR register whenever the received data is read from the FIFO buffer assigned to the specified pipe.

32.3.4.4 CFIFOCTR, DnFIFOCTR : FIFO Port Control Register (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x022 (CFIFOCTR)
0x02A + 0x4 × n (DnFIFOCTR)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BVAL	BCLR	FRDY	—	DTLN[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	DTLN[11:0]	Receive Data Length Indicate the length of the receive data.	R
12	—	This bit is read as 0. The write value should be 0.	R/W
13	FRDY	FIFO Port Ready Indicates whether the FIFO port can be accessed. 0: FIFO port is not accessible. 1: FIFO port is accessible.	R
14	BCLR	CPU Buffer Clear Specifies 1 to clear the FIFO buffer on the CPU side for the selected pipe. Only 1 can be written to this bit. This bit is read as 0. 0: Invalid 1: Clears the CPU buffer memory on the CPU side.	R/W
15	BVAL	Buffer Memory Valid Flag Specifies 1 when writing has ended in the CPU-side FIFO buffer for the pipe specified in CURPIPE (selected pipe). Only 1 can be written to this bit. 0: Invalid 1: Writing ended	R/W

Note: When a USB bus reset is detected, this register value is undefined.

DTLN[11:0] bits (Receive Data Length)

The DTLN bits indicate the length of the receive data. While the FIFO buffer is being read, the DTLN bits indicate different values depending on the RCNT bit value as described below.

1. When RCNT = 0:
   The length of received data is set in these bits, and the value is retained until all received data has been read from a single FIFO buffer plane.
   While BFRE is 1, these bits retain the length of the receive data until BCLR is set to 1 even after all data has been read.
2. When RCNT = 1:
   This controller decrements the value indicated by these bits each time data is read from the FIFO buffer.
   (The value is decremented by 1 when MBW is 00b, by 2 when MBW is 01b, and by 4 when MBW is 10b.)
   This controller sets these bits to 0 when all data has been read from one FIFO buffer plane. However, in double-buffer operation, if data has been received in one FIFO buffer plane before all data has been read from the other plane, this controller sets these bits to indicate the length of the receive data in the latter plane when all data has been read from the former plane.
   When reading these bits during FIFO buffer reading while RCNT = 1, note that these bits are updated within 150 ns after a read cycle for the FIFO port.

FRDY bit (FIFO Port Ready)

This bit indicates whether the FIFO port can be accessed from the CPU (DMAC).

In the following cases, the controller sets FRDY to 1, but data cannot be read via the FIFO port because there is no data to be read. In these cases, set BCLR to 1, clear the FIFO buffer, and then enable transmission and reception of the next data.

1. A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
2. A short packet is received and the data is completely read while BFRE is 1.

BCLR bit (CPU Buffer Clear)

When this bit is set to 1, this controller clears the FIFO buffer on the CPU for the selected pipe.

When double-buffer operation is set for the FIFO buffer assigned to the selected pipe, this controller clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the selected pipe is the DCP, setting BCLR to 1 allows this controller to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. When clearing the buffer on the SIE side, set the PID bits for the DCP to NAK before setting BCLR to 1.

When the selected pipe is not the DCP, writing 1 to this bit should be done while FRDY indicates 1. When checking the FRDY bit after executing BCLR, allow an interval of at least 80 ns after executing BCLR before referencing FRDY.

BVAL bit (Buffer Memory Valid Flag)

When the pipe specified in the CURPIPE bits (selected pipe) is for transmission, write 1 to this bit in the following cases. This controller sets the FIFO buffer from the CPU side to the SIE side, enabling transmission.

1. To transmit the short packet, set this bit to 1 after data has been written.
2. To transmit a zero-length packet, set this bit to 1 before writing data to FIFO.
3. Set this bit to 1 after the number of data bytes has been written for the pipe in continuous transfer mode, where the number is a natural integral multiple of the maximum packet size and less than the buffer size.

When the data of the maximum packet size has been written for the pipe in non-continuous transfer mode, this controller sets this bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When 1 is written to the BVAL and BCLR bits at the same time when the specified pipe is in the transmitting direction, the controller clears the old data and enables the transmission of a zero-length packet.

Writing 1 to this bit should be done while FRDY indicates 1. When checking the FRDY bit after executing BVAL, allow an interval of at least 80 ns after executing BVAL before referencing FRDY.

When the specified pipe is in the receiving direction, do not write 1 to this bit.

32.3.5 Enabling Interrupts

32.3.5.1 INTENB0 : Interrupt Enable Register 0

Base address: USBF = 0x9204_1000

Offset address: 0x30

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	BRDYE	Buffer Ready Interrupt Enable Enables or disables the USB interrupt output when the BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
9	NRDYE	Buffer Not Ready Response Interrupt Enable Enables or disables the USB interrupt output when the NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
10	BEMPE	Buffer Empty Interrupt Enable Enables or disables the USB interrupt output when the BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
11	CTRE	Control Transfer Stage Transition Interrupt Enable Enables or disables the USB interrupt output when the CTRT interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
12	DVSE	Device State Transition Interrupt Enable Enables or disables the USB interrupt output when the DVST interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
13	SOFE	Frame Number Update Interrupt Enable Enables or disables the USB interrupt output when the SOF interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
14	RSME	Resume Interrupt Enable Enables or disables the USB interrupt output when the RESM interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
15	VBSE	VBUS Interrupt Enable Enables or disables the USB interrupt output when the VBINT interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note: When a USB bus reset is detected, this register value is undefined.

32.3.5.2 INTENB1 : Interrupt Enable Register 1

Base address: USBF = 0x9204_1000

Offset address: 0x32

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PDDETINTE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PDDETINTE	PDDETINT Detection Interrupt Enable Enables or disables the USB interrupt output when a PDDETINT detection interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

32.3.5.3 BRDYENB : BRDY Interrupt Enable Register

Base address: USBF = 0x9204_1000

Offset address: 0x36

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPEBRDYE[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBRDYE[9:0]	BRDY Interrupt Enable for Each Pipe Enable or disable the BRDY interrupt for each pipe. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note: The bit number corresponds to the pipe number.

PIPEBRDYE[9:0] bits (BRDY Interrupt Enable for Each Pipe)

On detecting the BRDY interrupt for the pipe corresponding to the bit in this register which has been set to 1 by software, this controller sets 1 to the corresponding PIPEBRDY bit in BRDYSTS and the BRDY bit in INTSTS0, and asserts the interrupt signal through USB_FI.

While at least one PIPEBRDY bit in BRDYSTS indicates 1, this controller asserts the interrupt signal through USB_FI when the corresponding interrupt enable bit of this register is changed from 0 to 1 by software.

32.3.5.4 NRDYENB : NRDY Interrupt Enable Register

Base address: USBF = 0x9204_1000

Offset address: 0x38

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPENRDYE[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPENRDYE[9:0]	NRDY Interrupt Enable for Each Pipe Enable or disable the NRDY interrupt for each pipe. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note: The bit number corresponds to the pipe number.

PIPENRDYE[9:0] bits (NRDY Interrupt Enable for Each Pipe)

On detecting the NRDY interrupt for the pipe corresponding to the bit in this register which has been set to 1 by software, this controller sets 1 to the corresponding PIPENRDY bit in NRDYSTS and the NRDY bit in INTSTS0, and asserts the interrupt signal through the USB Function Interrupt (USB_FI).

While at least one PIPENRDY bit in NRDYSTS indicates 1, this controller asserts the interrupt signal through USB_FI when the corresponding interrupt enable bit of this register is changed from 0 to 1 by software.

32.3.5.5 BEMPENB : BEMP Interrupt Enable Register

Base address: USBF = 0x9204_1000

Offset address: 0x3A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPEBEMPE[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBEMPE[9:0]	BEMP Interrupt Enable for Each Pipe Enable or disable the BEMP interrupt for each pipe. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note:
- When a USB bus reset is detected, this register value is undefined.
- The bit number corresponds to the pipe number.

PIPEBEMPE[9:0] bits (BEMP Interrupt Enable for Each Pipe)

On detecting the BEMP interrupt for the pipe corresponding to the bit in this register which has been set to 1 by software, this controller sets 1 to the corresponding PIPEBEMP bit in BEMPSTS and the BEMP bit in INTSTS0, and asserts the interrupt signal through USB_FI.

While at least one PIPEBEMP bit in BEMPSTS indicates 1, this controller asserts the interrupt signal through USB_FI when the corresponding interrupt enable bit of this register is changed from 0 to 1 by software.

32.3.6 SOF Control Register

32.3.6.1 SOFCFG : SOF Pin Configuration Register

Base address: USBF = 0x9204_1000

Offset address: 0x3C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	BRDYM	INTL	EDGESTS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	EDGESTS	Interrupt Edge Processing Status Indicates the state of processing in response to interrupts when edge-sensing has been selected. 0: Processing in response to an edge-sensed interrupt is not in progress. 1: Processing in response to an edge-sensed interrupt is in progress	R

Bit	Symbol	Function	R/W
5	INTL	Interrupt Output Sense Select*2 Selects sense mode for USB interrupt output. 0: Edge sense 1: Level sense	R/W
6	BRDYM	PIPEBRDY Interrupt Status Clear Timing*1 Specifies the timing for clearing the PIPEBRDY interrupt state. This bit can be set only in the initial setting (before communications). The setting cannot be changed once communication starts. 0: Software clears the state. 1: Hardware clears the state by reading from the FIFO buffer or by writing to the FIFO buffer.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note 1. When setting the BRDYM bit to 1, set the INTL bit to 1 (level sense).
Note 2. With the INTL bit set to 0, to stop the system clock (SUSPM = 0) after clearing the interrupt state, write 0 to the SUSPM bit after checking that the EDGESTS bit is set to 0.

32.3.7 Interrupt Status

32.3.7.1 INTSTS0 : Interrupt Status Register 0

Base address: USBF = 0x9204_1000

Offset address: 0x40

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CTSQ[2:0]	Control Transfer Stage Indicate the control transfer stage. 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error 1 1 1: Reserved	R
3	VALID	USB Request Reception Indicates whether reception of the USB request has been detected. When writing to this bit, only 0 can be written. 0: Not detected 1: Setup packet reception	R/W
6:4	DVSQ[2:0]	Device State Indicate the device state. 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspended state	R
7	VBSTS	VBUS Input Status Indicates the input state of the USB_VUBUSIN pin. 0: USB_VUBUSIN pin is at the low level 1: USB_VUBUSIN pin is at the high level	R

Bit	Symbol	Function	R/W
8	BRDY	BRDY Interrupt Status Indicates the BRDY interrupt state. 0: BRDY interrupts not generated 1: BRDY interrupts generated	R
9	NRDY	NRDY Interrupt Status Indicates the NRDY interrupt state. 0: NRDY interrupts not generated 1: NRDY interrupts generated	R
10	BEMP	BEMP Interrupt Status Indicates the BEMP interrupt state. 0: BEMP interrupts not generated 1: BEMP interrupts generated	R
11	CTRT	Control Transfer Stage Transition Interrupt Status Indicates the control transfer stage transition interrupt state. When writing to this bit, only 0 can be written. 0: Control transfer stage transition interrupts not generated 1: Control transfer stage transition interrupts generated	R/W*1
12	DVST	Device State Transition Interrupt Status Indicates the device state transition interrupt state. When writing to this bit, only 0 can be written. 0: Device state transition interrupts not generated 1: Device state transition interrupts generated	R/W*1
13	SOFR	Frame Number Update Interrupt Status Indicates the frame number update interrupt state. When writing to this bit, only 0 can be written. 0: SOF interrupts not generated 1: SOF interrupts generated	R/W*1
14	RESM	Resume Interrupt Status*2 Indicates the resume detection interrupt state. When writing to this bit, only 0 can be written. 0: Resume interrupts not generated 1: Resume interrupts generated	R/W*1
15	VBINT	VBUS Change Detect Interrupt Status*2 Indicates the VBUS change detection interrupt state. When writing to this bit, only 0 can be written. 0: VBUS interrupts not generated 1: VBUS interrupts generated	R/W*1

Note: When a USB bus reset is detected, The DVST is set to 1 and DVSQ[2:0] are set to 001b and other bits are undefined.
Note 1. To clear the state indicated by the VBINT, RESM, SOFR, DVST or CTRT bits, write 0 only to the bit to be cleared, and write 1 to the other bits. Do not write 0 to the status bit set to 0.
Note 2. The controller detects the change in state indicated by the VBINT and RESM bits of this register, even while the clock is being stopped (SUSPM = 0), and conveys the interrupt if the corresponding interrupt is enabled. When the clock is enabled, clear the state using software.

BRDY bit (BRDY Interrupt Status)

This controller sets a BRDY bit to 1 when, among the PIPEBRDY bits for which the corresponding PIPEBRDYE bits in BRDYENB have been set to 1, at least one PIPEBRDY bit in BRDYSTS is set to 1 (that is, when this controller detects the BRDY interrupt state in at least one pipe among the pipes for which software has enabled the output of BRDY interrupts).

For the conditions for the setting of PIPEBRDY bits, see the description of the BRDYSTS register.

When a BRDY bit has been set to 1, the controller clears the bit in response to software writing 0 to all PIPEBRDY bits corresponding to PIPEBRDYE bits which have been set to 1.

A BRDY bit will not be cleared to 0 in response to software only writing 0 to individual bits that have been set to 1.

NRDY bit (NRDY Interrupt Status)

This controller sets an NRDY bit to 1 when, among the PIPEBRDY bits for which the corresponding PIPENRDYE bits in NRDYENB have been set to 1, at least one PIPENRDY bit in BNRDYSTS is set to 1 (that is, when this controller detects the NRDY interrupt state in at least one pipe among the pipes for which software has enabled the output of NRDY interrupts).

For the conditions for the setting of PIPENRDY bits, see the description of the NRDYSTS register.

When an NRDY bit has been set to 1, the controller clears the bit in response to software writing 0 to all PIPENRDY bits corresponding to PIPENRDYE bits which have been set to 1.

An NRDY bit will not be cleared to 0 in response to software only writing 0 to individual bits that have been set to 1.

BEMP bit (BEMP Interrupt Status)

This controller sets a BEMP bit to 1 when, among the PIPEBRDY bits for which the corresponding PIPEBEMPE bits in BEMPENB have been set to 1, at least one PIPEBEMP bit in BEMPSTS is set to 1 (that is, when this controller detects the BEMP interrupt state in at least one pipe among the pipes for which software has enabled the output of BEMP interrupts).

For the conditions for the setting of PIPEBEMP bits, see the description of the BEMPSTS register.

When a BEMP bit has been set to 1, the controller clears the bit in response to software writing 0 to all PIPEBEMP bits corresponding to PIPEBEMPE bits which have been set to 1.

A BEMP bit will not be cleared to 0 in response to software only writing 0 to individual bits that have been set to 1.

CTRT bit (Control Transfer Stage Transition Interrupt Status)

If this controller detects the stage transition of control transfer, it updates the CTSQ value and sets 1 to this bit.

When this interrupt occurs, clear the state before the controller detects the next control transfer stage transition.

DVST bit (Device State Transition Interrupt Status)

If this controller detects a change in the device state, it updates the DVSQ value and sets 1 to this bit.

When this interrupt occurs, clear the state before the controller detects the next device state transition.

SOFR bit (Frame Number Update Interrupt Status)

The conditions when the controller sets 1 in this bit are below.

When updating the frame number, this controller sets 1 to this bit (this interrupt is detected every 1 ms).

The controller detects the SOFR interrupt by internal interpolation even if the SOF packet from the USB host is corrupted.

RESM bit (Resume Interrupt Status)

If this controller is in the suspended state (DVSQ = 1xxb) and the DP pin falling edge is detected, 1 is set to this bit.

VBINT bit (VBUS Change Detect Interrupt Status)

When this controller detects a change in the USB_VUBUSIN pin input value (from high to low and from low to high), 1 is written to this bit. The controller writes the input value of the USB_VUBUSIN pin to the VBSTS bit. When the VBINT interrupt occurs, use the software to execute a consistency check several times during reading the VBSTS bit, and remove the chattering effect.

32.3.7.2 INTSTS1 : Interrupt Status Register 1

Base address: USBF = 0x9204_1000

Offset address: 0x42

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PDDETINT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PDDETINT	PDDET Detection Interrupt Status*1 Indicates the PDDETINT interrupt status. 0: PDDET interrupts not generated 1: PDDET interrupts generated	R/W*2
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, The bit 12 is set to 1 and other bits are undefined.

Note 1. To clear the status indicated by the PDDETINT bit, write 0 to bit0 only and 1 to other bits. Do not write 0 to status bits that indicate 0.

Note 2. This controller detects changes in the status indicated by the PDDETINT bit in this register even while the clock is stopped (SUSPM = 0), and reports corresponding interrupts if they are enabled. Perform a software status clear after the clock is enabled.

PDDETINT bit (PDDET Detection Interrupt Status)

This bit is set to 1 when a change (high to low or low to high) in EXH_PDDETSTS (BC1/2-PD detection status signal) is detected. When a PDDETINT interrupt is generated, read the PDDETSTS bit several times by software to perform debounce processing.

32.3.7.3 BRDYSTS : BRDY Interrupt Status Register

Base address: USBF = 0x9204_1000

Offset address: 0x46

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPEBRDY[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBRDY[9:0]	BRDY Interrupt Status for Each Pipe*1 Indicate the BRDY interrupt state for each pipe. When writing to these bits, only 0 can be written. 0: Interrupts not generated 1: Interrupts generated	R/W*2*3
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note 1. The bit number corresponds to the pipe number.
Note 2. When BRDYM = 0, to clear the state of each bit of this register, write 0 only to the bit to be cleared and 1 to other bits.
Note 3. When BRDYM = 0, clearing these interrupt status bits should be done before accessing the FIFO.

PIPEBRDY[9:0] bits (BRDY Interrupt Status for Each Pipe)

When the BRDY interrupt is detected for a pipe by this controller, the controller sets 1 in the corresponding PIPEBRDY[9:0] bit of the BRDYSTS register. Here, when 1 is written to the corresponding bit of BRDYENB register by using the software, the controller sets 1 to the BRDY bit of the INTSTS0 register.

The conditions for generating and clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit and PIPECFG.BFRE bit for each pipe as described below.

- When BRDYM = 0 and BFRE = 0
  With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
  On any of the following conditions, this controller generates the internal BRDY interrupt request trigger and sets 1 to the PIPEBRDY bit corresponding to the selected pipe.

  1. For the pipe in the transmitting direction

     (a) When the software has modified the DIR bit from 0 to 1

     (b) When the controller ends the packet transmission of the selected pipe in the condition where writing is not possible from the CPU to the FIFO buffer that has been assigned to the selected pipe (when the BSTS bit is read as 0).
     In continuous transmission/reception mode, the request trigger is generated when data of one plane of the FIFO buffer is complete.

     (c) In double-buffer operation, one FIFO buffer being empty on completion of the writing of data to the other FIFO buffer.
     The request trigger is not generated until completion of writing data to the currently-written FIFO buffer plane even if transmission to the other FIFO buffer is completed.

     (d) When the hardware flushes the buffer of the pipe for isochronous transfers.

     (e) The FIFO buffer making the transition from the write-disabled to the write-enabled state in response to writing of 1 to the ACLRM bit.
     The request trigger is not generated for the DCP (that is, during data transmission for control transfers).

2. For the pipe in the receiving direction

   (a) When packet reception is completed successfully thus enabling the FIFO buffer to be read when read-access from the CPU to the FIFO buffer for the selected pipe is disabled (when the BSTS bit is read as 0).
   The request trigger is not generated for the transaction in which DATA-PID disagreement occurs.
   When a short packet is received, the request trigger is generated even if the FIFO buffer has available space.
   When the transaction counter is used, the request trigger is generated on receiving the specified number of packets.
   In this case, the request trigger is generated even if the FIFO buffer has available space.

   (b) In double-buffer operation, one FIFO buffer being ready for reading on completion of the reading of data from the other FIFO buffer.
   The request trigger is not generated until completion of reading data from the currently-read FIFO buffer plane even if reception by the other FIFO buffer is completed.
   The BRDY interrupt is not generated in the status stage of control transfers.
   The PIPEBRDY interrupt state of the selected pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY interrupt status bit in the BRDYSTS register. In this case, 1s should be written to the PIPEBRDY interrupt status bits for the other pipes.
   Be sure to clear the BRDY state before accessing the FIFO buffer.

- When BRDYM = 0 and BFRE = 1
With these settings, this controller generates the BRDY interrupt on completion of the reading of all data for a single transfer using the pipe in the receiving direction, and sets 1 to the PIPEBRDY bit corresponding to the selected pipe.
On any of the following conditions, this controller determines that the last data for a single transfer has been received.

  1. When a short packet including a zero-length packet is received.
  2. When the transaction counter register (TRNCNT bits) is used and the number of packets specified by the TRNCNT bits is completely received.

  When the pertinent data is completely read out after any of the above determination conditions has been satisfied, this controller determines that all data for a single transfer has been completely read out.
  When a zero-length packet is received when the FIFO buffer is empty, the controller determines that all data for a single transfer has been completely read out upon the FRDY and the DTLN bits of the FIFO port control register are set to 1 and 0, respectively.
  In this case, write 1 to the BCLR bit of the corresponding FIFOCTR register by software to start the next transfer.
  With these settings, this controller does not detect the BRDY interrupt for the pipe in the transmitting direction.
  The PIPEBRDY interrupt state of the selected pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY interrupt status bit. In this case, 1s should be written to the PIPEBRDY interrupt status bits for the other pipes.
  In this mode, the BFRE bit setting should not be modified until all data for a single transfer has been processed.
  When it is necessary to modify the BFRE bit before completion of processing, all the FIFO buffers for the selected pipe should be cleared using the ACLRM bit.

- When BRDYM = 1 and BFRE = 0
With these settings, the PIPEBRDY values are linked to the BSTS bit settings for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by this controller depending on the FIFO buffer state.

  1. For the pipe in the transmitting direction
     The setting of a BRDY interrupt status bit is 1 while the port register of the corresponding FIFO buffer is ready for written data and 0 when it is not able to accept written data.
     However, the BRDY interrupt is not generated in response to setting of the bit to 1 if writing to the DCP in the transmitting direction can proceed.
  2. For the pipe in the receiving direction
     The setting of a BRDY interrupt status bit is 1 while the port register of the FIFO buffer is ready for reading and 0 when all data have been read.

  When a zero-length packet is received when the FIFO buffer is empty, the corresponding bit is set to 1 and the BRDY interrupt continues to be effective until 1 is written to BCLR.
  With this setting, the PIPEBRDY bit cannot be cleared to 0.
  When BRDYM is set to 1, all of the BFRE bits (for all pipes) should be cleared to 0.
  When BRDYM is set to 1, the INTL bit should be set to 1 (level control).

32.3.7.4 NRDYSTS : NRDY Interrupt Status Register

Base address: USBF = 0x9204_1000

Offset address: 0x48

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPENRDY[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPENRDY[9:0]	NRDY Interrupt Status for Each Pipe*1 Indicate the NRDY interrupt state for each pipe. When writing to these bits, only 0 can be written. 0: Interrupts not generated 1: Interrupts generated	R/W*2
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note 1. The bit number corresponds to the pipe number.
Note 2. To clear the state indicated by each bit of this register, write 0 only to the bit to be cleared and 1 to the other bits.

PIPENRDY[9:0] bits (NRDY Interrupt Status for Each Pipe)

On generating the internal NRDY interrupt request for the pipe whose PID bits are set to BUF, this controller sets the corresponding PIPENRDY bit in NRDYSTS in the register to 1. If the corresponding bit in the NRDYENB register is set to 1, this controller sets the NRDY bit in INTSTS0 in the register to 1.

The conditions on which this controller generates the internal NRDY interrupt request for a given pipe are described below.

However, the internal NRDY interrupt request is not generated during status stage execution of the control transfer.

1. When the pipe is in the transmitting direction
   (a) When an IN Token is received while there is no transmission data in the FIFO buffer and the corresponding PIPE PID bit is set to BUF (01):
   In this case, this controller generates an NRDY interrupt request at the reception of the IN token, setting the PIPENRDY bit to 1.
   For the pipe for the isochronous transfers in which an interrupt is generated, this controller transmits a zerolength packet, setting the OVRN bit to 1.
2. For the pipe in the receiving direction
   (a) When the PID bits for the corresponding pipe are set to BUF (01) and an OUT token is received while there is no open space in the FIFO buffer:
   For the pipe for the isochronous transfers in which an interrupt is generated, this controller generates an NRDY interrupt request at the reception of the OUT token, setting the PIPENRDY bit to 1 and OVRN bit to 1.
   For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, this controller generates an NRDY interrupt request when an NAK handshake is transferred after the data following the OUT token was received, setting the PIPENRDY bit to 1.
   However, during re-transmission (due to DATA-PID disagreement), the NRDY interrupt request is not generated.
   In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.
   (b) When the PID bits for the corresponding pipe are set to BUF (01) and a PING token is received while there is no open space in the FIFO buffer:
   In this case, this controller generates an NRDY interrupt request at the reception of the PING token, setting the PIPENRDY bit to 1.
   (c) In an isochronous transfer pipe, when the PID bits are set to BUF (01) and data is not received successfully within the interval frame:
   In this case, this controller generates an NRDY interrupt request at the reception of an SOF, and sets the PIPENRDY bit to 1.

32.3.7.5 BEMPSTS : BEMP Interrupt Status Register

Base address: USBF = 0x9204_1000

Offset address: 0x4A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPEBEMP[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBEMP[9:0]	BEMP Interrupt Status for Each Pipe*1 Indicate the BEMP interrupt state for each pipe. When writing to these bits, only 0 can be written. 0: Interrupts not generated 1: Interrupts generated	R/W*2
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note 1. The bit number corresponds to the pipe number.
Note 2. To clear the state indicated by each bit of this register, write 0 only to the bit to be cleared and 1 to the other bits.

PIPEBEMP[9:0] bits (BEMP Interrupt Status for Each Pipe)

On detecting the BEMP interrupt for the pipe whose PID bits are set to BUF by software, this controller sets the corresponding PIPEBEMP bit in BEMPSTS to 1. If the corresponding bit in BEMPENB is set to 1 by software, this controller sets the BEMP bit in INTSTS0 to 1.

The following describes the conditions on which this controller generates the internal BEMP interrupt request.

1. For the pipe in the transmitting direction, when the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).
   In single-buffer operation, the internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP.
   However, the internal BEMP interrupt request is not generated on any of the following conditions.
   (a) When writing data to the FIFO buffer on the CPU (DMAC) side is started by software on completion of transmitting data of one plane in double-buffer operation.
   (b) When the buffer is cleared (emptied) by setting the ACLRM or BCLR bit to 1.
   (c) When IN transfer (zero-length packet transmission) is performed during the control transfer status stage.
2. For the pipe in the receiving direction
   When the successfully-received data packet size exceeds the specified maximum packet size.
   In this case, this controller generates the BEMP interrupt request, setting the corresponding PIPEBEMP bit to 1, and discards the received data and modifies the setting of the PID bits of the corresponding pipe to STALL (11). However, the internal BEMP interrupt request is not generated on any of the following conditions.
   (a) When a CRC error or bit stuffing error is detected in the received data.
   (b) When a setup transaction is being performed.

Writing 0 to the PIPEBEMP bit clears the state; writing 1 to the PIPEBEMP bit has no effect.

32.3.8 Frame Number Registers

32.3.8.1 FRMNUM : Frame Number Register

Base address: USBF = 0x9204_1000

Offset address: 0x4C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	FRNM[10:0]	Frame Number Indicate the latest frame number.	R
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	CRCE	CRC Error Detection Status Indicates whether a CRC error has been detected in the pipe during isochronous transfer. When writing to this bit, only 0 can be written. 0: No error 1: An error occurred.	R/W
15	OVRN	Overrun/Underrun Detection Status*1 Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer. When writing to this bit, only 0 can be written. 0: No error 1: An error occurred.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note 1. The OVRN bit is for use in debugging. When designing a system, control the timing so that neither overrun nor underrun occurs.

FRNM[10:0] bits (Frame Number)

This controller sets these bits to indicate the latest frame number, which is updated every time an SOF packet is issued or received (every 1 ms).

When reading these bits by software, repeat reading until the same value is read twice.

CRCE bit (CRC Error Detection Status)

In an isochronous transfer pipe, this controller sets 1 to this bit when a CRC error or bit stuffing error has been detected.

This bit can be cleared by writing 0 to it by software. In this case, write 0x8000 when OVRN is not to be cleared at the same time.

This controller generates an internal NRDY interrupt request when it detects a CRC error. For details, see NRDY Interrupt Enable Bit for Each Pipe (PIPENRDYE).

OVRN bit (Overrun/Underrun Detection Status)

For the isochronous transfer pipe, this controller sets 1 to this bit when an overrun/underrun is detected.

When an overrun/underrun is detected, the controller issues an internal NRDY request. See NRDY Interrupt Enable Bit for Each Pipe (PIPENRDYE) for details.

This bit can be cleared writing 0 to it by software. In this case, write 0x4000 when CRCE is not to be cleared at the same time.

This controller sets this bit to 1 on any of the following conditions.

1. For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.
2. For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.

32.3.8.2 UFRMNUM : Frame Number Register

Base address: USBF = 0x9204_1000

Offset address: 0x4E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	UFRNM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	UFRNM[2:0]	Microframe Number Indicate the µframe number.	R
15:3	—	These bits are read as 0.	R

Note: When a USB bus reset is detected, this register value is undefined.

UFRNM[2:0] bits (Microframe Number)

This controller sets these bits to indicate the µframe number during high-speed operation. During operation other than high-speed operation, this controller sets these bits to 0x00.

When reading these bits, repeat reading until the same value is read twice.

32.3.9 USB Address

32.3.9.1 USBADDR : USB Address Register

Base address: USBF = 0x9204_1000

Offset address: 0x50

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	USBADDR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	USBADDR[6:0]	USB Address Indicate the USB address assigned by the host.	R
15:7	—	These bits are read as 0.	R

Note: When a USB bus reset is detected, The USBADDR[6:0] is set to 0000000b and other bits are undefined.

USBADDR[6:0] bits (USB Address)

These bits indicate the USB address assigned by the host when the SetAddress request is successfully processed.

If a USB bus reset is detected by the controller, 0x00 is set to this bit.

32.3.10 USB Request Registers

The USB request registers are used to store the setup requests for control transfers. The values of the USB request that have been received are stored.

32.3.10.1 USBREQ : USB Request Type Register

Base address: USBF = 0x9204_1000

Offset address: 0x54

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	BMREQUESTTYPE[7:0]	Request Type USB request bmRequestType value	R
15:8	BREQUEST[7:0]	Request USB request bRequest value	R

BMREQUESTTYPE[7:0] bits (Request Type)

These bits indicate the USB request data value received by this controller in the setup transaction. Writing to these bits by software has no effect.

BREQUEST[7:0] bits (Request)

These bits indicate the USB request data value received by this controller in the setup transaction. Writing to these bits by software has no effect.

32.3.10.2 USBVAL : USB Request Value Register

Base address: USBF = 0x9204_1000

Offset address: 0x56

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WVALUE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WVALUE[15:0]	Value USB request wValue value	R

WVALUE[15:0] bits (Value)

These bits are used to read the value of the USB request wValue. Bits 7 to 0 are lower-order byte.

These bits indicate the USB request wValue value received by this controller in the setup transaction. Writing to these bits by software has no effect.

32.3.10.3 USBINDX : USB Request Index Register

Base address: USBF = 0x9204_1000

Offset address: 0x58

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WINDEX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WINDEX[15:0]	Index USB request wIndex value	R

WINDEX[15:0] bits (Index)

These bits are used to read the value of the USB request wIndex. Bits 7 to 0 are lower-order byte.

These bits indicate the USB request wIndex value received by this controller in the setup transaction. Writing to these bits by software has no effect.

32.3.10.4 USBLENG : USB Request Length Register

Base address: USBF = 0x9204_1000

Offset address: 0x5A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WLENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WLENGTH[15:0]	Length USB request wLength value	R

WLENGTH[15:0] bits (Length)

These bits are used to read the value of the USB request wLength. Bits 7 to 0 are lower-order byte.

These bits indicate the USB request wLength value received in setup transaction by the controller. Writing to these bits by software has no effect.

32.3.11 DCP Configuration

32.3.11.1 DCPCFG : DCP Configuration Register

Base address: USBF = 0x9204_1000

Offset address: 0x5C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CNTMD	SHTNAK	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	SHTNAK	Disabling PIPE at the End of Transfer Specifies whether to change PID to NAK at the end of transfer when receiving direction is specified for the default control pipe. 0: The pipe continues at the end of transfer. 1: The pipe is disabled at the end of transfer.	R/W
8	CNTMD	Continuous Transfer Mode Specifies whether to enable the default control pipe communication in continuous transfer mode. 0: Continuous transfer mode disabled 1: Continuous transfer mode enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

32.3.11.2 DCPMAXP : DCP Maximum Packet Size Register

Base address: USBF = 0x9204_1000

Offset address: 0x5E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	MXPS[6:0]	Maximum Packet Size Specify the maximum data payload (maximum packet size) for the DCP.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

MXPS[6:0] bits (Maximum Packet Size)

Set the maximum data payload (maximum packet size) for the DCP in these bits. 0x40 (64 bytes) is the value after a reset.

The MXPS bits should be set to the value based on the USB specification.

The MXPS bits should be set while PID = NAK and before setting the CURPIPE bits.

Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

While MXPS is 0, do not write to the FIFO buffer or do not set PID to BUF.

32.3.11.3 DCPCTR : DCP Control Register

Base address: USBF = 0x9204_1000

Offset address: 0x60

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	—	—	—	—	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID These bits control the response type of this controller during control transfer. 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
2	CCPL	Control Transfer End Enable Setting this bit to 1 enables the status stage of the control transfer to be completed. 0: End of the control transfer disabled 1: End of the control transfer enabled	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	PIPE Busy Indicates whether the selected pipe is currently being used by the USB bus. 0: The pipe is not being used by the USB bus. 1: The pipe is being used by the USB bus.	R

Bit	Symbol	Function	R/W
6	SQMON	Sequence Toggle Bit Monitor Indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer. 0: DATA0 1: DATA1	R
7	SQSET	Toggle Bit Set Specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. This bit is read as 0. When writing to this bit, only 1 can be written. 0: Writing has no effect 1: Specifies DATA1	R/W
8	SQCLR	Toggle Bit Clear Specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. This bit is read as 0. When writing to this bit, only 1 can be written. 0: Writing has no effect 1: Specifies DATA0	R/W
10:9	—	These bits are read as 0. The write value should be 0.	R/W
11	—	This bit is read as 0. The write value should be 1.	R/W
12	—	The read value is undefined. The write value should be 0.	R/W
14:13	—	These bits are read as 0. The write value should be 1.	R/W
15	BSTS	Buffer Status Indicates whether access to the DCP FIFO buffer is enabled or disabled. 0: Buffer access disabled 1: Buffer access enabled	R

Note: When a USB bus reset is detected, The CCPL is set to 0 and PID[1:0] are set to 0 and other bits are undefined.

PID[1:0] bits (Response PID)

Change the setting of these bits from NAK to BUF by software during data stage or status stage of control transfers.

This controller changes the setting of these bits in the following conditions:

1. This controller changes the setting of the PID bits to select NAK response (00) on receiving the setup packet. Here, this controller sets VALID to 1. The setting of the PID bits cannot be changed until VALID is set to 0.
2. This controller sets the PID bits to select STALL response (11) on receiving the data of the size exceeding the maximum packet size when the PID bits have been set to select BUF response.
3. This controller sets the PID bits to select STALL response (1x) on detecting the control transfer sequence error.
4. This controller sets the PID bits to select NAK response on detecting the USB bus reset.

This controller does not reference to the setting of the PID bits while the SET_ADDRESS request is processed (auto processing).

CCPL bit (Control Transfer End Enable)

Setting this bit to 1 while the corresponding PID bits are set to BUF enables the stage of the control transfer to be completed.

Specifically, during control read transfer, this controller transmits the ACK handshake in response to the OUT transaction from the USB host, and outputs the zero-length packet in response to the IN transaction from the USB host during control writing or no-data control transfer. However, on detecting the SET_ADDRESS request, this controller operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of this bit.

This controller modifies this bit from 1 to 0 on receiving the new setup packet.

1 cannot be written by software to this bit while VALID is 1.

PBUSY bit (PIPE Busy)

This controller modifies this bit from 0 to 1 upon start of the USB transaction for the selected pipe, and modifies the bit from 1 to 0 upon completion of one transaction.

Reading this bit after the PID bits have been set to NAK allows checking that modification of the pipe settings is possible.

SQMON bit (Sequence Toggle Bit Monitor)

This bit indicates the expected value of the sequence toggle bit of the selected pipe.

This bit is toggled upon normal completion of the transaction. However, this bit is not toggled when a DATA-PID disagreement occurs during the receiving transfer.

This controller sets this bit to 1 (specifies DATA1 as the expected value) upon normal reception of the setup packet.

This controller does not reference to this bit during the IN/OUT transaction of the status stage, and does not allow this bit to toggle upon normal completion.

SQSET bit (Toggle Bit Set)

Setting this bit to 1 by software allows this controller to set DATA1 as the expected value of the sequence toggle bit of the selected pipe. This bit always indicates 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

Set this bit to 1 while PID is NAK and before setting the CURPIPE bits.

Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

SQCLR bit (Toggle Bit Clear)

Setting this bit to 1 by software allows this controller to set DATA0 as the expected value of the sequence toggle bit of the selected pipe. This bit always indicates 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

Set this bit to 1 while PID is NAK and before setting the CURPIPE bits.

Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

BSTS bit (Buffer Status)

This bit indicates whether access from the CPU to the FIFO buffer assigned to the DCP is possible.

The meaning of the BSTS bit depends on the ISEL bit setting as follows.

1. When ISEL = 0: indicates whether the received data can be read from the buffer.
2. When ISEL = 1: indicates whether the data for transmission can be written to the buffer.

32.3.12 Pipe Configuration Register

Pipes 1 to 9 should be set using the PIPESEL, PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI, PIPExCTR, PIPExTRE and PIPExTRN registers.

After selecting the pipe using the PIPESEL register, functions of the pipe should be set using the PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI registers. The PIPExCTR, PIPExTRE and PIPExTRN registers can be set regardless of the pipe selection in the PIPESEL register.

32.3.12.1 PIPESEL : Pipe Window Select Register

Base address: USBF = 0x9204_1000

Offset address: 0x64

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PIPESEL[3:0]	Pipe Window Select Specify the pipe corresponding to the PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI registers. 0x0: No pipe selected 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Reserved	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

PIPESEL[3:0] bit (Pipe Window Select)

When a value between 0001b and 1001b is set in these bits, the information and settings for the corresponding pipe can be read from the PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI registers. After a pipe is specified by setting these bits, the values set by software to the PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI registers are reflected in the corresponding pipe transfer type by this controller.

When 0000b is set in PIPESEL, 0 is read from all of the bits in the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers. When 0000b is set in PIPESEL, writing to the bits in these registers has no effect.

32.3.12.2 PIPECFG : Pipe Configuration Register

Base address: USBF = 0x9204_1000

Offset address: 0x68

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	EPNUM[3:0]	Endpoint Number Specify endpoint number for the selected pipe.	R/W
4	DIR	Transfer Direction Specifies pipe transfer direction for the selected pipe. 0: Receiving direction 1: Transmitting direction	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SHTNAK	Pipe Disable at the End of Transfer Specifies whether to change PID to NAK at the end of transfer when the selected pipe is in the receiving direction. 0: Pipe continued at end of transfer 1: Pipe disabled at end of transfer	R/W
8	CNTMD	Continuous Transfer Mode Specifies whether to use the selected pipe in continuous transfer mode. 0: Non-continuous transfer mode 1: Continuous transfer mode	R/W
9	DBLB	Double Buffer Mode Selects either single- or double-buffer operation for the FIFO buffer used by the selected pipe. 0: Single-buffer operation 1: Double-buffer operation	R/W

Bit	Symbol	Function	R/W
10	BFRE	BRDY Interrupt Operation Specification Specifies the BRDY interrupt generation timing from this controller to the CPU with respect to the selected pipe. 0: BRDY interrupt notification upon transmitting or receiving data 1: BRDY interrupt notification upon reading data	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W
15:14	TYPE[1:0]	Transfer Type Specify the transfer type for the pipe selected by the PIPESEL bits (selected pipe). 0 0: Pipe is not in use. 0 1: Bulk transfer 1 0: Interrupt transfer 1 1: Isochronous transfer	R/W

Note: When a USB bus reset is detected, this register value is undefined.

EPNUM[3:0] bits (Endpoint Number)

Specify the endpoint number for the selected pipe in these bits by software.

Setting 0000b indicates that the pipe is not being used.

These bits should be modified while PID is NAK, and the pipe is not selected by the CURPIPE bits.

Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

Do not make the settings such that the combination of the set values in the DIR and EPNUM bits should be the same for two or more pipes (EPNUM = 0000b (the selected pipe is not in use) can be set for all the pipes).

DIR bit (Transfer Direction)

When this bit is set to 0 by software, this controller uses the selected pipe in the receiving direction, and when this bit is set to 1, this controller uses the selected pipe in the transmitting direction.

This bit should be modified while PID is NAK and the pipe is not selected by the CURPIPE bits.

To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE bits are in the abovedescribed state.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

SHTNAK bit (Pipe Disable at the End of Transfer)

This bit is valid when pipes 1 to 5 are selected in the receiving direction.

When this bit is set to 1 for the selected pipe in the receiving direction, this controller modifies the PID bits for the selected pipe to NAK on determining the end of the transfer. This controller determines that the transfer has ended on any of the following conditions.

1. A short packet (including a zero-length packet) is successfully received.
2. The transaction counter is used and the number of packets specified for the counter is successfully received.

This bit should be modified while PID is NAK.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

This bit should be cleared to 0 for the pipe in the transmitting direction.

CNTMD bit (Continuous Transfer Mode)

This bit is valid when pipes 1 to 5 are selected and the transfer type of the selected pipe is set to bulk.

According to the setting value of this bit, this controller determines transmission/reception completion for the FIFO buffer assigned to the selected pipe. See Table 32.12.

Table 32.12 Relationship between transfer mode settings by CNTMD bit and timings at which reading data or transmitting data from FIFO buffer is enabled

CNTMD bit setting	When reading data or transmitting data is enabled
0	In the receiving direction (DIR = 0), the FIFO buffer is ready for reading in the following case. This controller receives one packet.
	In the transmitting direction (DIR = 1), the FIFO buffer is ready for transmission in either of the following cases. 1. Data of the maximum packet size is written to the FIFO buffer by software (or DMAC). 2. Data of the short packet size (including 0 byte data) is written to the FIFO buffer and then 1 is written to BVAL by software (or DMAC).
1	In the receiving direction (DIR = 0), the FIFO buffer is ready for reading in any of the following cases. 1. The number of the data bytes received in the FIFO buffer assigned to the selected pipe is equal to the number of assigned data bytes ((BUFSIZE + 1) × 64). 2. This controller receives a short packet other than a zero-length packet. 3. This controller receives a zero-length packet when data is already stored in the FIFO buffer assigned to the selected pipe. 4. This controller receives the number of packets equal to the transaction counter value specified for the selected pipe by software.
	In the transmitting direction (DIR = 1), the FIFO buffer is ready for transmission in any of the following cases 1 to 3. 1. The number of the data bytes written to the FIFO buffer by software (or DMAC) is equal to the number of data bytes in a single FIFO buffer plane assigned to the selected pipe. 2. The number of data bytes less than the size of a single FIFO buffer plane (including 0 byte data) assigned to the selected pipe is written to the FIFO buffer and then 1 is written to BVAL by software (or DMAC). 3. The number of data bytes less than the size of a single FIFO buffer plane (including 0 byte data) assigned to the selected pipe is written to the FIFO buffer and then the DENDx_N signal is asserted at the same time as writing the last data by software (or DMAC).

This bit should be modified while PID is NAK and the pipe is not selected by the CURPIPE bits.

To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE bits are in the abovedescribed state.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

DBLB bit (Double Buffer Mode)

This bit is valid when pipes 1 to 5 are selected.

When this bit is set to 1, this controller assigns two planes of the FIFO buffer size specified by the BUFSIZE bits in PIPEBUF to the selected pipe.

Specifically, the following expression determines the FIFO buffer size assigned to the selected pipe by this controller.

(BUFSIZE + 1) × 64 × (DBLB + 1) [bytes]

This bit should be modified while PID is NAK and the pipe is not selected by the CURPIPE bits.

To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously by software to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE bits are in the above-described state.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

BFRE bit (BRDY Interrupt Operation Specification)

This bit is valid when pipes 1 to 5 are selected.

When this bit is set to 1 and the selected pipe is in the receiving direction (the DIR bit is set to 0), this controller detects the transfer completion and generates the BRDY interrupt on having read the pertinent packet.

When the BRDY interrupt is generated with the above conditions, 1 must be written to BCLR. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to BCLR.

When this bit is set to 1 and the selected pipe is in the transmitting direction (the DIR bit is set to 1), this controller does not generate the BRDY interrupt. For details, see the description of the PIPEBRDY interrupt register.

This bit should be modified while PID is NAK and the pipe is not selected by the CURPIPE bits.

To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE bits are in the abovedescribed state.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

TYPE[1:0] bits (Transfer Type)

Specify the USB transfer type for the pipe selected by the PIPESEL bits (selected pipe) in the TYPE bits.

Table 32.13 lists the selected pipes and transfer types that can be set in the TYPE bits.

Table 32.13 Selected pipes and transfer types that can be set in TYPE bits

Selected pipe	TYPE bits	USB transfer type
Pipe 1 or Pipe 2	01b or 11b	Bulk or isochronous transfer
Pipe 3 to Pipe 5	01b	Bulk transfer
Pipe 6 to Pipe 9	10b	Interrupt transfer

Before setting PID to BUF for the selected pipe (before starting USB communication using the selected pipe), be sure to set these bits to the value other than 00b.

These bits should be modified while the PID bits for the selected pipe are set to NAK. Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

32.3.12.3 PIPEBUF : Pipe Buffer Specification Register

Base address: USBF = 0x9204_1000

Offset address: 0x6A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	BUFSIZE[4:0]					—	—	BUFNMB[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	BUFNMB[7:0]	Buffer number Specify the pipe FIFO buffer number for the selected pipe. (0x04 to 0x80)	R/W
9:8	—	These bits are read as 0. The write value should be 0.	R/W
14:10	BUFSIZE[4:0]	Buffer size Specify the FIFO buffer size for the pipe specified in PIPESEL bits (selected pipe). 0x00: 64 bytes 0x01: 128 bytes ⋮ 0x1F: 2 KB	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note: The bits in this register should be modified while PID is NAK and the pipe is not selected by the CURPIPE bits.
Note: Before modifying the bits of this register after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

BUFNMB[7:0] bits (Buffer number)

The first block number in the FIFO buffer to be allocated for the selected pipe should be set in these bits. The FIFO buffer blocks allocated for the selected pipe by this controller are determined as follows:

Block number: BUFNMB to block number of BUFNMB + (BUFSIZE + 1) × (DBLB + 1) − 1

These bits must be set to a value that does not exceed the size of installed memory (0 [0x00] to 8192 [0x80] for 8 KB memory). Observe the following conditions:

- 0x00 is used exclusively for DCP.
- 0x04 is used exclusively for pipe 6. However, when pipe 6 is not used, 0x04 can be used for other pipes. When pipe 6 is selected, writing to these bits has no effect and 0x04 is automatically assigned by this controller.
- 0x05 is used exclusively for pipe 7. However, when pipe 7 is not used, 0x05 can be used for other pipes. When pipe 7 is selected, writing to these bits has no effect and 0x05 is automatically assigned by this controller.
- 0x06 is used exclusively for pipe 8. However, when pipe 8 is not used, 0x06 can be used for other pipes. When pipe 8 is selected, writing to these bits has no effect and 0x06 is automatically assigned by this controller.
- 0x07 is used exclusively for pipe 9. However, when pipe 9 is not used, 0x07 can be used for other pipes. When pipe 9 is selected, writing to these bits has no effect and 0x07 is automatically assigned by this controller.

BUFSIZE[4:0] bits (Buffer size)

Specify the size of the FIFO buffer for the selected pipe in these bits in terms of blocks, where one block comprises 64 bytes. When the DBLB bit is set to 1 by software, this controller assigns two planes of the FIFO buffer size specified by the BUFSIZE bits to the selected pipe. Specifically, the following expression determines the FIFO buffer size assigned to the selected pipe by this controller.

(BUFSIZE + 1) × 64 × (DBLB + 1) [bytes]

Set the following value in these bits.

1. When pipes 1 to 5 are selected, any value from 0x00 to 0x1F can be set.
2. When pipes 6 to 9 are selected, only 0x0 should be set.

When used with CNTMD = 1, set an integral multiple of the maximum packet size to the BUFSIZE bits.

32.3.12.4 PIPEMAXP : Pipe Maximum Packet Size Register

Base address: USBF = 0x9204_1000

Offset address: 0x6C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MXPS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	MXPS[10:0]*1	Maximum Packet Size Specify the maximum data payload (maximum packet size) of the selected pipe. Pipes 6 to 9 can be set to 0x01 to 0x40 bytes.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

Note 1. The value after a reset of the MXPS bit is 0x00 when no pipe is selected with the PIPESEL bits in PIPESEL and 0x40 when a pipe is selected with the PIPESEL bits in PIPESEL.

MXPS[10:0] bits (Maximum Packet Size)

Set the maximum data payload (maximum packet size) for the selected pipe in these bits.

For pipes 1 and 2, a value between 1 byte (0x1) to 1024 bytes (0x400) can be set.

For pipes 3 to 5, any value from 8 bytes (0x8), 16 bytes (0x10), 32 bytes (0x20), 64 bytes (0x40) and 512 bytes (0x200) can be set (bits [2:0] are not provided).

The value after a reset is 0x040 (64 bytes).

These bits should be set to the appropriate value for each transfer type based on the USB specification.

Set these bits to 1 while PID is NAK and before setting the CURPIPE bits.

Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

While MXPS is 0, do not write to the FIFO buffer or set PID to BUF.

32.3.12.5 PIPEPERI : Pipe Timing Control Register

Base address: USBF = 0x9204_1000

Offset address: 0x6E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	IITV[2:0]	Interval Error Detection Spacing Specify the interval error detection timing for the selected pipe in terms of frames, which is expressed as n-th power of 2.	R/W
11:3	—	These bits are read as 0. The write value should be 0.	R/W
12	IFIS	Isochronous IN Buffer Flush Specifies whether to flush the buffer when the pipe selected by the PIPESEL bits (selected pipe) is used for isochronous IN transfers. 0: The buffer is not flushed 1: The buffer is flushed	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

IITV[2:0] bits (Interval Error Detection Spacing)

Specify the interval error detection timing in terms of frames, which is expressed as n-th power of 2, in these bits.

These bits should be modified while PID is NAK and before setting the CURPIPE bits.

Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

Before modifying these bits after USB communication has been completed with these bits set to a certain value, set PID to NAK and then set ACLRM to 1 to initialize the interval timer.

The IITV bits are invalid for pipes 3 to 5; set these bits to 0 for these pipes.

These bits can be set when the selected pipe transfer is for isochronous transfers.

When the selected pipe is for isochronous OUT transfers

This controller generates the NRDY interrupt when it fails to receive a data packet within the interval set for (μ) frames by the IITV bits.

This controller generates the NRDY interrupt when this controller fails to receive a data packet because of a CRC error or other errors contained in the packet, or because of the FIFO buffer being full because reading data from the FIFO buffer by software (DMAC) is slow.

This controller generates the NRDY interrupt on receiving an SOF packet. Even if the SOF packet is corrupted, the internal interpolation is used and allows the interrupt to be generated at the timing to receive the SOF packet. However, when the IITV bits are set to the value other than 0, this controller generates the NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation. When the PID bits are set to NAK by software after starting the

interval timer, this controller does not generate the NRDY interrupt on receiving an SOF packet. The interval counting starts at the different timing depending on the IITV bit setting as follows.

- When IITV = 0: The interval counting starts when the PID bits for the selected pipe are set to BUF. See Figure 32.2.
- When IITV ≠ 0: The interval counting starts on completion of successful reception of the first data packet after the PID bits for the selected pipe have been modified to BUF. See Figure 32.3

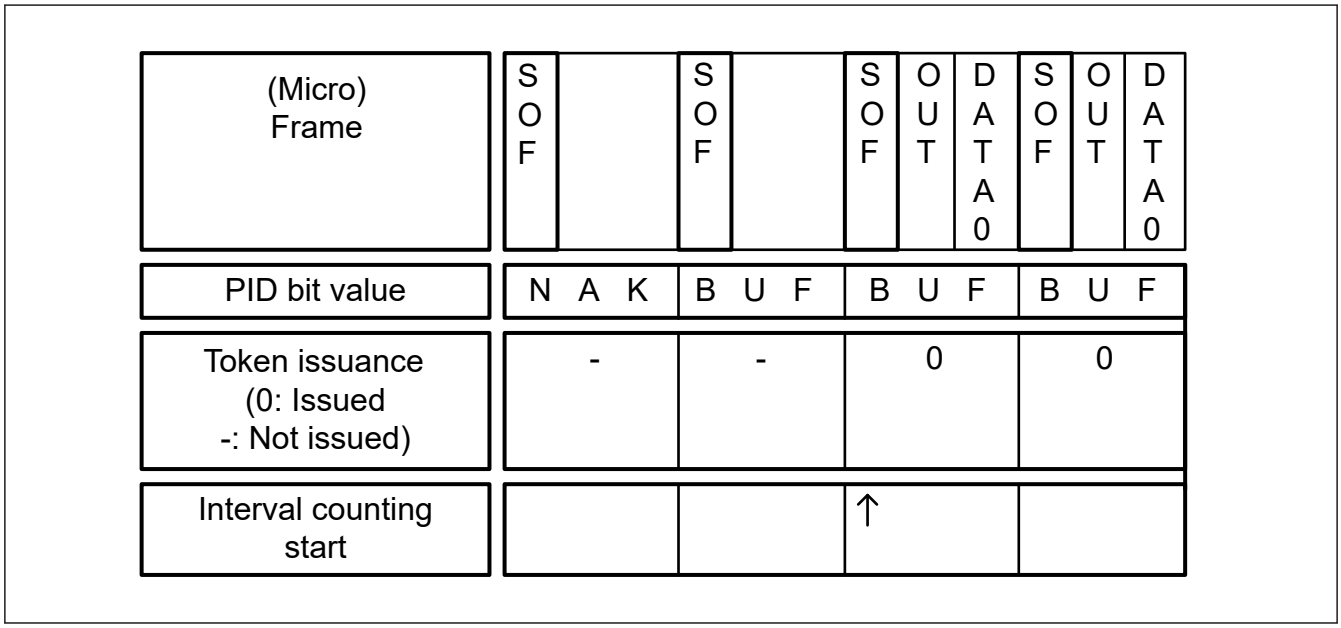


Figure 32.2 Relationship between (μ) frames and expected token reception when IITV = 0

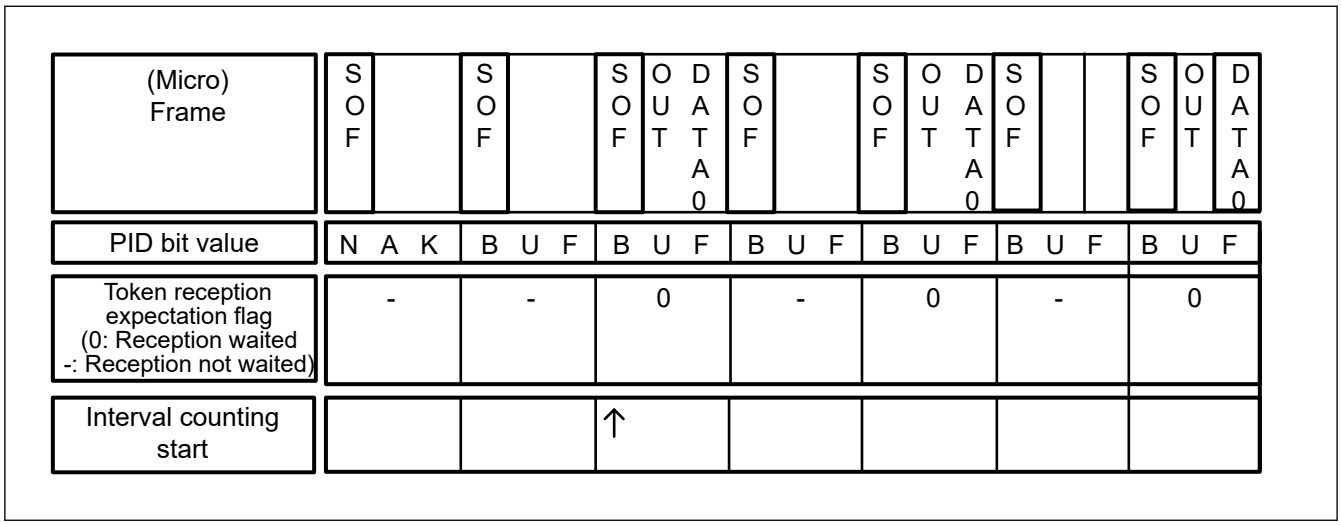


Figure 32.3 Relationship between (μ) frames and expected token reception when IITV = 1

When the selected pipe is for isochronous IN transfers

The IFIS bit should be 1 for this use. When IFIS = 0, this controller transmits a data packet in response to the received IN token irrespective of the IITV bit setting.

When IFIS = 1, this controller clears the FIFO buffer when this controller fails to receive an IN token within the interval set for (μ) frames by the IITV bits in a state in which there is data to be transmitted in the FIFO buffer.

This controller also clears the FIFO buffer when this controller fails to receive an IN token successfully because of a bus error such as a CRC error contained in the token.

This controller clears the FIFO buffer on receiving an SOF packet. Even if the SOF packet is corrupted, the internal interpolation is used and allows the FIFO buffer to be cleared at the timing to receive the SOF packet. The interval counting starts at the different timing depending on the IITV bit setting (similar to the timing during OUT transfers).

The counting conditions for the interval counter are any of the following.

- When a hardware reset is applied to this controller (at this point, the value of the IITV bits is also cleared to 0).
- When the ACLRM bit is set to 1 by software.
- When the controller detects a USB bus reset.

IFIS bit (Isochronous IN Buffer Flush)

When the selected pipe is for isochronous IN transfers, this controller automatically clears the FIFO buffer when this controller fails to receive the IN token from the USB host within the interval set by the IITV bits in terms of (μ) frames.

In double-buffer operation (DBLB = 1), this controller only clears the data in the plane used earlier.

This controller clears the FIFO buffer on receiving the SOF packet immediately after the (μ) frame in which this controller has expected to receive the IN token. Even if the SOF packet is corrupted, this controller also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation.

When the selected pipe is not for the isochronous transfer, set this bit to 0.

32.3.13 Pipe Control Registers

32.3.13.1 PIPEnCTR : PIPEn Control Register (n = 1 to 5)

Base address: USBF = 0x9204_1000

Offset address: 0x070 + 0x2 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID Specify the response type for the next transaction of the selected pipe. 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	PIPE Busy Indicates whether or not the selected pipe is being currently used for the USB bus. 0: The pipe is not being currently used for the USB bus. 1: The pipe is being currently used for the USB bus.	R
6	SQMON	Toggle Bit Confirm Sets the expected value of the sequence toggle bit in the next transaction of the selected pipe. 0: DATA0 1: DATA1	R
7	SQSET	Toggle Bit Set Specifies 1 when setting the expected value of the sequence toggle bit in the next transaction of the selected pipe, to DATA1. This bit is read as 0. When writing to this bit, only 1 can be written. 0: Writing has no effect. 1: Specifies DATA1	R/W
8	SQCLR	Toggle Bit Clear Specifies 1 when clearing the expected value of the sequence toggle bit in the next transaction of the selected pipe, to DATA0. This bit is read as 0. When writing to this bit, only 1 can be written. 0: Writing has no effect. 1: Specifies DATA0	R/W

Bit	Symbol	Function	R/W
9	ACLRM	Auto Buffer Clear Mode Enables or disables automatic buffer clear mode for the selected pipe. 0: Disabled 1: Enabled (all buffers are initialized)	R/W
10	ATREPM	Auto Response Mode Enables or disables auto response mode for the selected pipe. 0: Disabled 1: Enabled (a zero-length packet response while sending, NAK response and an NRDY interrupt is issued while receiving)	R/W
12:11	—	These bits are read as 0. The write value should be 0.	R/W
13	—	This bit is read as 0. The write value should be 1.	R/W
14	INBUFM	Transfer Buffer Monitor Indicates the selected FIFO buffer state when the selected pipe is in the transmitting direction. 0: FIFO buffer contains no transmittable data. 1: FIFO buffer contains transmittable data.	R
15	BSTS	Buffer Status Indicates the FIFO buffer state of the selected pipe. 0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note: When a USB bus reset is detected, The PID[1:0] is set to 00b and other bits are undefined.

PID[1:0] bits (Response PID)

Set a response of the controller in each pipe in this bit by the software.

The default value of this bit is NAK. When executing a USB transfer through the selected pipe, change the setting of the PID bits to select BUF. The basic operations (operations when there is no error in the communication packet) of this controller for each value of the PID bit are given in Table 32.14.

If the selected pipe is in USB communication, when the setting of the PID bits is changed from BUF to NAK by software, after writing NAK, to check whether the USB transfer of the selected pipe is actually shifted to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

In following cases, the controller modifies the value of this bit:

1. When the selected pipe is for receiving and when the software has written 1 to the SHTNAK bit of the selected pipe, the controller sets PID = NAK upon identifying the transfer end.
2. For the selected pipe, when the data packet of payload exceeding the maximum packet size is received, the controller sets PID = STALL (11b).
3. When the USB bus reset is detected, the controller sets PID = NAK.

Write 10b to shift from PID = NAK (00b) to PID = STALL.

Write 11b to shift from BUF (01b) to STALL.

First write 10b and then write 00b to shift form STALL (11b) to NAK.

First, shift to NAK and then to BUF to shift from STALL to BUF.

Table 32.14 List of controller operations according to the PID bit (1 of 2)

PID bit value	Transfer type (Type bits value)	Transfer direction (DIR bit value)	Controller operations
00b (NAK)	Bulk (TYPE = 01b) or Interrupt (TYPE = 10b)	Not dependent on setup value	NAK response is sent to the token from the USB host.
	Isochronous (TYPE = 11b)	Reception (DIR = 0)	Does not respond to the token from the USB host.
		Transmission (DIR = 1)	A zero-length packet is sent to the token from the USB host.

Table 32.14 List of controller operations according to the PID bit (2 of 2)

PID bit value	Transfer type (Type bits value)	Transfer direction (DIR bit value)	Controller operations
01b (BUF)	Bulk (TYPE = 01b)	Reception (DIR = 0)	When an OUT token is sent from the USB host, if the FIFO buffer for the selected pipe is ready for reception, the data is received and an ACK or NYET response is returned. A NAK response is returned if not ready. When a PING token is sent from the USB host, if the FIFO buffer of the selected pipe is ready for reception, an ACK response is returned. A NAK response is returned if not ready.
	Interrupt (TYPE = 10b)	Reception (DIR = 0)	For the Out token from the USB host, if the FIFO buffer of the selected pipe is ready for reception, the data is received and an ACK response is sent. A NAK response is returned if not ready.
	Bulk (TYPE = 01b) or Interrupt (TYPE = 10b)	Transmission (DIR = 1)	If the corresponding FIFO buffer is ready for transmission, the data is sent for the token from the USB. A NAK response is returned if not ready.
	Isochronous (TYPE = 11b)	Reception (DIR = 0)	For the Out token from the USB host, if the FIFO buffer of the selected pipe is ready for reception, the data is received. The data is deleted if not ready.
		Transmission (DIR = 1)	If the corresponding FIFO buffer is ready for transmission, the data is sent for the token from the USB. A zero-length packet is sent if not ready.
10b (STALL) or 11b (STALL)	Bulk (TYPE = 01b) or Interrupt (TYPE = 10)	Not dependent on setup value	A STALL response is sent to the token from the USB host.
	Isochronous (TYPE = 11b)	Not dependent on setup value	Returns nothing in response to the token from the USB host.

PBUSY bit (PIPE Busy)

This controller modifies this bit from 0 to 1 upon start of the USB transaction for the selected pipe, and modifies the bit from 1 to 0 upon completion of one transaction.

Reading this bit after PID has been set to NAK allows checking that modification of the pipe settings is possible.

SQMON bit (Toggle Bit Confirm)

This bit indicates the expected value of the sequence toggle bit of the selected pipe.

When the selected pipe is not for the isochronous transfer, this bit is toggled upon normal completion of the transaction. However, this bit is not toggled when a DATA-PID disagreement occurs during the receiving transfer.

SQSET bit (Toggle Bit Set)

Setting this bit to 1 by software allows this controller to set DATA1 as the expected value of the sequence toggle bit of the selected pipe. This bit always indicates 0.

Set this bit to 1 while PID is NAK.

Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

SQCLR bit (Toggle Bit Clear)

Setting this bit to 1 by software allows this controller to set DATA0 as the expected value of the sequence toggle bit of the selected pipe. This bit always indicates 0.

Set this bit to 1 while PID is NAK.

Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

ACLRM bit (Auto Buffer Clear Mode)

To clear the contents in the FIFO buffer assigned to the selected pipe completely, write 1 and then 0 to this bit continuously.

Table 32.15 shows the contents cleared by writing 1 and 0 to this bit continuously.

Table 32.16 shows the cases in which clearing the contents is necessary.

Table 32.15 Contents cleared by this controller by setting ACLRM = 1

No.	Contents cleared by ACLRM bit manipulation
(1)	All the contents in the FIFO buffer assigned to the selected pipe (all the information in two FIFO buffer planes in double-buffer operation)
(2)	The interval count value when the selected pipe is for isochronous transfer

Table 32.16 Cases where setting ACLRM = 1 is required

No.	Case where clearing is required
(1)	All the contents in the FIFO buffer assigned to the selected pipe are to be cleared.
(2)	The interval counter is to be reset.
(3)	When the BFRE setting is modified
(4)	When the DBLB setting is modified
(5)	When the transaction count function is forcibly terminated

Set this bit to 1 while PID is NAK and before specifying the pipe in CURPIPE bits. Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

ATREPM bit (Auto Response Mode)

When the selected pipe is for bulk transfer, this bit can be set to 1.

When this bit is set to 1, this controller responds to the token from the USB host as described below.

1. When the selected pipe is for bulk IN transfer (when TYPE = 01b and DIR = 1)
   When ATREPM = 1 and PID = BUF, this controller transmits a zero-length packet in response to the IN token. This controller updates (toggles) the sequence toggle bit (DATAPID) each time this controller receives the ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received). In this case, this controller does not generate the BRDY or BEMP interrupt.
2. When the selected pipe is for bulk OUT transfer (when TYPE = 01b and DIR = 0)
   When ATREPM = 1 and PID = BUF, this controller returns NAK in response to the OUT token (or PING token) and generates the NRDY interrupt.

This bit should be modified while PID is NAK. Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

For USB communication in auto response mode, set this bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.

When the selected pipe is for isochronous transfer, be sure to set this bit to 0.

INBUFM bit (Transfer Buffer Monitor)

When the selected pipe is in the transmitting direction (DIR = 1), this controller sets this bit to 1 when writing to at least one FIFO buffer plane is completed by software (or DMAC).

This controller sets this bit to 0 when this controller completes transmitting the data from the FIFO buffer plane to which all data has been written. In double-buffer operation (DBLB = 1), this controller sets this bit to 0 after this controller has completed transmitting the data from both FIFO buffer planes but before it has completed writing data to a single FIFO buffer plane.

This bit indicates the same value as the BSTS bit when the selected pipe is in the receiving direction (DIR = 0).

BSTS bit (Buffer Status)

This bit indicates whether access from the CPU to the FIFO buffer assigned to the selected pipe is enabled or disabled. The meaning of the BSTS bit depends on the setting of the DIR, BFRE and DCLRM bits as follows.

Table 32.17 BSTS bit operations

DIR bit value	BFRE bit value	DCLRM bit value	Meaning of BSTS bit
0	0	0	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	1: The received data can be read from the FIFO buffer. 0: The BCLR bit has been set to 1 after the received data has been completely read from the FIFO buffer.
		1	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
1	0	0	1: The transmit data can be written to the FIFO buffer. 0: The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

32.3.13.2 PIPEnCTR : PIPEn Control Register (n = 6 to 9)

Base address: USBF = 0x9204_1000

Offset address: 0x07A + 0x2 × (n - 6)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID Specify the response type for the next transaction of the selected pipe. 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	PIPE Busy Indicates whether or not the selected pipe is being currently used for the USB bus. 0: The pipe is not being currently used for the USB bus. 1: The pipe is being currently used for the USB bus.	R
6	SQMON	Toggle Bit Confirm Sets the expected value of the sequence toggle bit in the next transaction of the selected pipe. 0: DATA0 1: DATA1	R
7	SQSET	Toggle Bit Set Specifies 1 while clearing the expected value of the sequence toggle bit in the next transaction of the selected pipe to DATA1.This bit is read as 0. When writing to this bit, write 1. 0: Writing has no effect. 1: Specifies DATA1	R/W

Bit	Symbol	Function	R/W
8	SQCLR	Toggle Bit Clear Specifies 1 while clearing the expected value of the sequence toggle bit in the next transaction of the selected pipe to DATA0. This bit is read as 0. When writing to this bit, write 1. 0: Writing has no effect. 1: Specifies DATA0	R/W
9	ACLRM	Auto Buffer Clear Mode Enables or disables automatic buffer clear mode for the selected pipe. 0: Disabled 1: Enabled (all buffers are initialized)	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
13	—	This bit is read as 0. The write value should be 1.	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	BSTS	Buffer Status Indicates the FIFO buffer state of the selected pipe. 0: Buffer access is disabled 1: Buffer access is enabled	R

Note: When a USB bus reset is detected, The PID[1:0] are set to 00b and other bits are undefined.

As for the bits listed below, see the description of each bit in section 32.3.13.1. PIPEnCTR : PIPEn Control Register (n = 1 to 5)

- Response PID Bits (PID[1:0])
- PIPE Busy Bit (PBUSY)
- Buffer Status Bit (BSTS)
- Sequence Toggle Bit Monitor Bit (SQMON)
- Sequence Toggle Bit Set Bit (SQSET)
- Sequence Toggle Bit Clear Bit (SQCLR)

ACLRM bit (Auto Buffer Clear Mode)

To clear all contents from the FIFO buffer allocated to the selected pipe, write 1 and 0 sequentially in the ACLRM bit.

Table 32.18 shows the contents of data to be cleared by the controller when 1 and 0 are written to this bit sequentially.

The cases that require this processing are listed in Table 32.19.

Table 32.18 Contents cleared by the controller when ACLRM = 1

No.	Contents cleared by ACLRM bit manipulation
(1)	Entire contents of the FIFO buffer allocated to the selected pipe

Table 32.19 Cases where setting ACLRM = 1 is required

No.	Cases where clearing is required
(1)	All the contents in the FIFO buffer assigned to the selected pipe are to be cleared.
(2)	The interval counter is to be reset.
(3)	The value of the BFRE bit is changed.
(4)	The transaction count function is forcibly terminated.

The ACLRM bit should be set while PID = NAK and before specifying the pipe in the CURPIPE bits.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

32.3.14 Transaction Counters

32.3.14.1 PIPEnTRE : PIPEn Transaction Counter Enable Register (n = 1 to 5)

Base address: USBF = 0x9204_1000

Offset address: 0x090 + 0x4 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	TRCLR	Transaction Counter Clear The transaction counter can be cleared to 0 by writing 1 to this bit. This bit is read as 0. When writing to this bit, write 1. 0: Invalid 1: The current counter value is cleared	R/W
9	TRENB	Transaction Counter Enable Enable or disables the transaction counter 0: The transaction counter is disabled 1: The transaction counter is enabled	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

Note: Modify each bit in this register while PID is NAK.
Before modifying each bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

TRCLR bit (Transaction Counter Clear)

When this bit is set to 1 by software, this controller clears the current counter value of the transaction counter corresponding to the selected pipe and then sets this bit to 0.

TRENB bit (Transaction Counter Enable)

For the pipe in the receiving direction, setting this bit is set to 1 by software after setting the total number of the packets to be received in the TRNCNT bits allows the following control on having received the number of packets equal to the set value in the TRNCNT bits.

1. In continuous transmission/reception mode (CNTMD = 1), this module switches the FIFO buffer to the CPU side even if the FIFO buffer is not full on completion of reception.
2. While SHTNAK is 1, this module changes the setting of the PID bits to NAK for the corresponding pipe on having received the number of packets equal to the set value in the TRNCNT bits.
3. While DENDE is 1 and PKTMD is 0, the DEND signal is asserted when the number of packets specified in the TRNCNT bits is received and the last data is to be read.
4. While BFRE is 1, this module asserts the BRDY interrupt on having received the number of packets equal to the set value in the TRNCNT bits and then reading out the last received data.

For the pipe in the transmitting direction, set this bit to 0.

When the transaction counter is not used, set this bit to 0.

When the transaction counter is used, set the TRNCNT bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

32.3.14.2 PIPEnTRN : PIPEn Transaction Counter Register (n = 1 to 5)

Base address: USBF = 0x9204_1000

Offset address: 0x092 + 0x4 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRNCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TRNCNT[15:0]	Transaction Counter When written to: Specify the total number of reception packets (number of transactions) to be received by the selected pipe. When read from: Indicate the specified number of transactions if TRENB is 0. Indicate the number of currently counted transaction if TRENB is 1.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

TRNCNT[15:0] bits (Transaction Counter)

For the pipe in the receiving direction, setting these bits to 1 by software after setting the total number of the packets to be received in the TRNCNT bits allows the control of the transaction counter clear bit (TRCLR).

When TRENB = 0, these bits indicate the number of transactions set by software.

When TRENB = 1, these bits indicate the number of transactions being currently counted.

This controller increments the value of these bits by one when all of the following conditions are satisfied on receiving the packet.

- TRENB = 1
- (TRNCNT setting ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the set value in the MXPS bits.

This module clears the value of these bits to 0 when any of the following conditions are satisfied.

When all the following conditions are satisfied:

- TRENB = 1
- (TRNCNT setting ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the set value in the MXPS bits.

When both of the following conditions are satisfied:

- TRENB = 1
- A short packet is received.

When the following condition is satisfied:

- When the TRCLR bit is set to 1 by software

For the pipe in the transmitting direction, set these bits to 0. When the transaction counter is not used, set these bits to 0.

These bits should be modified while PID is NAK and TRENB is 0.

Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

To modify the value of these bits, set TRCLR to 1 before setting TRENB to 1.

32.3.14.3 LPSTS : Low Power Status Register

Base address: USBF = 0x9204_1000

Offset address: 0x102

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SUSPM	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	—	These bits are read as 0. The write value should be 0.	R/W
14	SUSPM	UTMI SuspendM Control Controls the SuspendM signal to the UTMI. 0: UTMI suspend mode 1: UTMI normal mode	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

SUSPM bit (UTMI SuspendM Control)

The USBf module controls output of the clock signal from the PLL in the USB-PHY layer. Accordingly, supply of the clock signal to the USBf module is stopped while SUSPM = 0.

After setting this bit to 1, wait for at least 100 µs to allow oscillation of the UTMI clock to become stable.

Writing to this controller is disabled when the SUSPM bit is set to 0 (the UTMI clock is stopped). Reading from the controller area is allowed. However, writing to the registers listed in Table 32.20 is enabled even when the SUSPM bit is set to 0.

Table 32.20 List of registers that can be written by software when SUSPM = 0

Address	Register name
0x8020_1000	SYSCFG0
0x8020_1002	SYSCFG1
0x8020_1032	INTENB1*1
0x8020_1102	LPSTS

Note 1. Bit 0 in the INTENB1 register can be written when SUSPM = 0.

However, the value written to the SYSCFG0 register while the UTMI clock is stopped (SUSPM = 0) is reflected after the UTMI clock oscillation is started (SUSPM = 1).

32.3.15 Next Register Set

32.3.15.1 N0SA_n, N1SA_n : Next Source Address Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x400 + 0x40 × n (N0SA_n)
0x40C + 0x40 × n (N1SA_n)

Bit position:	31	0
Bit field:	SAWD[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	SAWD[31:0]	Source Address or Write Data These bits set the start address of the DMA transfer source in normal mode. These bits set write data in write-only mode.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note: The N0SA_n register is overwritten by descriptor read data during a transfer in link mode.

This register is used to set write data in write-only mode (when the WONLY field in the CHCFG_n register = 1). For details, see section 32.15.1.3. Write-Only Mode.

32.3.15.2 N0DA_n, N1DA_n : Next Destination Address Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x404 + 0x40 × n (N0DA_n)
0x410 + 0x40 × n (N1DA_n)

Bit position:	31																												0
Bit field:	DA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DA[31:0]	Destination Address These bits set the start address of the DMA transfer destination.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note: The N0DA_n register is overwritten by descriptor read data during a transfer in link mode.

32.3.15.3 N0TB_n, N1TB_n : Next Transaction Byte Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x408 + 0x40 × n (N0TB_n)
0x414 + 0x40 × n (N1TB_n)

Bit position:	31																												0
Bit field:	TB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TB[31:0]	Transaction Byte These bits set the total number of transaction bytes. Do not start a DMA transaction with these bits set to 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note: The N0TB_n register is overwritten by descriptor read data during a transfer in link mode.

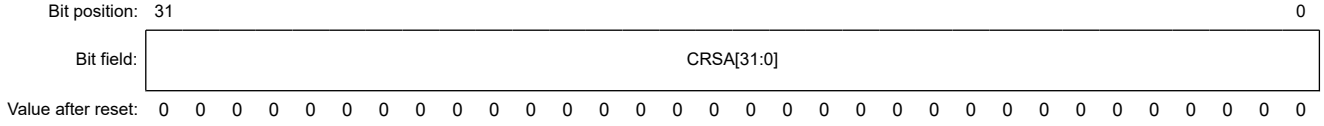
32.3.16 Current Register Set

The current register set indicates the DMA transfer source address, the DMA transfer destination address, and the total number of transfer bytes. The set value is loaded from the Next 0/1 register set (in register mode) or from the descriptor read data (in link mode) by hardware. This current register set cannot be written by software.

32.3.16.1 CRSA_n : Current Source Address Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x418 + 0x40 × n



Bit	Symbol	Function	R/W
31:0	CRSA[31:0]	Source Address These bits indicate the read address of the next DMA transaction. The CRSA[31:0] value is automatically incremented during a DMA transaction, but remains unchanged when the SAD bit in the CHCFG_n register is 1 and is undefined when the WONLY bit in the CHCFG_n register is 1. The initial value of these bits is loaded from the following register. In register mode: The transfer source address is loaded from the Next 0/1 register set. In link mode: The transfer source address is loaded from the descriptor. (Descriptor read data is input to the N0SA_n register by hardware, and is loaded to the CRSA_n register at the beginning of transfer.) The value of these bits is incremented upon completion of a read transfer. Read this register after the ongoing DMA stops (TACT = 0 in the CHSTAT_n register). (Handle the value of these bits during DMA operation as a reference value.)	R

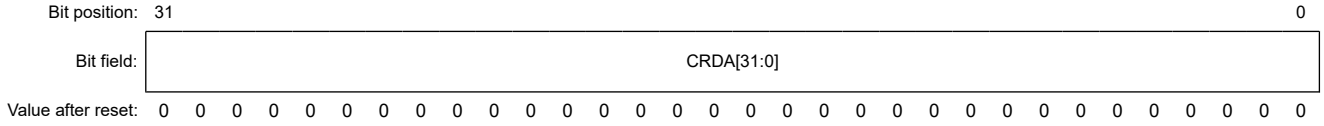
Note: When a USB bus reset is detected, this register value is undefined.

This register indicates the DMA transfer source address of DMA channel n (n = 1, 0).

32.3.16.2 CRDA_n : Current Destination Address Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x41C + 0x40 × n



Bit	Symbol	Function	R/W
31:0	CRDA[31:0]	Destination Address These bits indicate the write address of the next DMA transaction. The CRDA[31:0] value is automatically incremented during a DMA transaction, but remains unchanged when the DAD bit in the CHCFG_n register is 1. The initial value of these bits is loaded from the following register. In register mode: The transfer destination address is loaded from the Next 0/1 register set. In link mode: The transfer destination address is loaded from the descriptor. (Descriptor read data is input to the N0DA_n register by hardware, and is loaded to the CRDA_n register at the beginning of transfer.) The value of these bits is incremented upon completion of a write transfer. Read this register after the ongoing DMA stops (TACT = 0 in the CHSTAT_n register). (Handle the value of these bits during DMA operation as a reference value.)	R

Note: When a USB bus reset is detected, this register value is undefined.

This register indicates the DMA transfer destination address of DMA channel n (n = 1, 0).

32.3.16.3 CRTB_n : Current Transaction Byte Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x420 + 0x40 × n

Bit position:	31																												0
Bit field:	CRTB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CRTB[31:0]	Transaction Byte These bits indicate the remaining number of transfer bytes of the DMA transaction in progress. The CRTB[31:0] value is automatically decremented during a DMA transaction. The initial value of these bits is loaded from the following register. In register mode: The number of transfer bytes is loaded from the Next 0/1 register set. In link mode: The number of transfer bytes is loaded from the descriptor. (Descriptor read data is input to the N0TB_n register by hardware, and is loaded to the CRTB_n register at the beginning of transfer.) The value of these bits is decremented upon completion of a write transfer. Read this register after the ongoing DMA stops (TACT = 0 in the CHSTAT_n register). (Handle the value of these bits during DMA operation as a reference value.)	R

Note: When a USB bus reset is detected, this register value is undefined.

This register indicates the total number of transfer bytes of DMA channel n (n = 1, 0). This register is cleared to 0 at the end of a transfer.

32.3.17 Channel Register Set

32.3.17.1 CHSTAT_n : Channel Status Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x424 + 0x40 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DNUM[7:0]								—	—	—	—	—	SWPRQ	DMARQM	INTM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MODE	DER	DW	DL	SR	TC	END	ER	SUS	TACT	RQST	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN	Enable Indicates that operation of DMA channel n is enabled or disabled. 0: Operation disabled 1: Operation enabled	R
1	RQST	Request Indicates that a transfer request has been received. 0: No DMA transfer request has been received. 1: A DMA transfer request has been received.	R
2	TACT	Transaction Active Indicates that the DMAC is operating to confirm that the channel is completely stopped. For details, see section 32.16.8. Transfer Status. 0: DMA of channel_n is inactive. 1: DMA of channel_n is in progress.	R

Bit	Symbol	Function	R/W
3	SUS	Suspend Indicates that the channel is suspended. For details, see section 32.16.8.2. Suspending a Transfer. 0: Channel_n is not suspended. 1: Channel_n is suspended.	R
4	ER	Error Indicates that an error response was received during a DMA transfer and a USB_FDMAERR interrupt was generated. 0: No error response was received. 1: An error response was received.	R
5	END	USB_FDMAn Interrupted Indicates that a DMA transaction has been completed and an USB_FDMAn interrupt was generated. 0: DMA transfer not completed 1: DMA transfer completed	R
6	TC	Terminal Count Indicates that a DMA transaction has been completed. 0: DMA transfer not completed 1: DMA transfer completed	R
7	SR	Selected Register Set Indicates the selected register set in register mode. 0: Next0 register set 1: Next1 register set	R
8	DL	Descriptor Load Indicates that the internal state is the descriptor read state. This bit remains 1 when a bus error is received during descriptor read. 0: Other than descriptor read 1: (When ER = 0) While the descriptor is read in link mode (When ER = 1) A bus error occurred while the descriptor was read in link mode.	R
9	DW	Descriptor WriteBack Indicates that the internal state is the descriptor write-back state. This bit remains 1 when a bus error is received during descriptor write-back. 0: Other than header write-back in link mode 1: (When ER in the CHSTAT_n register is 0) During header write-back in link mode (When ER in the CHSTAT_n register is 1) A bus error occurred during header write-back in link mode.	R
10	DER	Descriptor Error Indicates that the read descriptor is invalid (LV = 0). (Does not depend on the level of the DIM bit in the CHCFG_n register.) 0: No descriptor error occurred. 1: A descriptor error occurred.	R
11	MODE	DMA Mode Indicates the DMA mode that is specified by the DMS bit in the CHCFG_n register. 0: Register mode 1: Link mode	R
15:12	—	These bits are read as 0.	R
16	INTM	Interrupt Mask Indicates that the USB_FDMAn interrupt is in the temporary mask status. 0: The temporary mask status is cleared. 1: In temporary mask status	R
17	DMARQM	DMAREQ Mask Indicates that a DMA transfer request from the USB control is in the temporary mask status. 0: The temporary mask status is cleared. 1: In temporary mask status	R

Bit	Symbol	Function	R/W
18	SWPRQ	Sweep Request Indicates the forced ejection request status. Indicates the software forced ejection request (activated by the SETSSWPRQ bit in the CHCTRL_n register). 0: The forced ejection request is not asserted. 1: The forced ejection request is asserted.	R
23:19	—	These bits are read as 0.	R
31:24	DNUM[7:0]	Data Number These bits indicate the amount of valid data in the buffer. Data is read from the transfer source, and then the amount of data that has not been written to the transfer destination is displayed (in bytes).	R

This register indicates the status of DMA channel n (n = 1, 0).

When the ER bit in the CHSTAT_n register is set to 1, handle the series of transfers as invalid.

To suspend the ongoing DMA transaction, mask or clear the transfer request or clear the EN bit in the CHSTAT_n register. For details on the procedure, see section 32.16.8.3. Aborting a Transfer.

If a DMA transfer request from the USB control and a transfer request by software (by setting the STG bit in the CHCFG_n register to 1) are used together for the same channel, enabled activation sources cannot be identified. Use only a transfer request in the system.

To make a transfer request by software, make sure that the previously requested DMA transfer operation has been completed (by checking the current register or in other ways), and then set the next STG bit.

EN bit (Enable)

[Setting condition]

- 1 is written to SETEN in the CHCTRL_n register.

[Clearing conditions]

- 1 is written to SWRST in the CHCTRL_n register.
- 1 is written to CLREN in the CHCTRL_n register.
- When an error response is received during a transfer.
- When all DMA transactions are completed in register mode (REN = 0 in the CHCFG_n register)
- When DMA transfer of the last descriptor (LE = 1) is completed in link mode (write-back when WBD = 0)
- When descriptor read operation stops in link mode (LV = 0 and DRRP = 0 in the CHCFG_n register)

RQST bit (Request)

[Setting conditions]

- 1 is written to STG in the CHCTRL_n register.
- When a DMA transfer request is received from the USB control

[Clearing conditions]

- 1 is written to SWRST in the CHCTRL_n register.
- 1 is written to CLRRQ in the CHCTRL_n register.
- When a transfer is performed on the side specified by REQD in the CHCFG_n register in single transfer mode (TM = 0 in the CHCFG_n register)
- When all DMA transactions are completed in register mode (REN = 0 in the CHCFG_n register)
- When DMA transfer of the last descriptor (LE = 1) is completed in link mode
- When descriptor read operation stops in link mode (LV = 0 and DRRP = 0 in the CHCFG_n register)
- When a DMA transaction is completed in link mode with DEM in the CHCFG_n register cleared to 0
- When the master interface receives a bus error

TACT bit (Transaction Active)

[Setting condition]

- 1 is written to SETEN in the CHCTRL_n register (descriptor read start or waiting for a DMA request)

[Clearing condition]

- Idle internal state (EN in the CHSTAT_n register is cleared to 0 and all transfers are completed.)

SUS bit (Suspend)

[Setting condition]

- When the internal state becomes suspended state after 1 is written to SETSUS in the CHCTRL_n register while DMA transfer of channel n is in progress.

[Clearing conditions]

- 1 is written to CLRSUS in the CHCTRL_n register.
- 1 is written to CLREN in the CHCTRL_n register.
- Conditions for clearing EN in the CHSTAT_n register are met.

ER bit (Error)

[Setting condition]

- When an error response is received during a bus cycle.

[Clearing condition]

- 1 is written to SWRST in the CHCTRL_n register.

END bit (USB_FDMAn Interrupted)

[Setting conditions]

- Conditions for setting the TC bit are met when DEM in the CHCFG_n register is 0.
- When LV = 0 in the header, DRRP = 0 in the CHCFG_n register, and DIM = 0 during the descriptor read operation in link mode.

[Clearing conditions]

- 1 is written to CLREND in the CHCTRL_n register.
- 1 is written to SWRST in the CHCTRL_n register.

TC bit (Terminal Count)

[Setting conditions]

- When transfers for the total number of transfer bytes specified in the CRTB register are completed in register mode.
- When transfers for the total number of transfer bytes specified in the CRTB register are completed in link mode while WBD = 1 in the descriptor's header.
- When descriptor write-back is completed in link mode while WBD = 0 in the descriptor's header.

[Clearing conditions]

- 1 is written to CLRTC in the CHCTRL_n register.
- 1 is written to SWRST in the CHCTRL_n register.

SR bit (Selected Register Set)

[Setting condition]

- 1 is written to the RSEL bit in the CHCFG_n register to set it.

[Clearing condition]

- 0 is written to the RSEL bit in the CHCFG_n register to clear it.

DL bit (Descriptor Load)

[Setting condition]

- When descriptor read starts in link mode

[Clearing conditions]

- Descriptor read is completed with an OK response in link mode.
- 1 is written to SWRST in the CHCTRL_n register. If 1 is retained with an error response, this bit can be cleared to 0 only with the SWRST bit.

DW bit (Descriptor WriteBack)

[Setting condition]

- When header write-back starts in link mode.

[Clearing conditions]

- Header write-back is completed with an OK response in link mode.
- 1 is written to SWRST in the CHCTRL_n register. If 1 is retained with an error response, this bit can be cleared to 0 only with the SWRST bit.

DER bit (Descriptor Error)

[Setting condition]

- LV of the read descriptor is 0 when DRRP in the CHCFG_n register is 0 in link mode.

[Clearing conditions]

- 1 is written to the CLRDER bit in the CHCTRL_n register.
- 1 is written to the SWRST bit in the CHCTRL_n register.

INTM bit (Interrupt Mask)

[Setting condition]

- 1 is written to the SETINTM bit in the CHCTRL_n register.

[Clearing conditions]

- 1 is written to the CLRINTM bit in the CHCTRL_n register.
- 1 is written to the SWRST bit in the CHCTRL_n register.

DMARQM bit (DMAREQ Mask)

[Setting condition]

- 1 is written to the SETDMARQM bit in the CHCTRL_n register to set it.

[Clearing conditions]

- Header write-back in link mode is completed with an OK response.
- 1 is written to the CLRDMARQM bit in the CHCTRL_n register.
- 1 is written to the SWRST bit in the CHCTRL_n register.

SWPRQ bit (Sweep Request)

[Setting condition]

- When SETSSWPRQ in the CHCTRL_n register is asserted

[Clearing conditions]

- The amount of data in the buffer becomes 0 because of forced ejection.

- 1 is written to the CLRHSWPRQM bit in the CHCTRL_n register.
- 1 is written to the SWRST bit in the CHCTRL_n register.

DNUM[7:0] bits (Data Number)

[Increment condition]

- When DMA read transfer is completed.

[Decrement condition]

- When DMA write transfer is completed.

[Clearing conditions]

- Conditions for clearing the EN bit are met.
- 1 is written to the SWRST bit in the CHCTRL_n register.

32.3.17.2 CHCTRL_n : Channel Control Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x428 + 0x40 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CLRDMARQM	SETDMARQM	CLRINTM	SETINTM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SETSWPRQ	—	SETREN	—	—	CLRSUS	SETSUS	CLRDER	CLRTC	CLRENDD	CLRRQ	SWRST	STG	CLREN	SETEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SETEN	Set Enable Enables DMA transfer of DMA channel n. If this bit is set together with the SWRST bit, clearing by the SWRST bit takes precedence, and a transfer does not start. 0: No effect 1: DMA transfer is enabled. (The EN bit in the CHSTAT_n register is set)	W
1	CLREN	Clear Enable Clears the EN bit in the CHSTAT_n register. For details, see section 32.16.8.3. Aborting a Transfer. 0: No effect 1: DMA transfer is disabled. (The EN bit in the CHSTAT_n register is cleared)	W
2	STG	Software Trigger Activates the software (setting the internal transfer request). If this bit is set together with the SWRST bit, clearing by the SWRST bit takes precedence. 0: No effect 1: A transfer request is set by software. (The RQST bit in the CHSTAT_n register is set)	W
3	SWRST	Software Reset Clears each bit in the CHSTAT_n register. (For details on bits to be cleared, see descriptions of each bit.) Set this bit to 1 when the EN bit and the TACT bit are 0. 0: No effect 1: Clears each bit in the CHSTAT_n register	W

Bit	Symbol	Function	R/W
4	CLRRQ	Clear Request Clears the RQST bit in the CHSTAT_n register. 0: No effect 1: Clears the RQST bit in the CHSTAT_n register	W
5	CLREND	Clear End Clears the END bit in the CHSTAT_n register, and clears USB_FDMAn to low level. 0: No effect 1: Clears the END bit in the CHSTAT_n register	W
6	CLRTC	Clear TC Clears the TC bit in the CHSTAT_n register. 0: No effect 1: Clears the TC bit in the CHSTAT_n register	W
7	CLRDER	Clear DER Clears the DER bit in the CHSTAT_n register, and clears USB_FDMAn to low level. 0: No effect 1: Clears the DER bit in the CHSTAT_n register	W
8	SETSUS	Set Suspend Setting this bit to 1 when the EN bit in the CHSTAT_n register is 1 suspends the ongoing DMA transfer. 0: No effect 1: Suspends the ongoing DMA transfer	W
9	CLRSUS	Clear Suspend Setting this bit to 1 when the SUS bit in the CHSTAT_n register is 1 clears the suspended state. 0: No effect 1: Clears the suspended status of the ongoing DMA transfer	W
11:10	—	The write value should be 0.	W
12	SETREN	Set Register Set Enable Sets the REN bit in the CHCFG_n register. 0: No effect 1: Sets the REN bit in the CHCFG_n register to 1	W
13	—	The write value should be 0.	W
14	SETSSWPRQ	Set Software Sweep Request Forcibly ejects the data in the buffer to the transfer destination. See section 32.16.3. Forced Ejection Request. If the transfer destination asserts a hardware request (REQD = 1), forced ejection is not available (disabled by hardware). 0: No effect 1: Writes the buffer data (that has not been written to the transfer destination) to the transfer destination	W
15	—	The write value should be 0.	W
16	SETINTM	Set Interrupt Mask Temporarily masks USB_FDMAn and sets the INTM bit in the CHSTAT_n register to 1. 0: No effect 1: Masks USB_FDMAn	W
17	CLRINTM	Clear Interrupt Mask Clears the NT_DMA [n] pin output mask status, and clears the INTM bit in the CHSTAT_n register to 0. 0: No effect 1: Clears the mask status set by the SETINTM bit	W
18	SETDMARQM	SET DMAREQ Mask Temporarily masks a DMA transfer request from the USB control, and sets the DMARQM bit in the CHSTAT_n register to 1. 0: No effect 1: Masks a DMA transfer request from the USB control	W

Bit	Symbol	Function	R/W
19	CLRDMARQM	Clear DMAREQ Mask Clears the temporary mask status of the DMA transfer request from the USB control, and clears the DMARQM bit in the CHSTAT_n register to 0. 0: No effect 1: Clears the mask status set by the SETDMARQM bit	W
31:20	—	The write value should be 0.	W

The CHCTRL_n register controls DMA transfer of DMA channel n (n = 1, 0). This register is used to activate each function, and it does not retain written values.

32.3.17.3 CHCFG_n : Channel Configuration Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x42C + 0x40 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMS	REN	RSW	RSEL	SBE	DIM	—	DEM	WONLY	TM	DAD	SAD	DDS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SDS[3:0]				DRRP	AM[2:0]			—	LVL	HIEN	LOEN	REQD	—	—	SEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SEL	Terminal Select Selects the FIFO channel to be used on the USB control side. 0: D0FIFO is used 1: D1FIFO is used	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	REQD	Request Direction Specifies whether the USB control side is source or destination. 0: The USB control side is source (initial value) 1: The USB control side is destination	R/W
4	LOEN	Sets the transfer request signal between the USB control and the DMAC. Always set this bit to 0.	R/W
5	HIEN	Sets the transfer request signal between the USB control and the DMAC. Always set this bit to 1.	R/W
6	LVL	Sets the transfer request signal between the USB control and the DMAC. Always set this bit to 1.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	AM[2:0]	These bits set the transfer request signal between the USB control and the DMAC. Always set these bits to 000b.	R/W
11	DRRP	Descriptor Read Repeat Switches operation when LV in the header is 0 after reading the descriptor. (See Figure 32.22.) 0: Sets the DER bit in the CHSTAT_n register to stop operation (initial value). 1: Reads the same descriptor continuously until LV becomes 1. When LV = 1, DMA transfer using the descriptor value starts. The descriptor read interval is controlled by the DSCITVL register.	R/W

Bit	Symbol	Function	R/W
15:12	SDS[3:0]	Source Data Size These bits set the DMA transfer size. When SDS[3] is set to 0, transfer mode is changed to normal mode. When SDS[3] is set to 1, transfer mode is changed to skip mode. Transfer size is set by SDS[2:0]. (For settable values, see Table 32.21.) 0 0 0: 8 bits (initial value) 0 0 1: 16 bits 0 1 0: 32 bits 0 1 1: 64 bits 1 0 0: 128 bits 1 0 1: 256 bits 1 1 0: 512 bits 1 1 1: 1024 bits	R/W
19:16	DDS[3:0]	Destination Data Size These bits set the DMA transfer size. When the USB control side is transfer destination, use the normal mode. When DDS[3] is set to 0, transfer mode is changed to normal mode. When SDS[3] is set to 1, transfer mode is changed to skip mode. Transfer size is set by DDS[2:0]. (For settable values, see Table 32.21.) 0 0 0: 8 bits (initial value) 0 0 1: 16 bits 0 1 0: 32 bits 0 1 1: 64 bits 1 0 0: 128 bits 1 0 1: 256 bits 1 1 0: 512 bits 1 1 1: 1024 bits	R/W
20	SAD	Source Address Direction Sets the count direction of the transfer source address of DMA channel n. When the USB control side is transfer source, set this bit to 1 (fixed). When using the skip mode on the transfer source side or when the transfer source is beat-unaligned, do not specify SAD = 1 (fixed). 0: Increment (initial value) 1: Fixed	R/W
21	DAD	Destination Address Direction Sets the count direction of the transfer destination address of DMA channel n. When the USB control side is transfer destination, set this bit to 1 (fixed). When using the skip mode on the transfer destination side or when the transfer destination is beat-unaligned, do not specify DAD = 1 (fixed). 0: Increment (initial value) 1: Fixed	R/W
22	TM	Sets the transfer request signal between the USB control and the DMAC. Always set this bit to 0.	R/W
23	WONLY	Write Only Mode Sets the write-only mode. For details, see section 32.15.1.3. Write-Only Mode. 0: Normal operation (initial value) 1: Write-only mode	R/W
24	DEM	USB_FDMAn Mask Masks USB_FDMAn interrupt detection. When this bit is 1 at USB_FDMAn timing, USB_FDMAn and END in the CHSTAT_n register are not asserted. In register mode, the DEM bit is automatically cleared to 0. In link mode, this bit is not cleared. 0: Does not mask USB_FDMAn interrupt detection (initial value) 1: Masks USB_FDMAn interrupt detection	R/W
25	—	This bit is read as 0. The write value should be 0.	R/W
26	DIM	Descriptor Interrupt Mask Sets masking for USB_FDMAn when LV = 0 after the descriptor's header is read. 0: Does not mask USB_FDMAn (initial value) 1: Masks USB_FDMAn	R/W

Bit	Symbol	Function	R/W
27	SBE	Sweep Buffer Enable Selects whether to sweep (by writing) the read data in the buffer before aborting the transfer when EN in the CHSTAT_n register is cleared to 0 during a DMA transaction. Sweep mode can be used only when REQD = 0. 0: Does not sweep buffer data and aborts the transfer (initial value) 1: Sweeps buffer data and aborts the transfer	R/W
28	RSEL	Register Set Select Selects the next register set to be executed next. This bit is valid only in register mode. When RSW = 1, this bit is automatically reversed upon completion of a DMA transaction. 0: Executes Next0 register set (initial value) 1: Executes Next1 register set	R/W
29	RSW	Register Select Switch The RSEL bit is automatically reversed upon completion of a DMA transaction. This bit is valid only in register mode. 0: Does not reverse the RSEL bit upon completion of a DMA transaction (initial value) 1: Reverses the RSEL bit upon completion of a DMA transaction	R/W
30	REN	Register Set Enable Performs DMA transaction of the next register set selected by the RSEL bit continuously after the current DMA transaction is completed. This bit is valid only in register mode. 0: Does not perform DMA transaction continuously 1: Performs DMA transaction continuously	R/W
31	DMS	DMA Mode Select Sets DMA mode. 0: Register mode (initial value) 1: Link mode	R/W

The CHCFG_n register controls DMA transfer of DMA channel n (n = 1, 0).

The settable size ranges of the SDS[2:0] and DDS[2:0] bits depend on the data bus width, number of implemented buffers, and alignment type of addresses to be accessed (beat-aligned or beat-unaligned). The following table shows the settable size ranges. These ranges are specified to prevent this controller from malfunctioning. Set appropriate values according to the module to be accessed.

Table 32.21 Settable size ranges of SDS[2:0] and DDS[2:0] bits

Data bus width [bit]	Number of buffers	Transfer address	Settable values of SDS[2:0] and DDS[2:0] bits	
32	16	Beat-aligned	8 to 32 bits 128 to 512 bits	(000b to 010b) (100b to 110b)
		Beat-unaligned	8 to 32 bits 128 to 256 bits	(000b to 010b) (100b to 101b)
		Beat-unaligned	8 to 32 bits	(000b to 010b)

If both transfer source and destination or either of these is beat-unaligned, set the SDS[2:0] and DDS[2:0] bits to values within the settable ranges for the beat-unaligned addresses from among the CHCFG_n register settings.

- Even when beat-aligned setting is made at the beginning of DMA transaction, if a skip transfer is used, beat-aligned setting may change to beat-unaligned setting during the DMA transaction. In this case, make beat-unaligned setting from the beginning.
- If the software cannot distinguish beat-aligned or beat-unaligned setting, use the beat-unaligned setting range.

DEM bit (USB_FDMAn Mask)

[Clearing condition]

A DMA transaction is completed when DEM = 1.

RSEL bit (Register Set Select)

[Transition condition]

A DMA transaction is completed when RSW = 1.

REN bit (Register Set Enable)

[Setting conditions]

- 1 is written to this bit.
- 1 is written to the SETREN bit in the CHCTRL_n register.

[Clearing conditions]

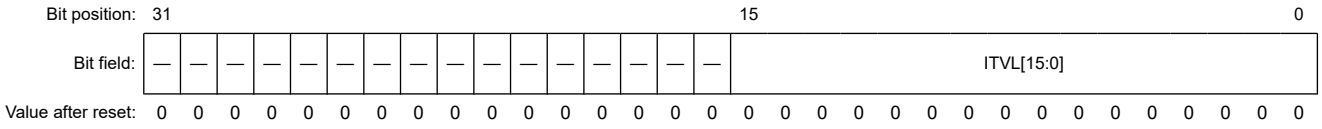
- 0 is written to this bit.
- A DMA transaction is completed when REN = 1.

To re-set the REN bit during a DMA transaction, use the SETREN bit in the CHCTRL_n register.

32.3.17.4 CHITVL_n : Channel Interval Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x430 + 0x40 × n



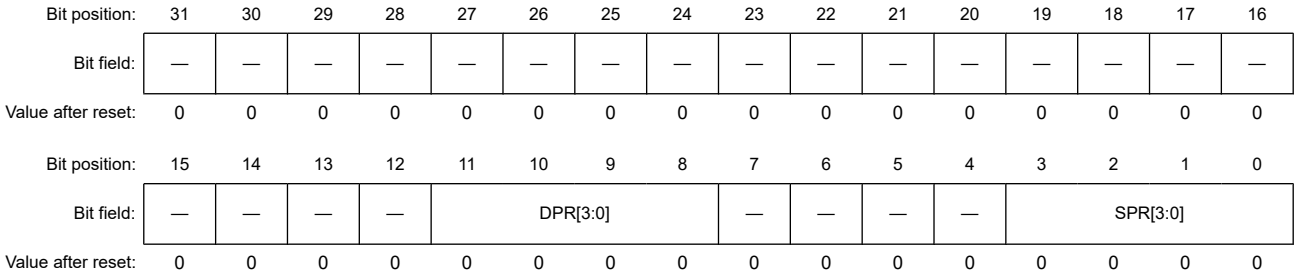
Bit	Symbol	Function	R/W
15:0	ITVL[15:0]	Interval These bits set the DMA transfer interval.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The CHITVL_n register sets the DMA transfer interval of DMA channel n (n = 1, 0).

32.3.17.5 CHEXT_n : Channel Extension Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x434 + 0x40 × n



Bit	Symbol	Function	R/W
3:0	SPR[3:0]	Source PROT These bits set the value to be output to the MHPROT[3:0] pins during DMA read transfer.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	DPR[3:0]	Destination PROT These bits set the value to be output to the MHPROT[3:0] pins during DMA write transfer.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The CHEXT_n register is an extension register for DMA channel n.

32.3.18 Link Register Set

The link register set is used to set and display the descriptor address in link mode.

When the descriptor address is set in the NXLA_n register by software to activate the DMAC, the NXLA_n register value is loaded to the CRLA_n register by hardware and the descriptor is read. The DMAC starts a DMA transaction according to the read descriptor value. The NXLA_n register value is automatically updated by the next link address value in the read descriptor to be used as the descriptor address for the next DMA transaction.

32.3.18.1 NXLA_n : Next Link Address Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x438 + 0x40 × n

Bit position:	31																												0
Bit field:	NXLA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	NXLA[31:0]	Next Link Address These bits set the address of the link destination. Because the two lower-order bits are fixed to 0, only word-aligned addresses can be set.	R/W

The NXLA_n register sets the link address of DMA channel n (n = 1, 0).

32.3.18.2 CRLA_n : Current Link Address Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x43C + 0x40 × n

Bit position:	31																												0
Bit field:	CRLA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CRLA[31:0]	Current Link Address These bits display the address of the descriptor that is being executed.	R

The CRLA_n register displays the link address of DMA channel n (n = 1, 0).

32.3.19 Skip Register Set

The skip register set is used for settings for skip (scatter/gather) transfer.

32.3.19.1 SCNT_n : Source Continuous Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x600 + 0x20 × n

Bit position:	31																												0
Bit field:	SCNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SCNT[31:0]	Source Continuous These bits set the size (in bytes) of continuous access space for access to the transfer source address.	R/W

The SCNT_n register sets the space size for continuous access to the transfer source address of DMA channel n (n = 1, 0).

Use this register together with the SSKP_n register (see Figure 32.4).

When using this mode, set the SDS[3] bit in the CHCFG_n register to 1.

When performing a skip transfer on the transfer source side, do not set the SAD bit in the CHCFG_n register to 1 (fixed). In addition, do not perform a skip transfer when SCNT_n is set to 0x00000000.

32.3.19.2 SSKP_n : Source Skip Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x604 + 0x20 × n

Bit position:	31																												0
Bit field:	SSKP[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SSKP[31:0]	Source Skip These bits set the skip amount (in bytes) during access to the source address.	R/W

The SSKP_n register sets the skip amount for skipping the next source address after accessing the data size set by the SCNT_n register during access to the source address of DMA channel n (n = 1, 0).

Use this register together with the SCNT_n register (see Figure 32.4).

When using this mode, set the SDS[3] bit in the CHCFG_n register to 1.

When performing a skip transfer on the transfer source side, do not set the SAD bit in the CHCFG_n register to 1 (fixed).

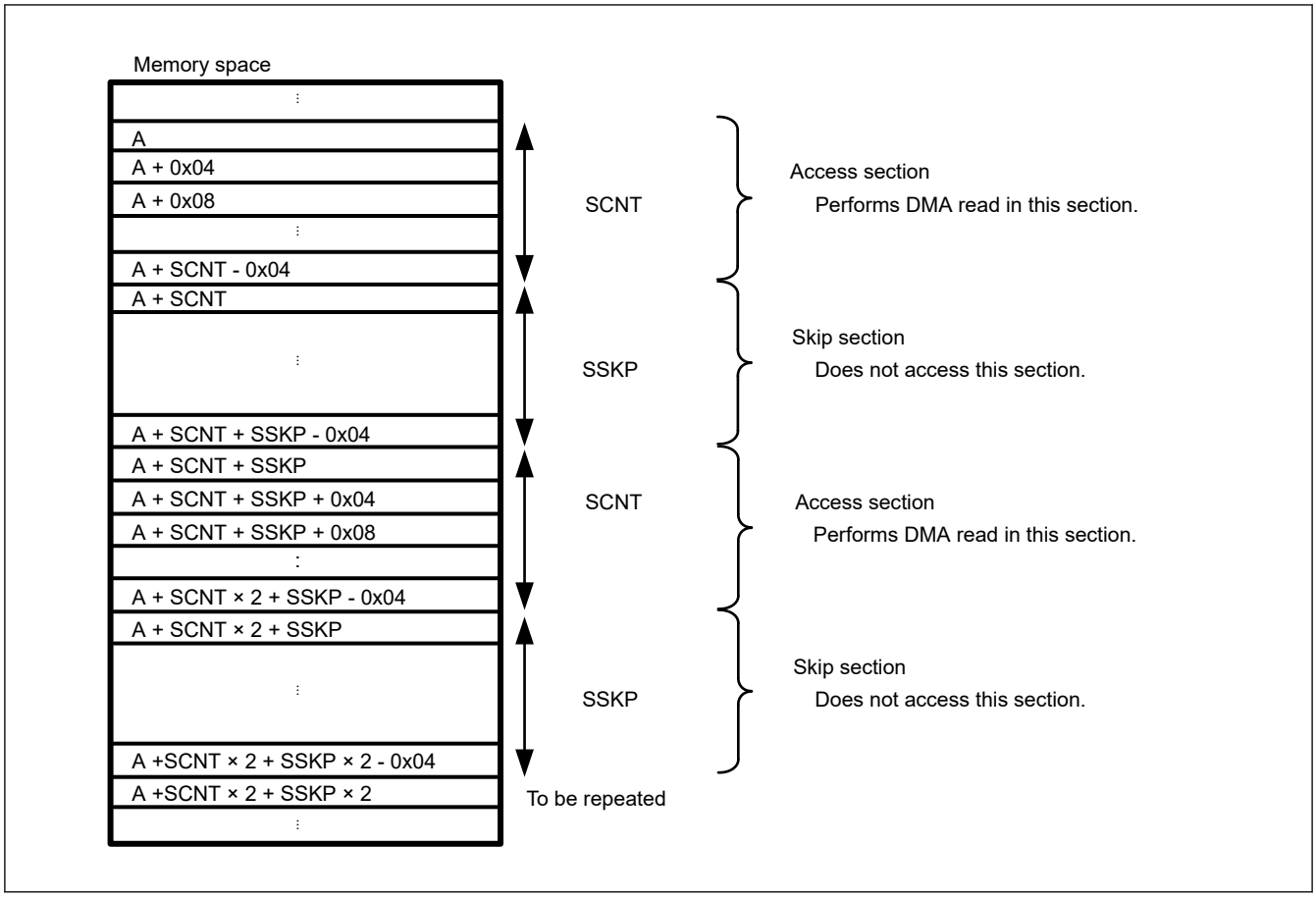


Figure 32.4 Relationship between SSKP and SCNT

The SCNT and SSKP_n values can be set regardless of the source address and the set value of the SDS field in the CHCFG_n register. The DMAC accesses a memory space with the size set in the SDS field, and acquires only valid data into the buffer. For details, see section 32.17.1.1. Read Access.

32.3.19.3 DCNT_n : Destination Continuous Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x608 + 0x20 × n

Bit position:	31																												0
Bit field:	DCNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DCNT[31:0]	Destination Continuous These bits set the space size (in bytes) for continuous access during an access to the destination address.	R/W

The DCNT_n register sets the space size for continuous access during an access to the destination address of DMA channel n (n = 1, 0).

Use this register together with the DSKP_n register (see Figure 32.5).

When using this mode, set the DDS[3] bit in the CHCFG_n register to 1.

When performing a skip transfer on the transfer destination side, do not set the DAD bit in the CHCFG_n register to 1 (fixed). In addition, do not perform a skip transfer when DCNT is set to 0x00000000.

32.3.19.4 DSKP_n : Destination Skip Register n (n = 0, 1)

Base address: USBF = 0x9204_1000

Offset address: 0x60C + 0x20 × n

Bit position:	31																												0
Bit field:	DSKP[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DSKP[31:0]	Destination Skip These bits set the skip amount (in bytes) during access to the destination address.	R/W

The DSKP_n register sets the skip amount for skipping the next destination address after accessing the data size set by the DCNT_n register during access to the destination address of DMA channel n (n = 1, 0).

Use this register together with the DCNT_n register (see Figure 32.5).

When using this mode, set the DDS[3] bit in the CHCFG_n register to 1.

When performing a skip transfer on the transfer destination side, do not set the DAD bit in the CHCFG_n register to 1 (fixed).

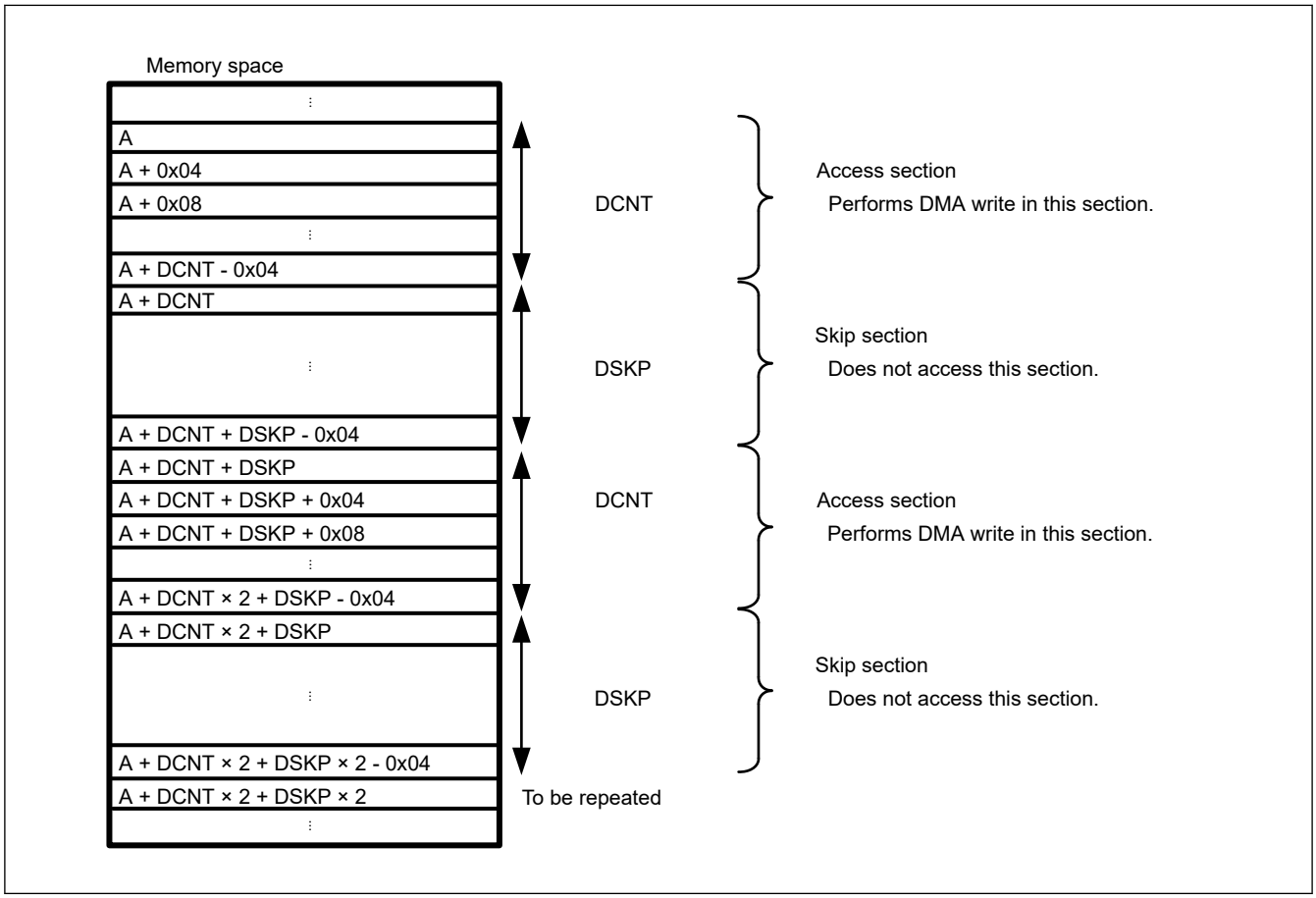


Figure 32.5 Relationship between DSKP and DCNT

The DCNT and DSKP_n values can be set regardless of the destination address and the set value of the DDS field in the CHCFG_n register. The DMAC performs write access only to the specified space with the size equal to or smaller than the value set in the DDS field. For details, see section 32.17.1.2. Write Access.

32.3.20 DMA Register Set

The following registers are common to all channels.

32.3.20.1 DCTRL : DMA Control Register

Base address: USBF = 0x9204_1000

Offset address: 0x700

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	LWPR[3:0]				—	—	—	—	LDPR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PR	Priority Sets transfer priority control mode. See section 32.16.2. DMA Channel Priority Control. 0: Fixed priority mode 1: Round-robin mode	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
19:16	LDPR[3:0]	Link Descriptor PROT These bits set the value to be output to the MHPROT[3:0] pins during descriptor read in link mode.	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	LWPR[3:0]	Link WriteBack PROT These bits set the value to be output to the MHPROT[3:0] pins during descriptor writeback in link mode.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The DCTRL register sets the transfer type for descriptor access and the arbitration between channels.

32.3.20.2 DSCITVL : Descriptor Interval Register

Base address: USBF = 0x9204_1000

Offset address: 0x704

Bit position:	31								15							8								0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DITVL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
15:8	DITVL[7:0]	Descriptor Interval These bits set the descriptor read interval. The descriptor is read again with an interval of (DITVL × 256) cycles.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The DSCITVL register sets the descriptor read interval.

By setting the DRRP bit in the CHCFG_n register to 1, the descriptor is continuously read until it reaches LV = 1. This register sets the read interval.

32.3.20.3 DSTAT_EN : DMA Status EN Register

Base address: USBF = 0x9204_1000

Offset address: 0x710

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EN1	EN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN0	Channel 0 EN Displays the EN bit status of DMA channel 0.	R
1	EN1	Channel 1 EN Displays the EN bit status of DMA channel 1.	R
31:2	—	These bits are read as 0.	R

The DSTAT_EN register displays the EN bit status of all channels.

Writing data to this register does not change the value of each bit.

32.3.20.4 DSTAT_ER : DMA Status ER Register

Base address: USBF = 0x9204_1000

Offset address: 0x714

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ER1	ER0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER0	Channel 0 ER Displays the ER bit status of DMA channel 0.	R
1	ER1	Channel 1 ER Displays the ER bit status of DMA channel 1.	R
31:2	—	These bits are read as 0.	R

The DSTAT_ER register displays the ER bit status of all channels.

Writing data to this register does not change the value of each bit.

32.3.20.5 DSTAT_END : DMA Status END Register

Base address: USBF = 0x9204_1000

Offset address: 0x718

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	END1	END0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	END0	Channel 0 END Displays the END bit status of DMA channel 0.	R
1	END1	Channel 1 END Displays the END bit status of DMA channel 1.	R
31:2	—	These bits are read as 0.	R

The DSTAT_END register displays the END bit status of all channels.

Writing data to this register does not change the value of each bit.

32.3.20.6 DSTAT_TC : DMA Status TC Register

Base address: USBF = 0x9204_1000

Offset address: 0x71C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TC1	TC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TC0	Channel 0 TC Displays the TC bit status of DMA channel 0.	R
1	TC1	Channel 1 TC Displays the TC bit status of DMA channel 1.	R
31:2	—	These bits are read as 0.	R

The DSTAT_TC register displays the TC bit status of all channels.

Writing data to this register does not change the value of each bit.

32.3.20.7 DSTAT_SUS : DMA Status SUS Register

Base address: USBF = 0x9204_1000

Offset address: 0x720

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SUS1	SUS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SUS0	Channel 0 SUS Displays the SUS bit status of DMA channel 0.	R
1	SUS1	Channel 1 SUS Displays the SUS bit status of DMA channel 1.	R
31:2	—	These bits are read as 0.	R

The DSTAT_SUS register displays the SUS bit status of all channels.

Writing data to this register does not change the value of each bit.

32.4 Operation

32.4.1 System Controls and Oscillation Controls

This section describes the register operations required for initial settings of this module, and the registers necessary for power consumption control. The startup sequence is as follows.

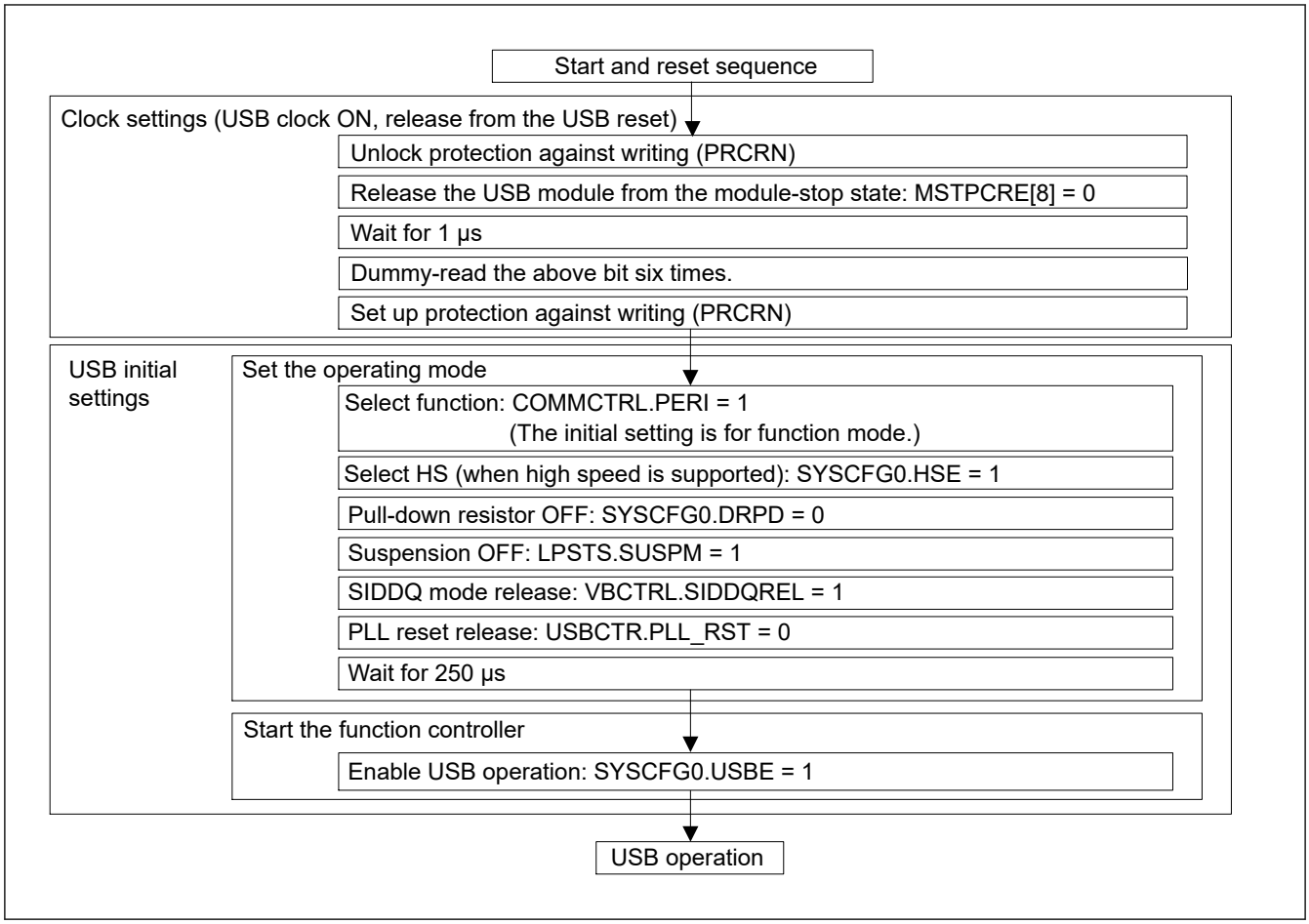


Figure 32.6 Startup sequence

32.4.2 Resets

Table 32.22 lists the types of reset for this controller. For the initial states of the registers following the reset operations, see .

Table 32.22 Types of reset

Name	Operation
Hardware reset	Low level input from the RES# pin
USB bus reset	Automatically detected by this module from the D+ and D- lines

32.4.3 USB Data Bus Resistor Controller

This controller controls the pull-up resistors for the D+ signal of the USB2.0-PHY port.

Make pull-up setting for the D+ signal using the DPRPU and DRPD bits of the SYSCFG0 register.

Confirm the connection to the USB Host before setting the DPRPU bit of the SYSCFG0 register to 1 to pull up D+.

This controller incorporates termination resistors for the D+ and D− signals (in high-speed operation mode) and output resistors (in full-speed operation mode). The switching of the internal resistors following the connection to the PC is automatically done by the controller during reset handshake, suspend, and resume processing.

If the DPRPU bit of the SYSCFG0 register is set to 0 during communication with the PC, the controller disables the pull-up resistors (or termination resistors) of the USB data lines, so that it can notify the host controller of the disconnection of the device.

32.4.4 Notes on Stopping Clocks

- The PHY clock can be stopped with the SUSPM register in suspend mode.

- If a clock is to be stopped while the controller is in USB suspended state, however, it is necessary to resume the supply of the clock during resume processing. The resumption of the PHY clock need be accomplished within 5.5 ms after a resume interrupt occurs.
- When the USB function is not in use, place the USB module in the module-stop state (MSTPCRE[8] = 1).

32.5 Interrupt Functions

32.5.1 Overview of Interrupt Functions

Table 32.23 lists the interrupt functions of this controller.

Table 32.23 List of interrupt functions

Bit	Interrupt name	Interrupt source	Related status
VBINT	VBUS interrupt	When a change in the state of the USB_VUBUSIN input pin has been detected (low to high or high to low).	VBSTS
RESM	Resume interrupt	When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	—
SOFR	Frame number update interrupt	If SOFRM = 0: When an SOF packet with a different frame number is received. If SOFRM = 1: When an SOF packet for a μ frame number of 0 cannot be received due to a corrupted packet.	—
DVST	Device state transition interrupt	When transition in device state is detected: A USB bus reset detected The suspended state detected Set Address request received Set Configuration request received	DVSQ
CTRT	Control transfer stage transition interrupt	When a stage transition is detected in control transfer: Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred	CTSQ
BEMP	Buffer empty interrupt	When the buffer becomes empty after all data in the buffer memory has been transmitted. When a packet exceeding the maximum packet size is received.	PIPEBEMP
NRDY	Buffer not ready interrupt	When a token is received when PID = BUF and the buffer memory is not ready to transmit or receive data. When a CRC error or bit stuff error occurs while receiving data in isochronous transfer. When an interval error occurs while receiving data in isochronous transfer.	PIPENRDY
BRDY	Buffer ready interrupt	When the buffer is ready (readable or writable).	PIPEBRDY

Table 32.24 shows operations for a USBf interrupt output from this controller. In case more than one interrupt source is generated, the USBf interrupt output method can be set by using the INTL bit in the SOFCFG register. Set the USBf interrupt output operation according to the user system.

Table 32.24 Operations for USBf Interrupt (USB_FI)

INTL setting	USBf interrupt output pin operation	
	When one interrupt source is generated	When more than one interrupt source is generated
Edge sense (INTL = 0)	Holds the pin low until the interrupt source is cleared.	Negated (L pulse output) for 40 clock cycles at 60 MHz when one source is cleared.
Level sense (INTL = 1)	Holds the pin low until the interrupt source is cleared.	Holds the pin low until all sources are cleared.

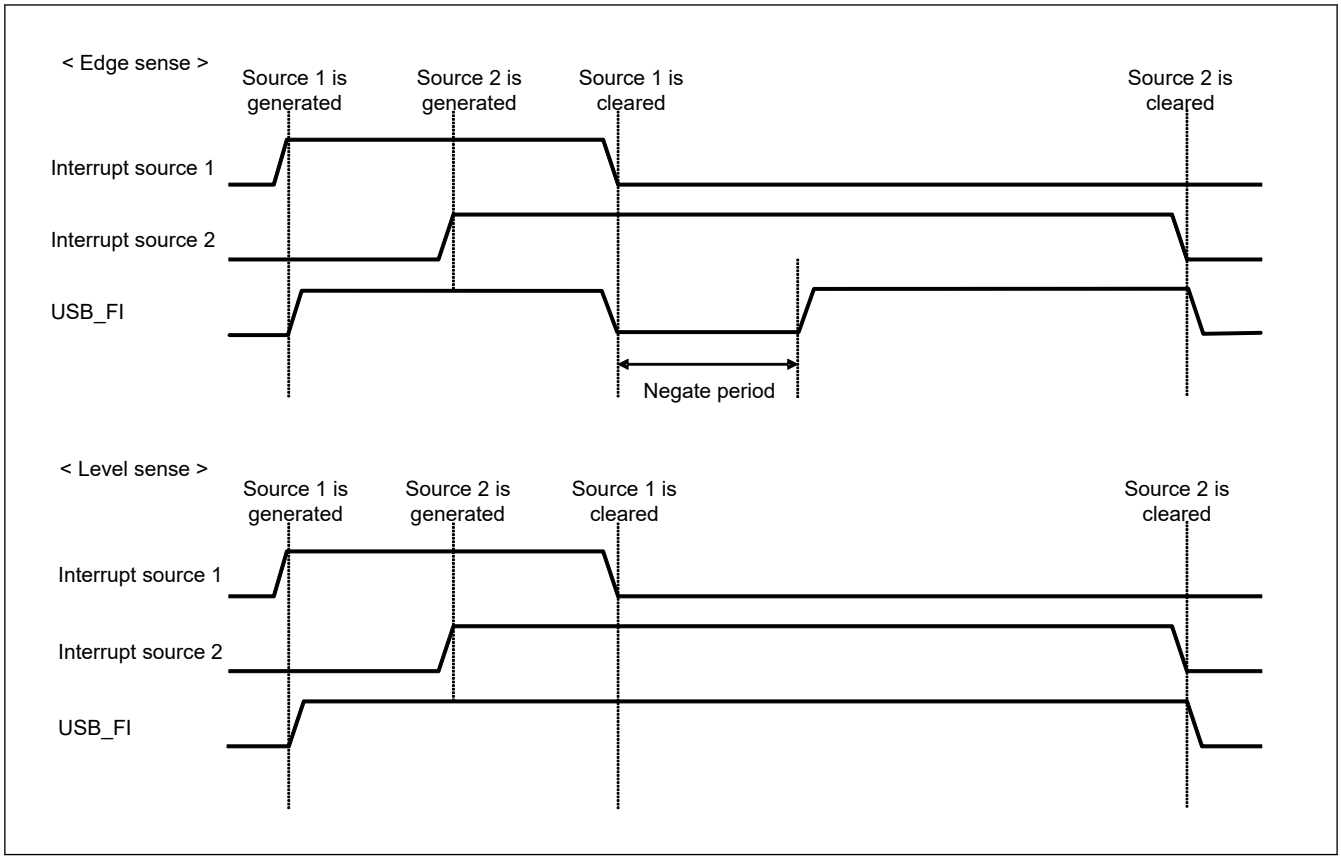


Figure 32.7 USBf interrupt output operation diagram

Figure 32.8 shows the interrupt configurations for the controller.

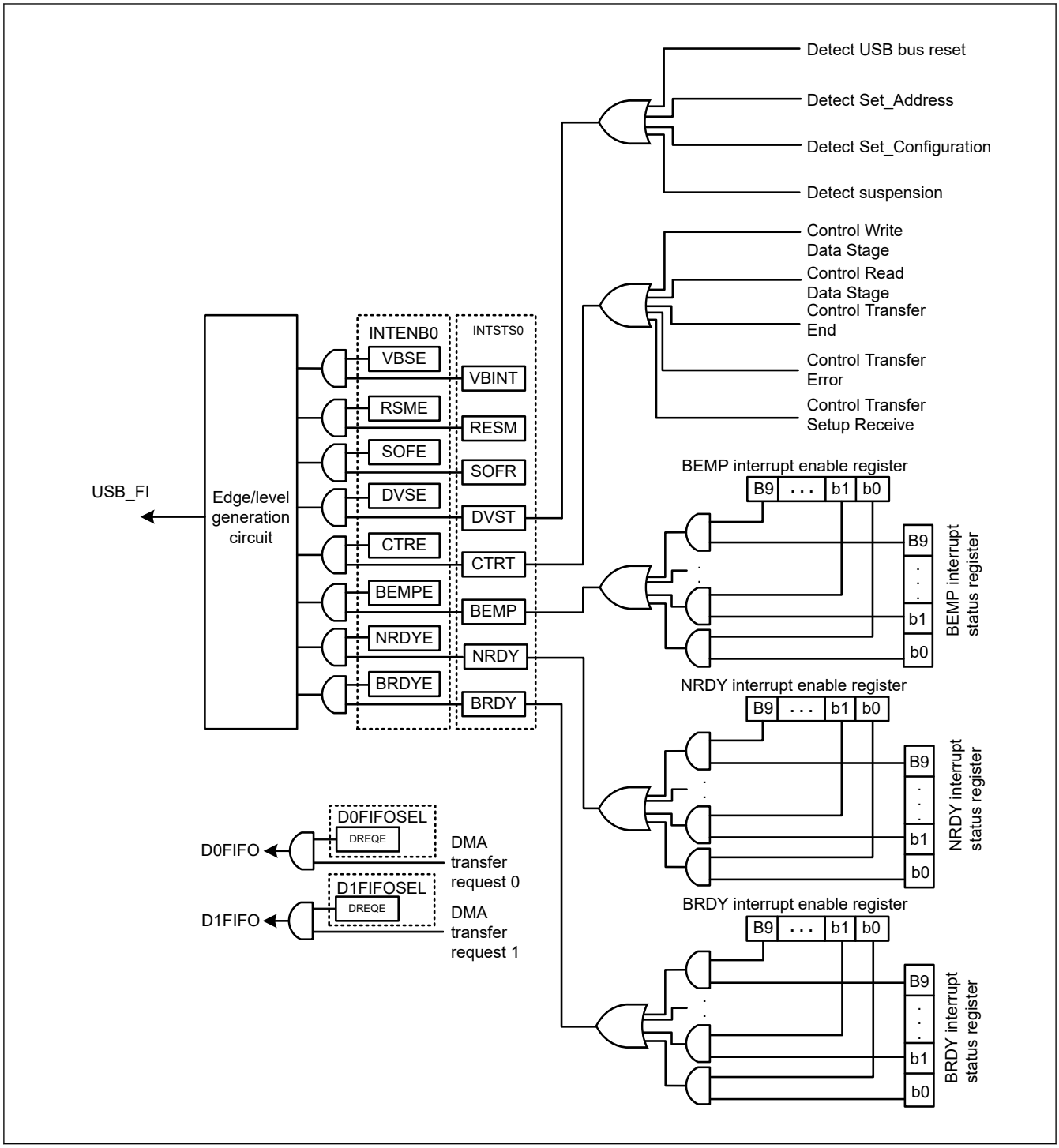
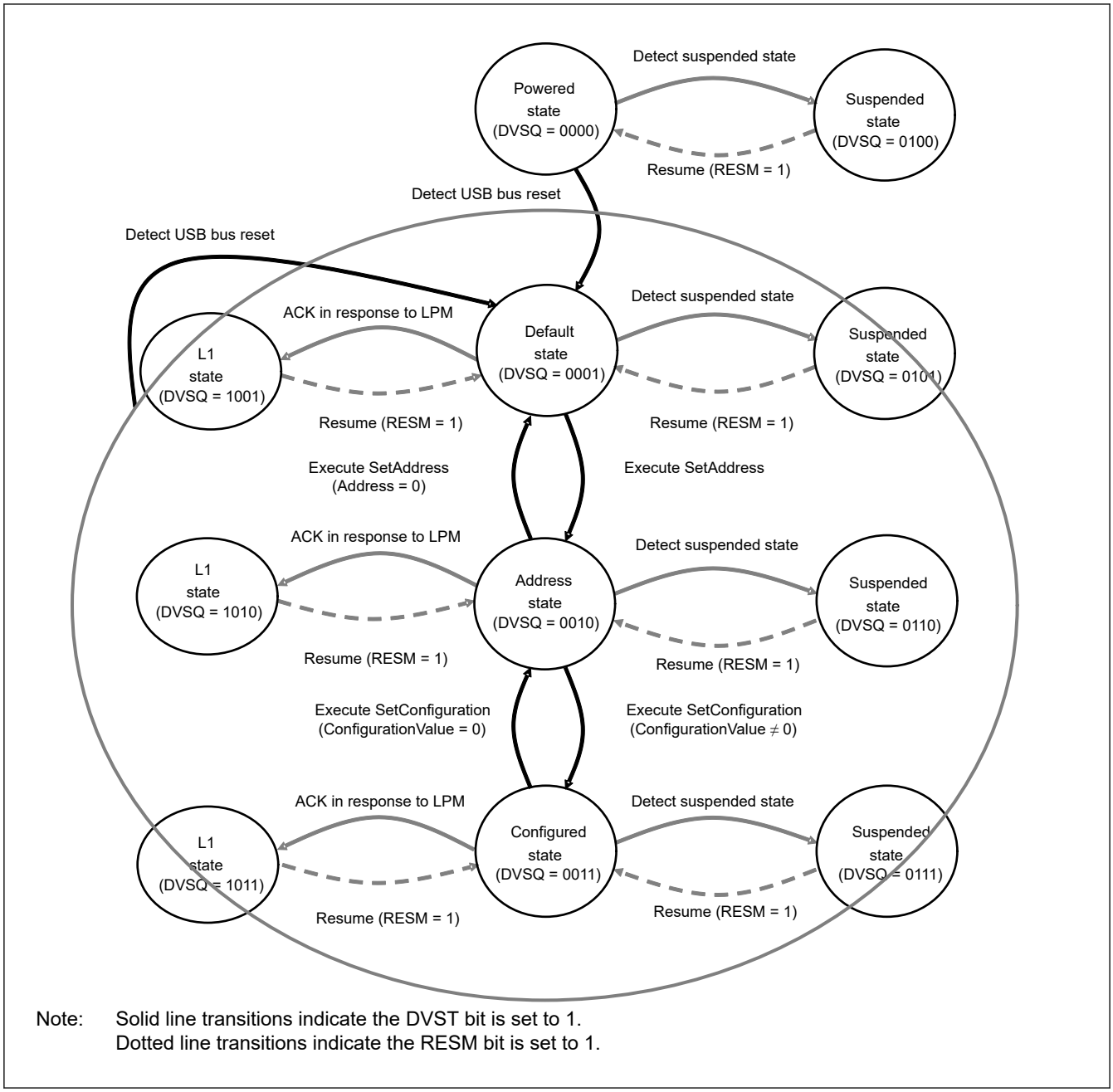


Figure 32.8 Interrupt configuration diagram

32.5.2 Device State Transition Interrupt

Figure 32.9 shows a diagram of how this module handles the device state transitions. The controller monitors the device states and generates the device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by the resume interrupt. The device state transition interrupts can be enabled or disabled individually by setting the INTENB0 register. The device state after a transition can be confirmed using the DVSQ bits in the INTSTS0 register.

When making a transition to the default state, the device state transition interrupt is generated after the reset handshake protocol has been completed.



Note: Solid line transitions indicate the DVST bit is set to 1.
Dotted line transitions indicate the RESM bit is set to 1.

Figure 32.9 Device state transitions

32.5.3 Control Transfer Stage Transition Interrupt

Figure 32.10 shows a diagram of how this module handles the control transfer stage transition. The controller monitors the control transfer sequence and generates the control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually in the INTENB0 register. The control transfer stage after a transition can be confirmed in the CTSQ bits in the INTSTS0 register.

The control transfer sequence errors are described below. If an error occurs, the PID bits in DCPCTR are set to 1xb (STALL).

1. During control read transfers
   (a) At the IN token of the data stage, an OUT or PING token is received when there have been no data transfers at all.
   (b) An IN token is received at the status stage.
   (c) A packet is received at the status stage for which the data packet is DATAPID = DATA0.
2. During control write transfers

(a) At the OUT token of the data stage, an IN token is received when there have been no ACK response at all.

(b) A packet is received at the data stage for which the first data packet is DATAPID = DATA0.

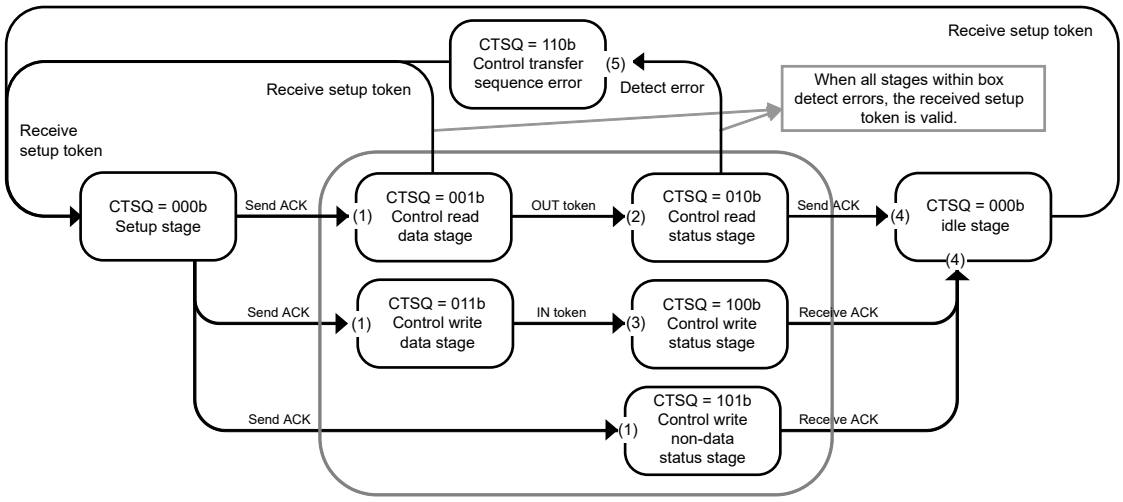
(c) At the status stage, an OUT or PING token is received.

3. During control write no-data transfers

(a) At the status stage, an OUT or PING token is received.

Note that in the control write transfer data stage, if the number of received data is more than the USB request wLength value, the control transfer sequence error cannot be recognized. Also, in the control read transfer status stage, when a packet other than a zero-length packet is received, an ACK response is returned and the transfer is successfully completed.

When a CTRT interrupt is generated due to a sequence error (SERR = 1), the value of CTSQ = 110b is retained until CTRT = 0 is written by the user system (clearing the interrupt state). Therefore, while CTSQ = 110b is retained, the CTRT interrupt for the completion of the setup stage is not generated, even when a new USB request is received. Events occurring after the setup stage are saved by the controller and the CTRT interrupt is generated after the interrupt state is cleared by software.



Note: CTRT interrupt
(1) Setup stage end
(2) Control read transfer status stage transition
(3) Control write transfer status stage transition
(4) Control transfer complete
(5) Control transfer sequence error

Figure 32.10 Control transfer stage transition

32.6 Pipe Control

Table 32.25 provides a list of pipe settings for the controller. In USB data transfers, data transmission is executed in logic pipes called endpoints. This controller comes with ten pipes for data transfer. Each pipe can be set to meet the requirements of the user system.

Table 32.25 Pipe settings

Register name	Bit name	Setting	Remarks
PIPECFG	TYPE	Specifies the transfer type	Can be set for pipes 1 to 9.
	BFRE	Selects BRDY interrupt mode	Can be set for pipes 1 to 5.
	DBLB	Selects double buffer configuration	Can be set for pipes 1 to 5.
	CNTMD	Selects continuous transfer or noncontinuous transfer	Can be set for pipes 1 and 2 only in bulk transfers. Can be set for pipes 3 to 5.
	DIR	Selects transfer direction	Set to IN or OUT
	EPNUM	Endpoint number	Can be set for pipes 1 to 9. Set to a value other than 0x0 when a pipe is in use.
	SHTNAK	Disables pipe when transfer is completed.	Can be set for pipes 1 and 2 only in bulk transfers. Can be set for pipes 3 to 5.
PIPEBUF	BUFSIZE	Buffer memory size	Cannot be set for DCP (fixed to 256 bytes). Up to 2 KB can be set for pipes 1 to 5. Cannot be set for pipes 6 to 9 (fixed to 64 bytes).
	BUFNMB	Buffer memory number	Cannot be set for DCP (fixed at areas 0x00 to 0x03). Can be set for pipes 1 to 5 (specifiable in area ranging from 0x08 to 0x80). Cannot be set for pipes 6 to 9 (fixed at areas ranging from 0x04 to 0x07).
DCPMAXP PIPEMAXP	MXPS	Maximum packet size	Setting conforming to the USB standard
PIPEPERI	IFIS	Buffer flush	Can be set for pipes 1 and 2 only in isochronous transfers. Cannot be set for pipes 3 to 5. Cannot be set for pipes 6 to 9.
	IITV	Interval counter	Can be set for pipes 1 and 2 only in isochronous transfers. Cannot be set for pipes 3 to 5. Cannot be set for pipes 6 to 9.
DCPCTR PIPExCTR	BSTS	Buffer status	DCP state switched between receive and transmission buffer by ISEL bit
	INBUFM	IN buffer monitor	Available only for pipes 3 to 5.
	ATREPM	Auto response mode	Can be set for pipes 1 to 5.
	ACLRM	Auto buffer clear	Can be set for pipes 1 to 9.
	SQCLR	Sequence clear	Clears data toggle bit.
	SQSET	Sequence set	Sets data toggle bit.
	SQMON	Sequence check	Monitors data toggle bit.
	PBUSY	Pipe busy check	—
	PID	Response PID	—
PIPExTRE	TRENB	Transaction count enable	Can be set for pipes 1 to 5.
	TRCLR	Current transaction counter clear	Can be set for pipes 1 to 5.
PIPExTRN	TRNCNT	Transaction counter	Can be set for pipes 1 to 5.

32.6.1 Maximum Packet Size Setting

The MXPS bits in DCPMAXP and PIPEMAXP are used to specify the maximum packet size for each pipe. DCP and PIPE1 to PIPE5 can be set to any of the maximum packet sizes defined by the USB Specification. For PIPE6 to PIPE9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

DCP: 64 should be set when using high-speed operation.

DCP: Select and set 8, 16, 32, or 64 when using full-speed operation.

Pipes 1 to 5: 512 should be set when using high-speed bulk transfer.

Pipes 1 to 5: Select and set 8, 16, 32, or 64 when using full-speed bulk transfer.

Pipes 1 to 2: Set a value between 1 and 1024 when using high-speed isochronous transfer.

Pipes 1 to 2: Set a value between 1 and 1023 when using full-speed isochronous transfer.

Pipes 6 to 9: Set a value between 1 and 64.

The high bandwidth transfers used with interrupt transfers and isochronous transfers are not supported.

32.6.2 Response PID

Set the response PID for each pipe with the PID bits of the DCPCTR and PIPExCTR registers.

1. Response PID setting
   The response PID specifies the response to a transaction from the Host.
   (a) NAK setting: Always sends a NAK response when a transaction is issued.
   (b) BUF setting: Responds to the transaction in accordance with the buffer memory state.
   (c) STALL setting: Always sends a STALL response when a transaction is issued.

   Regardless of the value set in the PID bit, an ACK is always sent as a response to a setup transaction and the USB request is stored in corresponding registers.
   Based on the results of the transaction, the controller may trigger the PID bits to be written.
   The controller will trigger a write event to the PID bit in the following cases.
2. Hardware setting of response PID
   (a) NAK setting:
      (i) When SETUP token is received normally (only DCP)
      (ii) In bulk transfers when the SHTNAK bit of the PIPECFG register is set to 1 and short packet is received.
      (iii) In bulk transfers when SHTNAK bit is set to 1 and the transaction counter is completed.
   (b) BUF setting: The BUF cannot be written by the controller.
   (c) STALL setting:
      (i) When a maximum packet size over error is detected in the received data packet
      (ii) When a control transfer sequence error is detected

32.6.3 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be re-written only when USB transmission is disabled (PID = NAK). Figure 32.11 shows the procedure for switching the pipe control register from the USB transmission enabled (PID = BUF) state.

Registers for which settings are prohibited when the USB transmission is enabled (PID = BUF):

- All bits of the DCPMAXP register
- Bits SQCLR and SQSET of the DCPCTR register
- All bits of the PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI registers
- ATREPM, ACLRM, SQCLR, and SQSET bits of the PIPExCTR register
- All bits of the PIPExTRE and PIPExTRN registers

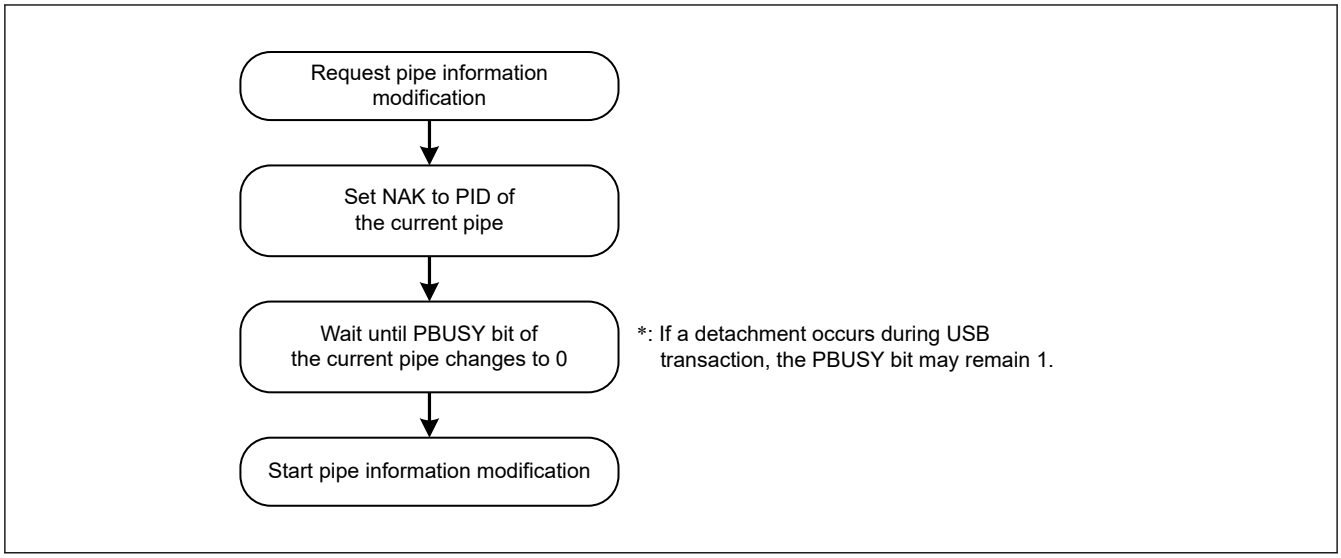


Figure 32.11 Procedure for changing pipe information from USB transmission enabled (PID = BUF) state

In addition, the following bits of the pipe control registers can only be re-written with pipe information that is not set in the CURPIPE bits of CPU/DMA0/DMA1-FIFO ports.

- Register for which settings are prohibited during setting of the CURPIPE bits in the FIFO port register:
  - All bits of the DCPMAXP register
  - All bits of the PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI registers
  - Bit ACLRM of the PIPExCTR register

When modifying information of a pipe, specify other pipe number in the CURPIPE bits. Also, after setting the DCP pipe information, clear the buffer using the BCLR bit.

32.6.4 Data PID Sequence Bit

When a normal data transfer occurs in the control transfer data stage, bulk transfer or interrupt transfer, the controller automatically toggles the data PID sequence bit. The next data PID sequence bit for data transfer can be confirmed in the SQMON bit in the DCPCTR or PIPExCTR registers. The sequence bit is switched in the ACK handshake receive timing when data is sent or in the ACK handshake send timing when data is received. The data PID sequence bit can also be modified for the SQCLR and SQSET bits of the DCPCTR and PIPExCTR registers.

In control transfers, the controller automatically sets the sequence bit for stage transitions. DATA1 is established at the end of the setup stage. In the status stage, the controller does not reference the sequence bit and returns a response with PID = DATA1. Therefore, the bit does not need to be set with software. Note that the data PID sequence bit must be set with software when a ClearFeature request is received.

Finally, the sequence bit cannot be manipulated through the SQSET bit for the isochronous transfer setup pipe.

32.7 FIFO Buffers

This section describes the operation of the FIFO buffers in this controller.

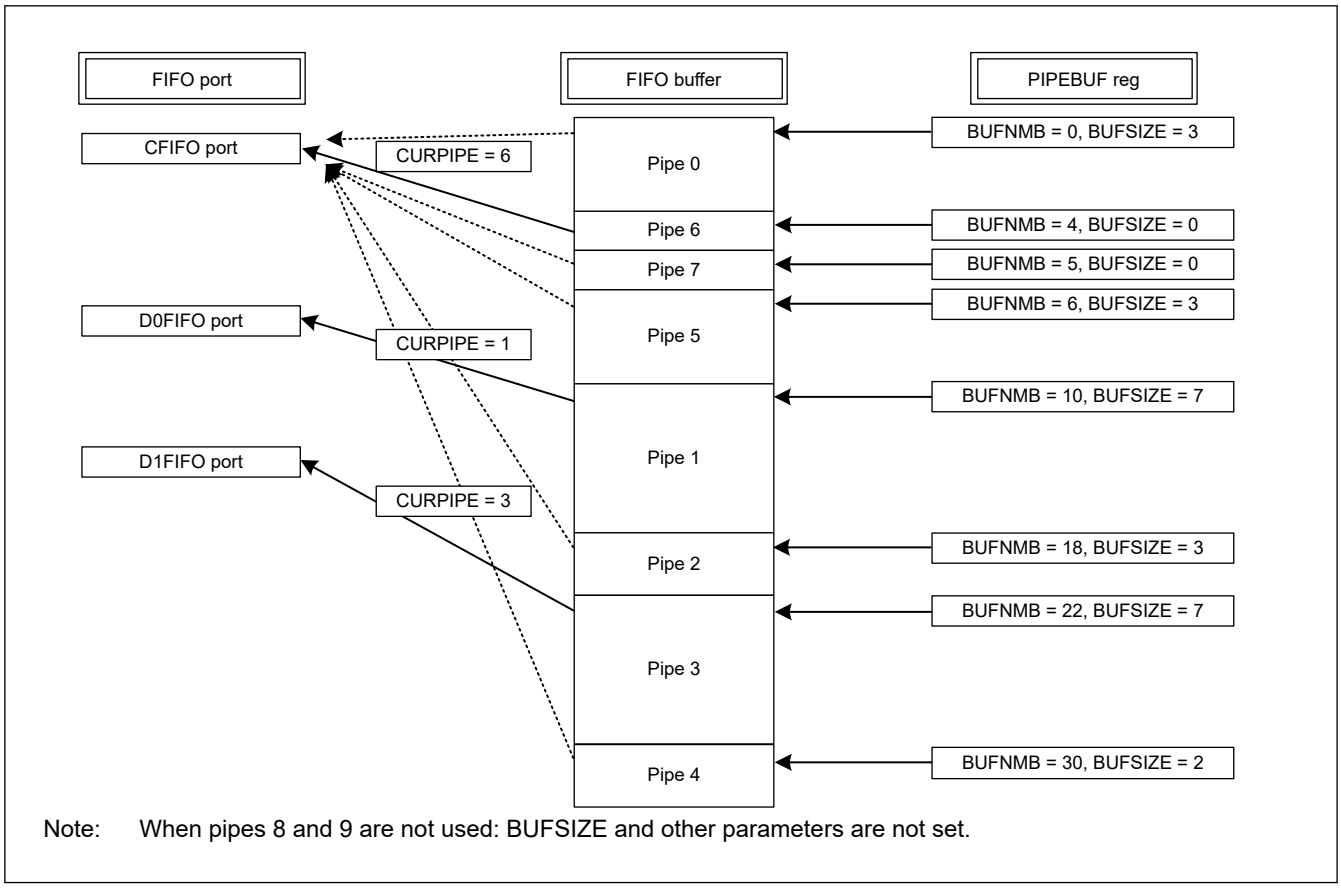
32.7.1 FIFO Buffer Allocation

Figure 32.12 shows an example of memory map for the FIFO buffers of this controller. The FIFO buffer area is shared by the CPU controlling the user system and this controller. The access right of the FIFO buffers may be given to the user system (CPU side) or to this controller (SIE side).

An independent FIFO buffer area is allocated for each pipe. The memory area is made up of memory blocks of 64 bytes and defined by the starting block number (1 block is 64 bytes long) and the number of blocks (specified by the BUFNMB and BUFSIZE bits of the PIPEBUF register). If the CNTMD bit of the PIPExCFG register is set to continuous transfer mode, the value specified in the BUFSIZE bits must be an integral multiple of the maximum packet size. If the double buffer configuration is selected through the DBLB bit of the PIPExCFG register, two planes of memory area the size of which is specified by the BUFSIZE bits of the PIPEBUF register are allocated to a single pipe.

Two FIFO ports are used to access an FIFO buffer (data read/write). The pipe to be assigned to an FIFO port is designated by specifying the pipe number in the CURPIPE bits of the CFIFOSEL or DxFIFOSEL register.

The FIFO buffer state of each pipe can be confirmed by using the BSTS and INBUFM bits of the DCPCTR and PIPExCTR registers. The access right of a FIFO port can be confirmed by using the FRDY bit of the CFIFOCTR or DxFIFOCTR register.



Note: When pipes 8 and 9 are not used: BUFSIZE and other parameters are not set.

Figure 32.12 Example of FIFO buffer memory map

32.7.2 Clearing FIFO Buffers

Table 32.26 shows a list of modes of clearing the FIFO buffers by this controller. The FIFO buffers can be cleared by the 3 bits that are listed in the table.

Table 32.26 List of FIFO buffer clearing modes

Bit name	BCLR	DCLRM	ACLRM
Register	CFIFOCTR register DxFIFOCTR register	DxFIFOSEL register	PIPExCTR register
Function	The FIFO buffer on the CPU side is cleared.	The FIFO buffer is automatically cleared after the data is read from the designated pipe.	The buffer is automatically cleared to discard all the received packets.
Clearing method	Write 1 to clear.	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

32.8 FIFO Port Function

This section describes FIFO port functions. Table 32.27 lists settings of FIFO port functions of this controller. If writing data is continued until the buffer becomes full (up to the maximum packet size in discontinuous transfer mode) during the data write access, the FIFO port automatically enters the state where data can be transmitted to the USB bus. To make data with a size less than the buffer full (the maximum packet size in discontinuous transfer mode) transmittable, write completion must

be set by the BVAL bit in the CFIFOCTR or DxFIFOCTR register. Furthermore, to transmit a zero-length packet, clear the buffer by the BCLR bit in the CFIFOCTR or DxFIFOCTR register and set write completion by the BVAL bit.

When all data has been read during the read access, the FIFO port automatically enters the state where a new packet can be received. However, when a zero-length packet is received (DTLN = 0), the data cannot be read. In that case, clear the buffer by the BCLR bit in the CFIFOCTR or DxFIFOCTR register. The receive data length is checked by the DTLN bits in the CFIFOCTR or DxFIFOCTR register.

Table 32.27 FIFO Port function settings

Register name	Bit name	Function	Notes
C/DxFIFOSEL	RCNT	DTLN read mode selection	
	REW	Buffer memory rewind (re-read, re-write)	
	DCLRM	Automatic clearing of buffer memory after specified pipe received data is read	DxFIFO only
	DREQE	DREQ signal assertion	DxFIFO only
	MBW	FIFO port access bit width	
	BIGEND	FIFO port endian selection	
	ISEL	FIFO port access direction	DCP only
	CURPIPE	Current pipe selection	
C/DxFIFOCTR	BVAL	Buffer memory write completion	
	BCLR	Clearing of CPU-side buffer memory	
	FRDY	Monitoring of FIFO port ready	
	DTLN	Confirmation of received data length	

32.8.1 FIFO Port Selection

Table 32.28 shows the list of pipes that can be selected in each FIFO port. The pipes to be accessed are selected with the CURPIPE bits of the CFIFOSEL or DxFIFOSEL register. After selecting the pipes, confirm that the value of the CURPIPE bits written was read correctly (if the previous pipe number is read out, this indicates the controller is still changing the pipe), then confirm that FRDY = 1 and access the FIFO port. Figure 32.13 shows the procedure for switching pipes for access to the FIFO port.

Also, select the bus width for the FIFO port access with the MBW bit. The buffer memory access direction is determined by the ISEL bit for DCP, and the DIR bit of the PIPExCFG register for all other pipes.

Table 32.28 FIFO port access by pipe

Pipe	Access method	Usable ports
DCP	CPU access	CFIFO port register
Pipes 1 to 9	CPU access	CFIFO port register
	DMA access	DxFIFO port register

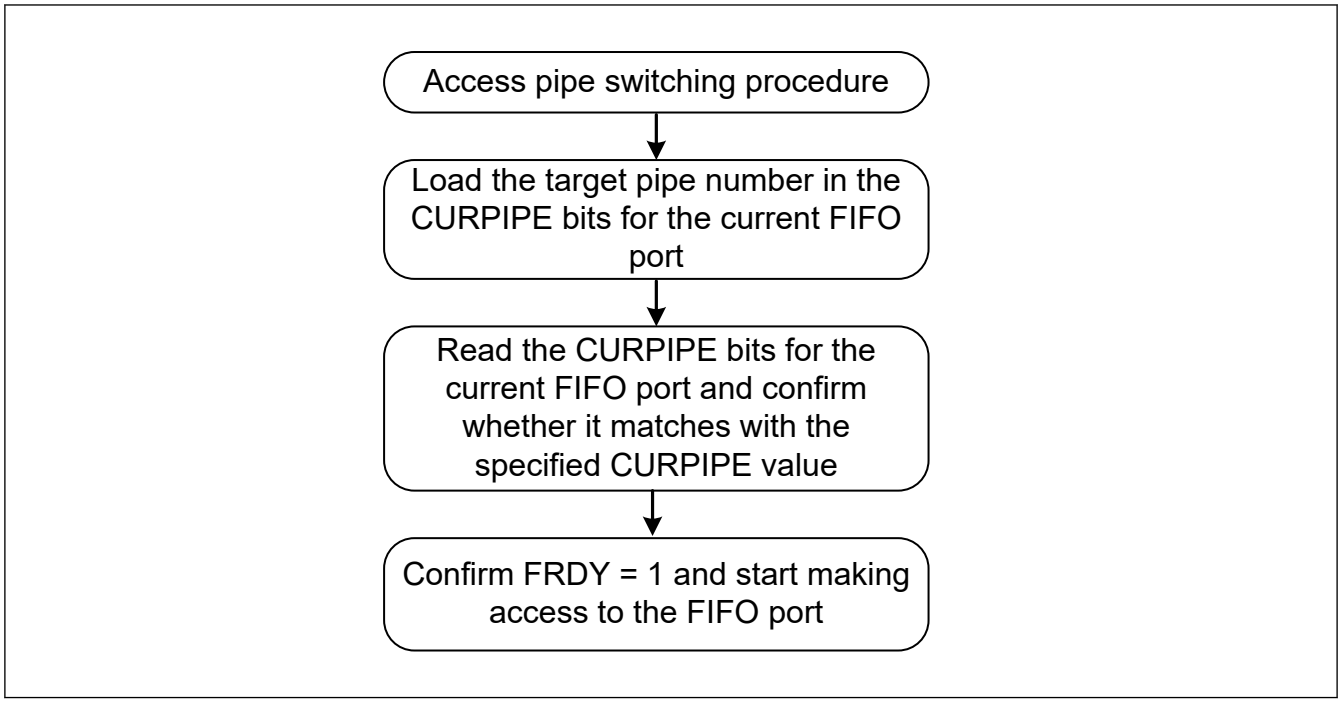


Figure 32.13 Pipe switching procedure for FIFO port access

32.8.2 DxFIFO Automatic Clear Mode (DxFIFO Port Read Direction)

When a data read event of the controller buffer memory is completed with setting the DCLRM bit of the DxFIFOSEL register to 1, the buffer memory of the corresponding pipe is automatically cleared.

Table 32.29 shows the correspondence between packet reception and buffer memory clearing by software in each setting.

As indicated in Table 32.29, the buffer clear conditions differ according to the set value of the BFRE bit, even for states in which clear is normally required, using the DCLRM bit eliminates the need for clearing of the buffer by software, enabling DMA transfers without the use of software.

Note that this function only supports the buffer memory read direction setting.

Table 32.29 Correspondence of packet reception and buffer memory clearing by software

Buffer state when packet is received	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Clearing not required	Clearing not required	Clearing not required	Clearing not required
Zero-length packet received	Clearing required	Clearing required	Clearing not required	Clearing not required
Normal short packet received	Clearing not required	Clearing required	Clearing not required	Clearing not required
Transaction count end	Clearing not required	Clearing required	Clearing not required	Clearing not required

32.8.3 BRDY Interrupt Timing Selection

The BFRE bit of the PIPECFG register can be set so that the BRDY interrupt is not generated when a data packet of maximum packet size is received.

When using a DMA transfer, this function enables an interrupt to be generated only when the last data is received. The last data indicates either a short packet reception or the transaction count end. By setting BFRE = 1, the BRDY interrupt will be generated after the received data is read. By reading the DTLN bit of the DnFIFOCTR register, the receive data length of last data packet received just before the BRDY interrupt was generated can be confirmed.

Table 32.30 shows the timing of the BRDY interrupt.

Table 32.30 BRDY interrupt generation timing

Buffer state when packet is received	BFRE = 0	BFRE = 1
Buffer full (normal packet received)	When packet is received	No interrupt generated
Zero-length packet received	When packet is received	When packet is received
Normal short packet received	When packet is received	When read event of data received from buffer memory is completed
Transaction count end	When packet is received	When read event of data received from buffer memory is completed

The BFRE bit function is only valid in reading direction of the buffer memory. When in writing direction, fix the BFRE bit to 0.

32.9 Control Transfer (DCP)

Data transfers of the data stage of control transfers are done using the default control pipe (DCP). The DCP buffer memory is a 64 byte single buffer, and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

32.9.1 Setup Stage

The controller always responds with an ACK when it receives a normal setup packet. The controller operations in the setup stage are as follows.

1. When a new setup packet is received, the controller sets the following bits.
   (a) Sets the VALID bit of the INTSTS0 register to 1.
   (b) Sets the PID bits of the DCPCTR register to NAK.
   (c) Sets the CCPL bit of the DCPCTR register to 0.
2. When a data packet is received following the setup packet, the USB request parameters are stored in the following registers: USBREQ, USBVAL, USBINDX and USBLENG.

Always set VALID = 0 before the response process to a control transfer. While VALID = 1, PID = BUF will not be set and the data stage cannot be completed.

The function of the VALID bit allows the controller to temporarily stop a request in-process when it receives a new USB request during a control transfer, and respond to the newest request.

In addition, the controller automatically judges the direction bit (bmRequestType bit 8) and the request data length (wLength) of the received USB request and determines whether it is a control read transfer, control write transfer or control write no-data transfer, and then handles the stage transition. If the sequence is incorrect, a sequence error for the control transfer stage transition interrupt is generated and is notified to the software. For more information concerning the controller stage management, see Figure 32.10.

32.9.2 Data Stage

Use the DCP for data transfers in response to receiving a USB request. Before accessing the DCP buffer memory, set the access direction in the ISEL bit of the CFIFOSEL register.

The transaction is executed by setting the PID bits of the DCPCTR register to BUF.

Data transfer completion is detected by the BRDY and BEMP interrupts. Use the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

For control write transfers in high-speed operation, a NYET handshake is sent in accordance with the buffer memory state.

32.9.3 Status Stage

When the setting of the PID bits of the DCPCTR register is BUF, set the CCPL bit to 1 to complete the control transfer.

After the above settings, the controller automatically executes the status stage in accordance with the data transfer direction fixed in the setup stage. The detailed process is as follows.

1. Control read transfers:

The controller receives a zero-length packet from the USB host and sends an ACK response.

2. Control write transfers and no-data control transfers:
   The controller sends a zero-length packet and receives an ACK response from the USB host controller.

32.9.4 Control Transfer Automatic Response

The controller automatically sends a response to a normal SET_ADDRESS request. If one of the following errors occurs, a response must be sent by software.

1. bmRequestType ≠ 0x00
2. wIndex ≠ 0x00
3. wLength ≠ 0x00
4. wValue > 0x7F
5. DVSQ = 011b (configured)

All requests other than the SET_ADDRESS request must be responded to by software.

32.10 Bulk Transfer (Pipes 1 to 5)

The user can select the buffer memory usage method (single/double buffer, continuous/non-continuous transfer mode) for bulk transfer. The maximum size that can be set for the buffer memory is 2 KB. The controller manages the buffer memory state and automatically responds to PING packets and NYET handshakes.

32.10.1 NYET Handshake Control

Table 32.31 shows the list of responses to a token received in a bulk or control transfer. When an OUT token is received in a bulk or control transfer and there is only enough open space for one packet in the buffer memory, the controller sends a NYET response. However, when a short packet is received, the controller sends an ACK response instead of a NYET response, even under these conditions.

Table 32.31 List of responses to received tokens

PID bits setting	Buffer memory state	Received token	Response	Notes
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY*1	OUT/PING	ACK	When OUT token is received, data packet is received.
	RCV-BRDY*2	OUT	NYET	Data packet is received.
	RCV-BRDY*2	OUT (Short)	ACK	Data packet is received.
	RCV-BRDY*2	PING	ACK	
	RCV-NRDY*3	OUT/PING	NAK	
	TRN-BRDY*4	IN	DATA0/1	Data packet is transmitted.
	TRN-NRDY*5	IN	NAK	

Note 1. RCV-BRDY: Buffer memory has enough space for 2 packets or more when an OUT or PING token is received.
Note 2. RCV-BRDY: Buffer memory has only enough space for one packet when an OUT token is received.
Note 3. RCV-NRDY: Buffer memory has not enough space when a PING token is received.
Note 4. TRN-BRDY: Buffer memory has data for transmission when an IN token is received.
Note 5. TRN-NRDY: Buffer memory does not have data for transmission when an IN token is received.

32.11 Interrupt Transfer (Pipes 6 to 9)

This controller executes an interrupt transfer in accordance with the period managed by the host controller. The controller ignores (no response) PING packets in interrupt transfers. In addition, the controller does not send a NYET handshake, but responds with ACK, NAK or STALL.

The controller does not support high-bandwidth interrupt transfers.

32.12 Isochronous Transfer (Pipes 1 and 2)

The controller provides the following functions for isochronous transfers.

1. Isochronous transfer error information notification
2. Interval counter (IITV bit setting)
3. Isochronous IN transfer data setup control (IDLY function)
4. Isochronous IN transfer buffer flush function (IFIS bit setting)
5. SOF pulse output function

The controller does not support high-bandwidth isochronous transfers.

32.12.1 Isochronous Transfer Error Detection

The controller manages isochronous transfer errors by software and therefore has the following error information detection functions. Table 32.32 and Table 32.33 describe the procedure in which errors are confirmed and the interrupts that are generated.

1. PID error
   When PID of the received packet is corrupted
2. CRC error and bit stuffing error
   When an error occurs in CRC of the received packet or when the bit stuffing is corrupted
3. Maximum packet size over
   This indicates the data size of the received packet is larger than the value set for the maximum packet size.
4. Overrun and underrun
   When there is no data in the buffer memory when an IN token is received in an IN-direction (send) transfer When there is no empty space in the buffer memory when an OUT token is received in an OUT-direction (receive) transfer
5. Interval error
   The following will generate interval errors.
   (a) When an IN token could not be received in the interval frame of an isochronous IN transfer
   (b) When an OUT token could not be received in the interval frame of an isochronous OUT transfer

Table 32.32 Errors detected in token reception/transmission

Detection priority	Error type	Generated interrupts and states at time of error detection
1	PID error	No interrupt generated (ignored as corrupted packet)
2	CRC error or bit stuffing error	No interrupt generated (ignored as corrupted packet)
3	Overrun or underrun errors	NRDY interrupt is generated and OVRN bit is set. A zero-length packet is sent in response to an IN token. A data packet is not received in response to an OUT token.
4	Interval error	NRDY interrupt generated

Table 32.33 Errors detected in data packet reception

Detection priority	Error type	Generated interrupts and states
1	PID error	No interrupt generated (ignored as corrupted packet)
2	CRC error or bit stuffing error	NRDY interrupt is generated and CRCE bit is set.
3	Maximum packet size over error	BEMP interrupt is generated and PID is set to STALL.

32.12.2 DATA-PID

This controller does not support high-bandwidth transfers. The following occurs in response to a received PID.

1. IN direction:
   (a) DATA0: Transmitted as PID of the data packet
   (b) DATA1: Not transmitted
   (c) DATA2: Not transmitted
   (d) mDATA: Not transmitted
2. OUT direction (in full-speed operation):
   (a) DATA0: Received successfully as PID of the data packet
   (b) DATA1: Received successfully as PID of the data packet
   (c) DATA2: The packet is ignored.
   (d) mDATA: The packet is ignored.
3. OUT direction (in high-speed operation):
   (a) DATA0: Received successfully as PID of the data packet
   (b) DATA1: Received successfully as PID of the data packet
   (c) DATA2: Received successfully as PID of the data packet
   (d) mDATA: Received successfully as PID of the data packet

32.12.3 Interval Counter

32.12.3.1 Overview of Operation

The isochronous transfer interval can be set in the IITV bit of the PIPEPERI register. Table 32.34 shows the functions of the interval counter.

Table 32.34 Interval counter functions

Transfer direction	Function	Detection conditions
IN	Transfer buffer flush function	Cannot successfully receive IN token in interval frame during isochronous IN transfer.
OUT	Notifies that a token not being received	Cannot successfully receive OUT token in interval frame during isochronous OUT transfer.

Since the interval counting is performed upon reception of SOF or by the complemented SOF, the isochronism can be maintained even if the SOF is damaged. Frame intervals are set as 2^{IITV} (μ) frames.

32.12.3.2 Interval Counter Initialization

The controller initializes the interval counter under the following conditions.

1. Hardware reset
   Initializes the IITV bit.
2. Clearing of the buffer memory by the ACLRM bit
   This initializes the counter but not the IITV bit.
3. USB bus reset

After the interval counter is initialized and a packet is successfully transferred, the interval count starts under the following conditions.

1. SOF is received after data is sent in response to an IN token when PID = BUF.
2. SOF is received after data is received in response to an OUT token when PID = BUF.

Note that the interval counter is not initialized in the following conditions.

1. When the PID is set to NAK or STALL
   The interval timer is not stopped at this time. The transaction will be attempted at the next interval.

2. USB bus reset or USB suspend
   The IITV bit is not initialized at this time. When the SOF is received, the count starts from the value before the reception.

32.12.4 Isochronous Transfer Transmission Data Setup

In the isochronous data transmission by this controller, after data is written to the buffer memory, the data packet can be sent out in the next frame after the SOF packet is detected. This function, called the isochronous transfer transmission data setup, allows specification of the frame that started transmission.

When the buffer memory is used for double buffering and writing to both buffers has been completed, only transfer from the first buffer to have received data can proceed. Therefore, even when several IN tokens are received in the same frame, only one packet of data is sent by the buffer memory.

When an IN token is received, if the buffer memory is ready for transmission, the data is transferred and a normal response is returned. However, if the buffer memory is not ready for transmission, a zero-length packet is sent and an underrun error occurs.

Figure 32.14 shows an example of transmission using the isochronous transfer transmission data setup function with this controller when IITV = 0 (for each frame) is set.

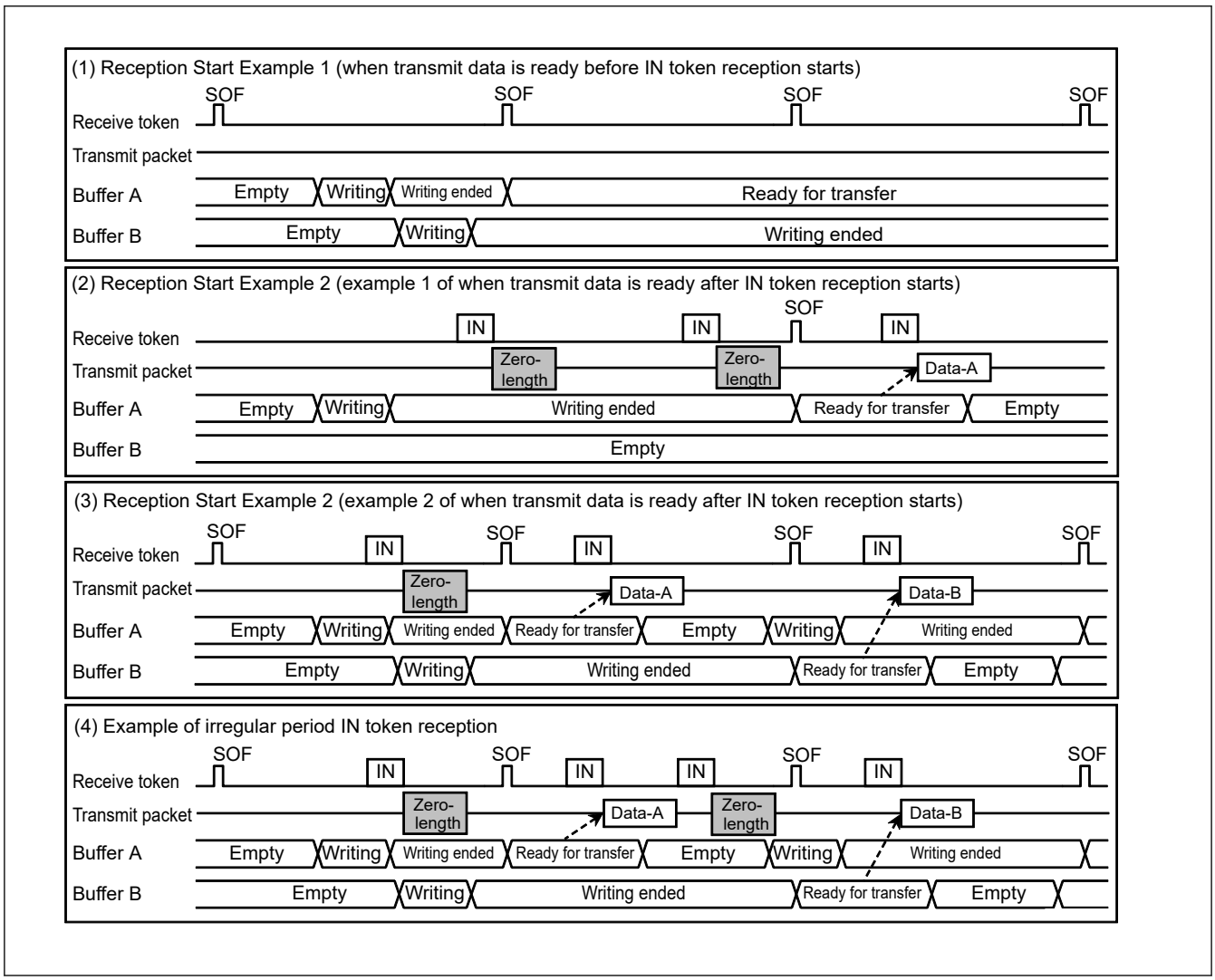


Figure 32.14 Example of data setup function operation

32.12.5 Isochronous Transfer Transmission Buffer Flush

When an SOF packet or a µSOF packet of the next frame is received without receiving an IN token in the interval frame during isochronous data transmission, this controller operates as if a corrupted IN token was received, and clears the buffer which is ready for transmission, making that buffer ready for writing.

If a double buffer is being used and writing to both buffers has been completed, data are considered to have been sent from the buffer memory that was cleared in the same interval frame, and transmission is enabled for the buffer memory that is not discarded with SOF or µSOF packets reception.

The timing at which the buffer flush function is activated varies depending on the setting of the IITV bits.

1. When IITV = 0
   The buffer flush operation proceeds from the first frame after the pipe becomes valid.
2. When IITV ≠ 0
   The buffer flush operation proceeds after the first successful transaction.

Figure 32.15 shows an operation example of the buffer flush function of this controller. When an unanticipated token is received prior to the interval frame, this controller sends the written data or a zero-length packet as an underrun error according to the data setup state.

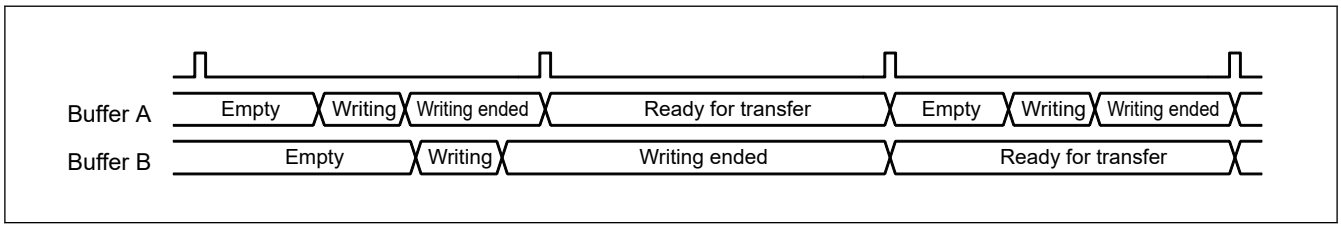


Figure 32.15 Buffer flush function operation example

Figure 32.16 shows an example of an interval error generated in the controller. There are five types of interval errors, as listed below. Timing 1 in the figure shows when the interval error occurs and how the buffer flush function operates.

When an interval error occurs during an IN transfer, the buffer flush function goes into operation; during an OUT transfer, the NRDY interrupt is generated.

Use the OVRN bit to determine whether an error is an NRDY interrupt, such as a receive packet error, or an overrun error.

Responses to the tokens in the shaded boxes are executed in accordance to the buffer memory state.

1. IN direction:
   (a) If the buffer is ready for transfer, data is transferred as a normal response.
   (b) If the buffer is not ready for transfer, a zero-length packet is sent and an underrun error occurs.
2. OUT direction:
   (a) If the buffer is ready for reception, data is received as a normal response.
   (b) If the buffer is not ready for reception, data is discarded and an overrun error occurs.

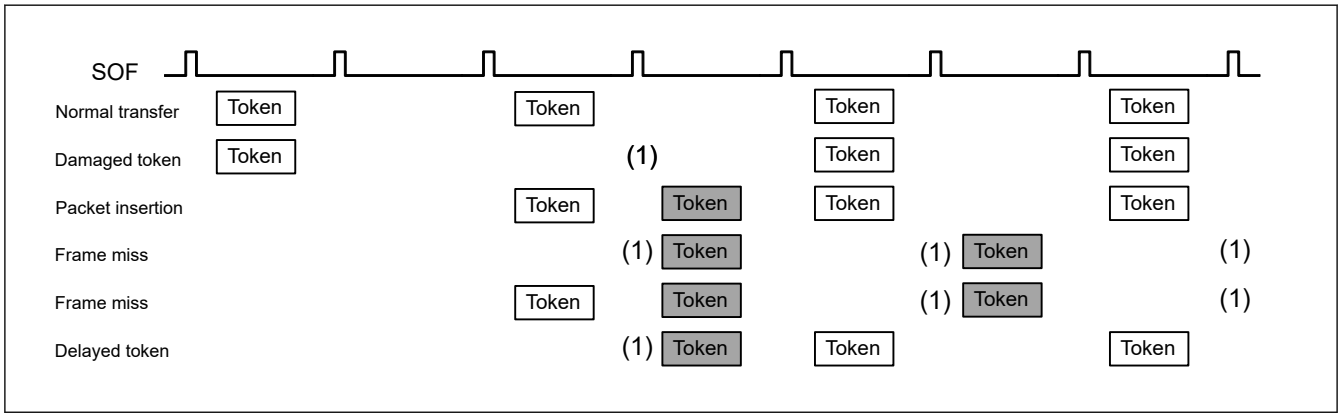


Figure 32.16 Example of interval error when IITV = 1

32.13 SOF Interpolation

If an SOF packet could not be received at intervals of 1 ms (in full-speed operation) or 125 µs (in high-speed operation) because of corruption or missing, this controller interpolates the SOF. The SOF interpolation operation begins when both the USBE and SUSPM bits have been set to 1 and an SOF packet is received. The interpolation is initialized under the following conditions.

1. Hardware reset
2. USB bus reset
3. Suspended state detected

The SOF interpolation operates according to the following specifications.

1. The frame interval (125 µs or 1 ms) is based on the results of the reset handshake protocol.
2. The interpolation does not operate until the SOF packet is received.
3. After a first SOF packet is received, interpolation of another SOF packet proceeds after a 125-µs or 1-ms interval counted in cycles of the internal clock running at 48 MHz clock has elapsed.
4. Interpolation is performed in the previous reception intervals after the 2nd and subsequent SOF packets are received.
5. Interpolation is not performed in the suspended state or during reception of a USB bus reset.
   When the controller goes to the suspended state in high-speed operation, interpolation continues for 3 ms from the last packet.

The SOF interpolation works for the following.

1. Updating of the frame number or micro-frame number
2. SOFR interrupt and µSOF lock
3. SOF pulse output
4. Isochronous transfer interval count

When an SOF packet is lost during full-speed operation, the FRNM bit of the FRMNUM register is not updated.

When a µSOF packet is lost during high-speed operation, the UFRNM bit of the UFRMNUM register is updated.

However, when a µSOF packet when µFRNM = 000b is lost, the FRNM bit is not updated. At this time, even if µSOF packets when µFRNM ≠ 000b are received successfully, the FRNM bit is not updated.

32.14 Link Power Management Processing

The Link Power Management standard redefines the existing suspended state as L2 state, and further defines L1 state that enables transitions and recovery with lower latency than the L2 (suspended) state.

The following table shows comparison between L2 (suspended) and L1.

Table 32.35 Comparison between L2 (suspended) and L1

Item	L1	L2 (suspended)
Transition	LPM transaction	3 ms idle
Recovery due to host	(Host) The host can specify the minimum drive period (75 µs to 1.175 ms). (Device) 10 µs K drive	(Host) Min 20 ms K drive (Device) 10 ms K drive
Recovery due to device	(Device) 50 µs K drive (Host) 60 µs to 990 µs K drive (Device) 10 µs K drive	(Device) 1 ms to 15 ms K drive (Host) Min 20 ms K drive (Device) 10 ms K drive
Signaling	Low and full speed idle	Low and full speed idle

The following describes the transition (to L1 state) processing and recovery processing.

32.14.1 Descriptor

When the GetDescriptor command is received, the own descriptor must be sent back.

It is necessary to change the contents of the descriptor to be returned depending on whether to respond to the transition to L1 and recovery processing because of the LPM transaction, as shown in the table below.

Table 32.36 Relationship between response to LPM and descriptor

Response to LPM	bcdUSB	Provision of USB2.0Ex.Desc	LPM of USB2.0Ex.Desc	Response when LPM is received	Remarks
Does not respond	0x0200	Not provided	—	Not Respond	Standard operation when the descriptor does not respond to LPM
	0x0201	Provided	LPM = 0	STALL	Clear declaration that the descriptor does not respond to LPM. In this case, no response must be avoided and STALL must be sent in response.
Responds	0x0201	Provided	LPM = 1	ACK or NYET	Standard operation when the descriptor responds to LPM

Whether to respond to the transition to L1 and recovery processing is declared with the LPM bit in the USB2.0 extension descriptor. To provide the USB2.0 extension descriptor, a value of 0x0201 or larger must be set for the bcdUSB field of the device descriptor.

If the descriptor does not respond to LPM, the USB2.0 extension descriptor is not provided and the bcdUSB field is set to 0x0200. In this case, even if LPM token is received, ignore it.

If the descriptor does not respond to LPM, it is also possible to set the bcdUSB field to 0x0201 and clear the LPM bit in the USB2.0 extension descriptor to 0 (does not respond). In this case, however, LPM cannot be ignored and STALL must be sent in response. When the descriptor responds to LPM, set the bcdUSB field to 0x0201 and set the LPM bit in the USB2.0 extension descriptor to 1 (responds). This enables acknowledgment of NYET or ACK response to the LPM token.

32.14.2 Basic Processing

The following processing is required.

1. Send no response, ACK, NYET, or STALL response to the LPM token received from the host in accordance with the own state.
2. If re-transmission of LPM token is not detected for 8 µs after an ACK response was sent, a transition to the L1 state is made.
3. Detect the K drive of the host and perform the processing for recovery to the idle state.
4. Perform the processing for recovery to the idle state by remote wakeup.

For step 1, hardware receives the LPM token, it sends the response. Hardware performs processing in step 2 in addition to re-transmission control and transition to the L1 state.

A transition to the L1 state can be checked by a DVST interrupt.

For step 3, a RESM interrupt is generated by detecting the K drive of the host in the L1 state.

For step 4, setting the WKUP bit to 1 by software gives an instruction to start remote wakeup to the hardware. In the specifications, software clears this bit at the time of recovery from the L2 state, but hardware clears this bit at the time of recovery from the L1 state.

32.14.3 HIRD Value Negotiation

The HIRD value included in the LPM token indicates the host K period for recovery from the L1 state. It is supposed that the desired HIRD value may vary depending on applicable applications. It is supposed that a small HIRD value is desired for applications focusing on improvement in transfer efficiency, but a large HIRD value is desired for applications focusing on low power consumption.

According to the L1NEGOMD and HIRDTHR bits in the L1CTRL register, an ACK response is returned if the received HIRD value is within the desired range. Otherwise, a NYET response is returned to request the host to correct the HIRD value.

Note: Using this HIRD value negotiation function requires that the negotiation processing be also supported on the host side.

32.15 DMA Mode

32.15.1 Register Mode/Link Mode

With the DMS bit in the CHCFG_n register, DMA mode can be switched between register mode and link mode.

Table 32.37 DMA mode setting

DMS(CHCFG)	Mode	Description
0	Register Mode	Performs a DMA transfer with the values set for the next register set.
1	Link Mode	Accesses the descriptor area and performs a DMA transfer with the value set for the descriptor. Repeats descriptor read and DMA transfer unless you set the descriptor or use the control register to stop them.

32.15.1.1 Register Mode

In register mode, you can perform a DMA transfer using the value set in the internal register. You can set two sets (Next0 register set and Next1 register set) of transfer source addresses, the transfer destination addresses, and the numbers of transfer bytes.

You can select the next register set to perform a transfer, or use two next register sets for continuous transfer.

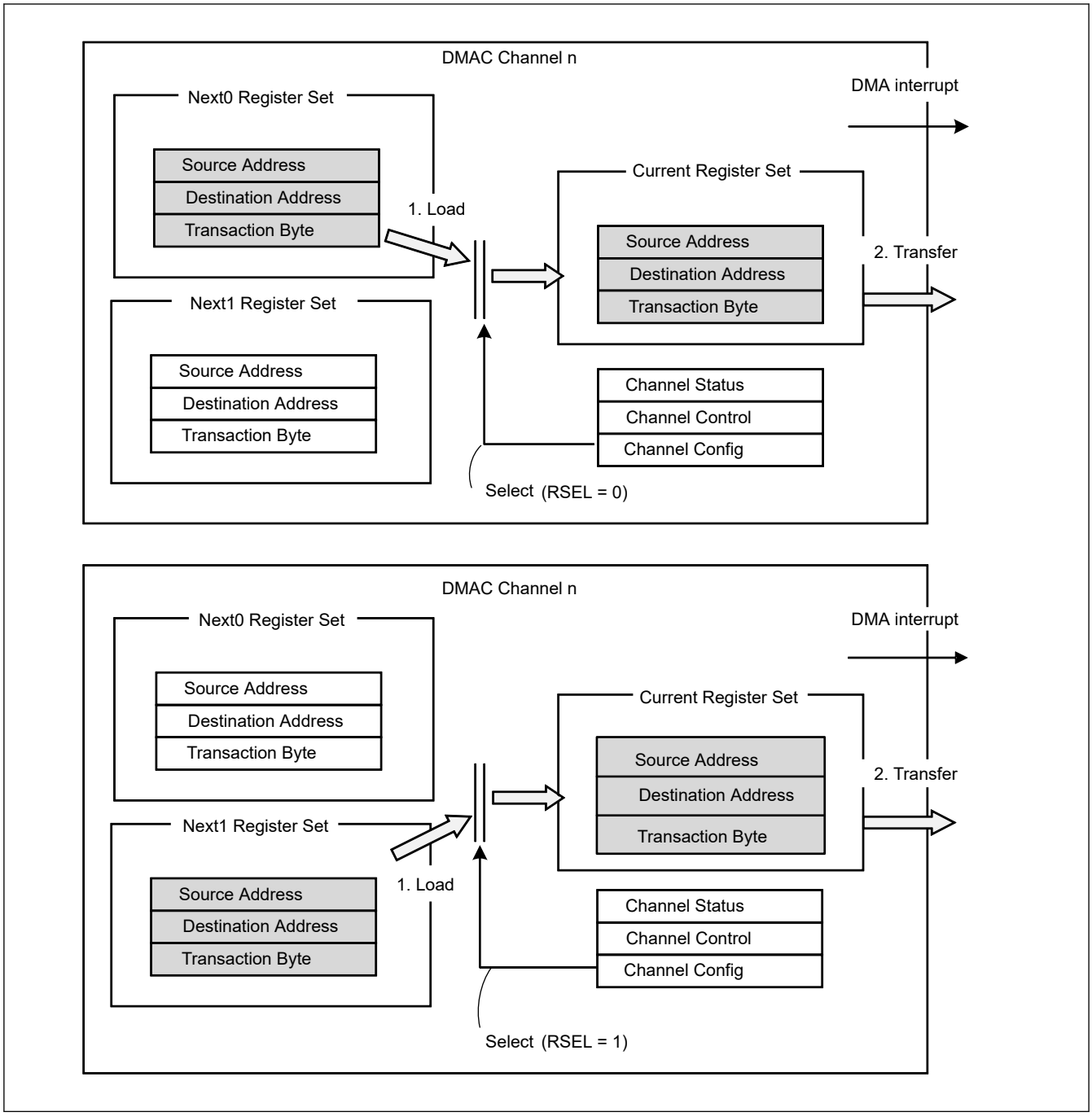


Figure 32.17 Overview of register normal mode

The above figure illustrates operation when Next0 register set is executed (above) and operation when Next1 register set is executed (below).

(1) Operation Flow in Register Mode

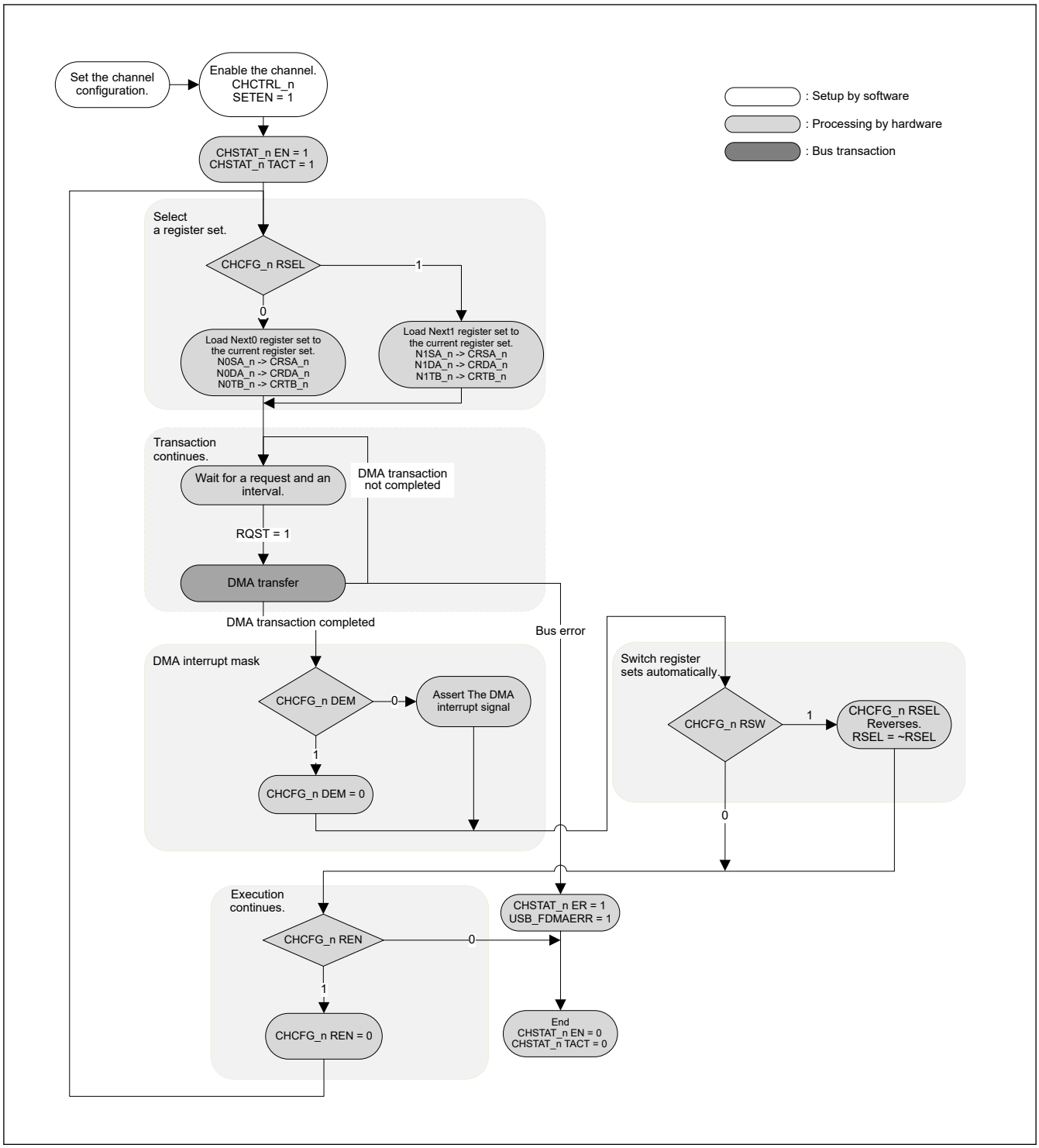


Figure 32.18 Register mode flow

<Description of register mode flow>

1. Channel setting (set channel configuration)
   Set Next0 or Next1 register set (the transfer destination address, the transfer source address, and the total number of transfer bytes). Also set the USB-controlled FIFO channels and the amount of transfer used for channel register set. (See section 32.16. DMA Transfer.)
2. Selecting register set

When 1 is written to the SETEN bit in the CHCTRL_n register, the EN bit and the TACT bit in the CHSTAT_n register are set to 1, and the set value of the next register set selected by the RSEL bit in the CHCFG_n register is loaded to the current register set.

3. DMA transaction (processing transaction)
   A DMA transaction is performed according to the set value. For details on transfer, see section 32.16. DMA Transfer.
4. Making USB_FDMAn (USB_FDMAn mask)
   The USB_FDMAn is masked according to the set DEM bit value in the CHCFG_n register. When DEM = 1, the USB_FDMAn is masked. Immediately after that, the DEM bit is automatically cleared to 0.
5. Switching register sets automatically
   Switching to the other next register set is determined according to the set RSW bit value in the CHCFG_n register.
6. Continuous execution
   Whether to perform DMA transfers continuously is determined according to the set REN bit value in the CHCFG_n register. When REN = 0, the EN bit and the TACT bit in the CHSTAT_n register are cleared to 0, and the DMAC stops operation. When REN = 1, DMA transfers are continued. Immediately after that, the REN bit is automatically cleared to 0.

(2) Setting Register Mode

Register mode settings

Select the register set to execute.

Table 32.38 Register mode settings

DMS (CHCFG_n)	RSEL (CHCFG_n)	Description
0	0	Executes Next0 register set.
	1	Executes Next1 register set.

USB_FDMAn mask settings

The USB_FDMAn can be masked.

Table 32.39 USB_FDMAn mask settings

DEM (CHCFG_n)	Description
0	When a DMA transaction is completed, the USB_FDMAn is asserted.
1	Even if a DMA transaction is completed, the USB_FDMAn is not asserted. Upon completion of a DMA transaction, the DEM bit is cleared to 0 by hardware.

Settings for executing register sets automatically

After a DMA transaction is completed, another DMA transaction can be performed successively.

Table 32.40 Settings for automatic execution of register sets

REN (CHCFG_n)	Description	Remarks
0	When a DMA transaction of the register set that is set in the RSEL bit is completed, the EN bit is cleared to 0 and the DMA operation ends.	Set this value if you want to perform a DMA transaction once.
1	After a DMA transaction is completed, a DMA transfer is performed for the register set that was selected to be performed successively. When the successive transfer is performed, the REN bit is cleared to 0.	Set this value if you want to perform DMA transactions for register sets successively.

Settings for switching register sets automatically

After a DMA transaction is completed, the register set to be used for the next DMA transaction can be switched.

Table 32.41 Settings for automatic switching of register sets

RSW (CHCFG_n)	Description	Remarks
0	After a DMA transaction is completed, register sets are not switched.	Set this value when you want to use only one register set.
1	When a DMA transaction is completed while REN = 1, the RSEL bit is reversed automatically and the other register set is selected.	Set this value when you want to switch register sets.

(3) Register Mode Setting Examples

Using the Next0 register set only

Table 32.42 Register mode setting example 1

DMS (CHCFG_n)	RSEL (CHCFG_n)	DEM (CHCFG_n)	RSW (CHCFG_n)	REN (CHCFG_n)
0 (Register mode)	0 (Next0)	0 (Do not mask)	0 (Do not switch)	0 (Continuous execution not in progress)

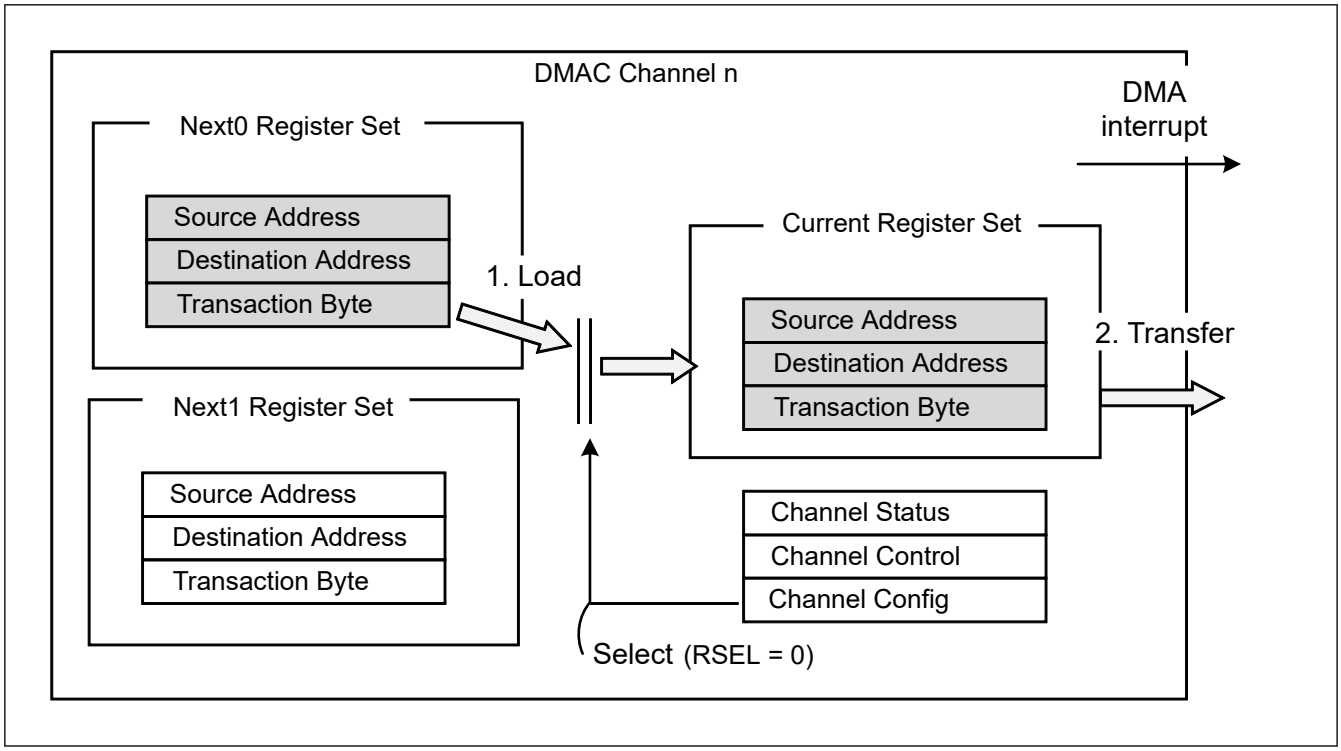


Figure 32.19 Register mode setting example 1

1. Writing 1 to the SETEN bit in the CHCTRL_n register sets the EN bit in the CHSTAT_n register to 1, and loads Next0 register set to the current register set.
2. According to the values set for the current register set and the channel register set, a DMA transaction is performed.
3. Because the DEM bit in the CHCFG_n register is 0, USB_FDMAn is asserted after a DMA transaction is completed.
4. Because the REN bit in the CHCFG_n register is 0, the EN bit in the CHSTAT_n register is cleared to 0, and the operation ends.

Using two register sets successively

Table 32.43 Register mode setting example 2

DMS (CHCFG_n)	RSEL (CHCFG_n)	DEM (CHCFG_n)	RSW (CHCFG_n)	REN (CHCFG_n)
0 (Register mode)	0 (Next0)	1 (Mask)	1 (Switch)	1 (Continuous execution in progress)

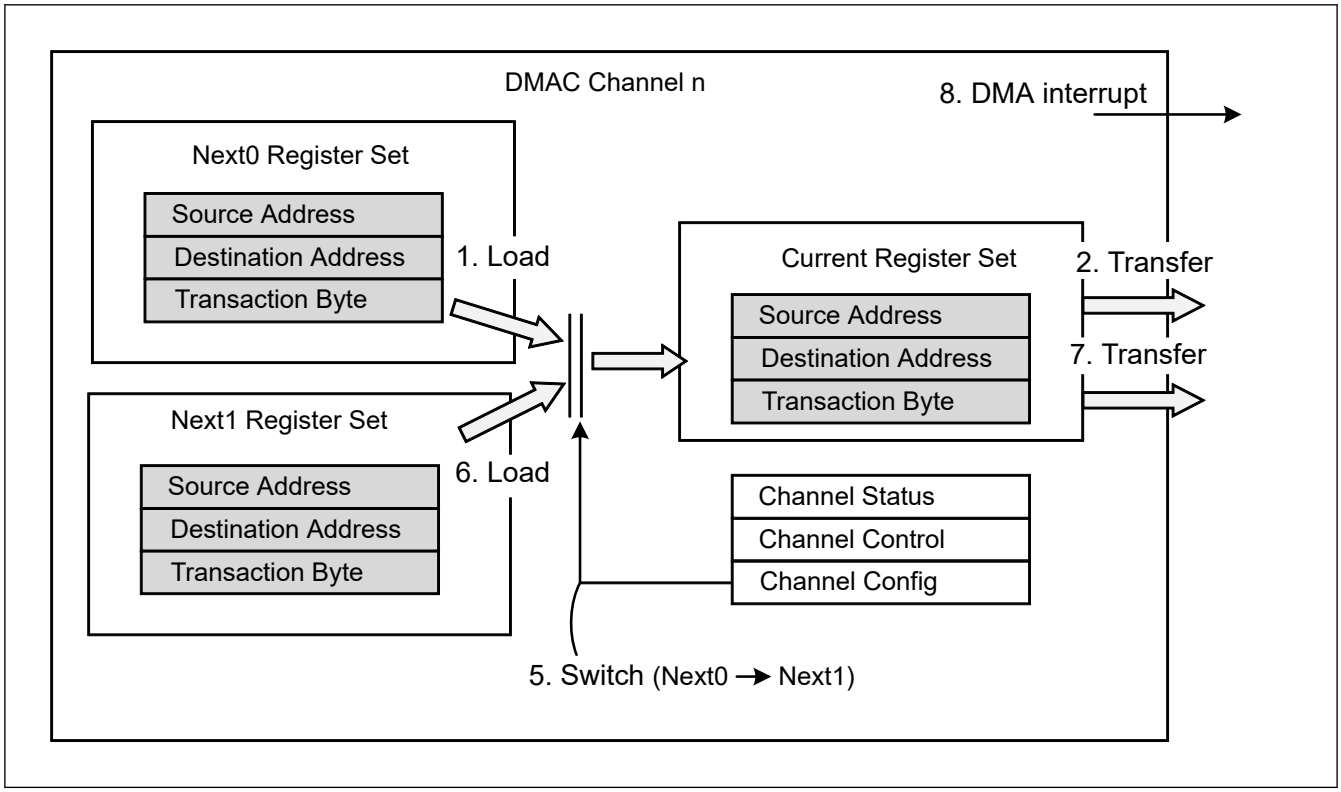


Figure 32.20 Register mode setting example 2

1. Writing 1 to the SETEN bit in the CHCTRL_n register sets the EN bit in the CHSTAT_n register to 1, and loads Next0 register set to the current register set.
2. According to the values of the current register set and the channel register set, a DMA transaction is performed.
3. Because the DEM bit in the CHCFG_n register is 1, USB_FDMAn is not asserted after a DMA transaction is completed, and the DEM bit is automatically cleared to 0.
4. Because the REN bit in the CHCFG_n register is 1, DMA transactions are performed successively, and the REN bit is automatically cleared to 0.
5. Because the RSW bit in the CHCFG_n register is 1, the register set to be executed next is switched (RSEL = 0 → 1).
6. Loads Next1 register set to the current register set.
7. According to the values of the current register set and the channel register set, DMA transactions are performed.
8. Because the DEM bit in the CHCFG_n register is 0, USB_FDMAn is asserted after a DMA transaction is completed.
9. Because the REN bit in the CHCFG_n register is 0, the EN bit in the CHSTAT_n register is cleared to 0, and the operation ends.

32.15.1.2 Link Mode

In link mode, a DMA transaction is performed by reading a descriptor in an external memory area as a setting value. The DMAC incorporates the next link address (NXLA_n) register and the current link address (CRLA_n) register for each channel. Each of them is used to set the address of the descriptor to be executed next and to display the descriptor address of the current DMA transaction respectively.

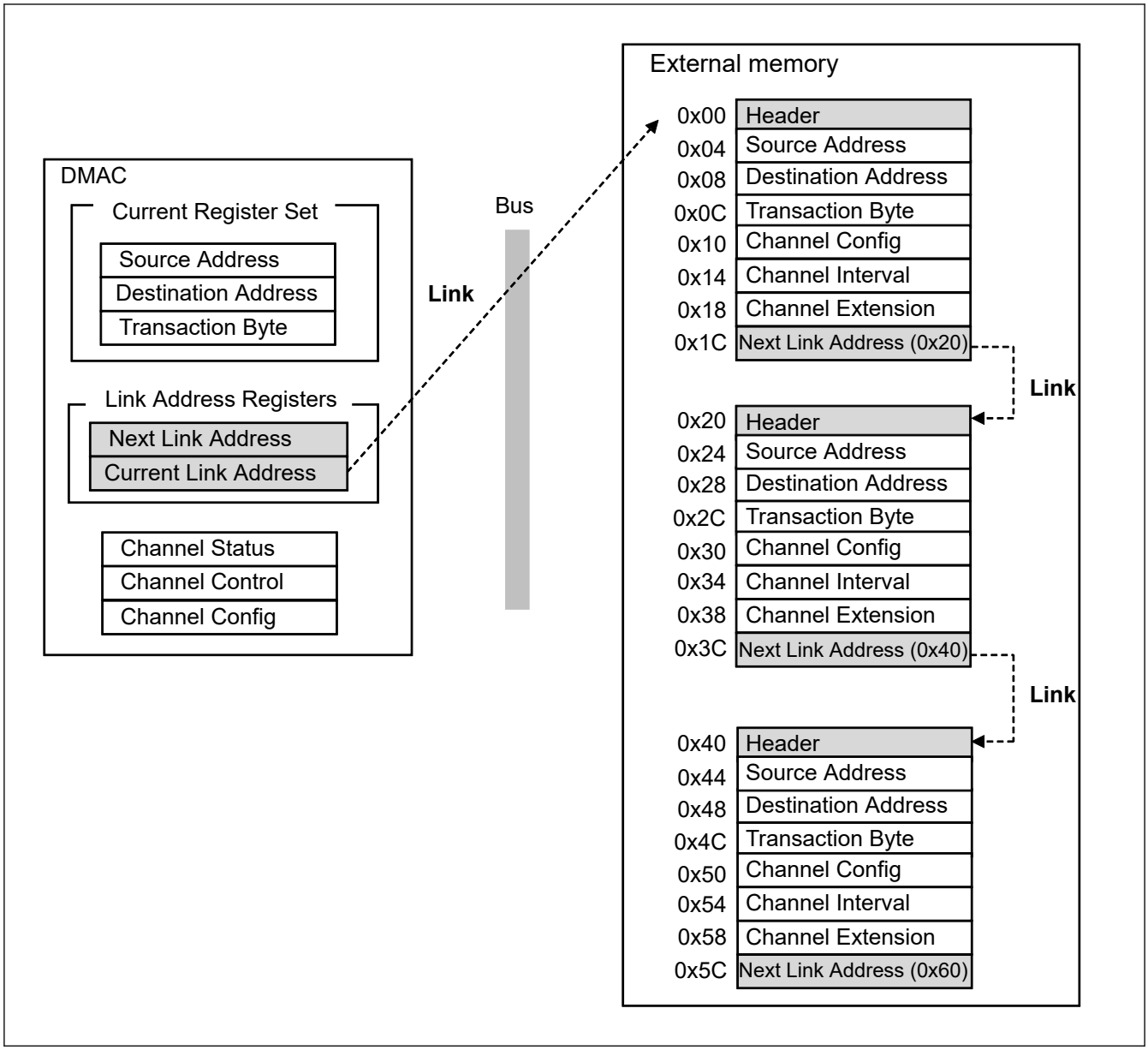


Figure 32.21 Overview of link mode

(1) Link Mode Operation Flow

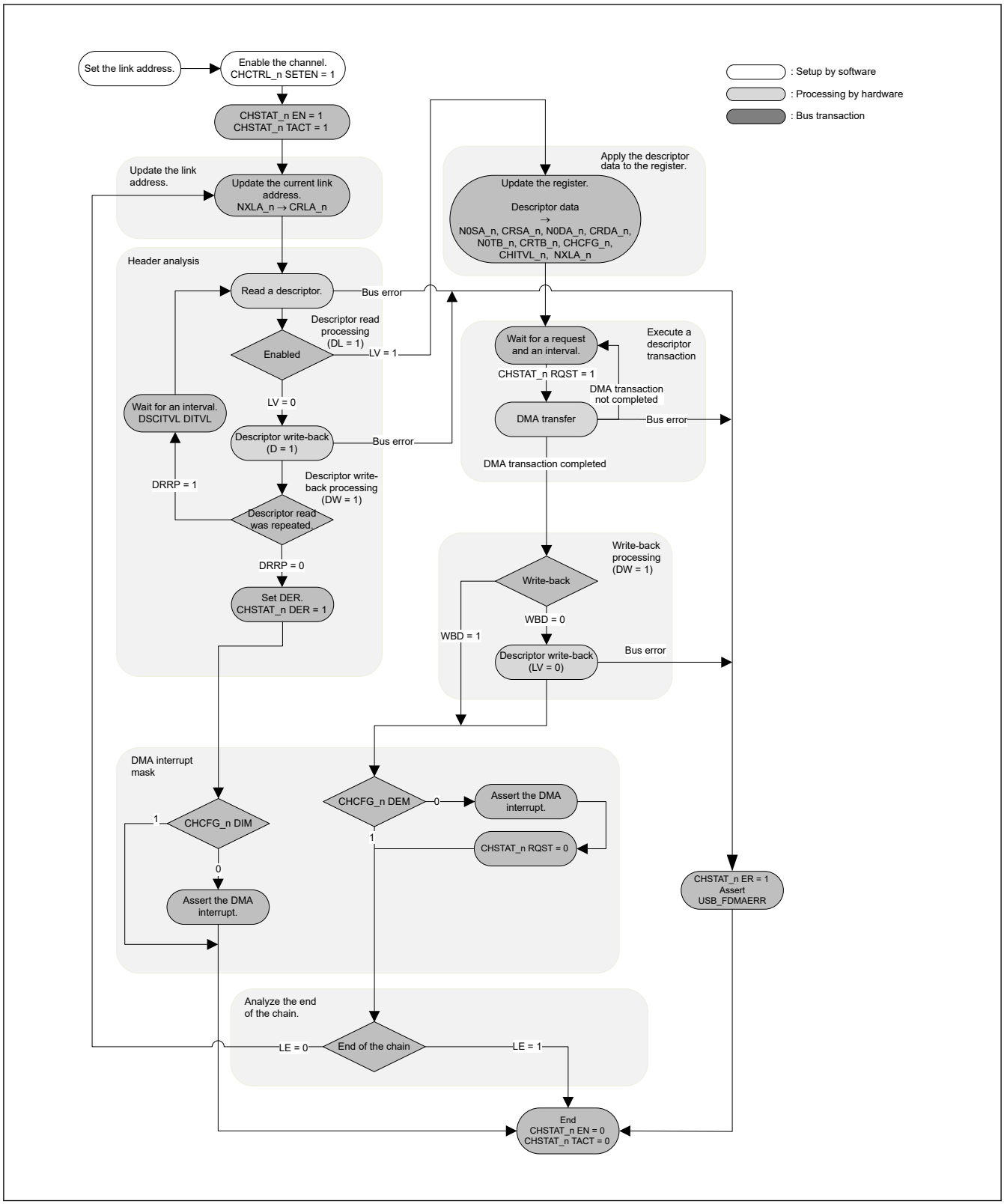


Figure 32.22 Link mode operation flow

Description of Link Mode Operation Flow

1. Channel setting
   Set the start address of the link destination in the NXLA_n register.

2. Updating the link address
   Writing 1 to the SETEN bit in the CHCTRL_n register sets the EN bit and the TACT bit in the CHSTAT_n register to 1, and loads the link address set for the NXLA_n register to the CRLA_n register.
3. Descriptor read and header judgment
   The DMAC starts reading a descriptor and checks the header contents. When LV = 0, 1 is written back to the D bit in the header. After that, when the DRRP bit in the CHCFG_n register = 1, the same descriptor is read again after the number of cycles set in the DSCITVL register elapsed. When DRRP = 0, DER in the CHSTAT_n register is set to 1. This indicates the end state (EN = 0 and TACT = 0 in the CHSTAT_n register). At that time, if the DIM bit in the CHCFG_n register is 0, USB_FDMAn is asserted.
4. Descriptor setting
   When LV = 1, the read descriptor data is loaded to the current register set and the channel register set. In addition, the next link destination is loaded to the NXLA_n register.
5. DMA transaction
   According to the set values, a DMA transaction is performed. For details on transfers, see section 32.16. DMA Transfer.
6. Writing back of header
   When WBD in the header = 0, the DMAC writes LV = 0 back to the header area.
7. USB_FDMAn mask
   When the DEM bit in the CHCFG_n register is 0, USB_FDMAn is asserted.
8. Link end judgment
   When LE in the header = 1, the EN bit and the TACT bit in the CHSTAT_n register are cleared to 0, and the DMAC stops operation. When LE = 0, the current register set is updated and the next descriptor read is started.

(2) Register Settings

Link mode settings

When using link mode, set the DMS bit in the CHCFG_n register to 1.

Table 32.44 Link mode settings

DMS (CHCFG_n)	Description
1	Operates in link mode.This bit cannot be modified by using a descriptor.

LINK address settings

As registers that indicate the link destination, the next link address (NXLA_n) register and the current link address (CRLA_n) register are available.

To start link mode, set the link destination in the NXLA_n register.

The NXLA_n register is updated to the next link after a descriptor is read. In addition, the CRLA_n register indicates the link address of the currently executed descriptor.

Table 32.45 Link address register set

Register	Description
Next link address register (NXLA_n)	Sets and displays the next link destination. Before starting link mode, set the address of the link destination in this register.
Current link address register (CRLA_n)	Displays the currently executed link destination. This register is read-only.

(3) Descriptor Settings

The DMAC supports multiple descriptor formats. To switch formats, use the DSCFM field of bits [31:28] in the first word (header) of a descriptor.

The following table shows the relationship between the DSCFM bit value and descriptor formats.

Table 32.46 Descriptor format

DSCFM	Descriptor size	Next link address	Channel extension	Channel interval	Channel config	Transaction size	Destination address	Source address	Header
0x3	Four words	Y	— (Reload)	— (Reload)	— (Reload)	— (Header)	Y	Y	Y (with STS)
0x1	Eight words	Y	Y	Y	Y	Y	Y	Y	Y (without STS)
Other than above	If DSCFM is set to a value other than 1 and 3, operation is not guaranteed. Do not set a value other than 0x1 and 0x3 for DSCFM.								

Table 32.47 Description of availability

Field	Availability	Description	Remarks
Header	Y(with STS)	Indicates that the STS field of [15:0] in the header is enabled. The value set in the STS field is used as the total number of transfer bytes (transaction size).	—
	Y(without STS)	The STS field of [15:0] in the header is disabled. The transaction size of the descriptor is used as the total number of transfer bytes.	—
Source address	Y	Specifies the source address.	—
Destination address	Y	Specifies the destination address.	—
Transaction size	Y	Specifies the transaction size.	—
	— (Header)	Omits the transaction size. The value set in the STS field in the header is used as the total number of transfer bytes (transaction size).	Because the STS field is 16 bits, up to 65,535 bytes can be set.
Channel config Channel interval Channel extension	Y	Specifies the channel config, channel interval, and channel extension.	—
	— (Reload)	Omits the channel config, channel interval, and channel extension. Inherits the previous setting values (of the CHCFG_n, CHITVL_n, and CHEXT_n registers at that time).	—
Next link address	Y	Specifies the next descriptor address (next link address) to be read after a DMA transfer of this descriptor.	—

The DMAC interprets data obtained through descriptor read in order. If a value less than eight words is specified in the DSCFM field, place descriptor data marked with Y in Table 32.46 in memory.

Table 32.48 Descriptor placement example

	Address (link address + n)							
DSCFM	+ 0x1C	+ 0x18	+ 0x14	+ 0x10	+ 0x0C	+ 0x08	+ 0x04	+ 0x00
0x3	—	—	—	—	Next link address	Destination address	Source address	Header
0x1	Next link address	Extension	Interval	Config	Transaction byte	Destination address	Source address	Header

Header

The header indicates descriptor states as shown below.

This area is read by the DMAC before starting a DMA transfer in link mode. Furthermore, the transfer status is written back by the DMAC after the DMA transfer.

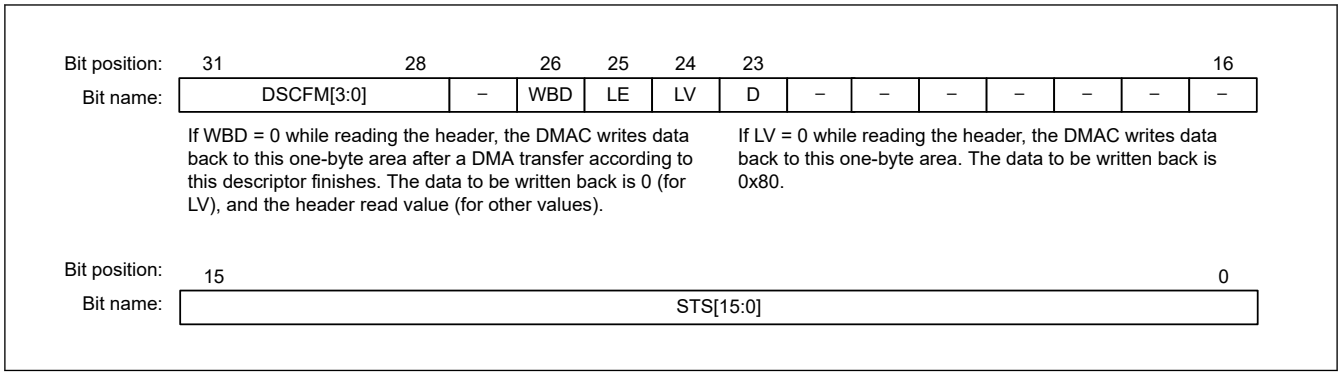


Figure 32.23 Header area

Table 32.49 Header area

Bit position	Bit name	Description
15:0	STS	Short Transaction Size When DSCFM = 0x3, these bits set the transaction size in bytes. Up to 65535 bytes can be set. When DSCFM = 0x3, do not set the STS bits to 0. If 0 is set, the operation is not guaranteed.
22:16	—	A reserved area. Set 0.
23	D	Descriptor Error Indicates an access error of a descriptor. When LV = 0 while a descriptor is being read, the DMAC writes 1 back to this bit. 0: No descriptor error 1: LV = 0 during descriptor read
24	LV	Link Valid Indicates that this descriptor is enabled. When WBD = 0, the DMAC writes 0 after the DMA transaction written in the descriptor is completed. Set 1 when setting header. 0: The descriptor is disabled. 1: The descriptor is enabled.
25	LE	Link End Indicates that the link ends with the DMA transaction of this descriptor. To indicate the end of the link, set this bit to 1. 0: The link continues. 1: The link ends.
26	WBD	Write Back Disable Masks write-back execution of the LV bit. When this bit is 1, the DMAC does not perform writeback operation. 0: The LV bit is written back to 0. 1: The LV bit is not written back.
27	—	A reserved area. Set 0.
31:28	DSCFM	Descriptor Format Specifies the descriptor format (descriptor length and combination). For details, see Table 32.46.

If descriptors are added sequentially while the DMAC is operating, an access of the CPU to set the LV bit to 1 and an access of the DMAC to write 1 back to the D bit may conflict. Because of this, the data that was written first is overwritten by the data that was written later.

To avoid this problem, the byte lane of the D bit and the byte lane of the LV bit are placed differently to separate write areas. The DMAC uses the byte write method to write back the D bit. Therefore, the CPU must set LV = 1 by using the byte write method.

Setting descriptors other than header

Data in descriptors except header has the same specifications as an on-chip register. For details on the on-chip register specifications, see section 32.3. Register Descriptions.

Setting AHB during descriptor access

The MHPROT pin output during the descriptor access can be set in the LWPR and LDPR fields in the DCTRL register. Set it according to the access destination for which descriptor is placed.

Descriptor areas and DMA transfer areas

Figure 32.24 provides an overview of descriptor areas and DMA transfer areas which the DMAC accesses.

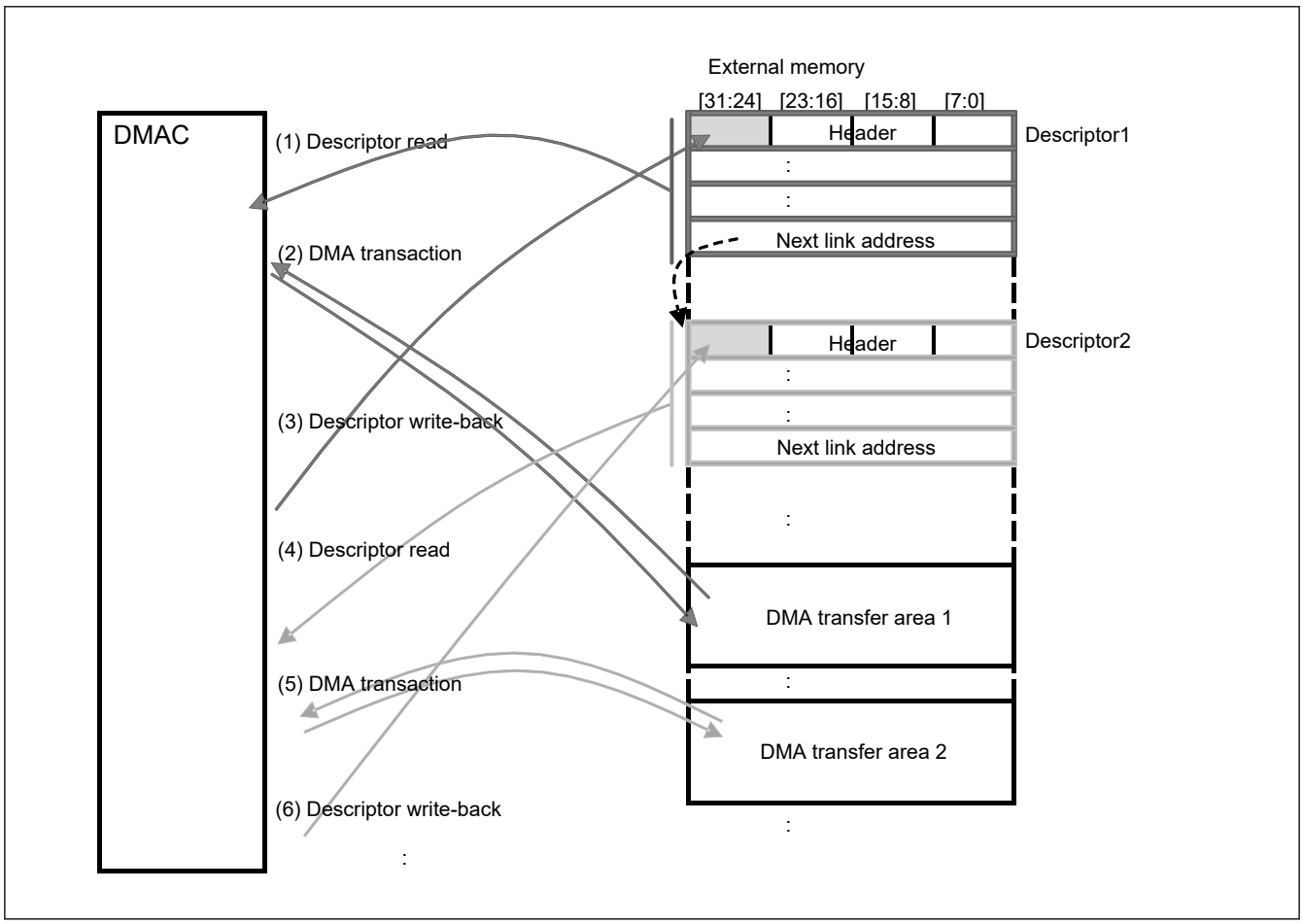


Figure 32.24 Overview of descriptor areas and DMA transfer areas

1. Descriptor read
   Loads the value set for the on-chip NXLA_n register to the CRLA_n register, and then reads the descriptor from the external memory space (descriptor 1) indicated by the CRLA_n register.
2. DMA transaction
   When the LV bit in the header of the descriptor is 1, performs a DMA transfer according to the descriptor information.
3. Descriptor write-back
   If the WBD bit in the header is 0 after a DMA transfer for the set number of bytes, data (LV bit = 0 and the data that was read in step 1 in other fields) is written back to header[31:24] of descriptor 1 with a byte size.
4. Descriptor read
   If the LE bit in the header of the descriptor that was read in step 1 is 0, reads the next descriptor from the address (descriptor 2) indicated by the next link address in the descriptor.
5. DMA transaction
   When the LV bit in the header of the descriptor is 1, performs a DMA transfer according to the descriptor information.
6. Descriptor write-back
   If the WBD bit in the header is 0 after a DMA transfer for the set number of bytes, data (LV bit = 0 and the data that was read in step 4 in other fields) is written back to header[31:24] of descriptor 2 with a byte size.

Hereafter, repeats steps 4 through 6.

When LE in the header is 1 and WBD = 0, a DMA transfer is performed with the descriptor settings, and 0 is written back to the LV bit in the header, and then operation ends.

When LE in the header is 1 and WBD = 1, a DMA transfer is performed with the descriptor settings, and then operation ends (without write-back).

When the LV bit in the header is 0, 1 is written back to the D bit in the header. After that, if the DRRP bit in the CHCFG_n register is 1, the descriptor is read again after the interval specified in the DITVL field in the DSCITVL_n register. When DRRP = 0, the operation stops.

Notes on descriptors

- In link mode, settings can be changed by reading a descriptor, but the timing for changing settings and for a hardware request cannot be synchronized. For this reason, when using a hardware request, set the AM, LVL, HIEN, LOEN, and SEL bits in the CHCFG_n register to 1 before setting the SETEN bit in the CHCTRL_n register. Besides, do not change these bits in a descriptor.
- In a descriptor, the settings of the DMS field in the CHCFG_n register cannot be changed (always link mode). In addition, settings of the REN, RSW, and RSEL fields in the CHCFG_n register can be changed in the descriptor, but that does not affect the operation.
- The DMAC references the DSCFM field and the LV bit in the header to determine if the descriptor is enabled or disabled. Therefore, initialize (DSCFM = 0x1 or 0x3 and LV = 1) the memory area equivalent to the DSCFM field and the LV bit of the descriptor before the DMAC accesses the memory area.
- If you want to set the next descriptor in the memory while the DMAC is operating, write 1 to the LV bit after setting the descriptors after header (source address, destination address, ... next link address). This is to avoid DMA transfer using descriptor values (source address, destination address, and so on) before the setting if descriptor settings by the CPU and descriptor read of the DMAC conflict and descriptor read by the DMAC interrupts descriptor settings by the CPU.
- If you want to keep the information written back to the D bit in the header, perform a byte access to write 1 to the LV bit in the header.

(4) LINK Configuration Example

In link mode, descriptors can be configured as shown below.

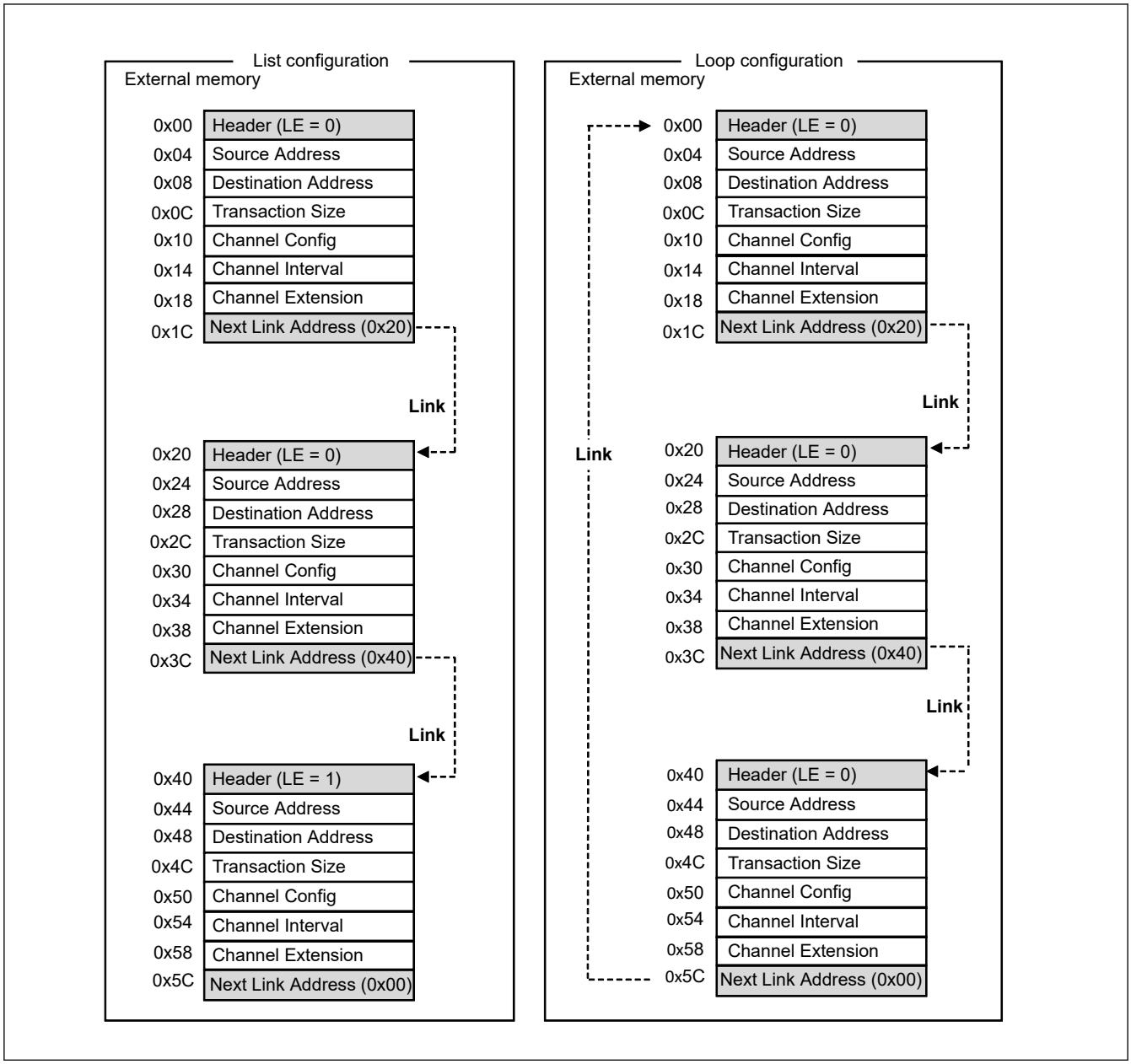


Figure 32.25 Link mode configuration example

List configuration

Writing 1 to the LE bit in the header of the last descriptor terminates the link.

Loop configuration

Setting the link destination of the last descriptor to the address of the previous descriptor configures descriptors in a loop. To terminate the loop, modify the LE bit in the header to 1 before the DMAC reads a descriptor, or follow the procedure to suspend a transfer.

32.15.1.3 Write-Only Mode

Setting the WONLY bit in the CHCFG_n register to 1 enables write-only mode.

Table 32.50 Write-only mode settings (1 of 2)

WONLY(CHCFG)	Mode	Description
0	Normal mode	Performs DMA transfers with the values set for the next register set.

Table 32.50 Write-only mode settings (2 of 2)

WONLY(CHCFG)	Mode	Description
1	Write-only mode	Performs DMA write transfers only without performing DMA read transfers.

In write-only mode, read operation for DMA transfers is not performed (descriptor read is performed in the same way as normal mode). In register mode, use the values set for the NxSA_n register (x = 0 when RSEL = 0, and x = 1 when RSEL = 1) as write data. In link mode, use the value in the SA field of a descriptor as write data.

Use this mode for initializing a memory area.

32.16 DMA Transfer

This section describes basic operation of DMA transfer.

32.16.1 Transfer Mode

Only single transfer mode is supported.

When a DMA transfer request from the USB control is accepted, a DMA transfer is performed on the side (transfer source or transfer destination) indicated by the REQD bit in the CHCFG_n register to assert the internal DMA acknowledge signal from the internal USB control to the DMAC control. Each time a transfer request is accepted, transfer proceeds. This operation is repeated until it reaches the transfer size loaded to the CRTB_n register. (Arbitration between channels is performed for each DMA transfer.)

The DMA acknowledge timing differs depending on the setting for the REQD bit in the CHCFG_n register or settings for the transfer size (DDS[2:0] and SDS[2:0] in the CHCFG_n register). For details, see section 32.16.7. Differences in Operation According to the Transfer Data Size.

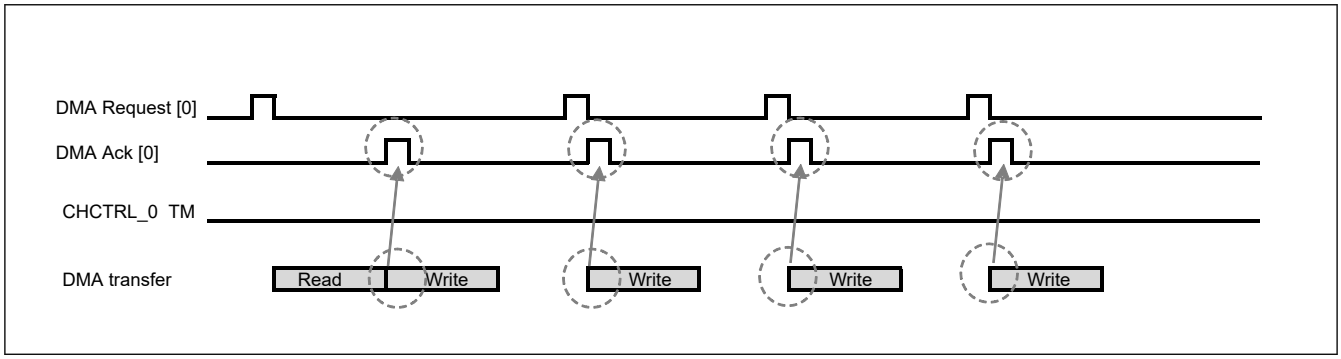


Figure 32.26 Single transfer mode (REQD = 1, SDS > DDS)

32.16.2 DMA Channel Priority Control

As an arbitration method between channels, fixed priority mode and round-robin mode are supported. To select a mode, use the PR bit in the DCTRL register. When the PR bit is 0, fixed priority mode is selected. When the PR bit is 1, round-robin mode is selected.

Table 32.51 Priority control settings

Mode	PR (DCTRL)	Description	Application
Fixed priority	0	Controls requests in fixed priority (high: CH0 > CH1: low) mode.	Use this mode if channels have priority.
Round-robin	1	Controls requests in round-robin mode.	Use this mode if you want to execute requests equally.

32.16.2.1 Fixed Priority Mode

In fixed priority mode, the priority among channels is fixed as shown below.

[High] CH0 > CH1 [Low]

If DMA transfer requests are generated in multiple channels simultaneously, a DMA transfer request with a smaller channel number takes precedence. The following figure provides an example where another DMA transfer request with higher priority is generated during a DMA transfer in fixed priority mode.

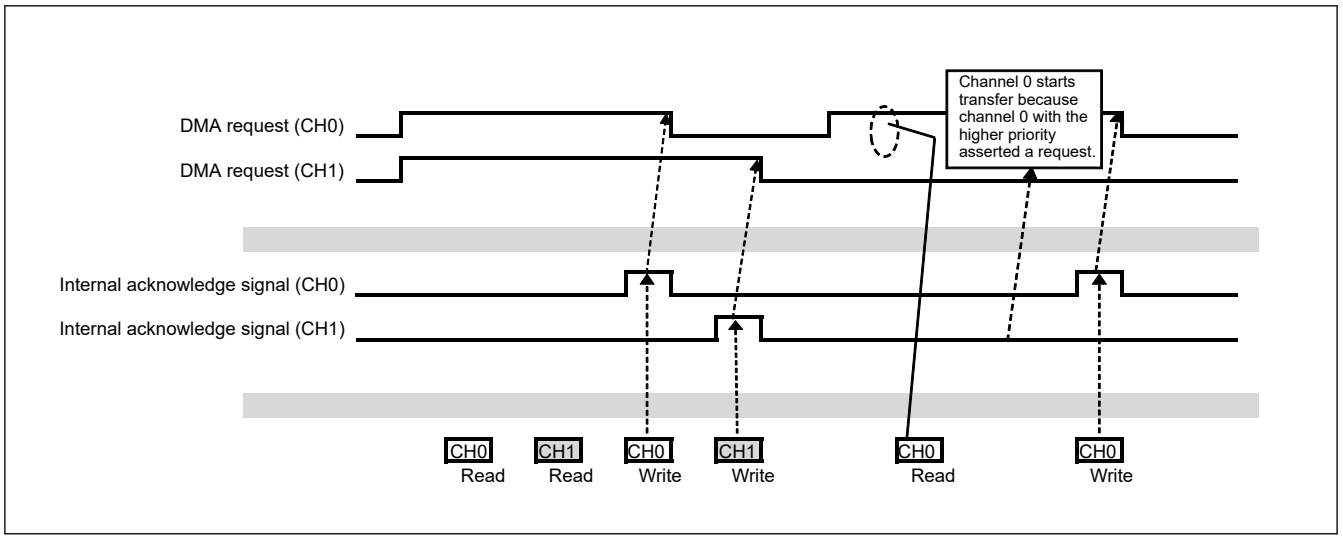


Figure 32.27 Fixed priority mode (four channels, REQD = 1)

Note: Channel 0 is handled with the highest priority, but the transfer with the next highest priority is performed to increase the bus usage rate while the transfer of channel 0 switches.

32.16.2.2 Round-Robin Mode

In round-robin mode, priority is changed each time a transfer of each channel is accepted so that the lowest priority is given to the channel in which the last transfer is performed.

The priority immediately after a reset is the same as the one in fixed priority mode as shown below.

[High] CH0 > CH1 [Low]

In this state, if no transfer request is generated in DMA channel 0, but a request is generated in DMA channel 1, a transfer of DMA channel 1 is performed. The priority after the transfer is shown below.

[High] CH1 > CH0 [Low]

The following figure provides an example of a DMA transfer in round-robin mode.

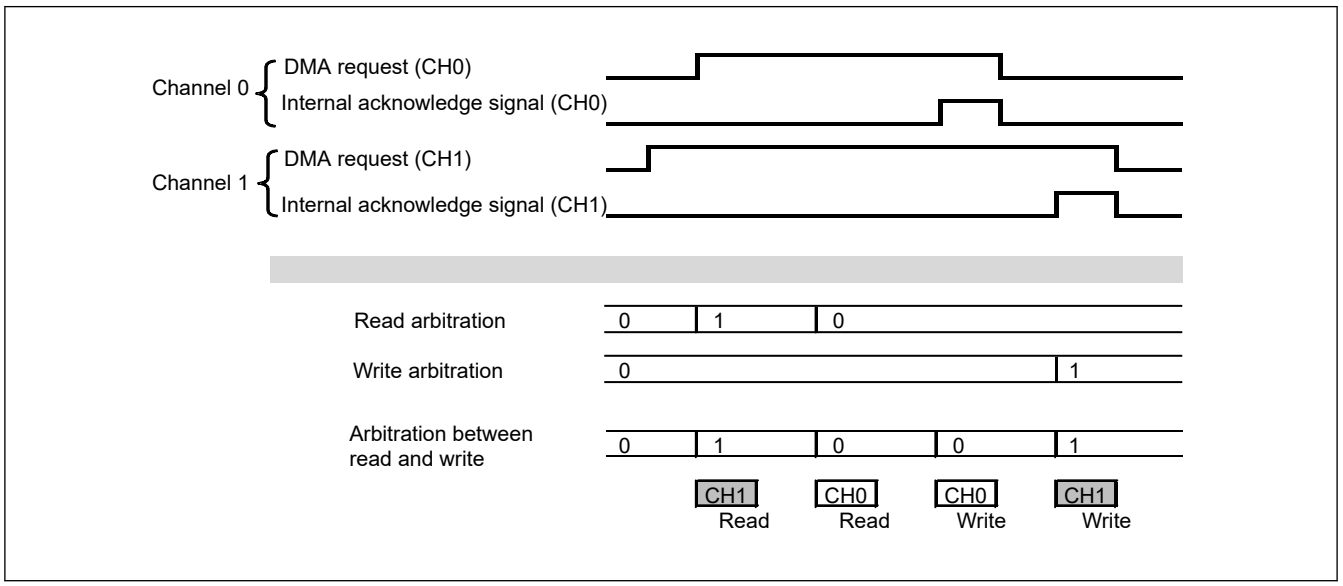


Figure 32.28 Round-robin mode (two channels, REQD = 1)

Note: In the DMAC, arbitration between read channels and arbitration between write channels are performed. As a result of these arbitrations, further arbitration is performed, and a bus access is issued.

32.16.3 Forced Ejection Request

When a forced ejection request is input, data that is not yet transferred in a buffer is transferred to the DMA transfer destination address. After data is flushed, the DMA transfer resumes.

The following describes notes on forced ejection requests:

- If a forced ejection request conflicts with a USB control DMA transfer request, the forced ejection takes precedence, and then the DMA transfer is performed.
- When the transfer destination is a system on the USB control side (the REQD bit in the CHCFG_n register = 1), the unit on the DMA transfer destination side receives data even if the USB control side does not request a DMA transfer. Therefore, a malfunction, such as buffer overflow, may occur. Because of this, when REQD =1, no forced ejection is performed by hardware.
- Differences from flush mode described in (2) Aborting a Transfer (with Buffer Flush: SBE = 1) (The EN bit is cleared to 0 when the SBE bit in the CHCFG_n register = 1.)
  Flush mode: The DMAC stops operation after data in a buffer is written.
  Forced ejection request: A DMA transfer continues after flush operation ends.

32.16.3.1 Software Forced Ejection Request

For software forced ejection requests, use the SETSSWPRQ bit in the CHCTRL_n register. To make a forced ejection request, write 1 to the SETSSWPRQ bit. The DMAC outputs data in a buffer to the DMA transfer destination.

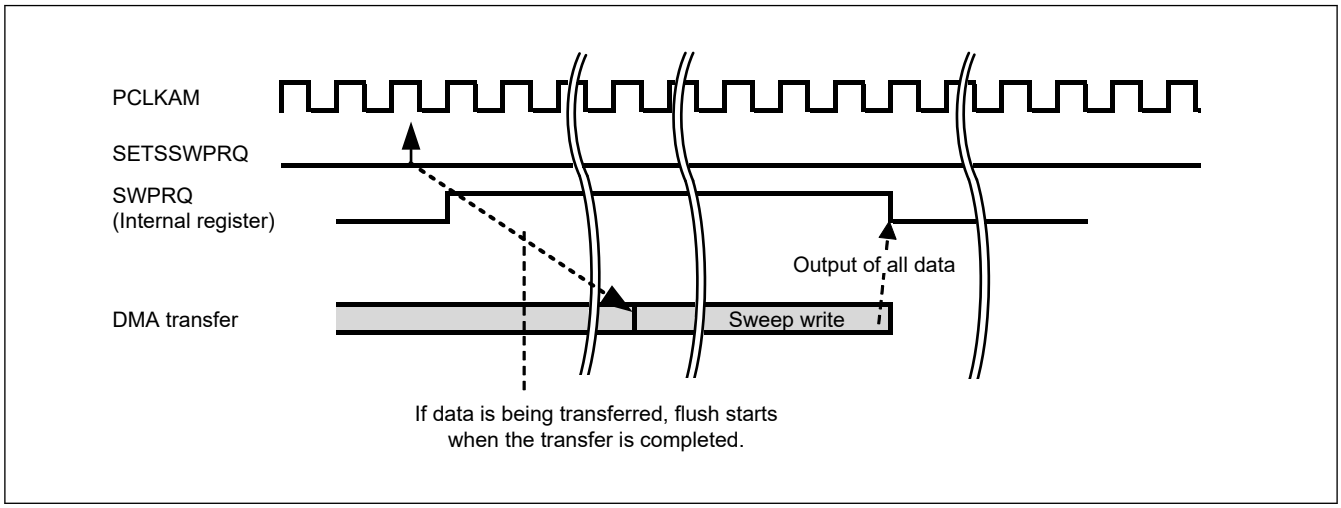


Figure 32.29 Software forced ejection timing

32.16.4 DMA Transfer Completion Interrupt (USB_FDMAn)

The USB_FDMAn[1:0] interrupt request signals indicate that a DMA transaction is completed.

Each bit of USB_FDMAn[1:0] corresponds to each channel.

When a transfer for the total number of transfer bytes loaded to the CRTB_n register is completed with an OKAY response, the END bit in the CHSTAT_n register is set to 1. If the DEM bit in the CHCFG_n register = 0 at this time, USB_FDMAn is asserted to high level (n = 1, 0). (When write-back is performed in link mode, this pin is asserted after the write-back operation.)

In addition, when the DRRP bit in the CHCFG_n register = 0 in link mode and the header of the read descriptor is LV = 0, the DER bit in the CHSTAT_n register is set to 1. If the DIM bit in the CHCFG_n register = 0 at this time, USB_FDMAn is asserted to high level.

Use these signals to detect a transfer completion interrupt in the interrupt controller.

Table 32.52 USB_FDMAn assert conditions

Source	Condition	USB_FDMAn mask signal
DMA transaction completed	When a transfer for the total number of transfer bytes loaded to the CRTB_n register is completed with an OKAY response (after write-back when write-back is performed in link mode)	DEM bit in the CHCFG_n register
Descriptor invalid	When the header of the read descriptor is LV = 0 while DRRP and DIM in the CHCFG_n register are 0 in link mode	DIM bit in the CHCFG_n register

32.16.5 DMA Error Interrupt (USB_FDMAERR)

If an error response is received during a DMA transfer or descriptor access, this module determines it as an error and aborts the transfer. When an error response is received, the EN bit in the CHSTAT_n register for channel n where a transfer is in progress is cleared to 0 and the ER bit is set to 1 (n = 1, 0). In addition, high level is output from the USB_FDMAERR pin.

The USB_FDMAERR signal cannot be masked.

Data for a series of error transfers is not guaranteed. Be sure to perform the following procedure to retry the transfer from the beginning.

1. Set the SWRST bit in the CHCTRL_n register to 1.
2. Set each register again.

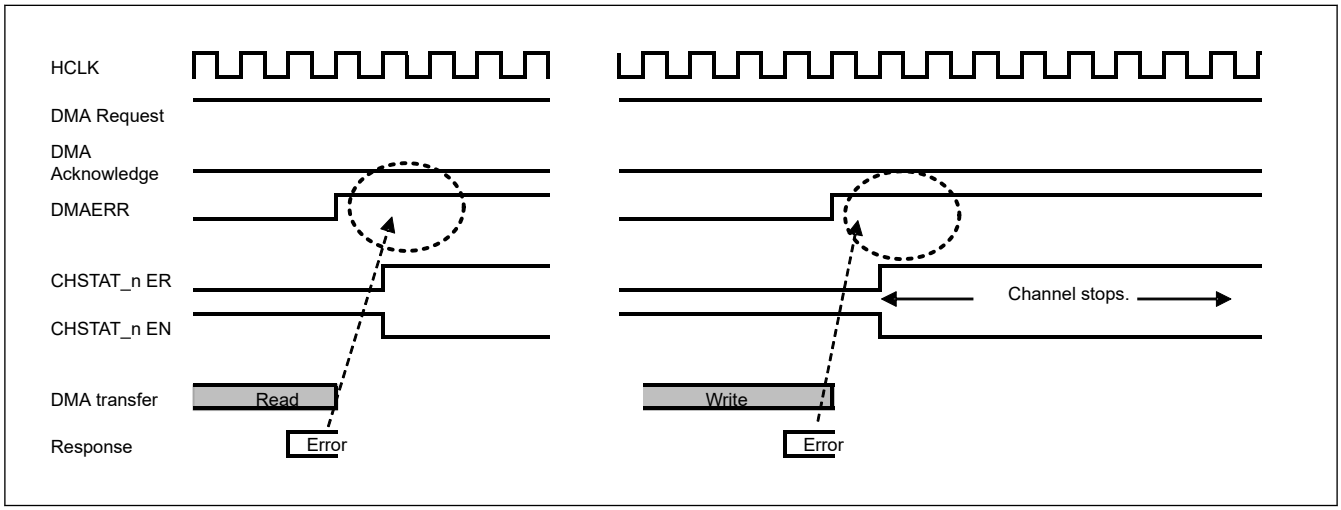


Figure 32.30 Transfer stop timing due to an error response (ERROR)

Note: The USB_FDMAERR pin output is not asserted even if an error occurs during an access to the slave of this controller.

32.16.6 Interval Count Function

By the setting for the ITVL field in the CHITVL_n register, the execution interval of DMA transfers can be adjusted. This function is used to avoid occupation of a bus by the DMAC. Until the count value becomes 0, no DMA transfer for the next DMA request is performed.

The following figure provides an operation example.

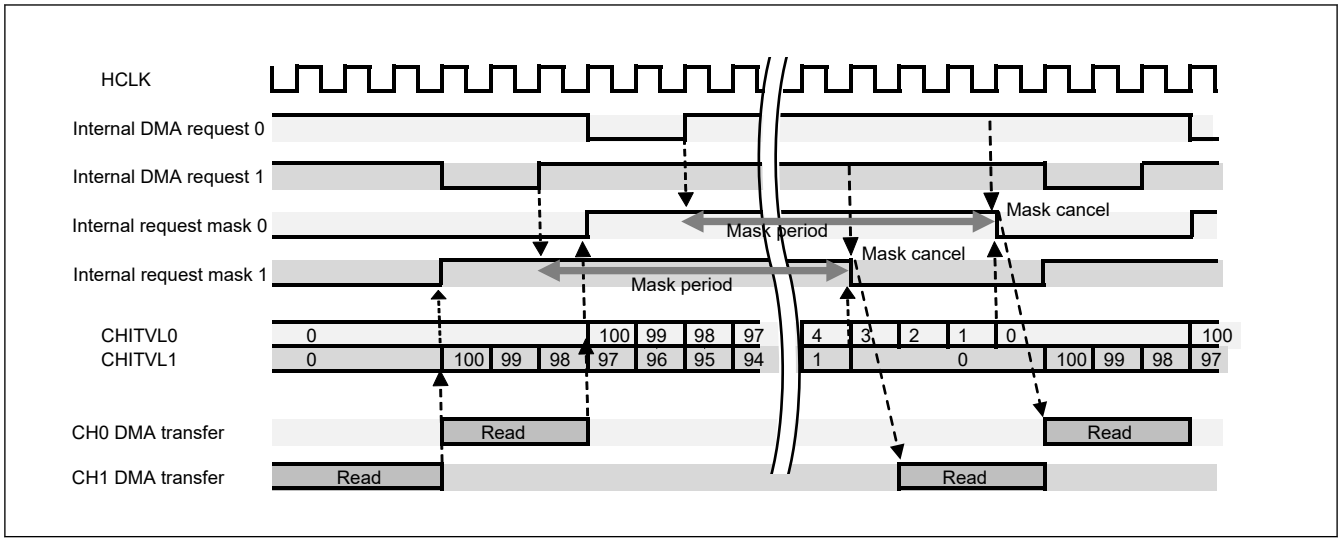


Figure 32.31 Interval count (REQD = 0, SDS < DDS)

An interval is inserted after a transfer specified by the REQD bit in the CHCFG_n register.

The following figure shows the relationship between the setting values for the REQD, SDS, and DDS bits in the CHCFG_n register and the cycles applied to intervals.

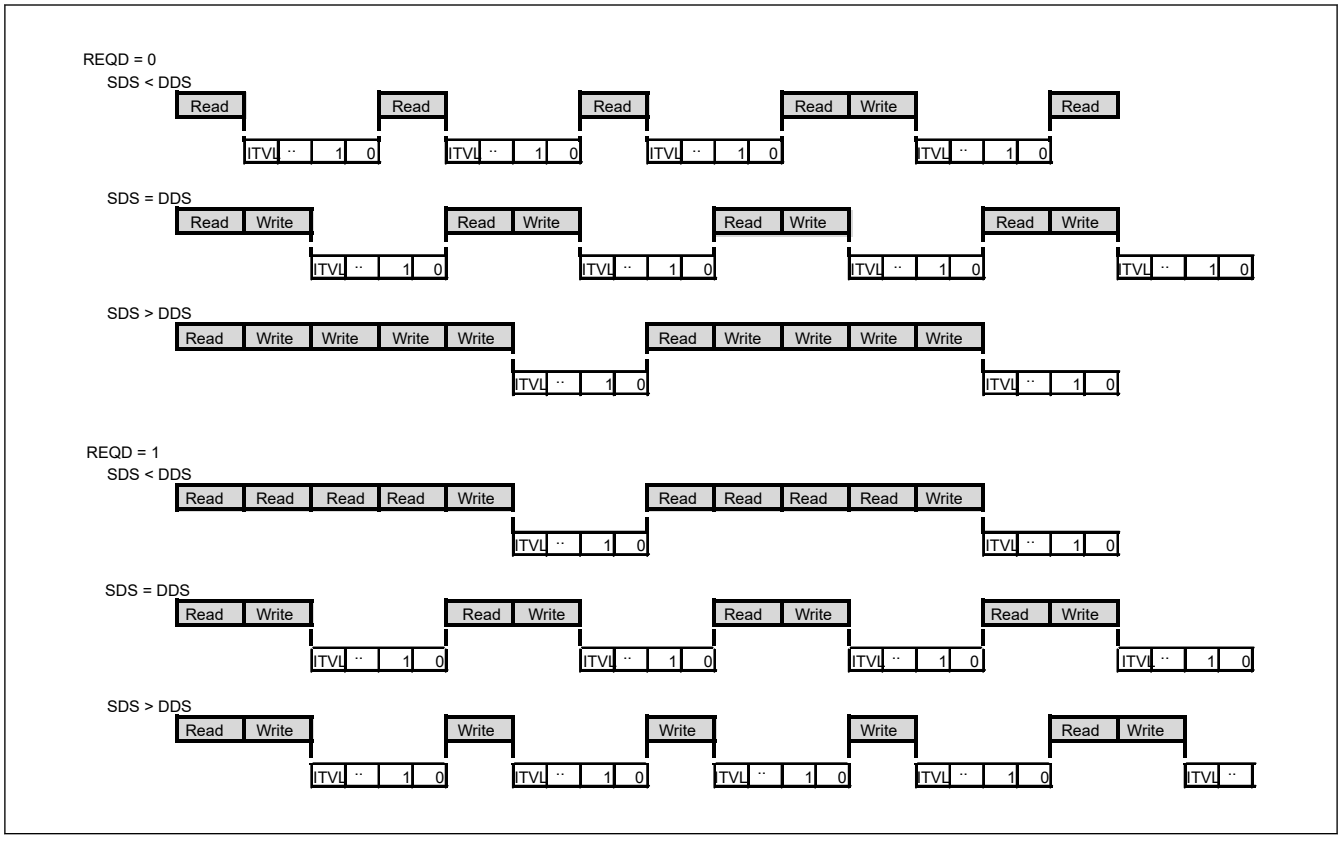


Figure 32.32 DMA transfer settings and interval counts

32.16.7 Differences in Operation According to the Transfer Data Size

32.16.7.1 When the Transfer Data Size on the Transfer Source Side is Small

When read operation for the transfer data size on the transfer destination side is completed, write operation to the transfer destination starts.

The following figure shows the timing chart (rising edge detection) when the transfer source data size is 8 bits and the transfer destination data size is 32 bits (SDS = 0 and DDS = 2 in the CHCFG_n register).

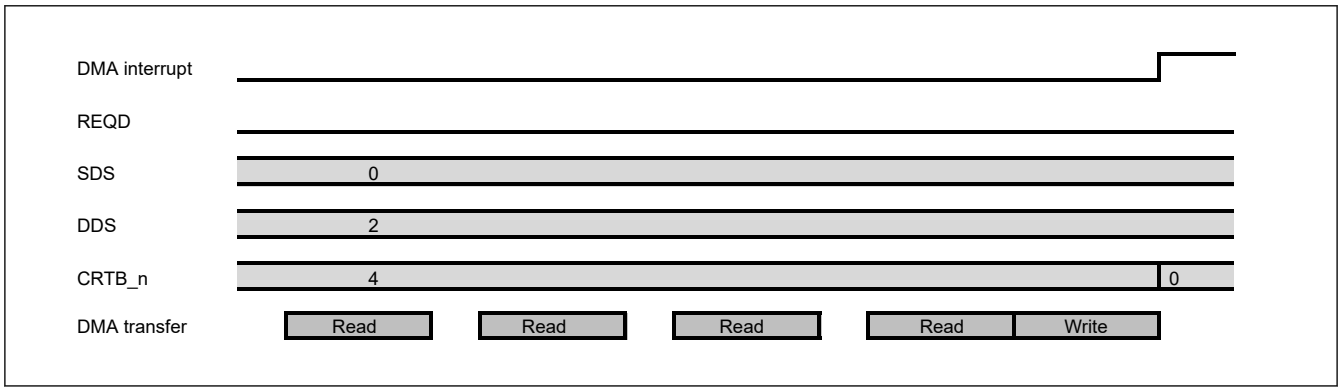


Figure 32.33 When the data size on the transfer source size is small (LVL = 0, HIEN = 1, REQD = 0, SDS < DDS in CHCFG_n)

32.16.7.2 When the Transfer Data Size on the Transfer Destination is Small

Because the transfer data size on the transfer source side is large, after a single read operation of the transfer source, write operation to the transfer destination is performed a few times. The following figure shows the timing chart (rising edge detection) when the transfer source data size is 64 bits and the transfer destination data size is 16 bits (SDS = 3 and DDS = 1 in the CHCFG_n register).

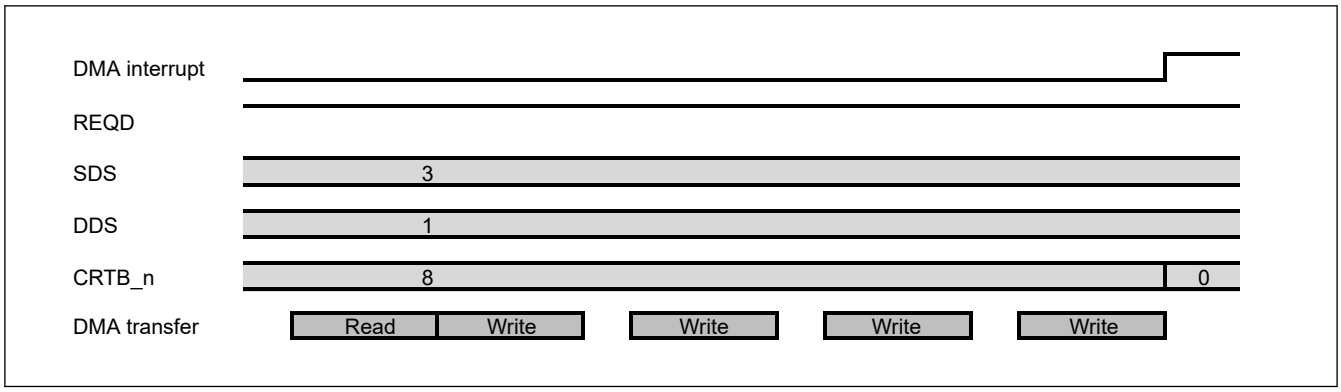


Figure 32.34 When the data size on the transfer destination side is small (LVL = 0, HIEN = 1, REQD = 1, SDS > DDS in CHCFG_n)

32.16.7.3 When Transfer Data Sizes on the Transfer Source Side and the Transfer Destination Side are Equal

Each time a DMA transfer request is detected, read operation is performed on the transfer source side and write operation is performed on the transfer destination side. The following figure shows the timing chart (rising edge detection) when the transfer source data size and the transfer destination data size are 8 bits (SDS = 0 and DDS = 0 in the CHCFG_n register).

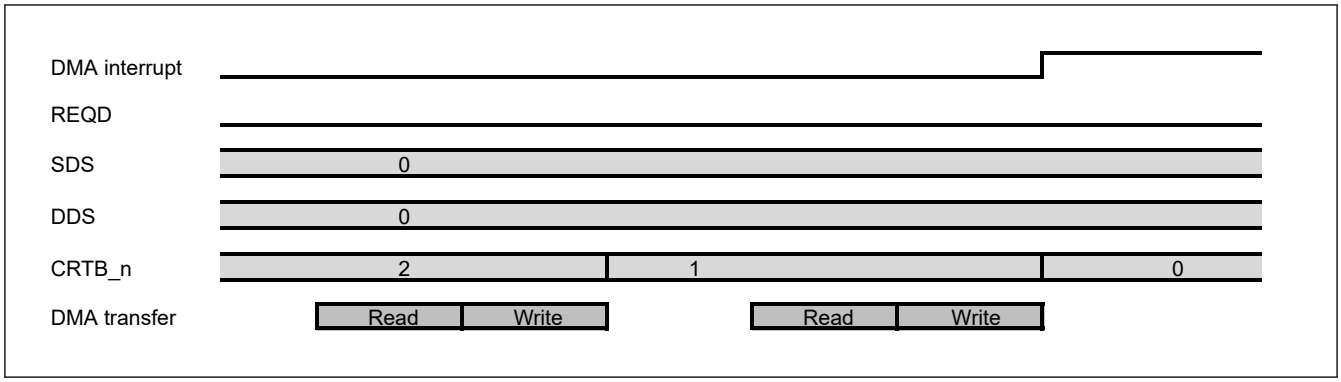


Figure 32.35 When transfer data sizes on the transfer source side and the transfer destination side are equal (LVL = 0, HIEN = 1, REQD = 0, and SDS = DDS in CHCFG_n)

32.16.8 Transfer Status

The CHSTAT_n register indicates the transfer status of each channel.

32.16.8.1 Transfer Status

The TACT bit in the CHSTAT_n register indicates that channel n is operating. Writing 1 to the SETEN bit in the CHCTRL_n register sets 1. The TACT bit remains 1 during an access to a descriptor or waiting for a DMA request.

The TACT bit is cleared when the EN bit in the CHSTAT_n register is cleared (for details on clearing conditions, see section 32.3.17.1. CHSTAT_n : Channel Status Register n (n = 0, 1), and the ongoing DMA transfer is completed.

The TACT bit is not cleared when the EN bit is not cleared (when the REN bit in the CHCFG_n register = 1 in register mode or the next descriptor access is performed in link mode) even if a DMA transaction finishes.

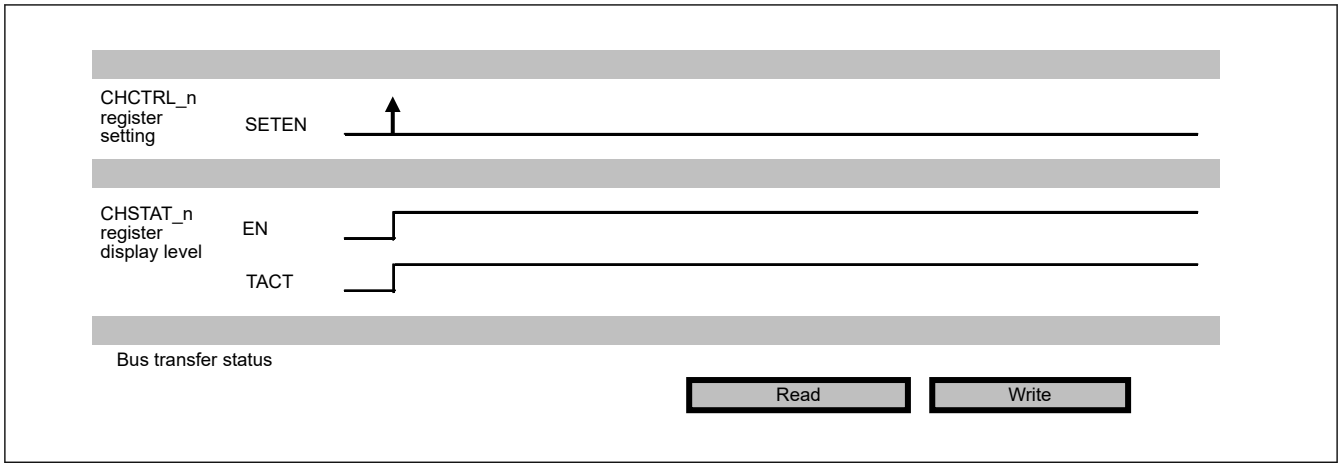


Figure 32.36 DMAC status example 1 (hardware request)

32.16.8.2 Suspending a Transfer

A DMA transfer can be suspended using the SETSUS bit in the CHCTRL_n register. At that time, if there is an active bus cycle, the DMA transfer is suspended after the bus cycle finishes. Writing 1 to the CLRSUS bit in the CHCTRL_n register resumes from the suspended state.

To check whether the transfer is suspended, set the SETSUS bit in the CHCTRL_n register, and then make sure that the SUS bit in the CHSTAT_n register or the SUS bit in the DSTAT_SUS register of the applicable channel is set to 1.

32.16.8.3 Aborting a Transfer

If 1 is written to the CLREN bit in the CHCTRL_n register during a DMA transaction, the DMA transaction of the channel can be aborted. As processing after aborting the transaction, you can use the SBE bit in the CHCFG_n register to select a mode to flush data remaining in a buffer when the transaction is aborted or a mode that does not flush buffer data. SBE = 0 (mode not to flush data) is selected by default.

When this mode (flush data) is enabled, if an ongoing transfer is aborted when the CLREN bit in the CHCTRL_n register = 1, data remaining in the buffer of the DMAC is flushed and the operation stops.

(1) Aborting a Transfer (without Buffer Flush: SBE = 0)

If 1 is written to the CLREN bit in the CHCTRL_n register during a DMA transfer, the DMA transfer is aborted and then stopped. The timing to stop the transfer depends on the set REQD bit value. After the transfer stops, write 1 to the SWRST bit in the CHCTRL_n register to clear the contents of the DMAC, and then make settings for the next transfer.

- When the TACT bit in the CHSTAT_n register is cleared to 0, you can confirm that the channel has completely stopped.
- If a DMA transfer is aborted, USB_FDMAn is not asserted.
- When the REQD bit in the CHCFG_n register = 0, the transfer stops when the next read operation is completed. (However, if data that can be written is remaining in the buffer, the data is written and then the transfer stops.)
- When the REQD bit in the CHCFG_n register = 1, the transfer stops when the next write operation is completed.

(2) Aborting a Transfer (with Buffer Flush: SBE = 1)

If 1 is written to the CLREN bit in the CHCTRL_n register during a DMA transfer, the DMA transfer is aborted. When the REQD bit in the CHCFG_n register = 0, already read data is flushed (written), and then the DMA transfer stops. When REQD = 1, flush mode cannot be used by hardware.

After the transfer stops, set the SWRST bit in the CHCTRL_n register to clear the contents of the DMAC, and then make settings for the next transfer.

- When the TACT bit in the CHSTAT_n register is cleared to 0, you can confirm that the channel has completely stopped.

(3) Checking Channel Stop

Even if 1 is written to the CLREN bit in the CHCTRL_n register, and the EN bit in the CHSTAT_n register is cleared to 0, if a transfer is already in progress on the bus, the DMAC cannot stop the transfer immediately. To check that DMA has completely stopped, make sure that the EN bit is 0 and the TACT bit in the CHSTAT_n register is 0.

(4) Transfer Aborting Procedure

The following describes the procedure for aborting a transfer:

1. Write 1 to the CLREN bit in the CHCTRL_n register.
2. When the SBE bit in the CHCFG_n register = 0, the transfer stops according to the REQD bit value in the CHCFG_n register. If SBE = 1 at this time, flush mode is enabled.
3. Read the CHSTAT_n register and confirm that the TACT bit is 0. When TACT = 0, DMA has completely stopped. When TACT = 1, continue polling until the TACT bit is cleared to 0.
4. If you want to perform the next DMA transfer after a transfer is aborted, be sure to set the SWRST (software reset) bit in the CHCTRL_n register to 1 before the next transfer starts.

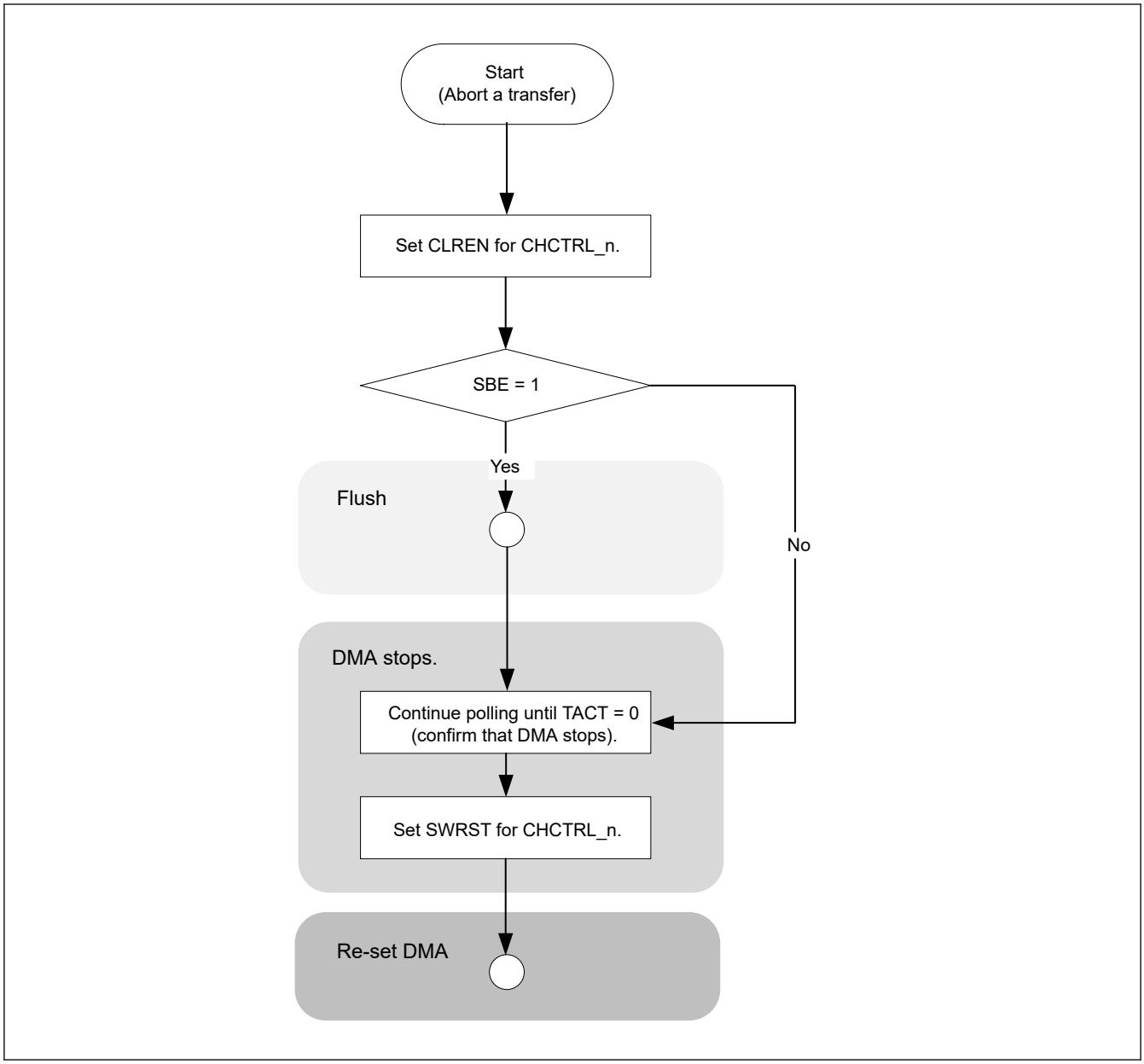


Figure 32.37 Transfer aborting flow

32.17 Access Type

32.17.1 AHB Master DMA Transfer Combinations

32.17.1.1 Read Access

The DMA read access issuance type has the following features.

- Read access is made to a beat-aligned space (including the source address indicated in the CRSA_n register) with the size specified by the SDS[2:0] bits in the CHCFG_n register. Beat-unaligned transfer is not performed for the bus. An unnecessary area may be read depending on the set values of the CRSA_n register and each SKIP register. In this case, necessary data in the read data is loaded to a buffer.
- The size and the burst length are determined by the value set for the SDS[2:0] field.
  When the value set for the SDS[2:0] field is equal to or smaller than the bus width:

Size:	SDS[2:0] set value

Burst type: Single

When the value set for the SDS[2:0] field is larger than the bus width:

Size: Bus width

Burst type: Fixed-length burst (burst length = SDS[2:0] set value/bus width)

The following table shows AHB bus access types.

Table 32.53 DMA read transfer combinations

SDS	Source address	AHB transfer					
		First transfer			Second transfer		
		Address	Size	Burst	Address	Size	Burst
0 (8 bits)	—	addr	8	Single	—	—	—
1 (16 bits)	2-byte align	{addr[31:1], 0b}	16	Single	—	—	—
	2-byte unalign				{addr[31:1], 0b} + 0x2	16	Single
2 (32 bits)	4-byte align	{addr[31:2], 00b}	32	Single	—	—	—
	4-byte unalign				{addr[31:2], 00b} + 0x4	32	Single
4 (128 bits)	16-byte align	{addr[31:4], 0x0}	32	INCR4	—	—	—
	16 bytes				{addr[31:4], 0x0} + 0x10	32	INCR4
5 (256 bits)	32-byte align	{addr[31:5], 0x00}	32	INCR8	—	—	—
	32 bytes				{addr[31:5], 0x00} + 0x20	32	INCR8
6 (512 bits)	64-byte align	{addr[31:6], 0x00}	32	INCR16	—	—	—
	64 bytes				{addr[31:6], 0x00} + 0x40	32	INCR16

Note: If the transfer becomes EBT during the burst transfer, remaining transfer is performed by 32-bit INCR.

32.17.1.2 Write Access

The DMA write access issuance type has the following features.

- Write access is made from the destination address (indicated in the CRDA_n register) to a beat-aligned boundary with the size specified by the DDS[2:0] bits in the CHCFG_n register.
- The size and the burst length are determined by the value set for the DDS[2:0] field.
  When the value set for the DDS[2:0] field is equal to or smaller than the bus width:

  Size: DDS[2:0] set value

  Burst type: Single

  When the value set for the DDS[2:0] field is larger than the bus width:

  Size: Bus width

  Burst type: Fixed-length burst (burst length = DDS[2:0] set value/bus width)
- Write access to any space other than the specified space is not performed. In the following cases, write access is made with a combination of sizes smaller than the value set for the DDS[2:0] field.
  - When the destination address is beat-unaligned with the value set for the DDS[2:0] field
  - When an access with the DDS field size is made over the SKIP boundary
  - When the DDS[2:0] size is larger than the remaining number of transfer bytes

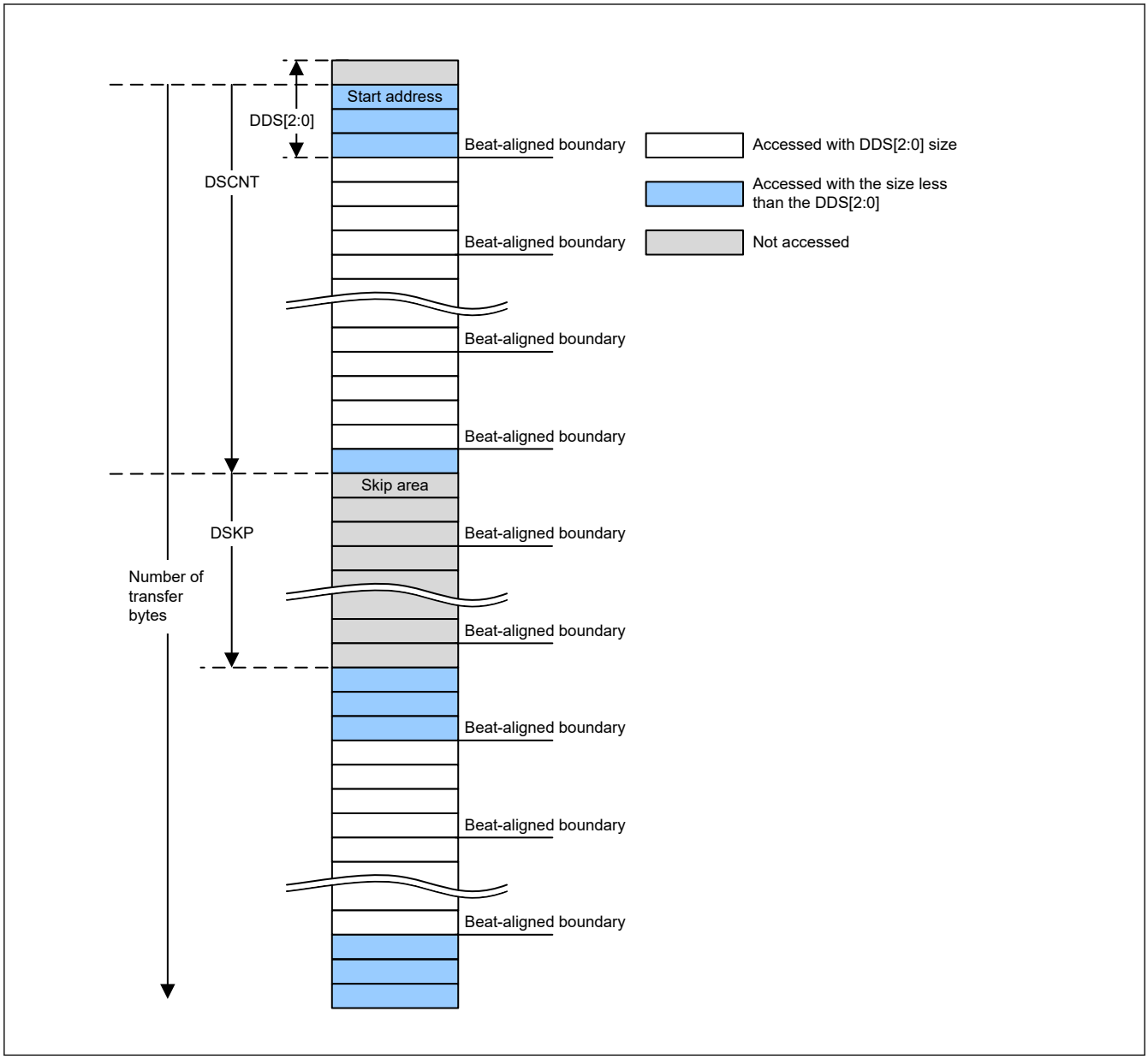


Figure 32.38 Example of DMA write access space and access type

33. Serial Communications Interface (SCI, SCIE)

33.1 Overview

This LSI has six independent serial communications interface (SCI) channels and 12 independent serial communications interface for encoder interface (SCIE) channels. The SCI is configurable to six asynchronous and synchronous serial interfaces:

- Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA))
- 8-bit clock synchronous interface
- Simple I2C (master-only)
- Simple SPI
- Smart card interface
- Manchester mode

The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. Each SCI has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using a baud rate generator.

The SCIE is configurable to Asynchronous, Clock synchronous, and Manchester mode only. Note that wording, SCI, stands for both SCI and SCIE in this chapter except significant difference.

Table 33.1 lists the SCI specifications, Figure 33.1 shows a block diagram of SCI channel n, and Table 33.2 lists the I/O pins.

Table 33.1 SCI specifications (1 of 3)

Item	Specifications
Number of channels	6 channels (ch0 to ch4 in NONSAFETY and ch5 in SAFETY) 12 channels for encoder (8 in LLPP0 and 4 in LLPP1)
Serial communication modes	• Asynchronous • Clock synchronous • Smart card interface (This mode is not supported in SCIE.) • Simple I2C (This mode is not supported in SCIE.) • Simple SPI (This mode is not supported in SCIE.) • Manchester mode
Transfer speed	Bit rate specifiable with the baud rate generator
Full-duplex communications	• Transmitter: Continuous transmission possible using double-buffering • Receiver: Continuous reception possible using double-buffering
Half-duplex communication	Possible by using TXD pins
Data transfer	Selectable as LSB-first or MSB-first transfer
RXD/TXD polarity	Selectable polarity for TXD and RXD independently
Interrupt sources	• Transmit end, transmit data empty, receive data full, receive error, receive data ready, and address match • Completion of generation of a start condition, restart condition, or stop condition (for simple I2C mode)
RS-485 driver control function	Output DE pin to enable transmission mode of external transceiver
Loopback function	Self-diagnosis of communication function inside module supported
Synchronizing circuit bypass function	Bus clock (PCLKM) can be used as operation clock (PCLKSCIn) when bypass function is enabled.
Module-stop function	Module-stop state can be set for each channel

Table 33.1 SCI specifications (2 of 3)

Item		Specifications
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Reception sampling timing adjustment	Adjustable reception sampling timing from default timing
	Transmission timing adjustment	Adjustable timing of transmission wave edge by register
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	Transmission and reception controllable with CTSn# and RTSn# pins
	Transmission/reception	Selectable to 1-stage register or 16-stage FIFO
	Address match	Interrupt request/event output can be issued upon detecting a match between received data and the value in the compare match register.
	Start-bit detection	Selectable to low level or falling edge detection
	Break detection	Breaks from framing errors detectable by reading register
	Clock source	Selectable to internal or external clock
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication enabled between multiple processors
	Noise cancellation	Digital noise filters included on signal paths from RXDn pin inputs
Clock synchronous mode	Data length	8 bits
	Reception sampling timing adjustment	Adjustable reception sampling timing to backward from default timing in case internal clock used
	Receive error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Hardware flow control	Transmission and reception controllable with CTSn# and RTSn# pins
	Transmission/reception	Selectable to 1-stage register or 16-stage FIFO
Smart card interface mode	Error processing	Error signal can be automatically transmitted on detecting a parity error during reception.
		Data can be automatically retransmitted on receiving an error signal during transmission.
	Data type	Both direct and inverse convention supported
Simple I2C mode	Transfer format	I2C bus format (MSB-first only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 kbps
	Noise cancellation	The signal paths from input on the SCLn and SDAn pins incorporate digital noise filters and provide an adjustable interval for noise cancellation.

Table 33.1 SCI specifications (3 of 3)

Item		Specifications
Simple SPI mode	Data length	8 bits
	Error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Reception sampling timing adjustment	Adjustable reception sampling timing to backward from default timing in case internal clock used
	SS input pin function	High impedance state can be invoked on the output pins by driving the SSn# pin high.
	Clock settings	Configurable between four clock phase and clock polarity settings
	Transmission/reception	Selectable to 1-stage register or 16-stage FIFO
Manchester mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Receive error detection	Parity, overrun, framing, Manchester errors, Preface, Start Bit, SYNC
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission
	Clock source	Only internal clock can be used. (The setting of external clock is prohibited because it is not the object of operation guarantee.)
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communication function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RxDn pins incorporate digital noise filters
	Manchester encoding/decoding function	Function to perform Manchester encoding/decoding of transmission/reception data and communicate using Manchester code
	Preface setting/detection function	The function outputs the configured the preface pattern and detects it.
	Start Bit setting/detection function	The function outputs the configured the Start Bit pattern and detects it.
	Reception retiming function	Function to perform timing correction for each bit center edge by using Manchester code having edge at bit center
Bit rate modulation function		Error reduction through correction of outputs from the baud rate generator

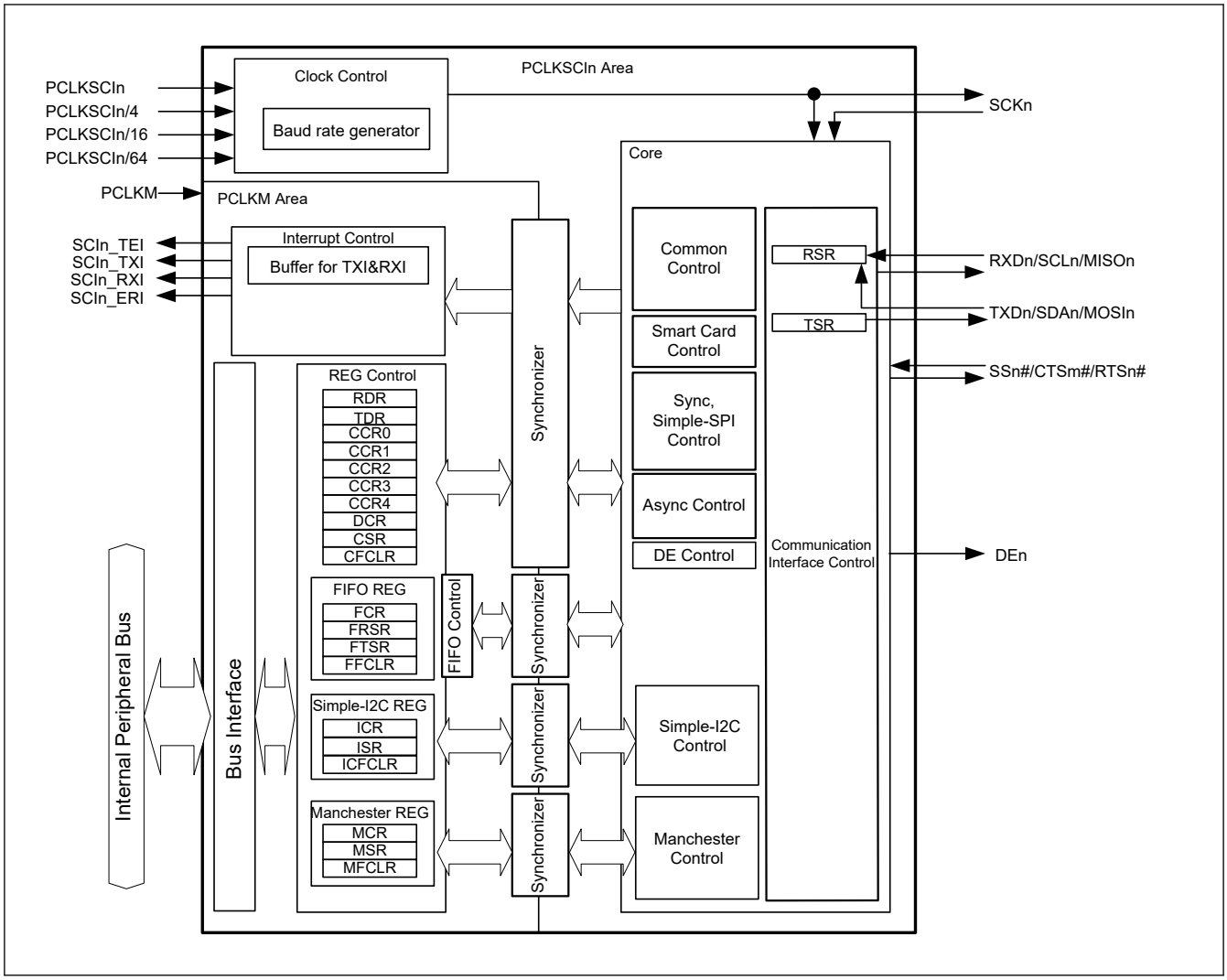


Figure 33.1 SCI block diagram

Table 33.2 SCI input/output pins

Channel	Pin name	I/O	Function
SCIn	SCKn	I/O	SCIn clock input/output
	RXDn	Input	SCIn receive data input
	SCLn	I/O	SCIn I2C clock input/output
	MISOn	I/O	SCIn data input (master), data output (slave)
	TXDn	Output	SCIn transmit data output
	SDAn	I/O	SCIn I2C data input/output
	MOSIn	I/O	SCIn data output (master), data input (slave)
	SSn#	Input	SCIn chip select input, active-low
	RTSn#	Output	SCIn transfer start control output, active-low
	CTSn#	Input	SCIn transfer start control input, active-low
	DEn	Output	SCIn Driver Enable output
SCIEm	SCKEm	Output	SCIEm clock output for encoder
	RXDEm	Input	SCIEm receive data input for encoder
	TXDEm	Output	SCIEm transmit data output for encoder
	DEEm	Output	SCIEm Driver Enable output for encoder

Note: n = 0 to 5, m = 0 to 11

33.2 Register Map

Table 33.3 SCI register map (1 of 2)

Address		Register symbol	Register name	Write protection
SCI	0x8000_5000 + 0x0400 × n (n = 0 to 4) 0x8100_5000 (n = 5)	RDR	Receive Data Register	—
SCIE	0x9003_0000 + 0x0400 × n (n = 0 to 7) 0x9013_0000 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5000 + 0x0400 × n (n = 0 to 4) 0x8100_5000 (n = 5)	RDR	Receive Data Register	—
SCIE	0x9003_0000 + 0x0400 × n (n = 0 to 7) 0x9013_0000 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5004 + 0x0400 × n (n = 0 to 4) 0x8100_5004 (n = 5)	TDR	Transmit Data Register	—
SCIE	0x9003_0004 + 0x0400 × n (n = 0 to 7) 0x9013_0004 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5008 + 0x0400 × n (n = 0 to 4) 0x8100_5008 (n = 5)	CCR0	Common Control Register 0	—
SCIE	0x9003_0008 + 0x0400 × n (n = 0 to 7) 0x9013_0008 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_500C + 0x0400 × n (n = 0 to 4) 0x8100_500C (n = 5)	CCR1	Common Control Register 1	—
SCIE	0x9003_000C + 0x0400 × n (n = 0 to 7) 0x9013_000C + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5010 + 0x0400 × n (n = 0 to 4) 0x8100_5010 (n = 5)	CCR2	Common Control Register 2	—
SCIE	0x9003_0010 + 0x0400 × n (n = 0 to 7) 0x9013_0010 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5014 + 0x0400 × n (n = 0 to 4) 0x8100_5014 (n = 5)	CCR3	Common Control Register 3	—
SCIE	0x9003_0014 + 0x0400 × n (n = 0 to 7) 0x9013_0014 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5018 + 0x0400 × n (n = 0 to 4) 0x8100_5018 (n = 5)	CCR4	Common Control Register 4	—
SCIE	0x9003_0018 + 0x0400 × n (n = 0 to 7) 0x9013_0018 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5020 + 0x0400 × n (n = 0 to 4) 0x8100_5020 (n = 5)	ICR	Simple I2C Control Register	—
SCIE	0x9003_0020 + 0x0400 × n (n = 0 to 7) 0x9013_0020 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5024 + 0x0400 × n (n = 0 to 4) 0x8100_5024 (n = 5)	FCR	FIFO Control Register	—
SCIE	0x9003_0024 + 0x0400 × n (n = 0 to 7) 0x9013_0024 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_502C + 0x0400 × n (n = 0 to 4) 0x8100_502C (n = 5)	MCR	Manchester Control Register	—
SCIE	0x9003_002C + 0x0400 × n (n = 0 to 7) 0x9013_002C + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5030 + 0x0400 × n (n = 0 to 4) 0x8100_5030 (n = 5)	DCR	Driver Control Register	—
SCIE	0x9003_0030 + 0x0400 × n (n = 0 to 7) 0x9013_0030 + 0x0400 × (n - 8) (n = 8 to 11)			

Table 33.3 SCI register map (2 of 2)

Address		Register symbol	Register name	Write protection
SCI	0x8000_5048 + 0x0400 × n (n = 0 to 4) 0x8100_5048 (n = 5)	CSR	Common Status Register	—
SCIE	0x9003_0048 + 0x0400 × n (n = 0 to 7) 0x9013_0048 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_504C + 0x0400 × n (n = 0 to 4) 0x8100_504C (n = 5)	ISR	Simple I2C Status Register	—
SCIE	0x9003_004C + 0x0400 × n (n = 0 to 7) 0x9013_004C + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5050 + 0x0400 × n (n = 0 to 4) 0x8100_5050 (n = 5)	FRSR	FIFO Receive Status Register	—
SCIE	0x9003_0050 + 0x0400 × n (n = 0 to 7) 0x9013_0050 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5054 + 0x0400 × n (n = 0 to 4) 0x8100_5054 (n = 5)	FTSR	FIFO Transmit Status Register	—
SCIE	0x9003_0054 + 0x0400 × n (n = 0 to 7) 0x9013_0054 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5058 + 0x0400 × n (n = 0 to 4) 0x8100_5058 (n = 5)	MSR	Manchester Status Register	—
SCIE	0x9003_0058 + 0x0400 × n (n = 0 to 7) 0x9013_0058 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5068 + 0x0400 × n (n = 0 to 4) 0x8100_5068 (n = 5)	CFCLR	Common Flag Clear Register	—
SCIE	0x9003_0068 + 0x0400 × n (n = 0 to 7) 0x9013_0068 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_506C + 0x0400 × n (n = 0 to 4) 0x8100_506C (n = 5)	ICFCLR	Simple I2C Flag Clear Register	—
SCIE	0x9003_006C + 0x0400 × n (n = 0 to 7) 0x9013_006C + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5070 + 0x0400 × n (n = 0 to 4) 0x8100_5070 (n = 5)	FFCLR	FIFO Flag Clear Register	—
SCIE	0x9003_0070 + 0x0400 × n (n = 0 to 7) 0x9013_0070 + 0x0400 × (n - 8) (n = 8 to 11)			
SCI	0x8000_5074 + 0x0400 × n (n = 0 to 4) 0x8100_5074 (n = 5)	MFCLR	Manchester Flag Clear Register	—
SCIE	0x9003_0074 + 0x0400 × n (n = 0 to 7) 0x9013_0074 + 0x0400 × (n - 8) (n = 8 to 11)			

Table 33.4 SCI related system control register (1 of 2)

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
SCI unit 0	—	MSTPCRA.MSTPCRA08	SLVACCCTL1.SCI0_SL
SCI unit 1	—	MSTPCRA.MSTPCRA09	SLVACCCTL1.SCI1_SL
SCI unit 2	—	MSTPCRA.MSTPCRA10	SLVACCCTL1.SCI2_SL
SCI unit 3	—	MSTPCRA.MSTPCRA11	SLVACCCTL1.SCI3_SL
SCI unit 4	—	MSTPCRA.MSTPCRA12	SLVACCCTL1.SCI4_SL
SCI unit 5	—	MSTPCRG.MSTPCRG00	SLVACCCTL2.SCI5_SL
SCIE unit 0	—	MSTPCRA.MSTPCRA16	SLVACCCTL7.LLPP_SL*1
SCIE unit 1	—	MSTPCRA.MSTPCRA17	SLVACCCTL7.LLPP_SL*1
SCIE unit 2	—	MSTPCRA.MSTPCRA18	SLVACCCTL7.LLPP_SL*1
SCIE unit 3	—	MSTPCRA.MSTPCRA19	SLVACCCTL7.LLPP_SL*1

Table 33.4 SCI related system control register (2 of 2)

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
SCIE unit 4	—	MSTPCRA.MSTPCRA20	SLVACCCTL7.LLPP_SL*1
SCIE unit 5	—	MSTPCRA.MSTPCRA21	SLVACCCTL7.LLPP_SL*1
SCIE unit 6	—	MSTPCRA.MSTPCRA22	SLVACCCTL7.LLPP_SL*1
SCIE unit 7	—	MSTPCRA.MSTPCRA23	SLVACCCTL7.LLPP_SL*1
SCIE unit 8	—	MSTPCRA.MSTPCRA24	SLVACCCTL7.LLPP_SL*1
SCIE unit 9	—	MSTPCRA.MSTPCRA25	SLVACCCTL7.LLPP_SL*1
SCIE unit 10	—	MSTPCRA.MSTPCRA26	SLVACCCTL7.LLPP_SL*1
SCIE unit 11	—	MSTPCRA.MSTPCRA27	SLVACCCTL7.LLPP_SL*1

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

33.3 Register Descriptions

This section explains each register specification.

Table 33.5 shows a list of registers/bits that can not be rewritten during communication. Also, check the notes on register rewriting described in section 33.18. Usage Notes.

Table 33.5 Bits that cannot be rewritten during communication

Register	Bit	Write constraint
CCR0	RE	In Clock synchronous mode, Simple-SPI and Simple-I2C, 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.
CCR0	TE	
CCR0	IDSEL	Writable only when TE = 0 and RE = 0.
CCR0	SSE	Writable only when TE = 0 and RE = 0.
CCR1	CTSE	Writable only when TE = 0 and RE = 0.
CCR1	CTSPEN	Writable only when TE = 0 and RE = 0.
CCR1	PE	Writable only when TE = 0 and RE = 0.
CCR1	PM	Writable only when TE = 0 and RE = 0.
CCR1	TINV	Writable only when TE = 0 and RE = 0.
CCR1	RINV	Writable only when TE = 0 and RE = 0.
CCR1	SPLP	Writable only when TE = 0 and RE = 0.
CCR1	SHARPS	Writable only when TE = 0 and RE = 0.
CCR1	NFCS[2:0]	Writable only when TE = 0 and RE = 0.
CCR1	NFEN	Writable only when TE = 0 and RE = 0.
CCR2	All bits	Writable only when TE = 0 and RE = 0.
CCR3	All bits	Writable only when TE = 0 and RE = 0.
CCR4	All bits	Writable only when TE = 0 and RE = 0.
ICR	IICDL[4:0]	Writable only when TE = 0 and RE = 0.
ICR	IICINTM	Writable only when TE = 0 and RE = 0.
ICR	IICCSC	Writable only when TE = 0 and RE = 0.
FCR	All bits	Writable only when TE = 0 and RE = 0.
MCR	All bits	Writable only when TE = 0 and RE = 0.
DCR	All bits	Writable only when TE = 0 and RE = 0.

33.3.1 RSR : Receive Shift Register

RSR is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data. When one frame of data has been received, it is automatically transferred to the RDR register.

The RSR register cannot be directly accessed by the CPU.

33.3.2 RDR : Receive Data Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	FER	PER	—	—	ORER	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	FFER	FPER	DR	MPB	RDAT[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	RDAT[8:0]	Serial receive data RDAT is a 9-bit register for storing received data. Received data is stored in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. And 0 is stored in the unused bit.	R
9	MPB	Multi-processor flag 0: Data transmission cycles 1: ID transmission cycles	R
10	DR	Receive data ready flag FRSR.DR can be read.	R
11	FPER	FIFO parity error flag (Valid only in Asynchronous mode) 0: There is no parity error in the data read from the receive FIFO. 1: There is parity error in the data read from the receive FIFO.	R
12	FFER	FIFO framing error flag (Valid only in Asynchronous mode) 0: There is no framing error in the data read from the receive FIFO. 1: There is framing error in the data read from the receive FIFO.	R
23:13	—	These bits are read as 0.	R
24	ORER	Overrun Error flag CSR.ORER can be read.	R
26:25	—	These bits are read as 0.	R
27	PER	Parity error flag CSR.PER can be read.	R
28	FER	Framing error flag CSR.FER can be read.	R
31:29	—	These bits are read as 0.	R

In FIFO mode (CCR3.FM = 1), this register has 16-stage FIFO structure.

RDAT[8:0] bits (Serial receive data)

After one frame of data is received, the received data is transferred from the RSR register to this registers, thus allowing the RSR register to receive the next data.

The RSR and RDR registers have a double-buffered structure to enable continuous reception.

Read RDR only once when a receive data full interrupt (RXI) request is issued. Without reading received data from RDR, if the next one frame is received, an overrun error occurs.

In FIFO mode, continuous reception is executed until 16 stages are stored. If data is read when there is no received data in the receive FIFO (RDR), the value is undefined. When the receive FIFO (RDR) are full of received data, subsequent serial receive data is lost.

The CPU cannot write to RDR.

"0" is stored in the bit position which is not received (RDR.bit8 or RDR.bit7) at the time of 7 bit or 8 bit communication of Asynchronous and Manchester mode.

FFER and FPER store error information of received data only in FIFO mode. When not FIFO, 0 is stored in RDR.bit 12 and bit 11.

MPB bit (Multi-processor flag)

In asynchronous mode and Manchester mode, during multi-processor communication (CCR3.MP = 1), the value of the multi-processor bit corresponding to the received data (RDAT[8:0]) can be read.

FPER bit (FIFO parity error flag)

Indicates whether the data read from the receive FIFO has a parity error.

FFER bit (FIFO framing error flag)

Indicates whether the data read from receive FIFO has a framing error.

33.3.3 TDR : Transmit Data Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TSYNC	—	—	MPBT	TDAT[8:0]								
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
8:0	TDAT[8:0]	Serial transmit data TDAT is a 9-bit register for setting transmit data. Transmit data is set in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. When byte access, write TDR[15:8] and then write TDR[7:0].	R/W
9	MPBT	Multi-processor transfer bit flag Value of the multi-processor bit in the transmission frame. This bit is use in Asynchronous mode and Manchester mode. When writing to this bit when not used, write the initial value. 0: Data transmission cycles 1: ID transmission cycles	R/W
11:10	—	These bits are read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
12	TSYNC	Transmit SYNC Data Bit It is valid when MCR.SBSEL = 1 and MCR.SYNSEL = 1 in Manchester mode. When writing to this bit when not used, write the initial value. 0: The Start Bit is transmitted as DATA SYNC. 1: The Start Bit is transmitted as COMMAND SYNC.	R/W
31:13	—	These bits are read as 1. The write value should be 1.	R/W

In FIFO mode (CCR3.FM = 1), this register has 16-stage FIFO structure.

TDAT[8:0] bits (Serial transmit data)

The TDR is a 9-bit register for storing transmit data.

When empty space is detected in the TSR register, the transmit data stored in the TDR registers is transferred to TSR, and transmitting is started.

The TSR and TDR registers have a double-buffer structure to realize continuous transmission. When the next data to be transmitted is stored in TDR after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

In FIFO mode, when the SCI detects that the transmit shift register (TSR) is empty, it transmits data in the transmit FIFO (TDR) into TSR and starts serial transmission. Continuous serial transmission is executed until there is no transmit data left in the transmit FIFO (TDR).

When transmit FIFO is full of transmit data (16 frames), no more data can be written. If new data is written, the data is ignored.

Write transmit data to the TDR only once, after TXI interrupt is issued when CCR0.TE is 1. And when using bytes access, write TDR[15:8] and then write TDR[7:0].

MPBT bit (Multi-processor transfer bit flag)

Selects the multi-processor bit of transmit frame.

TSYNC bit (Transmit SYNC Data Bit)

When Manchester mode and MCR.SBSEL = 1 and MCR.SYNSEL=1, the type of SYNC selected according to this bit becomes the Start Bit of the transmission frame.

33.3.4 TSR : Transmit Shift Register

TSR is a shift register that transmits serial data. TSR cannot be directly accessed by the CPU.

To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, and then sends the data to the TXDn pin.

33.3.5 CCR0 : Common Control Register 0

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	SSE	—	—	TEIE	TIE	—	—	—	RIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IDSEL	DCME	MPIE	—	—	—	TE	—	—	—	RE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RE	Receive Enable In clock synchronous mode and simple-SPI mode, receive only setting is prohibited (TE = 0 and RE = 1 setting prohibited). 0: Serial reception is disabled. 1: Serial reception is enabled.	R/W*1
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TE	Transmit Enable 0: Serial transmission is disabled. 1: Serial transmission is enabled.	R/W*1
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	MPIE	Multi-Processor Interrupt Enable (Valid in asynchronous mode and Manchester mode when CCR3.MP = 1.) This bit should set 0 in smart card interface mode. When the data with the multiprocessor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and non-multiprocessor reception is resumed. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame. (Consider the synchronization delay time.) 0: Non-Multi-Processor reception 1: Multi-Processor reception	R/W*2
9	DCME	Data Compare Match Enable (Valid only in asynchronous mode) 0: Address match function is disabled. 1: Address match function is enabled.	R/W*2
10	IDSEL	ID frame select (Valid only in asynchronous mode with multi-processor mode) 0: All data is to be compared. 1: The data with RDR.MPB = 1 is to be compared.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	RIE	Receive Interrupt Enable 0: RXI and ERI interrupt requests are disabled. 1: RXI and ERI interrupt requests are enabled.	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TIE	Transmit Interrupt Enable 0: TXI interrupt request is disabled. 1: TXI interrupt request is enabled.	R/W
21	TEIE	Transmit End Interrupt Enable This bit should set 0 in smart card interface mode. 0: TEI interrupt request is disabled. 1: TEI interrupt request is enabled.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	SSE	SSn# Pin Function Enable (Valid in Simple-SPI mode.) In slave mode (CCR3.CKE[1:0] = 1xb), set 1 to this bit. 0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In clock-synchronous mode (CCR3.MOD[2:0] = 010b), simple-SPI mode (CCR3.MOD[2:0] = 011b), and simple-I^{2}C mode (CCR3.MOD[2:0] = 100b), 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE. In other modes, writing is enabled under any condition.

Note 2. This bit is cleared by hardware. Note that unintentional setting of this bit may be occurred by read-modify-write operation.

RE bit (Receive Enable)

Enables or disables serial receive operation.

When this bit is set to 1, serial reception becomes possible after the synchronization delay time has elapsed in asynchronous mode or the synchronous clock input in clock synchronous mode or the neg-edge of RxD in manchester mode or start bit in smart-card-interface mode.

Note that CCR3 should be set prior to setting the RE bit to 1 in order to designate the reception format.

Except smart-card-interface-mode, even if reception is halted by setting the RE bit to 0, the CSR.RDRF, FER, PER, ORER, MSR.MER, SBER, SYER, PFER, FRSR.DR flags are not affected and the previous values are retained. In smart-card-interface-mode, even if reception is halted by setting the RE bit to 0, the CSR.FER, PER, ORER flags are not affected and the previous value is retained. Also, to stop the reception operation, synchronization delay time will be required from when the RE bit is set to 0 until the reception operation is stopped.

TE bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission becomes possible after the synchronization delay time has elapsed. After the synchronization delay time, transmission is started by writing transmit data to TDR. Note that CCR3 should be set prior to setting the TE bit to 1 in order to designate the transmission format. In addition, the synchronization delay time is required until the transmission control circuit is stopped after the TE bit is set to 0.

MPIE bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags (CSR.RDRF, ORER, FER, FRSR.DR, MSR.MER, SYER, PFER, SBER) are not set.

When the data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, see section 33.5. Multi-Processor Communication Function. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.

When the receive data includes the MPB bit set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER, FER, MER, SYER, PFER, and SBER to 1 is disabled.

When the receive data includes the MPB bit set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if CCR0.RIE is set to 1), and setting the flags ORER, FER, MER, SYER, PFER, and SBER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

DCME bit (Data Compare Match Enable)

It can select whether the Address match function (data compare match function) uses or not.

When DCME is 1, if SCI detects the match to the comparison data (CCR4.CMPD) with receive data, DCME is cleared automatically, and after that, SCI operation mode will be receive mode without data compare match function.

See section 33.4.6. Address Match (receive data match detection) Function.

The write value should be 0 other than asynchronous mode.

IDSEL bit (ID frame select)

It can select whether it's compared in spite of the value of MPB bit or it's compared only the data of MPB bit = 1 (ID frame) when the Address match function is valid. Please set at the same time as DCME.

RIE bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

RXI and ERI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the CSR.ORER, FER, or PER and then setting the flag to 0, or setting the RIE bit to 0.

In the case of Manchester mode, the MER, SYER, PFER, and SBER flags are also the cause of ERI interrupt request, so the same processing is necessary. For details of these flags, see section 33.3.12. MCR : Manchester Control Register and section 33.3.18. MSR : Manchester Status Register.

TIE bit (Transmit Interrupt Enable)

Enables or disables a TXI interrupt request.

A TXI interrupt request is disabled by setting the TIE bit to 0. At the beginning of transmission, set 1 to CCR0.TE and CCR0.TIE simultaneously. Then the TXI interrupt request is generated.

TEIE bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.

A TEI interrupt request is disabled by setting the TEIE bit to 0.

In simple I2C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI). In this case, the TEIE bit can be used to enable or disable the STI.

SSE bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Do not set both the SSE and CTSE (even if this setting is made, operation is the same as that when these bits are set to 0).

In the slave mode (CCR3.CKE[1:0] = 10b or 11b), SSE should be set 1.

In the master mode (CCR3.CKE[1:0] = 00b or 01b) and single-master, the SSn# pin on the master side is not required to control reception and transmission, so SSE should be set 0.

33.3.6 CCR1 : Common Control Register 1

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	NFEN	—	NFCS[2:0]			—	—	—	SHARPS	—	—	—	SPLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RINV	TINV	—	—	PM	PE	—	—	SPB2IO	SPB2DT	—	—	CTSPEN	CTSE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	CTSE	CTS Enable 0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W
1	CTSPEN	CTS external pin Enable 0: 1-pin mode, CTS and RTS share 1 pin. 1: 2-pin mode, CTS and RTS have dedicated pins.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SPB2DT	Serial port break data select The output level of TXD pin is selected when CCR0.TE = 0.*1 0: Low level is output in TXD pin, when TINV = 0. High level is output in TXD pin, when TINV = 1. 1: High level is output in TXD pin, when TINV = 0. Low level is output in TXD pin, when TINV = 1.	R/W
5	SPB2IO	Serial port break I/O This bit selects output or not output of TXD pin when the CCR0.TE = 0.*1 0: The value of SPB2DT bit is not output in TXD pin. 1: The value of SPB2DT bit is output in TXD pin.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	PE	Parity Enable (Valid only in asynchronous mode and Manchester mode. In Smart Card Interface mode, set 1 to this bit.) 0: Parity bit addition is not performed, when transmitting. Parity bit checking is not performed, when receiving. 1: The parity bit is added, when transmitting. The parity bit is checked, when receiving.	R/W
9	PM	Parity Mode (Valid only when PE = 1) 0: Selects even parity 1: Selects odd parity	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	TINV	TXD invert 0: Transmit data is not inverted and output to TXD.*2 1: Transmit data is inverted and output to TXD.	R/W
13	RINV	RXD invert 0: Received data from RXD is not inverted and input.*2 1: Received data from RXD is inverted and input.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	SPLP	Loopback Control This function is available in asynchronous mode, Manchester mode, and clock synchronous mode. 0: Normal mode 1: Loopback mode	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	SHARPS	Half-duplex communication select In the simple I2C mode or in the simple SPI mode, this bit should be set 0. 0: The TXD pin and the RXD pin independent 1: Half-duplex communication using TXD pin	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
26:24	NFCS[2:0]	Noise Filter Clock Select (Valid in asynchronous mode, Manchester mode, and simple-I2C mode.) Select for the noise filter's clock source. In simple-I2C mode, 000b setting is prohibited. "The baud rate generator source clock" means the clock selected by CCR2.CKS[1:0]. 0 0 0: The base clock signal divided by 1. 0 0 1: The baud rate generator source clock divided by 1. 0 1 0: The baud rate generator source clock divided by 2. 0 1 1: The baud rate generator source clock divided by 4. 1 0 0: The baud rate generator source clock divided by 8. Others: Setting prohibited	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	NFEN	Digital Noise Filter Function Enable (Valid in asynchronous mode, Manchester mode, and simple-I2C mode) Noise filter function is available for RXD only in asynchronous mode and Manchester mode, for SCL and SDA inputs in simple-I2C mode. 0: Noise filter disabled 1: Noise filter enabled	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. You should use this bit in asynchronous mode and Manchester mode only. Not guaranteed in other mode.
Note 2. RINV/TINV should be set 0b in smart card interface mode and simple I2C mode.

CTSE bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for CTS input. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple I2C mode. Do not set both the CTSE and SSE bits (even if this setting is made, operation is the same as that when these bits are set to 0).

CTSPEN bit (CTS external pin Enable)

When CTSE is 1, select the terminals usage method when using the CTS and RTS functions.

SPB2DT bit (Serial port break data select), SPB2IO bit (Serial port break I/O)

The TXD pin status decided by combination of CCR0.TE bit, CCR1.SPB2IO bit and CCR1.SPB2DT bit is indicated in Table 33.6.

Table 33.6 TXD pin status

CCR0.TE	CCR1.SPB2IO	CCR1.SPB2DT	TXD pin status (When CCR1.TINV = 0)
0	0	x	Hi-Z (initial value)
0	1	0	Low level output
0	1	1	High level output
1	x	x	Serial transmission data is output.

PE bit (Parity Enable)

When PE bit to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

In the multiprocessor format, the parity bit is not added or checked regardless of this bit setting.

PM bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd). In multi-processor mode, this bit is invalid.

For details on the usage of this bit in smart card interface mode, see section 33.7.2. Data Format (Except in Block Transfer Mode).

TINV bit (TXD invert), RINV bit (RXD invert)

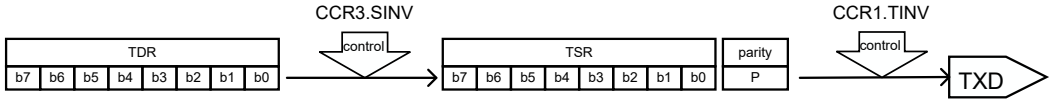
The data of RDR is controlled by RINV and CCR3.SINV. And the data from TXD pin is controlled by TINV and CCR3.SINV. The control by RINV/TINV are done to communication pins (RXD/TXD), so they can control not only data-bits but also other bits (start bit, stop bit, parity bit). For details, see Figure 33.2.

During half-duplex communication and slave operation in simple SPI mode, use the TXD pin for reception, so set the inversion control of the received data with the TINV bit.

Note: Description and timing charts in this chapter assumes CCR1.TINV = 0 and CCR1.RINV = 0 when TINV and RINV value are not specified.

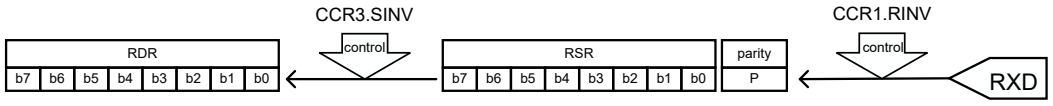
The receive/transmit data control (Data size = 8 bits, Even parity, MSB first)

The transmit data is controlled by CCR1.TINV and CCR3.SINV.



CCR3. SINV	CCR1. TINV	TDR	TSR	Parity (even)	TxD waveform
0	0	0xBE	0xBE	0	
0	1	0xBE	0xBE	0	
1	0	0xBE	0x41	0	
1	1	0xBE	0x41	0	

The received data is controlled by CCR1.RINV and CCR3.SINV.



CCR3. SINV	CCR1. TINV	RDR	RSR	parity (even)	RxD waveform
0	0	0xBE	0xBE	0	
1	0	0x41	0xBE	0	
0	1	0xBE	0xBE	0	
1	1	0x41	0xBE	0	

Figure 33.2 Example of the receive/transmit data control

SPLP bit (Loopback Control)

When SPLP = 1, TXD is connected to RXD internally.

Transmit data can be inverted and received by TINV = 1.

Set to 0 in asynchronous mode with external clock and clock synchronous mode slave.

SHARPS bit (Half-duplex communication select)

Setting this bit to 1 enables half-duplex communication using the TXD pin. However, it cannot be used in Simple-SPI mode, Simple-I2C mode, and Smart Card Interface mode.

NFCS[2:0] bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter.

To use the noise filter in asynchronous mode and Manchester mode, set these bits from 000b to 100b. In simple I2C mode, set the bits to a value in the range from 001b to 100b.

NFEN bit (Digital Noise Filter Function Enable)

This bit enables or disables the digital noise filter function. When the function is enabled, noise cancellation is applied to the RXDn input in asynchronous mode and Manchester mode, and noise cancellation is applied to the SDAn and SCLn input in simple I2C mode. In any mode other than above, set the NFEN bit to 0.

33.3.7 CCR2 : Common Control Register 2

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MDDR[7:0]								—	—	CKS[1:0]		—	—	—	BRME
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BRR[7:0]								—	ABCSE	ABCS	BGDM	—	BCP[2:0]		
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	BCP[2:0]	Base Clock Pulse Selects the number of base clock cycles in smart card interface mode. 0 0 0: 93 clock cycles (S = 93)*1 0 0 1: 128 clock cycles (S = 128)*1 0 1 0: 186 clock cycles (S = 186)*1 0 1 1: 512 clock cycles (S = 512)*1 1 0 0: 32 clock cycles (S = 32)*1 (Initial value) 1 0 1: 64 clock cycles (S = 64)*1 1 1 0: 372 clock cycles (S = 372)*1 1 1 1: 256 clock cycles (S = 256)*1	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	BGDM	Baud Rate Generator Double-Speed Mode Select Valid in asynchronous/clock-synchronous/simple-SPI/Manchester mode and CCR3.CKE[1] = 0. 0: Baud rate generator outputs the clock with single frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W
5	ABCS	Asynchronous Mode Base Clock Select (Valid only in Asynchronous mode and Manchester mode) 0: Selects 16 base clock cycles for 1-bit period. 1: Selects 8 base clock cycles for 1-bit period.	R/W
6	ABCSE	Asynchronous Mode Extended Base Clock Select (Valid only in asynchronous mode and CCR3.CKE[1] = 0) 0: Clock cycles for 1-bit period are decided with combination between CCR2.BGDM and CCR2.ABCS. 1: 6 base clock cycles for 1-bit period and doubled frequency is output from the baud rate generator.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
15:8	BRR[7:0]	Bit rate setting BRR is an 8-bit register that adjusts the bit rate.	R/W
16	BRME	Bit Rate Modulation Enable 0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
21:20	CKS[1:0]	Clock Select 0 0: PCLKSCIn clock (n = 0)*2 0 1: PCLKSCIn/4 clock (n = 1)*2 1 0: PCLKSCIn/16 clock (n = 2)*2 1 1: PCLKSCIn/64 clock (n = 3)*2	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	MDDR[7:0]	Modulation Duty setting MDDR corrects the bit rate adjusted by the BRR register.	R/W

Note 1. S is the value of S in Table 33.8.
Note 2. n is the decimal notation of the value of n in Table 33.8.

BCP[2:0] bits (Base Clock Pulse)

The BCP[2:0] bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

For details, see section 33.7.4. Receive Data Sampling Timing and Reception Margin.

BGDM bit (Baud Rate Generator Double-Speed Mode Select)

The BGDM bit is valid when the baud rate generator is selected as the clock source (CCR3.CKE[1] = 0) in asynchronous mode, Manchester mode, clock synchronous mode, simple-SPI mode. When external clock is selected (CCR3.CKE[1] = 1), set it to 0. For the clock output from the baud rate generator, either single or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.

Set this bit to 0 in modes other than asynchronous mode or Manchester mode or clock synchronous mode or simple-SPI.

ABCS bit (Asynchronous Mode Base Clock Select)

Selects the clock cycles for 1-bit period.

Set it to 0 in modes other than asynchronous mode and Manchester mode.

ABCSE bit (Asynchronous Mode Extended Base Clock Select)

The pulse number for a base clock at 1-bit period is 6 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 6 dividing frequency of the bus clock, please use this bit and set CCR2.CKS[1:0] = 00b and BRR = 0.

Set it to 0 in modes other than asynchronous mode. Even in asynchronous mode, set it to 0 when using external clock.

Table 33.7 Base clock cycle number per 1-bit

ABCSE bit	ABCS bit	BGDM bit	The base clock cycles / 1 bit	The frequency of the baud rate generator
0	0	0	16	×1
0	0	1	16	×2
0	1	0	8	×1
0	1	1	8	×2
1	x	x	6	×2

BRR[7:0] bits (Bit rate setting)

BRR is an 8-bit register that adjusts the bit rate.

SCI has independent baud rate generator control, different bit rates can be set for each. Table 33.8 shows the relationship between the setting (N) in the BRR and the bit rate (B) for asynchronous mode, multiprocessor transfer, Manchester mode, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I2C mode.

Table 33.8 Relationship between N setting in BRR and bit rate B

Mode	BGDM bit	ABCS bit	ABCSE bit	BRR[7:0] setting	Error (%)
Asynchronous, Multiprocessor transfer, Manchester	0	0	0	$N = \frac{PCLKSCIn \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \{\frac{PCLKSCIn \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
	1	0	0	$N = \frac{PCLKSCIn \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \{\frac{PCLKSCIn \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLKSCIn \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \{\frac{PCLKSCIn \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
	x	x	1*2	$N = \frac{PCLKSCIn \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error = \{\frac{PCLKSCIn \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
Clock synchronous, Simple-SPI	0	0	0	$N = \frac{PCLKSCIn \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	—
	1	0	0	$N = \frac{PCLKSCIn \times 10^6}{4 \times 2^{2n-1} \times B} - 1$	—
Smart card interface				$N = \frac{PCLKSCIn \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \{\frac{PCLKSCIn \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\} \times 100$
Simple-I2C*1				$N = \frac{PCLKSCIn \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	—

Note: B: Bit rate (bps)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLKSCIn: Operating frequency (MHz)
n and S: Determined by the settings of the CCR2 registers as listed in the table below. Please be careful about 2^{2n+1} is used in the expression for Smart card interface, 2^{2n-1} is used in other modes.
Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I2C mode satisfy the I2C standard.
Note 2. In Manchester mode, only ABCSE = 0 can be selected.

Table 33.9 Calculating widths at high and low level for SCL

Mode	SCL	Formula (result in seconds)
I^2C	Width at high level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLKSCIn \times 10^6}$
	Width at low level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLKSCIn \times 10^6}$

Table 33.10 Clock source settings

CCR2 setting	Clock source	n
CKS[1:0] = 00b	PCLKSCIn clock	0
CKS[1:0] = 01b	PCLKSCIn/4 clock	1
CKS[1:0] = 10b	PCLKSCIn/16 clock	2
CKS[1:0] = 11b	PCLKSCIn/64 clock	3

Table 33.11 Base clock settings in smart card interface mode (1 of 2)

CCR2 setting	Base clock cycles for 1-bit period	S
BCP[2:0] = 000b	93 clock cycles	93
BCP[2:0] = 001b	128 clock cycles	128
BCP[2:0] = 010b	186 clock cycles	186
BCP[2:0] = 011b	512 clock cycles	512

Table 33.11 Base clock settings in smart card interface mode (2 of 2)

CCR2 setting	Base clock cycles for 1-bit period	S
BCP[2:0] = 100b	32 clock cycles	32
BCP[2:0] = 101b	64 clock cycles	64
BCP[2:0] = 110b	372 clock cycles	372
BCP[2:0] = 111b	256 clock cycles	256

Table 33.12 lists examples of N settings in BRR in asynchronous mode and Manchester mode. Table 33.13 and Table 33.18 list the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in clock synchronous mode and simple SPI mode are listed in Table 33.15. Examples of BRR (N) settings in smart card interface mode are listed in Table 33.17. Examples of BRR (N) settings in simple I2C mode are listed in Table 33.19. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 33.7.4. Receive Data Sampling Timing and Reception Margin. Table 33.14 and Table 33.16 list the maximum bit rates with external clock input. Table 33.20 lists the minimum widths at high and low level for SCL at various bit rates.

When either the asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) is set to 1 in asynchronous mode and Manchester mode, the bit rate becomes twice that listed in Table 33.12.

When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 33.12 Examples of BRR settings for various bit rates (asynchronous mode and Manchester mode)

Bit rate (bps)	PCLKSCIn = 75 MHz			PCLKSCIn = 80 MHz			PCLKSCIn = 96 MHz			PCLKSCIn = 100 MHz			PCLKSCIn = 125 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	243	0.06	—	—	—	—	—	—	—	—	—	—	—	—
19200	0	121	0.06	0	129	0.16	0	155	0.16	0	162	-0.15	0	202	0.22
38400	0	60	0.06	0	64	0.16	0	77	0.16	0	80	0.47	0	101	-0.27
57600	0	40	-0.76	0	42	0.94	0	51	0.16	0	53	0.47	0	67	-0.27
115200	0	19	1.73	0	21	-1.36	0	25	0.16	0	26	0.47	0	33	-0.27

Note: This is an example when CCR2.ABCS = 0, CCR2.BGDM = 0 and CCR2.ABCSE = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS = 1 and BGDM = 1, the bit rate increases four times.

Table 33.13 Maximum bit rate for each operating frequency (asynchronous mode and Manchester mode) (1 of 2)

PCLKSCIn (MHz)	BGDM bit	ABCS bit	ABCSE bit	n	N	MAX baud rate (baud)
75	0	0	0	0	0	2,343,750
		1	0	0	0	4,687,500
	1	0	0	0	0	
		1	0	0	0	9,375,000
	x	x	1	0	0	12,500,000
80	0	0	0	0	0	2,500,000
		1	0	0	0	5,000,000
	1	0	0	0	0	
		1	0	0	0	10,000,000
	x	x	1	0	0	13,333,333
96	0	0	0	0	0	3,000,000
		1	0	0	0	6,000,000
	1	0	0	0	0	
		1	0	0	0	12,000,000
	x	x	1	0	0	16,000,000

Table 33.13 Maximum bit rate for each operating frequency (asynchronous mode and Manchester mode) (2 of 2)

PCLKSCIn (MHz)	BGDM bit	ABCS bit	ABCSE bit	n	N	MAX baud rate (baud)
100	0	0	0	0	0	3,125,000
		1	0	0	0	6,250,000
	1	0	0	0	0	
		1	0	0	0	12,500,000
	x	x	1	0	0	16,666,667
125	0	0	0	0	0	3,906,250
		1	0	0	0	7,812,500
	1	0	0	0	0	
		1	0	0	0	15,625,000
	x	x	1	0	0	20,833,333

Table 33.14 Maximum bit rate with external clock input (asynchronous mode)

PCLKSCIn (MHz)	External clock (MHz)	Max baud rate (baud)	
		CCR2.ABCS = 0	CCR2.ABCS = 1
75	18.75	1,171,875	2,343,750
80	20	1,250,000	2,500,000
96	24	1,500,000	3,000,000
100	25	1,562,500	3,125,000
125	31.25	1,953,125	3,906,250

Table 33.15 BRR settings for various bit rates (clock synchronous mode, simple SPI mode)

Bit rate (bps)	PCLKSCIn = 75 MHz			PCLKSCIn = 80 MHz			PCLKSCIn = 96 MHz			PCLKSCIn = 100 MHz			PCLKSCIn = 125 MHz		
	BGDM bit	n	N	BGDM bit	n	N	BGDM bit	n	N	BGDM bit	n	N	BGDM bit	n	N
2.5 M	1	0	14	0	0	7	—	—	—	0	0	9	1	0	24
4 M	—	—	—	0	0	4	0	0	5	—	—	—	—	—	—
5 M	—	—	—	0	0	3	—	—	—	0	0	4	—	—	—
7.5 M	1	0	4	—	—	—	—	—	—	—	—	—	—	—	—
8 M	—	—	—	1	0	4	0	0	2	—	—	—	—	—	—
10 M	—	—	—	1	0	3	—	—	—	1	0	4	—	—	—
12 M	—	—	—	—	—	—	0	0	1	—	—	—	—	—	—
12.5 M	1	0	2	—	—	—	—	—	—	0	0	1	1	0	4
24 M	—	—	—	—	—	—	0	0	0	—	—	—	—	—	—
18.75 M	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—
25 M	—	—	—	—	—	—	—	—	—	0	0	0	—	—	—
37.5 M	1	0	0	—	—	—	—	—	—	—	—	—	—	—	—

Note: '—' can be set, but an error over 10% will occur.

Table 33.16 Maximum bit rate with external clock input (clock synchronous mode, simple SPI mode) (1 of 2)

PCLKSCIn (MHz)	External clock (MHz)	MAX bit rate (Mbps)
75	37.50	37.50
80	20.00	20.00
96	24.00	24.00
100	25.00	25.00

Table 33.16 Maximum bit rate with external clock input (clock synchronous mode, simple SPI mode) (2 of 2)

PCLKSCIn (MHz)	External clock (MHz)	MAX bit rate (Mbps)
125	31.25	31.25

Note: 37.5 MHz is max AC spec in this LSI.

Table 33.17 BRR settings for various bit rates (smart card interface mode, n = 0, S = 372)

Bit rate (bps)	PCLKSCIn = 75 MHz			PCLKSCIn = 80 MHz			PCLKSCIn = 96 MHz			PCLKSCIn = 100 MHz			PCLKSCIn = 125 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	10	-4.54	0	10	1.82	0	12	3.39	0	13	0.01	0	17	-2.77

Table 33.18 Maximum bit rate for each operating frequency (smart card interface mode, S = 32)

PCLKSCIn (MHz)	MAX Bit rate (bps)	n	N
75	1,171,875	0	0
80	1,250,000	0	0
96	1,500,000	0	0
100	1,562,500	0	0
125	1,953,125	0	0

Table 33.19 BRR settings for various bit rates (simple I2C mode)

Bit rate (kbps)	PCLKSCIn = 75 MHz			PCLKSCIn = 80 MHz			PCLKSCIn = 96 MHz			PCLKSCIn = 100 MHz			PCLKSCIn = 125 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
100	0	22	1.90	0	24	0.00	0	29	0.00	0	30	0.81	0	38	0.16
400	0	5	-2.34	0	5	4.17	0	7	-6.25	0	7	-2.34	0	9	-2.34

Table 33.20 Minimum widths at high and low level for SCL at various bit rates (simple I2C mode)

Bit rate (kbps)	PCLKSCIn = 75 MHz			PCLKSCIn = 80 MHz			PCLKSCIn = 96 MHz			PCLKSCIn = 100 MHz			PCLKSCIn = 125 MHz		
	n	N	Minimum width of high/low level (µs)	n	N	Minimum width of high/low level (µs)	n	N	Minimum width of high/low level (µs)	n	N	Minimum width of high/low level (µs)	n	N	Minimum width of high/low level (µs)
100	0	22	4.29 / 4.91	0	24	4.38 / 5.00	0	29	4.38 / 5.00	0	30	4.34 / 4.96	0	38	4.37 / 4.99
400	0	5	1.12 / 1.28	0	5	1.05 / 1.20	0	7	1.17 / 1.33	0	7	1.12 / 1.28	0	9	1.12 / 1.28

BRME bit (Bit Rate Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by baud rate generator is evenly corrected when this function is enabled.

Set to 0 in Clock-synchronous mode, Simple-SPI mode, Smart Card Interface mode, and Manchester mode.

CKS[1:0] bits (Clock Select)

These bits select the clock source for the baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to BRR explanation.

MDDR[7:0] bits (Modulation Duty setting)

When the BRME bit is set to 1, the bit rate generated by the baud rate generator is evenly corrected according to the settings of MDDR (M/256). The relationship between the MDDR setting (M) and the bit rate (B) is given in Table 33.21.

The initial value of MDDR is 0xFF. Bit 7 in this register is fixed to 1.

Table 33.21 Relationship between MDDR setting (M) and bit rate (B) using bit rate modulation function

Mode*1	BGDM bit	ABCS bit	ABCSE bit	BRR setting	Error (%)
Asynchronous Multiprocessor transfer	0	0	0	$N = \frac{\mathrm{PCLKSCIn} \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\mathrm{Error} = \{\frac{\mathrm{PCLKSCIn} \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	1	0	0	$N = \frac{\mathrm{PCLKSCIn} \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\mathrm{Error} = \{\frac{\mathrm{PCLKSCIn} \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{\mathrm{PCLKSCIn} \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\mathrm{Error} = \{\frac{\mathrm{PCLKSCIn} \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	x	x	1	$N = \frac{\mathrm{PCLKSCIn} \times 10^6}{12 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\mathrm{Error} = \{\frac{\mathrm{PCLKSCIn} \times 10^6}{B \times 12 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
Simple-I2C*2				$N = \frac{\mathrm{PCLKSCIn} \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	

Note: B: Bit rate (bps)
M: MDDR setting (128 ≤ M ≤ 255)
N: BRR setting for bound rate generator (0 ≤ N ≤ 255)
PCLKSCIn: Operating frequency (MHz)
n and S: Determined by the settings of the CCR2.BCP[2:0] and CKS[1:0] as listed in Table 33.10 and Table 33.11.
Please be careful about 2^{2n+1} is used in the expression for Smart card interface, 2^{2n-1} is used in other modes.

Note 1. Do not use this function in Clock-synchronous mode, Simple-SPI mode, Smart card Interface mode, and Manchester mode.

Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I2C mode satisfy the I2C standard.

Table 33.22 lists examples of N settings in BRR and M settings in MDDR in asynchronous mode.

Table 33.22 Examples of BRR and MDDR settings for various bit rates (asynchronous mode)

Bit rate (bps)	PCLKSCIn = 75 MHz					PCLKSCIn = 80 MHz					PCLKSCIn = 96 MHz					PCLKSCIn = 100 MHz					PCLKSCIn = 125 MHz				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	35	151	0	0.00	0	58	232	0	0.00	0	46	154	0	-0.00	0	47	151	0	0.00	0	59	151	0	0.00
57600	0	23	151	0	0.00	0	39	236	0	0.03	0	47	236	0	0.03	0	31	151	0	0.00	0	39	151	0	0.00
115200	0	11	151	0	0.00	0	19	236	0	0.03	0	23	236	0	0.03	0	15	151	0	0.00	0	19	151	0	0.00
230400	0	5	151	0	0.00	0	9	236	0	0.03	0	11	236	0	0.03	0	7	151	0	0.00	0	9	151	0	0.00
460800	0	2	151	0	0.00	0	4	236	0	0.03	0	5	236	0	0.03	0	3	151	0	0.00	0	4	151	0	0.00

Note: In this example when CCR2.ABCS = 0 and CCR2.ABCSE = 0.
CCR2.BRME = 0 (M = 256) disables the bit rate modulation function.

33.3.8 CCR3 : Common Control Register 3

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	BLK	GM	—	—	CKE[1:0]		—	—	DEN	FM	MP	MOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDESEL	STP	SINV	LSBF	—	—	CHR[1:0]		BPEN	—	—	—	—	—	CPOL	CPHA
Value after reset:	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	CPHA	Clock Phase Select (Valid in Clock-synchronous mode and Simple-SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0.) 0: Data is sampled at an odd edge and changes at an even edge. (Clock is delayed.) 1: Data changes at an odd edge and is sampled at an even edge. (Clock is not delayed.)	R/W
1	CPOL	Clock Polarity Select (Valid in Clock-synchronous mode and Simple-SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0.) 0: SCKn in idle state is 0. 1: SCKn in idle state is 1.	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	BPEN	Synchronizer bypass enable This bit controls whether to bypass the synchronizer circuit between the bus clock and operation clock. When BPEN = 1, PCLKSPIn (operation clock) is input from PCLKM (Bus clock) and the synchronization circuit is bypassed. See section 7, Clock Generation Circuit. 0: Synchronizer circuit is not bypassed. 1: Synchronizer circuit is bypassed.	R/W
9:8	CHR[1:0]	Character Length (Valid in Asynchronous mode and Manchester mode)*1 Select the data length for transmission and reception. 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*2	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	LSBF	LSB First select Set this bit to 0 in simple I2C mode. 0: MSB first 1: LSB first	R/W
13	SINV	Transmitted/Received Data Invert The level of communication pins (RXD/TXD) are controlled by combination of this bit and CCR1.TINV/RINV. For details, see Figure 33.2. Set this bit to 0 in simple-I2C mode. 0: TDR contents are transmitted to TSR as they are. RSR contents are stored to RDR as they are. 1: TDR contents are inverted before being transmitted to TSR. RSR contents are inverted and stored to RDR.	R/W
14	STP	Stop Bit Length (Valid in Asynchronous mode and Manchester mode) 0: 1 stop bit 1: 2 stop bits	R/W

Bit	Symbol	Function	R/W
15	RXDESEL	Asynchronous Start Bit Edge Detection Select (Valid only in asynchronous mode) 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W
18:16	MOD[2:0]	Communication mode select Select the SCI communication mode. 0 0 0: Asynchronous mode (Multi-processor mode) 0 0 1: Smart card interface mode 0 1 0: Clock synchronous mode 0 1 1: Simple SPI mode 1 0 0: Simple I2C mode 1 0 1: Manchester mode Others: Setting prohibited	R/W
19	MP	Multi-Processor Mode (Valid in Asynchronous mode and Manchester mode) 0: Multi-processor communications function is disabled. 1: Multi-processor communications function is enabled.	R/W
20	FM	FIFO Mode select (Valid in Asynchronous mode (including multi-processor mode), Clock synchronous mode, Simple SPI mode) 0: Non-FIFO mode (TDR and RDR registers are non-FIFO). 1: FIFO mode (TDR and RDR registers are FIFO).	R/W
21	DEN	Driver enable (Valid only in Asynchronous mode) 0: RS-485 Driver control function disable. 1: RS-485 Driver control function enable.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	CKE[1:0]	Clock enable In the case of asynchronous mode When using the external clock 16 times the bit rate should be input from the SCKn pin when CCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the CCR2.ABCS bit is 1. 0 0: The baud rate generator The SCKn pin is available for use as an I/O port in accord with the I/O port settings. 0 1: The baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock In case of Manchester mode 0 0: On-chip baud rate generator The SCKn pin functions as I/O port Others: Prohibited In the case of Clock synchronous mode, Simple-SPI 0 x: Internal clock (Master operation) The SCKn pin functions as the clock output pin. 1 x: External clock (Slave operation) The SCKn pin functions as the clock input pin. In the case of Smart card interface mode When CCR3.GM = 0 0 0: Output disabled (The SCKn pin can be used for other peripherals.) 0 1: Clock output 1 x: Setting prohibited When CCR3.GM = 1 0 0: Output fixed low 1 0: Output fixed high x 1: Clock output	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	GM	GSM Mode (Valid only in Smart card interface mode) 0: Non-GSM mode operation 1: GSM mode operation	R/W

Bit	Symbol	Function	R/W
29	BLK	Block Transfer Mode (Valid only in Smart card interface mode) 0: Non-block transfer mode operation 1: Block transfer mode operation	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In other than asynchronous mode and Manchester mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 2. LSB first is fixed and the MSB (bit 7) in TDR is not transmitted in transmission.

CPHA bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. For details, see Figure 33.101.

Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

CPOL bit (Clock Polarity Select)

This bit selects the polarity of the clock signal output through the SCKn pin. For details, see Figure 33.101.

Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

BPEN bit (Synchronizer bypass enable)

The synchronization circuit can be bypassed by this bit only when the same clock is input to the bus clock and the operation clock (PCLKSCIn). For details, see section 33.16. Synchronizer Bypass Function.

CHR[1:0] bits (Character Length)

Selects the data length for transmission and reception.

Except of asynchronous mode and Manchester mode, a fixed data length of 8 bits is used.

LSBF bit (LSB First select)

Select whether to transmit/receive data in MSB first or LSB first.

SINV bit (Transmitted/Received Data Invert)

SINV can invert the transmit data-bit from TDR to TSR, and also can invert the received data from RSR to RDR. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the CCR1.PM.

STP bit (Stop Bit Length)

Selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

RXDESEL bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.

Set this bit to 0 in modes except of asynchronous mode.

MOD[2:0] bits (Communication mode select)

Selects the SCI communication mode.

Table 33.23 Relationship between communication mode selection bits (MOD[2:0]), other operation mode setting bits (1 of 2)

Communication mode	Asynchronous		SMIF*1	Clock synchronous	Simple SPI	Simple I2C	Manchester	
CCR3.MOD[2:0]	000b		001b	010b	011b	100b	101b	
CCR3.MP	0	1	—	—	—	—	0	1

Table 33.23 Relationship between communication mode selection bits (MOD[2:0]), other operation mode setting bits (2 of 2)

Communication mode	Asynchronous								SMIF*1	Clock synchronous		Simple SPI		Simple I2C	Manchester
CCR3.FM	0		1		0		1		—	0	1	0	1	—	—
CCR3.DEN	0	1	0	1	0	1	0	1	—	—		—		—	—
CCR0.SSE	—								—	—		0	1	—	—

Note: '— ' means setting prohibited.
Note 1. Smart card interface

MP bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

FM bit (FIFO Mode select)

When FM = 1, the TDR register / RDR register switches to FIFO mode.

DEN bit (Driver enable)

Select RS-485 Driver control function disable or enable.

CKE[1:0] bits (Clock enable)

These bits select the clock source and SCKn pin function.

In smart card interface mode, these bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, see section 33.7.8. Clock Output Control.

GM bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.

In GSM mode, the CSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, see section 33.7.6. Serial Data Transmission (Except in Block Transfer Mode) and section 33.7.8. Clock Output Control.

BLK bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.

For details, see section 33.7.3. Block Transfer Mode.

33.3.9 CCR4 : Common Control Register 4

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AET	ATT[2:0]			AJD	AST[2:0]			—	—	—	—	—	—	ATEN	ASEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CMPD[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	CMPD[8:0]	Compare Match Data (Valid only in Asynchronous mode) Set the compare data pattern for address match function.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ASEN	Adjust receive sampling timing enable (Valid in Asynchronous mode using internal clock, Clock-synchronous mode operating as master, Simple-SPI mode operating as master) 0: Adjust sampling timing disable. 1: Adjust sampling timing enable.	R/W
17	ATEN	Adjust transmit timing enable (Valid only in asynchronous mode using internal clock) 0: Adjust transmit timing disable. 1: Adjust transmit timing enable.	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
26:24	AST[2:0]	Adjustment value for receive Sampling Timing This bit enables only when ASEN is 1. In asynchronous mode using internal clock The sampling timing of RXDn pin is adjusted from the middle of bit by the following formula. Adjustment sampling timing = base clock × the setting value of AST[2:0]. In Clock-synchronous mode and Simple-SPI mode using internal clock The RxD sampling timing can be adjusted by delaying 1 to 4 PCLKSCIn. 0 0 0: 1 PCLKSCIn delay 0 0 1: 2 PCLKSCIn delay 0 1 0: 3 PCLKSCIn delay 0 1 1: 4 PCLKSCIn delay Others: Setting prohibited	R/W
27	AJD	Adjustment Direction for receive sampling timing (Valid in Asynchronous mode using internal clock) This bit enables only when ASEN is 1. Adjustment direction for RxD receive sampling timing is determined by this bit. For details, see section 33.4.10. The function of adjust receive sampling timing (Asynchronous Mode). 0: The sampling timing is adjusted backward to the middle of bit. 1: The sampling timing is adjusted forward to the middle of bit.	R/W
30:28	ATT[2:0]	Adjustment value for Transmit timing (Valid in Asynchronous mode using internal clock) This bit enables only when ATEN is 1. The selected edge timing of TxD is adjusted by the following formula. Adjustment edge timing = base clock × the setting value of ATT[2:0] This setting timing is limited by setting the base clock cycles. For details, see section 33.4.11. The function of adjust transmit timing (Asynchronous Mode).	R/W
31	AET	Adjustment edge for transmit timing (Valid in Asynchronous mode using internal clock) The adjustable edge is set by this bit. This bit enables only when ATEN = 1. For details, see section 33.4.11. The function of adjust transmit timing (Asynchronous Mode). 0: Adjust the rising edge timing, when CCR1.TINV = 0. Adjust the falling edge timing, when CCR1.TINV = 1. 1: Adjust the falling edge timing, when CCR1.TINV = 0. Adjust the rising edge timing, when CCR1.TINV = 1.	R/W

Note: If the description in this document and the timing chart do not specify the ASEN / ATEN setting value, it means that the reception sampling adjustment function / transmission timing adjustment function are OFF (CCR4.ASEN = 0, CCR4.ATEN = 0).

CMPD[8:0] bits (Compare Match Data)

Set the comparison data for receive data, when Address match function is enabled (CCR0.DCME = 1). CCR4.CMPD[8:0] should be written while CCR0.DCME is 0.

For the comparison data, it can select length from 3 types, they are CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.

ASEN bit (Adjust receive sampling timing enable)

When this bit is 1, the receive sampling timing adjustment function is enabled. Control is different in Asynchronous mode, Clock-synchronous mode, and Simple-SPI mode.

In Asynchronous mode using internal clock, see section 33.4.10. The function of adjust receive sampling timing (Asynchronous Mode) in details.

In Clock-synchronous mode as master, Simple-SPI mode operating as master, see section 33.9.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used in details. Only the digital delay of the master mode receive sampling clock (MRCLK) can be controlled by this bit.

ATEN bit (Adjust transmit timing enable)

When this bit is 1, the transmission timing adjustment function is enabled. The transmission timing adjustment function can adjust the edge timing of the waveform output from the TXD pin. See section 33.4.11. The function of adjust transmit timing (Asynchronous Mode) in details.

AST[2:0] bits (Adjustment value for receive Sampling Timing)

When ASEN = 1, the receive sampling timing can be adjusted according to this bit setting value.

In Asynchronous mode using internal clock.

The sampling timing of RXD pin is adjusted from the middle of bit by the following formula. This setting value is limited by setting the base clock cycles. For details, see section 33.4.10. The function of adjust receive sampling timing (Asynchronous Mode).

Adjustment sampling timing = base clock × the setting value of AST[2:0].

In Clock-synchronous mode and Simple-SPI mode using internal clock

The RxD sampling timing can be adjusted by delaying 1 to 4 PCLKSCIn. For details, see section 33.9.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used.

000: 1 PCLKSCIn delay

001: 2 PCLKSCIn delay

010: 3 PCLKSCIn delay

011: 4 PCLKSCIn delay

1xx: Setting prohibited

AJD bit (Adjustment Direction for receive sampling timing)

Set the RXD pin sampling timing adjustment direction from the bit center to the rear or front. See section 33.4.10. The function of adjust receive sampling timing (Asynchronous Mode) for details.

ATT[2:0] bits (Adjustment value for Transmit timing)

The edge timing of the TXD pin specified by the AET bit is adjusted by the base clock × ATT[2:0] setting value. The upper limit of the adjustment time that can be set is limited by the number of base clock cycles. See section 33.4.11. The function of adjust transmit timing (Asynchronous Mode) for details.

AET bit (Adjustment edge for transmit timing)

Set the TXD pin edge for timing adjustment. See section 33.4.11. The function of adjust transmit timing (Asynchronous Mode) for details.

33.3.10 ICR : Simple I2C Control Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	IICSCLS[1:0]		IICSDAS[1:0]		—	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IICACKT	—	—	—	IICCSC	IICINTM	—	—	—	IICDL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	IICDL[4:0]	SDA Delay Output Select (Cycles below are of the clock signal from the baud rate generator.) 0x00: No output delay 0x01: 0 to 1 cycle 0x02: 1 to 2 cycles 0x03: 2 to 3 cycles 0x04: 3 to 4 cycles 0x05: 4 to 5 cycles ⋮ 0x1E: 29 to 30 cycles 0x1F: 30 to 31 cycles	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	IICINTM	I2C Interrupt Mode Select 0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts.	R/W
9	IICCSC	Clock Synchronization 0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
13	IICACKT	ACK Transmission Data 0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	IICSTAREQ	Start Condition Generation 0: A start condition is not generated. 1: A start condition is generated. *1 *3 *4 *5	R/W
17	IICRSTAREQ	Restart Condition Generation 0: A restart condition is not generated. 1: A restart condition is generated. *2 *3 *4 *5	R/W
18	IICSTPREQ	Stop Condition Generation 0: A stop condition is not generated. 1: A stop condition is generated. *2 *3 *4 *5	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
21:20	IICSDAS[1:0]	SDA Output Select 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SDAn pin. 1 1: Place the SDAn pin in the high-impedance state.	R/W

Bit	Symbol	Function	R/W
23:22	IICSCLS[1:0]	SCL Output Select 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SCLn pin. 1 1: Place the SCLn pin in the high-impedance state.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In the bus free state, perform the start condition generation.
Note 2. In the bus busy state, perform restart or stop condition generation when the SCLn pin after acknowledgment described in section 33.8. Simple I2C mode in Figure 33.71 and Figure 33.72 is low level.
Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

IICDL[4:0] bits (SDA Delay Output Select)

These bits are used to set a delay for output on the SDAn pin relative to the falling edge of the output on the SCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the baud rate generator as the base. The signal obtained by frequency-dividing PCLKSCIn by the divisor set in CCR2.CKS[1:0] is supplied as the clock signal from the baud rate generator.

Set these bits to 0x00 unless operation is in simple I2C mode. In simple I2C mode, set these bits from 0x01 to 0x1F.

IICINTM bit (I2C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I2C mode.

IICCSC bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SCLn clock signal is to be synchronized when the SCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.

The SCLn clock signal is not synchronized if the IICCSC bit is 0. The SCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

IICSTAREQ bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

If you want to generate the start condition after generating the stop condition, start the generation of the start condition with a half cycle period of the bit rate from the stop condition generation interrupt request output.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

IICRSTAREQ bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of stop condition

IICSDAS[1:0] bits (SDA Output Select)

These bits control output from the SDAn pin.

IICSCLS[1:0] bits (SCL Output Select)

These bits control output from the SCLn pin.

33.3.11 FCR : FIFO Control Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	RSTRG[4:0]					RFRST	—	—	RTRG[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TFRST	—	—	TTRG[4:0]					—	—	—	—	—	—	—	DRES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DRES	Receive data ready error select (Valid in Asynchronous mode) This bit select the interrupt request for a reception data ready detection. 0: reception data full interrupt (RXI) 1: receive error interrupt (ERI)	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
12:8	TTRG[4:0]	Transmit FIFO data trigger number (Valid in Asynchronous mode (including multi-processor mode), Clock-synchronous mode, Simple-SPI mode) Trigger number must be set 15 or less. When DMAC transfer is used, set TTRG[4:0] = 0x0F. 0x00: Trigger number 0 ⋮ 0x0F: Trigger number 15 Others: Setting prohibited	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	TFRST	Transmit FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in transmit FIFO (TDR register) are made 0.	W

Bit	Symbol	Function	R/W
20:16	RTRG[4:0]	Receive FIFO data trigger number (Valid in Asynchronous mode (including multi-processor mode), Clock-synchronous mode, Simple-SPI mode) Trigger number must be set 15 or less. When DMAC transfer is used, set RTRG[4:0] = 0x00. 0x00: Trigger number 0 ⋮ 0x0F: Trigger number 15 Others: Setting prohibited	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	RFRST	Receive FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in receive FIFO (RDR register) are made 0.	W
28:24	RSTRG[4:0]	RTS# Output Active Trigger Number Select (Valid in Asynchronous mode (including multi-processor mode), Clock-synchronous mode) These bits enable only when CCR3.FM = 1 and CCR1.CTSE = 0 and CCR0.SSE = 0. Trigger number must be set 15 or less. 0x00: Trigger number 0 ⋮ 0x0F: Trigger number 15 Others: Setting prohibited	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

DRES bit (Receive data ready error select)

Select whether the detection of receive data ready (FRSR.DR = 1) is the cause of RXI interrupt request or the cause of ERI interrupt request.

TTRG[4:0] bits (Transmit FIFO data trigger number)

The TDRE flag is set to 1 when the quantity of transmit data in the transmit FIFO (TDR register) is equal to or less than the specified transmit triggering number. If CCR0.TIE = 1, TXI interrupt request is occurred.

Note: In case the trigger number is set 16 or more, unexpected TXI interrupt will occur.

TFRST bit (Transmit FIFO Data Register Reset)

When the TFRST bit is set to 1, the number of the transmission data stored in transmit FIFO (TDR register) is made 0.

RTRG[4:0] bits (Receive FIFO data trigger number)

The CSR.RDRF flag is set to 1 when the quantity of receive data in the receive FIFO (RDR register) is equal to or greater than the specified receive triggering number. If CCR0.RIE = 1, RXI interrupt request is occurred. When FCR.RTRG is set to 0, RDRF bit is set if the quantity of data in receive FIFO is greater than or equal to 1.

Note: In case the trigger number is set 16 or more, unexpected RXI interrupt will occur.

RFRST bit (Receive FIFO Data Register Reset)

When the RFRST bit is set to 1, the number of the reception data stored in receive FIFO (RDR register) is made 0.

RSTRG[4:0] bits (RTS# Output Active Trigger Number Select)

When the quantity of receive data stored in the receive FIFO (RDR register) is equal to or greater than this number, the RTS# signal is in the High state. When FCR.RSTRG is set to 0, RTS# is in the high state if the quantity of data in receive FIFO is greater than or equal to 1.

Note: In case the trigger number is set 16 or more, RTS# will go to High state at unexpected timing.

33.3.12 MCR : Manchester Control Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SBER EN	SYER EN	PFER EN	—	—	RPPAT[1:0]		RPLEN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TPPAT[1:0]		TPLEN[3:0]				—	SBSE L	SYNS EL	SYNV AL	—	ERTE N	TMPO L	RMPO L
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMPOL	Polarity of Received Manchester Code Sets the polarity of the received Manchester code 0: Logic 0 is coded as a zero-to-one transition in Manchester code Logic 1 is coded as a one-to-zero transition in Manchester code 1: Logic 0 is coded as a one-to-zero transition in Manchester code Logic 1 is coded as a zero-to-one transition in Manchester code	R/W
1	TMPOL	Polarity of Transmit Manchester Code Sets the polarity of the transmit Manchester code 0: Logic 0 is coded as a zero-to-one transition in Manchester code Logic 1 is coded as a one-to-zero transition in Manchester code 1: Logic 0 is coded as a one-to-zero transition in Manchester code Logic 1 is coded as a zero-to-one transition in Manchester code	R/W
2	ERTEN	Manchester Edge Retiming Enable Sets the receive retiming function 0: Disables the receive retiming function 1: Enables the receive retiming function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	SYNVAL	SYNC value Setting Sets the SYNC type of the start bit(s) in the Manchester code When the start bit area consists of one bit. (SBSEL = 0) ● when transmitting 0: The start bit is added as a zero-to-one transition. 1: The start bit is added as a one-to-zero transition. ● when receiving 0: Only when the start bit is a zero-to-one transition, the data is received. The other cases are judged as an error. 1: Only when the start bit is a one-to-zero transition, the data is received. The other cases are judged as an error. When the start bit area consists of three bits. (SBSEL = 1) ● when transmitting 0: The start bits are added as a zero-to-one transition. (DATA SYNC) 1: The start bits are coded as a one-to-zero transition. (COMMAND SYNC) ● when receiving When the start bit area consists of three bits, data is received regardless of the value of this bit.	R/W
5	SYNSEL	SYNC Select 0: The start bit pattern is set with the SYNVAL bit. 1: The start bit pattern is set with the TSYNC bit.	R/W
6	SBSEL	Start Bit Select 0: The start bit area consists of one bit. 1: The start bit area consists of three bits. (COMMAND SYNC or DATA SYNC)	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
11:8	TPLEN[3:0]	Transmit preface length Set the preface length of the transmit data in Manchester mode 0x0: Disables the transmit preface generation Others: Transmit preface length (bit length)	R/W
13:12	TPPAT[1:0]	Transmit preface pattern Set the preface pattern of the transmit data 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RPLEN[3:0]	Receive Preface Length Set the preface length in received frames when Manchester mode is enabled 0x0: Disables the receive preface generation Others: Receive preface length (bit length)	R/W
21:20	RPPAT[1:0]	Receive Preface Pattern Set the preface pattern of the received frames 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	PFEREN	Preface Error Enable Specifies whether to handle a preface error as an interrupt source 0: Does not handle a preface error as an interrupt source 1: Handles a preface error as an interrupt source	R/W
25	SYEREN	Receive SYNC Error Enable Specifies whether to handle a receive SYNC error as an interrupt source 0: Does not handle a receive SYNC error as an interrupt source 1: Handles a receive SYNC error as an interrupt source	R/W
26	SBEREN	Start Bit Error Enable Specifies whether to handle a start bit error as an interrupt source 0: Does not handle a start bit error as an interrupt source 1: Handles a start bit error as an interrupt source	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

RMPOL bit (Polarity of Received Manchester Code)

This bit sets the polarity of the received Manchester code. For details, see section 33.6.7. Serial Data Reception in Manchester mode.

TMPOL bit (Polarity of Transmit Manchester Code)

This bit sets the polarity of the transmit Manchester code. For details, see section 33.6.6. Serial Data Transmission in Manchester Mode.

ERTEN bit (Manchester Edge Retiming Enable)

This bit sets the receive retiming function in Manchester mode.

For information on the receive retiming function, see section 33.6.9. Receive Retiming.

SYNVAL bit (SYNC value Setting)

This bit is valid when the SYNSEL bit of this register is set to 0.

The SYNC type can be set by combining this bit and the SBSEL bit.

For the start bit area determined by the combination of this bit and the SBSEL bit, see Figure 33.44 and Figure 33.45.

SYNSEL bit (SYNC Select)

This bit is valid when the SBSEL bit of this register is set to 1. This bit determines the destination to be referred to for setting the SYNC type of the start bit area added to Manchester frames.

When this bit is set to 0, the SYNVAL bit of this register is referred to.

When this bit is set to 1, the TSYNC bit in the TDR register is referred to.

SBSEL bit (Start Bit Select)

This bit sets the start bit area in Manchester frames.

When this bit is set to 1, the start bit area added to each frame consists of three bits, and the SYNSEL and SYNVAL bits in this register are valid.

When this bit is set to 0, the start bit area added to each frame consists of one bit.

TPLEN[3:0] bits (Transmit preface length)

These bits set the preface bit length of the transmit data in Manchester mode.

The settable range is 0x0 to 0xF (0d to 15d). 0x0 disables the transmit preface, which is not added.

TPPAT[1:0] bits (Transmit preface pattern)

These bits set one of the four preface patterns in Manchester mode. For the transmit data when the TPPAT bits are set, see Figure 33.43.

When these bits are set to 00b, the preface area is set to all zeros.

When these bits are set to 01b, the preface area is set to the zero-one-zero-one pattern.

When these bits are set to 10b, the preface area is set to the one-zero-one-zero pattern.

When these bits are set to 11b, the preface area is set to all ones.

RPLEN[3:0] bits (Receive Preface Length)

These bits set the preface bit length of the received frames in Manchester mode.

The settable range is 0x0 to 0xF (0d to 15d). 0x0 disables the receive preface, which is not added. When 0x1 to 0xF is set, the set value is handled as the receive preface bit length.

RPPAT[1:0] bit (Receive Preface Pattern)

These bits set one of the four preface patterns in Manchester mode. For the transmit and receive data when the TPPAT bits are set, see Figure 33.43.

When these bits are set to 00b, the preface area is handled as all zeros.

When these bits are set to 01b, the preface area is handled as the zero-one-zero-one pattern.

When these bits are set to 10b, the preface area is handled as the one-zero-one-zero pattern.

When these bits are set to 11b, the preface area is handled as all ones.

PFEREN bit (Preface Error Enable)

This bit specifies whether to handle a preface error as an interrupt source.

When it is set to 0, a preface error is not handled as an interrupt source. When it is set to 1, a preface error is handled as an interrupt source.

SYEREN bit (Receive SYNC Error Enable)

This bit specifies whether to handle a receive SYNC error as an interrupt source.

When it is set to 0, a receive SYNC error is not handled as an interrupt source. When it is set to 1, a receive SYNC error is handled as an interrupt source.

SBEREN bit (Start Bit Error Enable)

This bit specifies whether to handle a start bit error as an interrupt source.

When it is set to 0, a start bit error is not handled as an interrupt source. When it is set to 1, a start bit error is handled as an interrupt source.

33.3.13 DCR : Driver Control Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	DENGT[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DEAST[4:0]					—	—	—	—	—	—	—	DEPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DEPOL	Driver effective polarity select (Valid only in Asynchronous mode) 0: The DE signal is active high. 1: The DE signal is active low.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DEAST[4:0]	Driver Assertion Time (Valid only in Asynchronous mode) Set the driver assertion time. When CCR3.DEN = 1, the following driver assertion time is inserted in addition to the normal transmission waiting time. Driver assertion time = DEAST[4:0] set value × base clock period Setting DEAST[4:0] = 0x00 is prohibited.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	DENGT[4:0]	Driver negate time (Valid only in Asynchronous mode) Set the driver negation time. When CCR3.DEN = 1, the following driver negate time is inserted after STOP bit transmission end. Driver negate time = DENGT[4:0] set value × base clock period Setting DENGT[4:0] = 0x00 is prohibited.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

DEPOL bit (Driver effective polarity select)

Select the active level of the DE signal.

DEAST[4:0] bits (Driver Assertion Time)

Set the driver assertion time (= time from the activation of the DE (Driver Enable) signal to the start of the start bit). It is expressed in base clock units (1/6, 1/8, or 1/16 bit period).

DENGT[4:0] bits (Driver negate time)

Set the driver negation time (= time from the end of the last stop bit of the transmitted message until the DE (Driver Enable) signal is disabled). It is expressed in base clock units (1/6, 1/8, or 1/16 bit period).

If the transmission data is written during the driver negate time, transmit starting operation is different depends on the writing timing. (The DE signal remains valid, and transmission of the start bit may start after the transmission wait time has elapsed. Also, the DE signal may become invalid once, and start bit transmission may start after the DEAST time + transmission wait time has elapsed.)

33.3.14 CSR : Common Status Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x48

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRF	TEND	TDRE	FER	PER	MFF	—	ORER	—	—	—	—	—	DFER	DPER	DCMF
Value after reset:	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDMON	—	—	—	—	—	—	—	—	—	—	ERS	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
4	ERS	Error Signal Status Flag (Valid only in Smart card interface mode) 0: Error signal Low not responded 1: Error signal Low responded	R
14:5	—	These bits are read as 0.	R
15	RXDMON	Serial input data monitor The state of the RXD pin without synchronizing by bus clock is shown. 0: RXD pin is the Low level, when CCR1.RINV = 0. RXD pin is the High level, when CCR1.RINV = 1. 1: RXD pin is the High level, when CCR1.RINV = 0. RXD pin is the Low level, when CCR1.RINV = 1.	R
16	DCMF	Data Compare Match Flag (Valid only in Asynchronous mode) 0: No matched 1: Matched	R
17	DPER	Data Compare Match Parity Error Flag (Valid only in Asynchronous mode) 0: No parity error occurred at address match detection. 1: A parity error has occurred at address match detection.	R
18	DFER	Data Compare Match Framing Error Flag (Valid only in Asynchronous mode) 0: No framing error occurred at address match detection. 1: A framing error has occurred at address match detection.	R
23:19	—	These bits are read as 0.	R
24	ORER	Overrun Error Flag 0: No overrun error occurred. 1: An overrun error has occurred.	R
25	—	This bit is read as 0.	R
26	MFF	Mode Fault Error Flag (Valid only in Simple-SPI mode) 0: No mode fault error 1: Mode fault error	R
27	PER	Parity Error Flag 0: No parity error occurred (Non-FIFO mode) No parity error in all received data in receive FIFO (FIFO mode) 1: A parity error has occurred (Non-FIFO mode) One or more parity errors occurred in received data in receive FIFO (FIFO mode)	R

Bit	Symbol	Function	R/W
28	FER	Framing Error Flag 0: No framing error occurred (Non-FIFO mode) No framing error in all received data in receive FIFO (FIFO mode) 1: A framing error has occurred (Non-FIFO mode) One or more framing errors occurred in received data in receive FIFO (FIFO mode)	R
29	TDRE	Transmit Data Empty Flag 0: Transmit data is in TDR register (Non-FIFO mode) The quantity of transmit data written in transmit FIFO exceeds the specified transmit triggering number. (FIFO mode) 1: No transmit data is in TDR register (Non-FIFO mode) The quantity of transmit data written in transmit FIFO is equal to or less than the specified transmit triggering number. (FIFO mode)	R
30	TEND	Transmit End Flag 0: A character is being transmitted or standing by for transmission. 1: Character transfer has been completed.	R
31	RDRF	Receive Data Full Flag 0: No received data is in RDR register (Non-FIFO mode) The quantity of receive data written in receive FIFO falls below the specified receive triggering number (FIFO mode) 1: Received data is in RDR register (Non-FIFO mode) The quantity of receive data written in receive FIFO is equal to or greater than the specified receive triggering number (FIFO mode)	R

ERS bits (Error Signal Status Flag)

[Setting condition]

- When an error signal LOW is sampled.

[Clearing condition]

- When write 1 to CFCLR.ERSC.

DCMF bit (Data Compare Match Flag)

Indicates that SCI detects the match to the comparison data (CCR4.CMPD) with receive data.

Clearing the CCR0.RE bit to 0 does not affect the DCMF flag, which retains its previous state.

[Setting condition]

- Matched to the comparison data (CCR4.CMPD) with receive data, while CCR0.DCME = 1.

[Clearing condition]

- When write 1 to CFCLR.DCMFC.

DPER bit (Data Compare Match Parity Error Flag)

It indicates that a parity error occurred at Address Match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DPER flag, which retains its previous state.

[Setting condition]

- When a parity error was detected by the frame in which an address match was detected.

[Clearing condition]

- When write 1 to CFCLR.DPERC.

DFER bit (Data Compare Match Framing Error Flag)

It indicates that a framing error occurred at Address Match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DFER flag, which retains its previous state.

[Setting condition]

- When a stop bit of the frame in which an Address Match was detected is 0.
  When it is a 2-stop mode, the 1st bit of stop bit judges only whether it's 1 and does not check the 2nd bit of stop bit.

[Clearing condition]

- When write 1 to CFCLR.DFERC.

ORER bit (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the ORER flag, which retains its previous state. In simple-I2C mode, this bit is not use.

[Setting condition with Non-FIFO mode (CCR3.FM = 0)]

- When the next data is received before reading out RDR with no-error reception data stored in RDR.
  In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, reception data is not forwarded to RDR register.
  Note that, in clock synchronous mode and simple-SPI mode, serial reception will be stop.

[Setting condition with FIFO mode (CCR3.FM = 1)]

- When the next serial reception is completed while the receive FIFO is full of N_{FIFO} (= 16) receive data.
  (N_{FIFO}: Indicates the number of available FIFO stages)

[Clearing condition]

- When write 1 to CFCLR.ORERC.

MFF bit (Mode Fault Error Flag)

This bit indicates mode fault errors. In a multi-master mode, determine the mode fault error occurrence by reading the MFF flag.

[Setting condition]

- Input on the SSn# pin being at the low level during master operation (CCR3.CKE[1:0] = 00b or 01b) in simple SPI mode.

[Clearing condition]

- When write 1 to CFCLR.MFFC.

PER bit (Parity Error Flag)

Indicates that a parity error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the PER flag, which retains its previous state.

When performing parity check in asynchronous mode, set the address match function disabled (CCR0.DCME = 0). In

Clock-synchronous mode, Simple-SPI mode, and Simple-I2C mode, this bit not used.

[Setting condition]

- When a parity error is detected during reception.
  In FIFO select mode, when one or more parity error is detected in receive FIFO data.
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When write 1 to CFCLR.PERC.

FER bit (Framing Error Flag)

Indicates that a framing error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the FER flag, which retains its previous state.

When performing framing check in asynchronous mode, set the address match function disabled (CCR0.DCME = 0).

In Clock-synchronous mode, Simple-SPI mode, and Simple-I2C mode, this bit not used.

[Setting condition]

- When 0 is sampled as the stop bit during reception.
  In FIFO select mode, when one or more framing error is detected in receive FIFO data.
  In Manchester mode, when both sampling results (1⁄4 and 3⁄4 sampling points) are not 1 for 1 stop bit.
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to RDR, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When write 1 to CFCLR.FERC.

TDRE bit (Transmit Data Empty Flag)

[Non-FIFO selected (CCR3.FM = 0)]

Indicates the presence of transmit data in the TDR register.

The condition of CCR0.TE = 0 has priority over the condition of 0.

If other conditions that become 1 and conditions that become 0 are satisfied at the same time, the TDRE flag is set to 0.

[Setting condition]

- When CCR0.TE is 0.
- When data is transmitted from the TDR register to the TSR register.

[Clearing condition]

- When write 1 to CFCLR.TDREC.
- When the transmission data is written to the TDR register during CCR0.TE = 1.

[FIFO selected (CCR3.FM = 1)]

- Indicates that data has been transferred from the transmit FIFO (TDR) into the transmit shift register (TSR), the quantity of data in transmit FIFO has fallen below the specified transmit triggering number.
  When the condition which becomes 1 and the condition which becomes 0 were formed at the same time, TDRE flag will be 0. After that, when the number of data stored in transmit FIFO is judged, and if that is equal to or less than TTRG value, TDRE is set to 1 after 1 PCLKM.

[Setting condition]

- When the quantity of transmit data written in transmit FIFO is equal to or less than the specified transmit triggering number.[*1]

[Clearing condition]

- When write 1 to CFCLR.TDREC.
- When the transmission data is written to transmit FIFO by the DMAC (except ch5).

Note 1. The transmit FIFO is 16-stage FIFO, the maximum number of data that can be written when the TDRE flag is 1 is "N_{FIFO} (= 16) - FTSR.T[5:0]". Even if data any more is written, data is discarded.

TEND bit (Transmit End Flag)

[Non-FIFO selected (CCR3.FM = 0), and Not Smart card interface mode (CCR3.MOD[2:0] ≠ 001b)]

Indicates completion of transmission.

[Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at transmission of the tail-end bit of a character.

- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When write 1 to CFCLR.TDREC during CCR0.TE = 1.

[Non-FIFO selected (CCR3.FM = 0), and Smart card interface mode (CCR3.MOD[2:0] = 001b)]

With no error signal from the receiving side, this bit is set to 1 when next data is ready to be transferred to the TDR register.

[Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When ERS flag is 0 and TDR register is not updated after 1-byte transmission + fixed time, the set timing is determined by setting CCR3 register as listed below.
  - When GM = 0 and BLK = 0, 12.5 etu after the start of transmission
  - When GM = 0 and BLK = 1, 11.5 etu after the start of transmission
  - When GM = 1 and BLK = 0, 11.0 etu after the start of transmission
  - When GM = 1 and BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When write 1 to CFCLR.TDREC during CCR0.TE = 1.

[FIFO selected (CCR3.FM = 1)]

Indicates that the transmit FIFO does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- TEND is set to 1 when transmit FIFO does not contain transmit data when the last bit of a one-byte serial character is transmitted.
- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.

RDRF bit (Receive Data Full Flag)

[Non-FIFO selected (CCR3.FM = 0)]

Indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing condition]

- When write 1 to CFCLR.RDRFC.
- When a data is read from the RDR register.

[FIFO selected (CCR3.FM = 1)]

Indicates that receive data has been transferred to the receive FIFO data register (RDR), and the quantity of data in receive FIFO is equal to or greater than the specified receive triggering number. When FCR.RTRG is set to 0, RDRF is set if receive FIFO is not empty.

[Setting condition]

RDRF is set to 1 when the quantity of receive data in receive FIFO is equal to or greater than the specified receive triggering number.*1

[Clearing condition]

- When write 1 to CFCLR.RDRFC.
- When the reception data is read from receive FIFO by the DMAC (except ch5).
  When setting condition and clearing condition are occurred at the same time, RDRF flag will be 0. After that, if the data count in receive FIFO is equal to or greater than RTRG value, RDRF is set to 1 after 1 PCLKM.

Note 1. Since the receive FIFO is N_{FIFO} (= 16) stage, the maximum quantity of data that can be read when RDRF is 1 is equal to FRSR.R[5:0]. Further data can be read, but the value is undefined.

Note: Except when interrupting communication, RDRF and TDRE should not be cleared by CFCLR register.

33.3.15 ISR : Simple I2C Status Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x4C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIF	—	—	IICACKR
Value after reset:	0	0	0	0	0	0	0	0	0	0	x	x	0	x	0	0

Bit	Symbol	Function	R/W
0	IICACKR	ACK Reception Data Flag 0: ACK received 1: NACK received	R
1	—	This bit is read as 0.	R
2	—	The read value is undefined.	R
3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag 0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R
4	—	The read value is undefined.	R
5	—	The read value is undefined.	R
31:6	—	These bits are read as 0.	R

IICACKR bit (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACKR flag is updated at the rising of SCLn clock for the ACK/NACK receiving bit.

IICSTIF bit (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ, do it after clearing the IICSTIF flag.

When the IICSTIF flag is 1 while CCR0.TEIE = 1, an STI request is output.

[Setting condition]

Completion of the generation of a start, restart, or stop condition (when setting condition and clearing condition are occurred at the same time, IICSTIF will be 0).

[Clearing condition]

- Writing 1 to ICFCLR.IICSTIFC bit
- When operation is not in simple-I2C
- Writing 0 to CCR0.TE bit

33.3.16 FRSR : FIFO Receive Status Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	FNUM[5:0]						—	—	PNUM[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	R[5:0]						—	—	—	—	—	—	—	DR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DR	Receive Data Ready flag 0: Receiving is in progress, or no received data has remained in receive FIFO after normally completed receiving. (Receive FIFO is empty.) 1: The following receive data does not come for a fixed period after storing data below the threshold.	R
7:1	—	These bits are read as 0.	R
13:8	R[5:0]	Receive FIFO Data Count (Valid in Asynchronous mode (including multi-processor), Clock synchronous mode, Simple-SPI mode, when CCR3.FM is 1.) Indicate the quantity of data in receive FIFO.	R
15:14	—	These bits are read as 0.	R
21:16	PNUM[5:0]	Parity Error Count (Valid only in Asynchronous mode) Indicates the quantity of data with a parity error among the data in receive FIFO.	R
23:22	—	These bits are read as 0.	R
29:24	FNUM[5:0]	Framing Error Count (Valid only in Asynchronous mode) Indicates the quantity of data with a framing error among the data in receive FIFO.	R
31:30	—	These bits are read as 0.	R

DR bit (Receive Data Ready flag)

Indicates that the quantity of data stored in the receive FIFO falls below the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in asynchronous mode. This bit is valid only asynchronous mode (including multi-processor mode) and FIFO selected. In other modes, this bit does not set to 1.

[Setting conditions]

DR is set to 1 when both of the following conditions are met.

- Data in receive FIFO is less than the specified receive triggering number, and no next data is received after the elapse of 15 etu[*1] from the last stop bit.
- CSR.FER, PER flags are 0.

[Clearing conditions]

- When all receive data in the receive FIFO is read and 1 is written to FFCLR.DRC.
- When CCR3.FM bit is 0.

Note 1. This is equivalent to one and a half (1.5) frames in the 8-bit format with one stop bit (etu: elementary time unit).

R[5:0] bits (Receive FIFO Data Count)

Indicate the quantity of receive data stored in receive FIFO.

0x00 means no receive data. N_{FIFO} (= 16) means receive FIFO is full.

PNUM[5:0] bits (Parity Error Count)

Indicate the quantity of data with parity error in the received FIFO.

FNUM[5:0] bits (Framing Error Count)

Indicate the quantity of data with framing error in the received FIFO.

33.3.17 FTSR : FIFO Transmit Status Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x54

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	T[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	T[5:0]	Transmit FIFO Data Count (Valid in Asynchronous mode (including multi-processor), Clock synchronous mode, Simple-SPI mode, when CCR3.FM is 1.) Indicate the quantity of non-transmitted data stored in transmit FIFO.	R
31:6	—	These bits are read as 0.	R

T[5:0] bits (Transmit FIFO Data Count)

Indicate the quantity of non-transmitted data stored in transmit FIFO.

0x00 means no un-transmit data. N_{FIFO} (= 16) means transmit FIFO is full.

33.3.18 MSR : Manchester Status Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x58

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RSYNC	—	MER	—	SBER	SYER	PFER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PFER	Preface Error flag This bit is set when a preface error (pattern mismatch) is detected 0: No preface error detected 1: Preface error detected	R
1	SYER	SYNC Error flag This bit is set when no edge is detected in the adjustable range during receive retiming 0: No receive SYNC error detected 1: Receive SYNC error detected	R
2	SBER	Start Bit Error flag This bit is set when a pattern mismatch in the start bit area is detected 0: No start bit error detected 1: Start bit error detected	R
3	—	This bit is read as 0.	R
4	MER	Manchester Error Flag Valid for Manchester mode only 0: No Manchester error occurred 1: Manchester error has occurred	R
5	—	This bit is read as 0.	R
6	RSYNC	Receive SYNC data bit It is valid when MCR.SBSEL = 1 in Manchester mode, 0 is read otherwise. 0: The received the Start Bit is DATA SYNC 1: The received the Start Bit is COMMAND SYNC	R
31:7	—	These bits are read as 0.	R

PFER bit (Preface Error flag)

This bit indicates that a preface error was detected when receiving frames in Manchester mode.

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the PFER flag is not affected and retains its previous value.

[Setting condition]

When detecting a preface error when receiving frames in Manchester mode.

The following operations are performed when a preface error occurs.

<When MCR.PFEREN = 1>

The received data is not transferred to the RDR register and no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the PFER flag is being set to 1, the subsequently received data is not transferred to the RDR register.

<When MCR.PFEREN = 0>

The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the PFER flag being set to 1.

[Clearing condition]

Write 1 to MFCLR.PFERC.

SYER bit (SYNC Error flag)

This bit indicates that a receive SYNC error was detected when receiving frames in Manchester mode with MCR.ERTEN = 1 (Manchester edge retiming enabled).

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the SYER flag is not affected and retains its previous value.

[Setting condition]

When detecting a receive SYNC error when receiving frames in Manchester mode.

The following operations are performed when a receive SYNC error occurs.

<When MCR.SYEREN = 1>

Although the received data is transferred to the RDR register, no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the SYER flag is being set to 1, the subsequently received data is not transferred to the RDR register.

<When MCR.SYEREN = 0>

The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SYER flag being set to 1.

[Clearing condition]

Write 1 to MFCLR.SYERC.

SBER bit (Start Bit Error flag)

This bit indicates that a start bit error was detected when receiving frames in Manchester mode.

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the SBER flag is not affected and retains its previous value.

[Setting condition]

When detecting a start bit error when receiving frames in Manchester mode

The following operations are performed when a start bit error occurs.

<When MCR.SBEREN = 1>

The received data is not transferred to the RDR register and no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the SBER flag is being set to 1, the subsequently received data is not transferred to the RDR register.

<When MCR.SBEREN = 0>

The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SBER flag being set to 1.

[Clearing condition]

Write 1 to MFCLR.SBERC.

MER bit (Manchester Error Flag)

When data is received in Manchester mode, Manchester error is detected, and it is displayed. Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the MER flag is not affected and retains its previous value.

[Setting conditions]

When receiving in Manchester mode and detecting Manchester code error in data area of received frame.

Received data when an error occurs is transferred to the RDR register, but the RXI interrupt request is not generated and the ERI interrupt request is generated.

When the Manchester error flag is set to 1, subsequent receive data is not transferred to the RDR register.

For details on Manchester error, see section 33.6.11. Errors in Manchester Mode.

[Clearing condition]

Write 1 to MFCLR.MERC.

RSYNC bit (Receive SYNC data bit)

When Manchester mode (CCR3.MOD[2:0] = 101b) and MCR.SBSEL = 1, this bit indicates the type of SYNC of the received the Start Bit. For other settings, it is fixed to 0.

33.3.19 CFCLR : Common Flag Clear Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRFC	—	TDREC	FERC	PERC	MFFC	—	ORERC	—	—	—	—	—	DFERC	DPERC	DCMFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ERSC	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	The write value should be 0.	W
4	ERSC	ERS clear 0: No effect 1: Clear the CSR.ERS bit	W
15:5	—	The write value should be 0.	W
16	DCMFC	DCMF clear 0: No effect 1: Clear the CSR.DCMF bit	W
17	DPERC	DPER clear 0: No effect 1: Clear the CSR.DPER bit	W
18	DFERC	DFER clear 0: No effect 1: Clear the CSR.DFER bit	W
23:19	—	The write value should be 0.	W
24	ORERC	ORER clear 0: No effect 1: Clear the CSR.ORER bit	W
25	—	The write value should be 0.	W
26	MFFC	MFF clear 0: No effect 1: Clear the CSR.MFF bit	W
27	PERC	PER clear 0: No effect 1: Clear the CSR.PER bit	W
28	FERC	FER clear 0: No effect 1: Clear the CSR.FER bit	W

Bit	Symbol	Function	R/W
29	TDREC	TDRE clear 0: No effect 1: Clear the CSR.TDRE bit	W
30	—	The write value should be 0.	W
31	RDRFC	RDRF clear 0: No effect 1: Clear the CSR.RDRF bit	W

33.3.20 ICFCLR : Simple I2C Flag Clear Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x6C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIFC	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	The write value should be 0.	W
3	IICSTIFC	IICSTIF clear 0: No effect 1: Clear the ISR.IICSTIF bit	W
31:4	—	The write value should be 0.	W

33.3.21 FFCLR : FIFO Flag Clear Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x70

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DRC	DR clear 0: No effect 1: Clear the FRSR.DR bit	W
31:1	—	The write value should be 0.	W

33.3.22 MFCLR : Manchester Flag Clear Register

Base address: SCIn = 0x8000_5000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_5000
SCIEi = 0x9003_0000 + 0x0400 × i (i = 0 to 7)
SCIEj = 0x9013_0000 + 0x0400 × (j - 8) (j = 8 to 11)

Offset address: 0x74

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MERC	—	SBERC	SYERC	PFERC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PFERC	PFER clear bit Setting this bit to 1 clears the MSR.PFER bit. This bit is read as 0.	W
1	SYERC	SYER clear bit Setting this bit to 1 clears the MSR.SYER bit. This bit is read as 0.	W
2	SBERC	SBER clear bit Setting this bit to 1 clears the MSR.SBER bit. This bit is read as 0.	W
3	—	The write value should be 0.	W
4	MERC	MER clear bit Setting this bit to 1 clears the MSR.MER bit. This bit is read as 0.	W
31:5	—	The write value should be 0.	W

33.4 Asynchronous Mode

Figure 33.3 shows the general format for asynchronous serial communications.

One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).

In asynchronous serial communications, the communications line is held in the mark state (high level) when not communicating.

The SCI monitors the communications line. When the SCI detects a start bit, it starts serial communication. The detection condition of the start bit changes according to the CCR3.RXDESEL setting. SCI regards space (Low level) as a start bit when CCR3.RXDESEL is 0. SCI regards a fall edge as a start bit when RXDESEL is 1.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure (it has also FIFO mode), so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

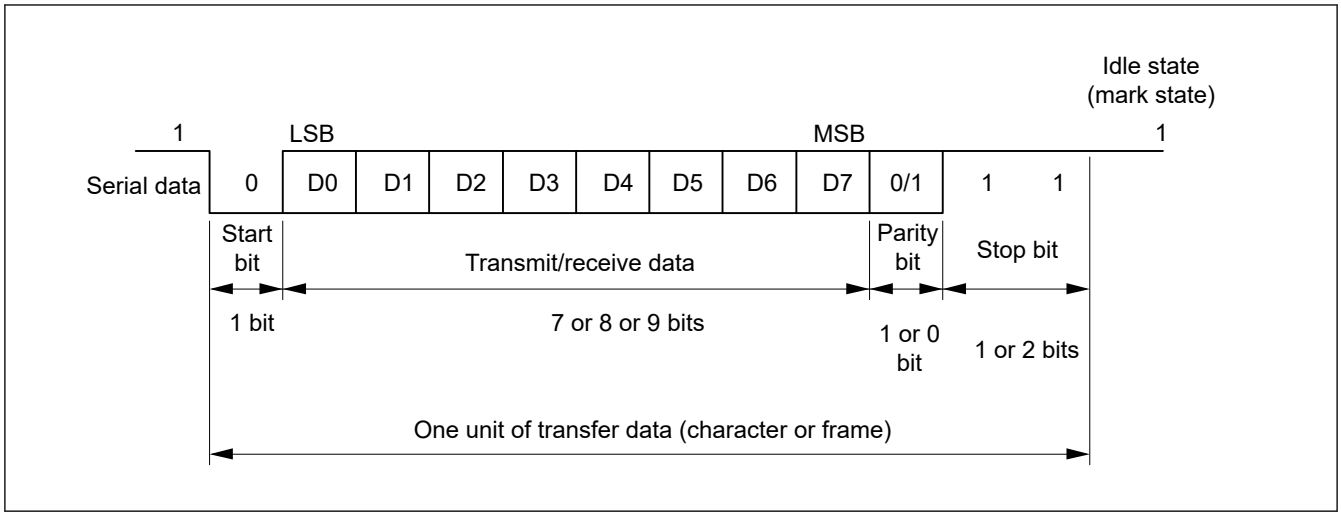


Figure 33.3 Data format in asynchronous serial communications (example with 8-bit data, parity, 2 stop bits)

33.4.1 Serial Data Transfer Format

Table 33.24 lists the serial data transfer formats that can be used in asynchronous mode. Any of 18 transfer formats can be selected according to the CCR1 and CCR3 settings.

For details of multi-processor function, see section 33.5. Multi-Processor Communication Function.

Table 33.24 Serial transfer formats (asynchronous mode) (1 of 2)

CCR3				CCR1	Serial transfer format and frame length
CHR[1]	CHR[0]	MP	STP	PE	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	St 9-bit data STP
0	0	0	1	0	St 9-bit data STP STP
0	0	0	0	1	St 9-bit data P STP
0	0	0	1	1	St 9-bit data P STP STP
1	0	0	0	0	St 8-bit data STP
1	0	0	1	0	St 8-bit data STP STP
1	0	0	0	1	St 8-bit data P STP
1	0	0	1	1	St 8-bit data P STP STP
1	1	0	0	0	St 7-bit data STP
1	1	0	1	0	St 7-bit data STP STP
1	1	0	0	1	St 7-bit data P STP
1	1	0	1	1	St 7-bit data P STP STP

Table 33.24 Serial transfer formats (asynchronous mode) (2 of 2)

CCR3				CCR1	Serial transfer format and frame length
CHR[1]	CHR[0]	MP	STP	PE	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	1	0	—	St 9-bit data MPB STP
0	0	1	1	—	St 9-bit data MPB STP STP
1	0	1	0	—	St 8-bit data MPB STP
1	0	1	1	—	St 8-bit data MPB STP STP
1	1	1	0	—	St 7-bit data MPB STP
1	1	1	1	—	St 7-bit data MPB STP STP

[Symbol explanation]

St : Start bit
STP : Stop bit
P : Parity bit
MPB : Multi-processor bit

33.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times[*1] the bit rate. In reception, the SCI samples the falling edge of the start bit using the base clock and performs internal synchronization[*2]. When sampling timing does not adjust ("CCR4.ASEN = 0b" or "CCR4.ASEN = 1b and CCR4.AST[2:0] = 000b"), receive data is sampled at the rising edge of the 8th pulse[*1] of the base clock, data is latched at the middle of each bit, as shown in Figure 33.4. Thus, the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left|(0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100[\%] \cdots \text{Equation (1)}$$

M: Reception margin

N: Ratio of bit rate to clock

(N = 16, when CCR2.ABCSE = 0 and CCR2.ABCS = 0, N = 8, when CCR2.ABCS = 1, N = 6, when CCR2.ABCSE = 1)

D: Duty cycle of clock (D = 0.5 to 1.0)

L: Frame length (L = 9 to 13)

F: Absolute value of clock frequency deviation

Assuming values of N = 16, F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100(\%) = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

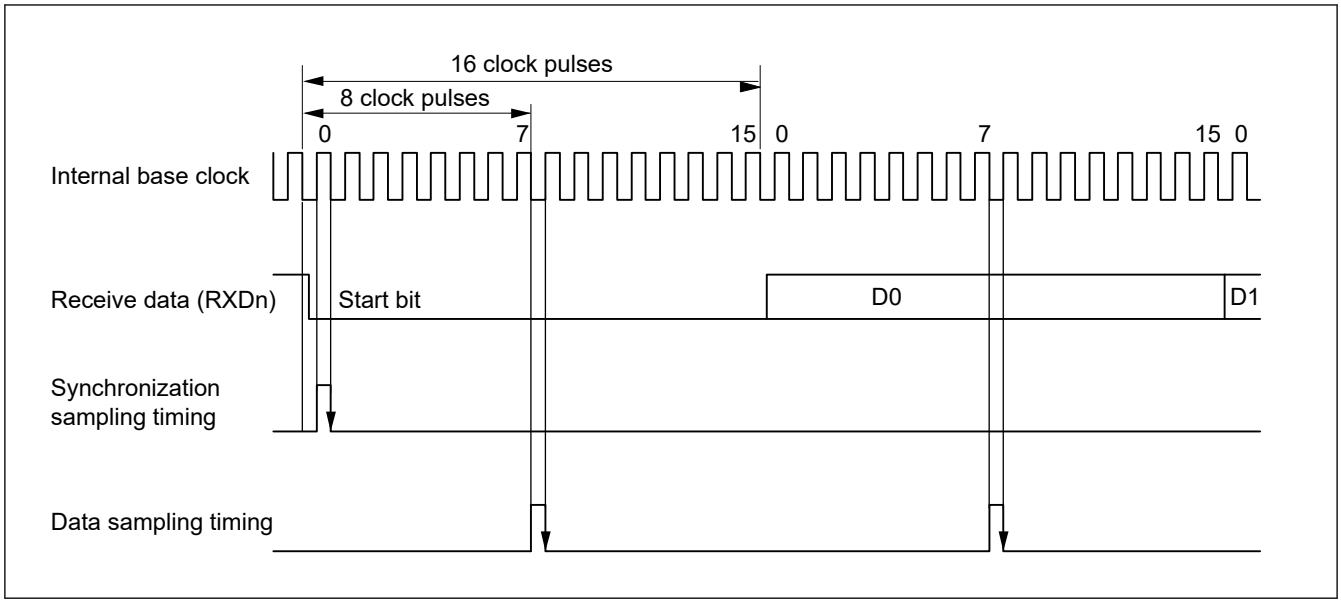


Figure 33.4 Receive data sampling timing in asynchronous mode

Note 1. This is an example when CCR2.ABCS = 0 and CCR2.ABCSE = 0. When CCR2.ABCS = 1 and CCR2.ABCS = 0, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock. When CCR2.ABCSE is 1, a sextuple frequency of a bit rate is a base clock and receive data is sampled at the rising edge of the 3rd pulse of the base clock.

Note 2. The determination condition of the start bit is as follows.

In the case of the function of adjust sampling timing is OFF (CCR4.ASEN = 0):

The determination condition of a start bit is that Low beyond half bit length continues. It is same as the sampling timing.

In Figure 33.4, the low period should be kept over 8 cycles to detect a start bit. If Low period does not keep over 8 cycles, the module judges this as a noise. So, the module does not start reception and wait start bit.

In the case of the function of adjust sampling timing is ON (CCR4.ASEN = 1):

The determination condition of a start bit is that Low keeps up until the sampling timing.

Adjusting the sampling timing forward (CCR4.AJD = 1) increases the possibility of erroneously determining a noise as the start bit.

33.4.3 Clock

Either an internal clock generated by the baud rate generator or an external clock input to the SCKn pin can be selected as the SCI's transfer clock, according to the setting of CCR3.CKE[1:0]. When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when CCR2.ABCS = 0) and 8 times the bit rate (when CCR2.ABCS = 1).

When the SCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 33.5.

If you selected an internal clock, the SCK pin is outputted after CCR0.TE is set to 1 or CCR0.RE is set to 1.

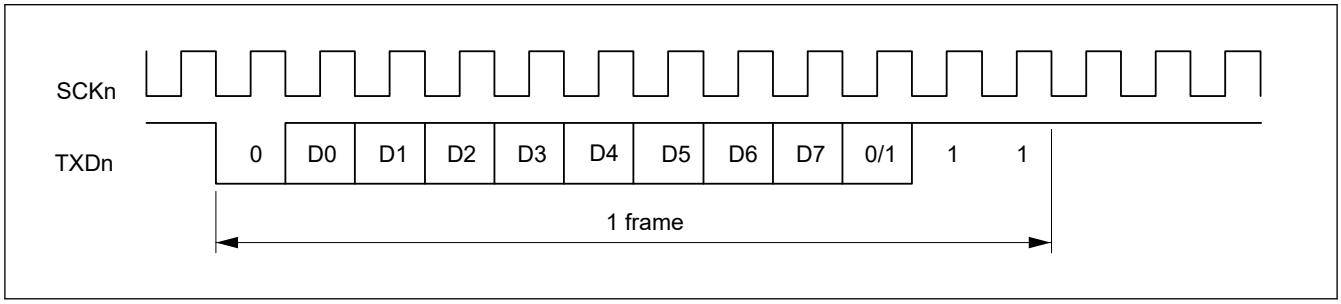


Figure 33.5 Phase relationship between output clock and transmit data (asynchronous mode: CHR[1:0] = 10b, PE = 1, MP = 0, STP = 1)

33.4.4 Double-Speed Operation and frequency of 6 times the bit rate

When CCR2.ABCS is set to 1, the SCI operates on the bit rate twice that in the case where ABCS is set to 0. And when CCR2.BGDM is set to 1, the cycle of the base clock is halved and the bit rate is doubled from that in the case where BGDM is set to 0. When CCR3.CKE[1] is set to 0 and the baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the SCI to operate on a bit rate four times that in the case ABCS = 0 and BGDM = 0.

When CCR2.ABCSE is set to 1, the number of base clock pulses are 6 during a period of 1 bit, and the base clock frequency is half. And SCI works 16/3 times of bit rate compared with a case of CCR2.ABCS = 0, CCR2.BGDM = 0 and CCR2.ABCSE = 0.

As shown by Formula (1) in section 33.4.2. Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when CCR2.ABCS is set to 1 or CCR2.ABCSE is set to 1. Therefore, if the desired bit rate can be obtained with CCR2.ABCS set to 0 and/or CCR2.ABCSE set to 0, it is recommended to use the SCI with CCR2.ABCS set to 0 and/or CCR2.ABCSE set to 0.

33.4.5 CTS and RTS Function

The CTS function is the transmission control function by the CTSn# pin. Setting the CCR1.CTSE bit to 1 enables the CTS function. For the functions of CTS and RTS, you can select the alternate setting that uses either function with one terminal or the dedicated setting that uses each function independently with two terminals. This setting is done with the CCR1.CTSPEN bit.

When the CTS function is enabled, placing the low level on the CTSn# pin, causes transmission to start.

Even if the CTSn# pin goes high after transmission starts, the frame being transmitted is not affected and transmission will continue.

The RTS function is the transmission request function by the RTSn# pin. In the RTS function, the RTSn# pin output low level, when reception becomes possible. Conditions for output of the low and high levels are shown below.

[Conditions of low-level output]

(1) Non-FIFO selected when all of the following conditions are satisfied.

- CCR0.RE is 1
- When in receivable state
- There are no received data yet to be read
- CSR.ORER, FER, PER flags are all 0

(2) FIFO selected when all of the following conditions are satisfied.

- CCR0.RE is 1
- When the quantity of receive data written in receive FIFO (RDR) are less than the setting value of FCR.RSTRG[4:0]
- CSR.ORER (RDR.ORER) is 0

[Condition for high-level output]

When the conditions of low-level output have not been satisfied.

33.4.6 Address Match (receive data match detection) Function

The address match function can be used only for the asynchronous mode.

If CCR0.DCME is set to 1,[*2] when one frame of data has been received, SCI compares that receive data with the data which is set to CCR4.CMPD. If SCI detects the match to the comparison data (CCR4.CMPD[*1]) with receive data, SCI can issue RXI interrupt request.

If CCR3.MP bit is set to 0, this comparative target in communication data is valid only data field in receive format. In multi-processor mode (CCR3.MP = 1), if CCR0.IDSEL bit is set to 1, the reception data at which MPB bit is 1 detects address match or unmatch, and the reception data at which MPB bit is 0 detects always unmatch. If CCR0.IDSEL bit is set to 0, SCI detects address match or unmatch despite the value of the MPB bit of the reception data at every reception complete.

Until SCI detects the match to the comparison data (CCR4.CMPD) with receive data, the communication data is skipped (discarded), and SCI can not detect parity error, framing error.

When SCI detects the match, the CCR0.DCME is automatically cleared, and CSR.DCMF is set to 1. If CCR0.IDSEL bit is set to 1 at this time, CCR0.MPIE bit is automatically cleared. If CCR0.IDSEL bit is set to 0 at this time, CCR0.MPIE bit is kept. At the same time, if CCR0.RIE is set to 1, SCI issues RXI interrupt request.

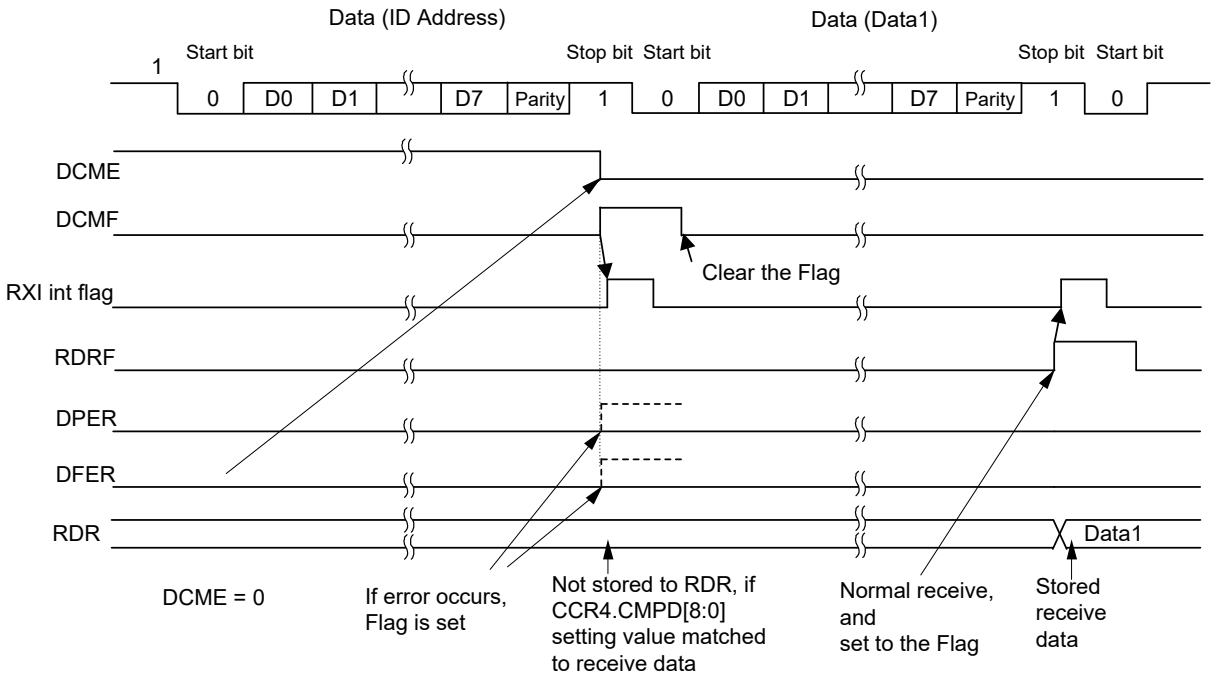
If SCI detects framing error in comparative receive data which is detected the match, CSR.DFER is set to 1, and if SCI detects parity error in that frame, CSR.DPER is set to 1. That comparative receive data and MPB bit are not stored to RDR register, and CSR.RDRF is retained to 0.

After SCI detects the match, and the CCR0.DCME is automatically cleared, it receives next data continuously in current register setting.

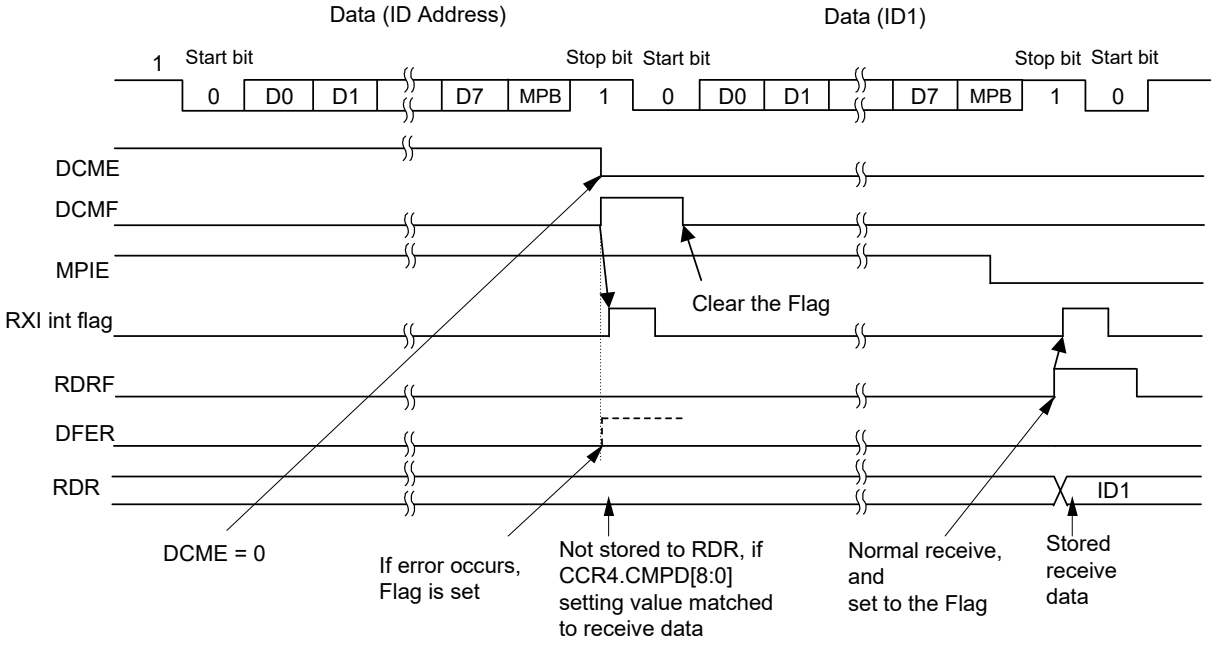
When CSR.DFER flag or CSR.DPER flag is set, the address much is not detected. Before making the address match function effective, please be sure to set CSR.DFER and CSR.DPER flag as 0.

Figure 33.6 and Figure 33.7 show the address match function example.

Note 1. This comparative target can select one length of 3 types, they are CMPD[6:0] with 7-bit length enable, CMPD[7:0] with 8-bit, and CMPD[8:0] with 9-bit length.
Note 2. Set the CCR0.DCME bit to 1 before receiving the start bit of the received frame that performs address matching.

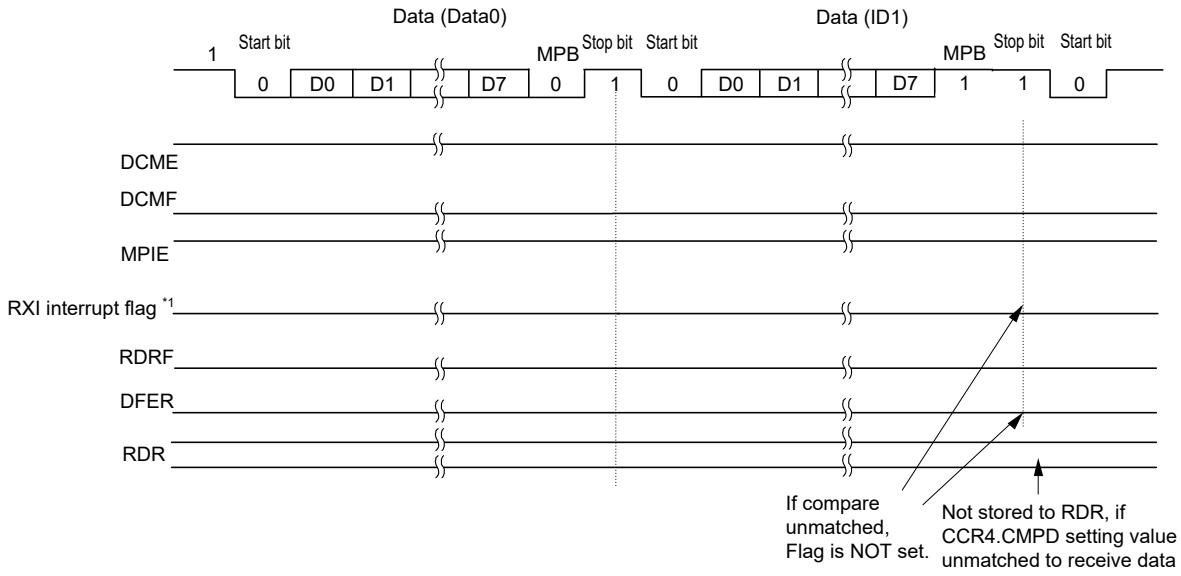


(a) Example of Compare matched between receive data and CCR4.CMPD[8:0] (8-bit length/parity/Normal mode)

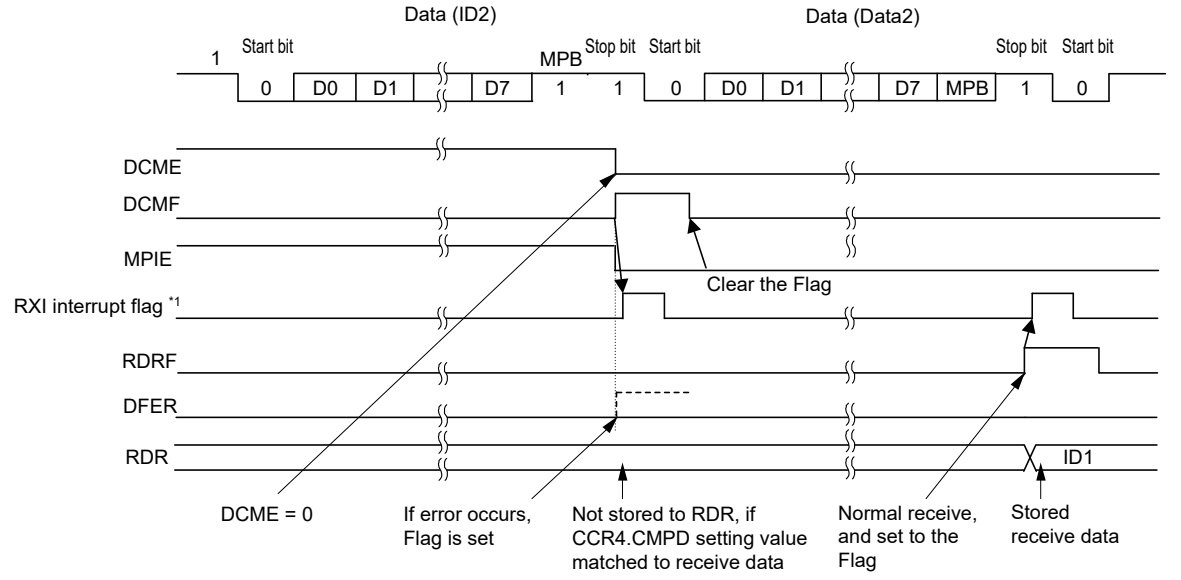


(b) Example of Compare matched between receive data and CCR4.CMPD[8:0] (8-bit length/Multi-processor mode)

Figure 33.6 Example of addresss match (1) non-multi-processor mode / 8-bit data



(a) Example of Compare unmatched between receive data and CCR4.CMPD (8-bit length / IDSEL = 1 / Multi-processor mode)



(b) Example of Compare matched between receive data and CCR4.CMPD (8-bit length / IDSEL = 1 / Multi-processor mode)

Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.7 Example of address match (2) multi-processor mode / 8bit-data

33.4.7 SCI Initialization (Asynchronous Mode)

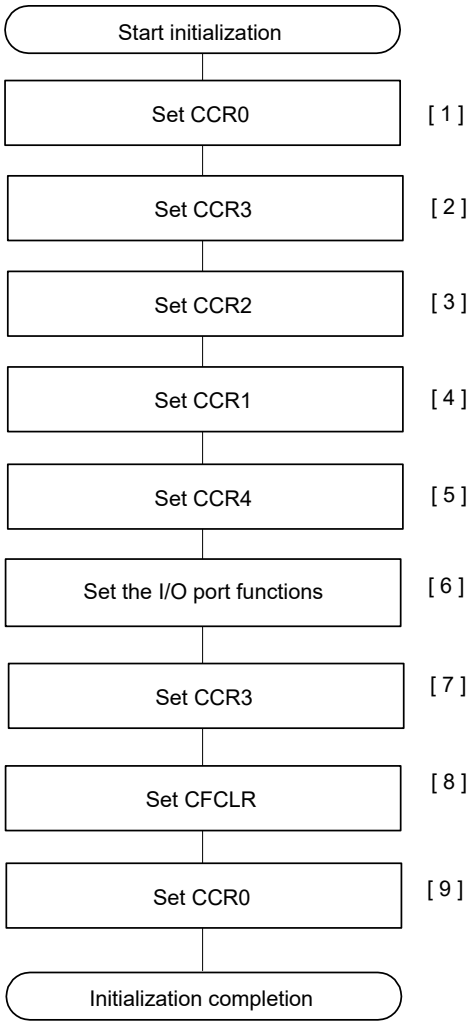
Before transmitting and receiving data, start by writing 0 to CCR0.TE and CCR0.RE (or writing the initial value to CCR0) and then continue through the Procedure (select to Non-FIFO or FIFO) for SCI given in Figure 33.8 or Figure 33.9. Whenever the operating mode or transfer format is changed, CCR0.TE and CCR0.RE must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that setting the CCR0.RE bit to 0 initializes neither the ORER, FER, PER, RDRF, DR and RDAT. When FIFO selected, even if the TE bit is set as 0, the TEND flag is not initialized, so please be careful.

Please be also careful at the time of change in the operation mode.

Moreover, note that switching the value of the CCR0.TE bit from 0 to 1 while the CCR0.TIE bit is 1 leads to the generation of a TXI interrupt request.



[1] Set the following bits of CCR0 to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set up following function and communication mode.
Driver control function for RS-485
FIFO no-use
Multi-Processor mode
Communication mode (MOD[2:0] = 000b)
Transmission / reception format
Clock enable (Leave the initial value in this step when outputting the clock)
Leave unused bits at their initial values.

[3] Set up the bit-rate-modulation function*1 *2, select clock, set bit rate*2.

[4] Set up the Noise-filter function, the loop-back function, communication terminal status, the parity check function, and the CTS/RTS function.

[5] Set up the adjust sampling timing function and the adjust transmit timing function.

[6] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[7] Set clock enable bit (CKE[1:0]) at this step when outputting the clock. After this register setting, a clock will be output immediately.

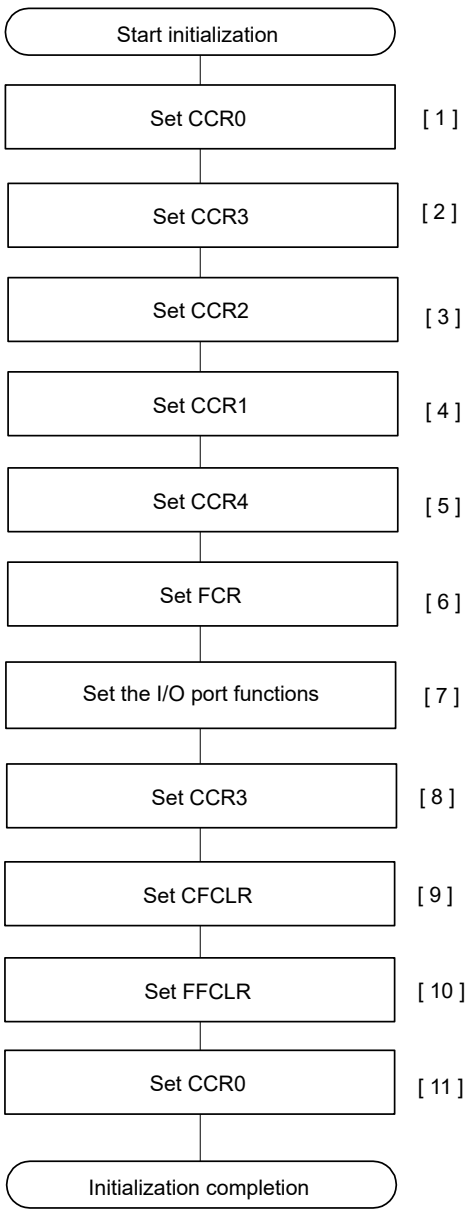
[8] Write 1 to the following bits and clear the corresponding flag.
CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
If it's initialization flow after a reset, you can skip this step.

[9] Set the TE or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note 1. If you do not use the bit rate modulation function, you do not need to set it.

Note 2. If you use an external clock, you do not need to set it.

Figure 33.8 Sample SCI initialization flowchart (asynchronous mode/non-FIFO selected)



Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

[1] Set the following bits of CCR0 to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set up following function and communication mode.
Driver control function for RS-485
FIFO use
Multi-Processor mode
Communication mode (MOD[2:0] = 000b)
Transmission / reception format
Clock enable (Leave the initial value in this step when outputting the clock)
Leave unused bits at their initial values.

[3] Set up the bit-rate-modulation function*1 *2, select clock, set bit rate*2.

[4] Set up the Noise-filter function, the loop-back function, communication terminal status, the parity check function, and the CTS/RTS function.

[5] Set up the adjust sampling timing function and the adjust transmit timing function.

[6] Set the TFRST and RFRST to 1 to empty FIFO.
Set the DRES, TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits

[7] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[8] Set clock enable bit (CKE[1:0]) at this step when outputting the clock. After this register setting, a clock will be output immediately.

[9] Write 1 to the following bits and clear the corresponding flag.
CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
If it's initialization flow after a reset, you can skip this step.

[10] Write 1 to the FFCLR.DRC and clear the corresponding flag.
If it's initialization flow after a reset, you can skip this step.

[11] Set the TE or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Figure 33.9 Sample SCI initialization flowchart (asynchronous mode/FIFO selected)

Figure 33.10 shows an example of the timing when data is transmitted after reset is released, and SCI is set to asynchronous mode according to Figure 33.8 or Figure 33.9. As shown in the figure, when the pin function is set to the TXD pin, the CCR0.TE bit is 0, so the pin is high impedance. When transmit data is written after setting the CCR0.TE bit to 1, data transmission starts. There is a transmission wait time from writing TDR to data transmission starts. In asynchronous mode, TXD is high during this period.

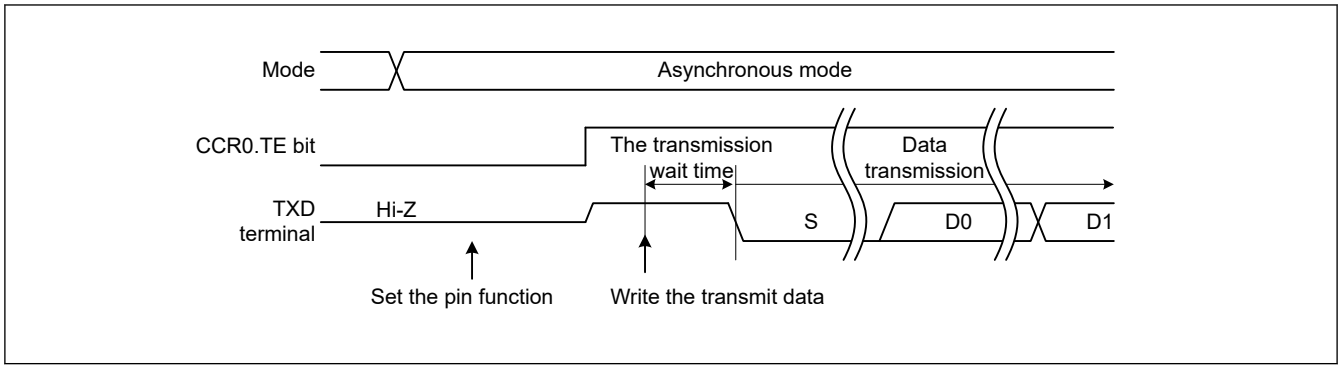


Figure 33.10 Data transmission timing example in asynchronous mode

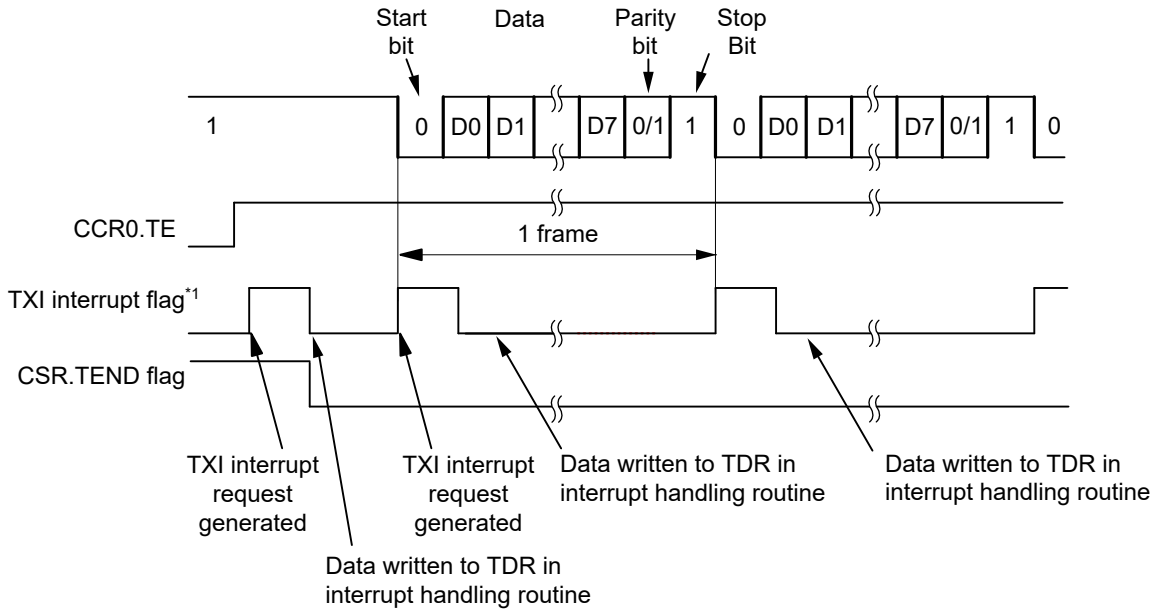
33.4.8 Serial Data Transmission (Asynchronous Mode)

(1) Non-FIFO selected

Figure 33.11, Figure 33.12, and Figure 33.13 show an example of the operation for serial transmission in asynchronous mode. In serial transmission, the SCI operates as described below.

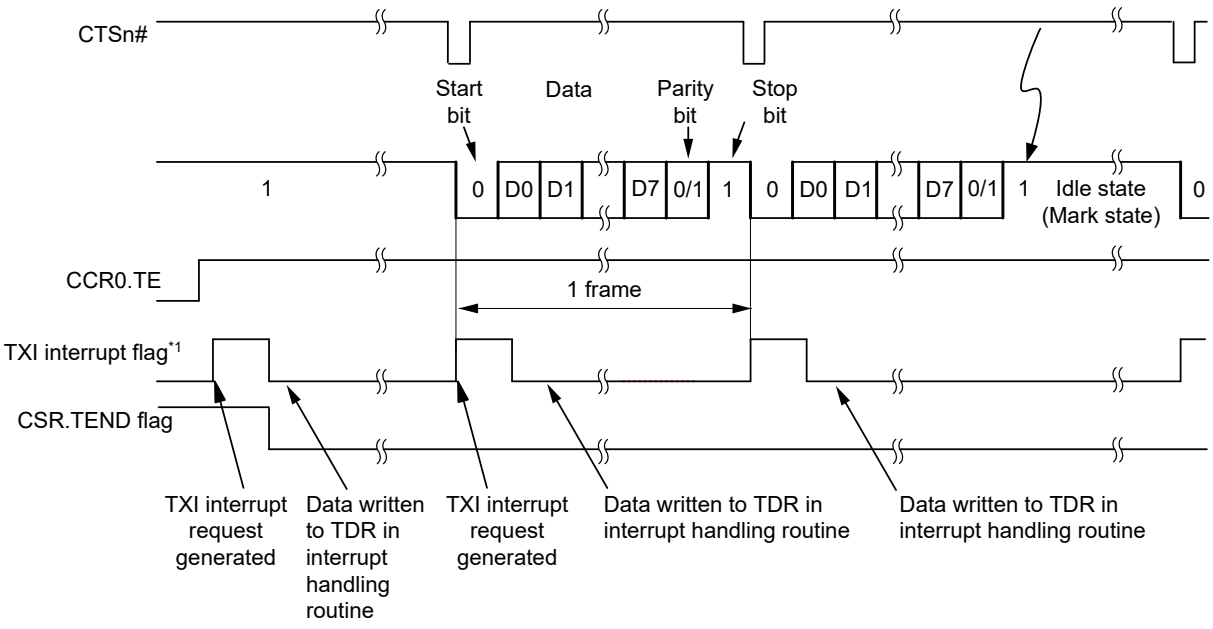
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt handling routine. At the beginning of transmission, set 1 to CCR0.TE and CCR0.TIE simultaneously. Then the TXI interrupt request is generated.
2. Transmission starts at data transfer from TDR to TSR when CCR1.CTSE = 0 (CTS function is disabled) or when the CTS# pin level is low. If CCR0.TIE is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to TDR in the TXI interrupt handling routine before transmission of the current transmit data is completed. Write the last transmission data, then the last data transmit start, and TXI is generated. When TEI interrupt requests are in use, before the last data transmit end, set the CCR0.TIE bit to 0 (a TXI interrupt request is disabled) and the CCR0.TEIE bit to 1 (a TEI interrupt request is enabled) using the TXI handling routine.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks for updating of (writing to) TDR at the time of stop bit output.
5. When TDR is updated, setting of CCR1.CTSE to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from TDR to TSR and sending of the stop bit, after which serial transmission of the next frame starts.
6. If TDR is not updated, CSR.TEND is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the CCR0.TEIE is 1 at this time, a TEI interrupt request is generated.

Figure 33.15 shows the example of Serial Transmission Flowchart.



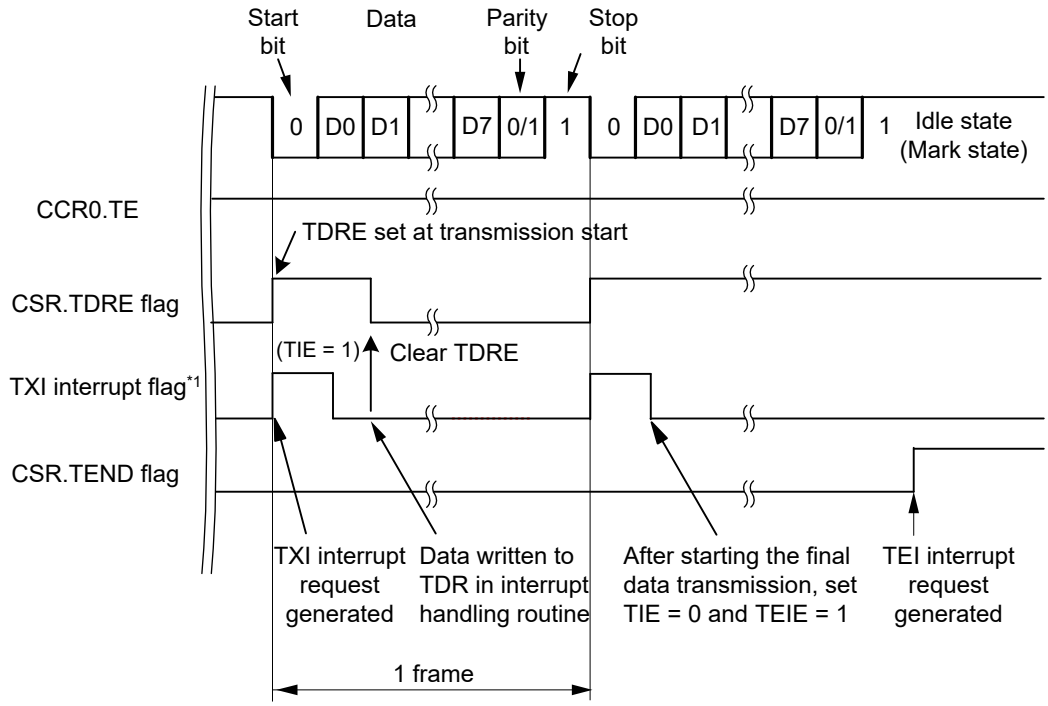
Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.11 Example of operation for serial transmission in asynchronous mode (1) (with 8-bit data, parity, 1 stop bit, CTS function not used, at the beginning of transmission)



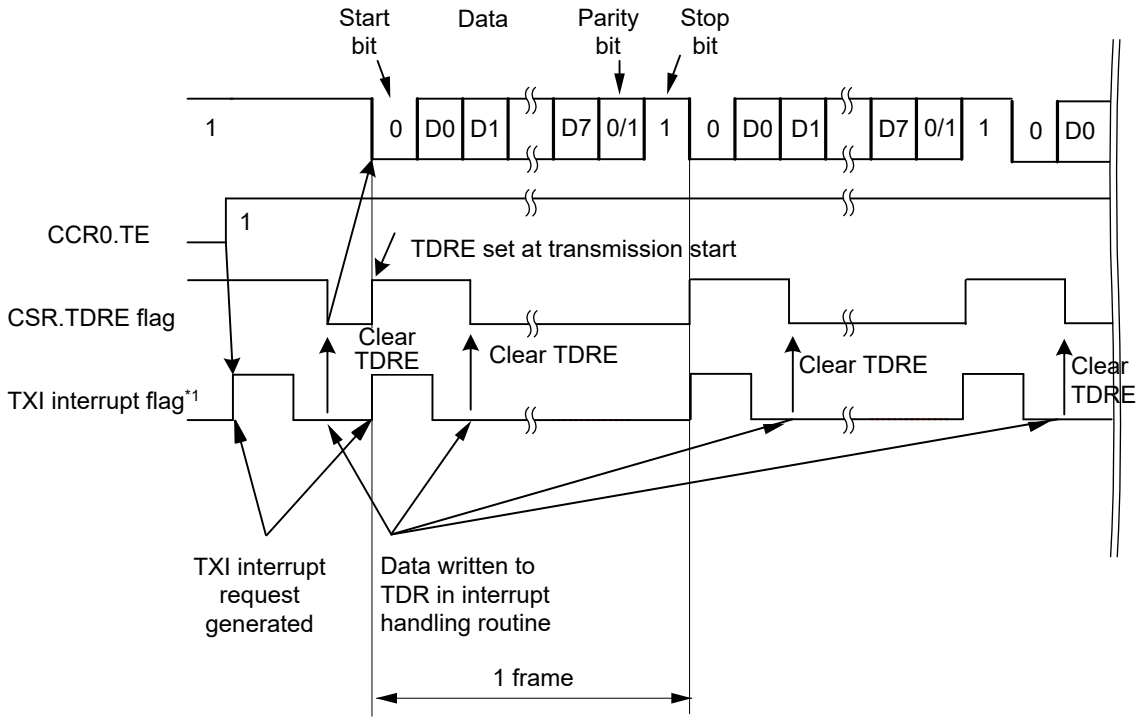
Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.12 Example of operation for serial transmission in asynchronous mode (2) (with 8-bit data, parity, 1 stop bit, CTS function used, at the beginning of transmission)



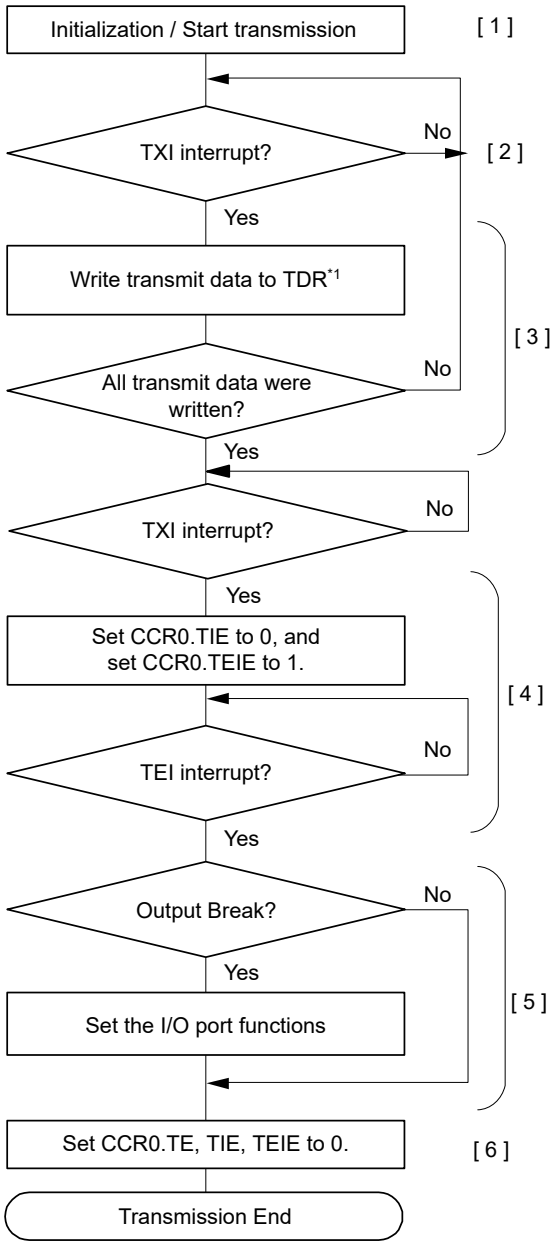
Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.13 Example of operation for serial transmission in asynchronous mode (3) (with 8-bit data, parity, 1 stop bit, CTS function not used, from the middle of transmission until transmission completion)



Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.14 Example of operation for serial transmission in asynchronous mode (4) (with 8-bit data, parity, 1 stop bit, CTS function not used, in the middle of transmission)



[1] Initialize SCI:
To enable TXI interrupt at the start of transmission, set TE and TIE to 1 with one instruction at the same time.

[2] TXI interrupt request:
The transmit-data-empty-interrupt (TXI) is generated at following conditions.
When set TE and TIE to 1 at same time.
When transmission-data forwards to TSR from TDR.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data to TDR*1 once by using a TXI.
Transmit data can also be written by activating the DMAC.

[4] If you use TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1 and transmission is started.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE, TIE, TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 33.15 Example of serial transmission flowchart in asynchronous mode (non-FIFO selected)

(2) FIFO selected

Figure 33.16 shows an example of data format that is written to transmit FIFO (TDR) in asynchronous mode with FIFO selected. MPBT write to transmit FIFO (TDR) bit 9. Data is set to TDR.TDAT[8:0] corresponded to data length. It should write to 0 for unused bits. It should write it in order of the TDR[15:8] register and the TDR[7:0] register at byte-access.

Data Length	Register setting		Transmit data in TDR[15:0]															
	CCR3. CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	–	–	–	–	–	–	MPBT	–	–	TDAT[6:0]						
8 bits	1	0	–	–	–	–	–	–	MPBT	–	TDAT[7:0]							
9 bits	0	x	–	–	–	–	–	–	MPBT	TDAT[8:0]								

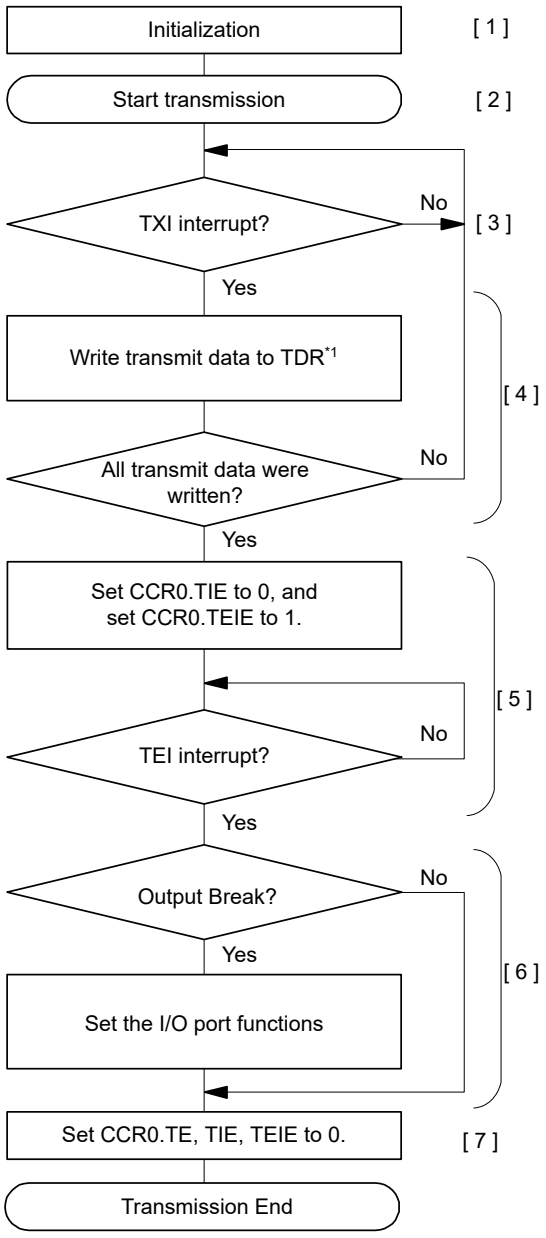
Note: –: Do not used. It should write to 0

Figure 33.16 Data format that is written to transmit FIFO (TDR) (FIFO selected)

In serial transmission, the SCI operates as described below.

1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt handling routine. The writable transmit data number is until (N_{FIFO} (= 16) - FTSR.T) bytes. At the beginning of transmission, set 1 to CCR0.TE and CCR0.TIE simultaneously. Then the TXI interrupt request is generated.
2. Transmission starts at data transfer from TDR to TSR when CCR1.CTSE = 0 (CTS function is disabled) or when the CTS# pin level is low. When the quantity of transmit data written in transmit FIFO (TDR) is equal to or less than the specified transmit triggering number, CSR.TDRE is set to 1. If CCR0.TIE is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to TDR in the TXI interrupt handling routine before transmission of the current transmit data is completed. Write the last transmission data, then the last data transmit start, and TXI is generated. When TEI interrupt requests are in use, before the last data transmit end, set the CCR0.TIE bit to 0 (a TXI interrupt request is disabled) and the CCR0.TEIE bit to 1 (a TEI interrupt request is enabled) using the TXI handling routine.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks whether non-transmitted data in transmit FIFO (TDR) or not at the time of stop bit output.
5. When data is set to transmit FIFO (TDR), setting of CCR1.CTSE to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from transmit FIFO (TDR) to TSR and sending of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set to transmit FIFO (TDR), CSR.TEND is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If CCR0.TEIE is 1 at this time, a TEI interrupt request is generated.

Figure 33.17 shows a sample flowchart for serial transmission in asynchronous mode at FIFO selected.



[1] Initialize SCI:
To enable TXI interrupt, set TE and TIE to 1 with one instruction at the same time.

[2] Set transmit-data to the transmit FIFO (TDR):
The number of transmittable data that can be set is 16.

[3] TXI interrupt request:
The transmit-data-empty-interrupt (TXI) is generated when transmission-data forwards to TSR from the transmit FIFO (TDR) and the number of transmit-data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[4] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data to the transmit FIFO (TDR) by using a TXI interrupt and clear TDRE flag to 0. (Transmit data can also be written to the transmit FIFO (TDR) by activating the DMAC. In this case, the TDRE flag is cleared automatically, so please do not operate the TDRE flag.)
The number of transmission data it's possible to write in is N_{FIFO} (= 16) - (the number of stored transmit FIFO data).

[5] If you use TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR.*1

[6] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[7] Set CCR0.TE, TIE, TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 33.17 Example of serial transmission flowchart in asynchronous mode (FIFO selected)

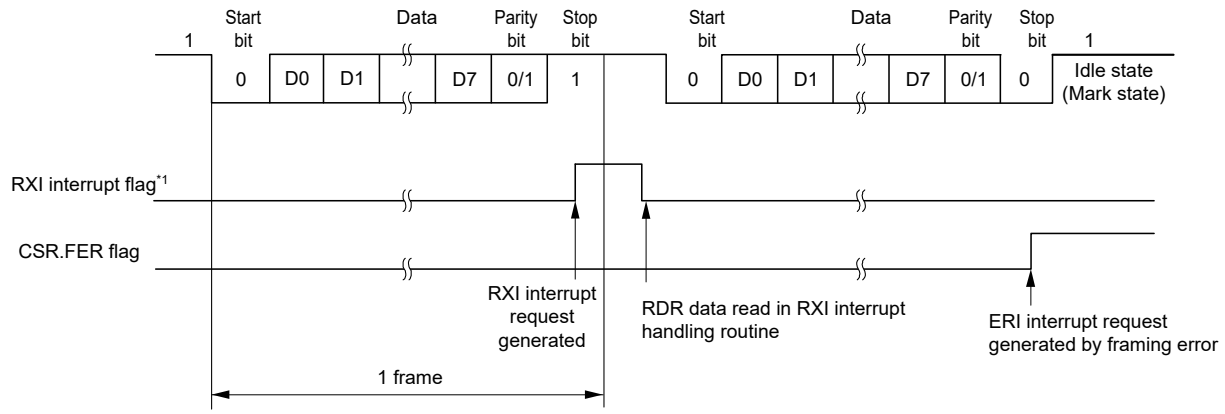
33.4.9 Serial Data Reception (Asynchronous Mode)

(1) Non-FIFO selected

Figure 33.18 and Figure 33.19 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

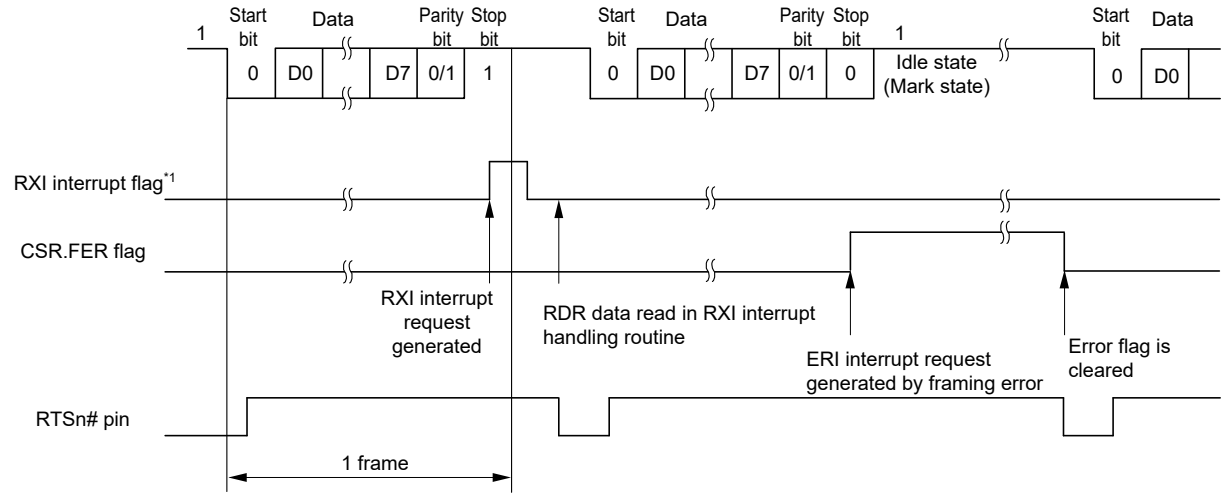
1. When the value of CCR0.RE becomes 1, the output signal on the RTSn# pin goes to the low level in the case of RTS function use.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, CSR.ORER is set to 1. If CCR0.RIE is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.

4. If a parity error is detected, CSR.PER is set to 1 and receive data is transferred to RDR. If the CCR0.RIE is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, CSR.FER is set to 1 and receive data is transferred to RDR. If CCR0.RIE is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR. If CCR0.RIE is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to RDR causes the RTSn# pin to output the low level in the case of RTS function use. If you do not want to turn the RTSn# pin output low after receiving the last data, set CCR0.RE bit to 0, before reading the RDR.



Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.18 Example of SCI operation for serial reception in asynchronous mode (1) (example with 8-bit data, parity, 1 stop bit, RTS function is not used)



Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

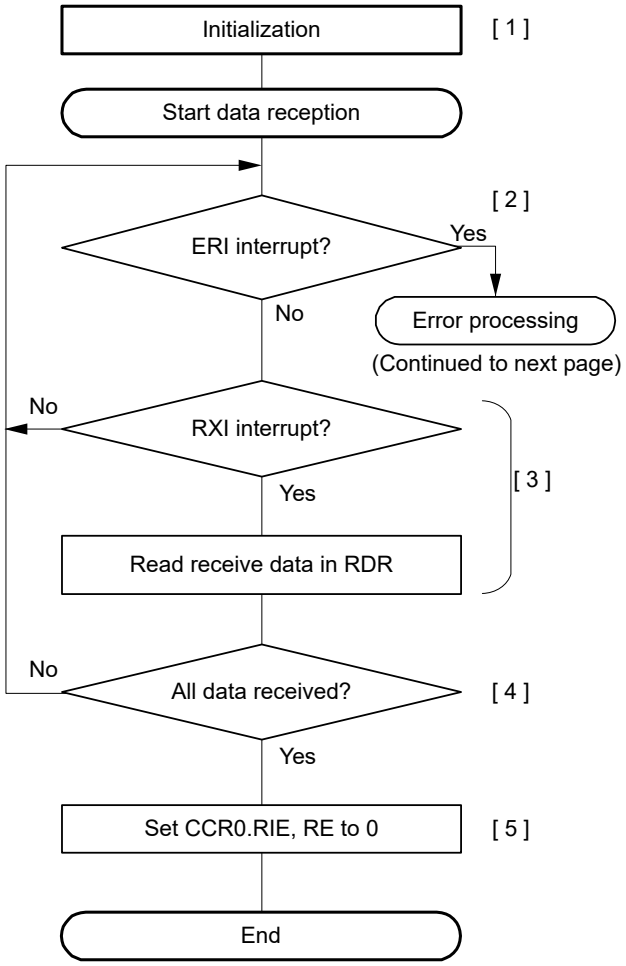
Figure 33.19 Example of SCI operation for serial reception in asynchronous mode (2) (example with 8-bit data, parity, 1 stop bit, RTS function is used)

Table 33.25 lists the states of the flags in CSR status register and receive data handling when a receive error is detected. If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing. When a reception is forcibly terminated by

setting CCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in RDR. Figure 33.20 and Figure 33.21 show samples of flowcharts for serial data reception.

Table 33.25 Flags in the CSR status register and receive data handling

Flags in the CSR status register			Receive data	Receive error type
ORER	FER	PER		
1	0	0	Lost	Overrun error
0	1	0	Transferred RDR	Framing error
0	0	1	Transferred RDR	Parity error
1	1	0	Lost	Overrun error + Framing error
1	0	1	Lost	Overrun error + Parity error
0	1	1	Transferred RDR	Framing error + Parity error
1	1	1	Lost	Overrun error + Framing error + Parity error



[1] Initialize SCI.
(Set CCR0.RE, RIE to 1.)

[2] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXDn pin.

[3] Read the receive data in RDR once in the RXI interrupt handling routine.

[4] Serial reception continuation procedure:
To continue serial reception, before the stop bit of the current frame is received, read data from RDR in the RXI interrupt handling routine. The RDR data can also be read by activating the DMAC.

[5] Set CCR0.RIE, RE to 0.

Figure 33.20 Example flowchart of serial reception in asynchronous mode (non-FIFO selected) (1)

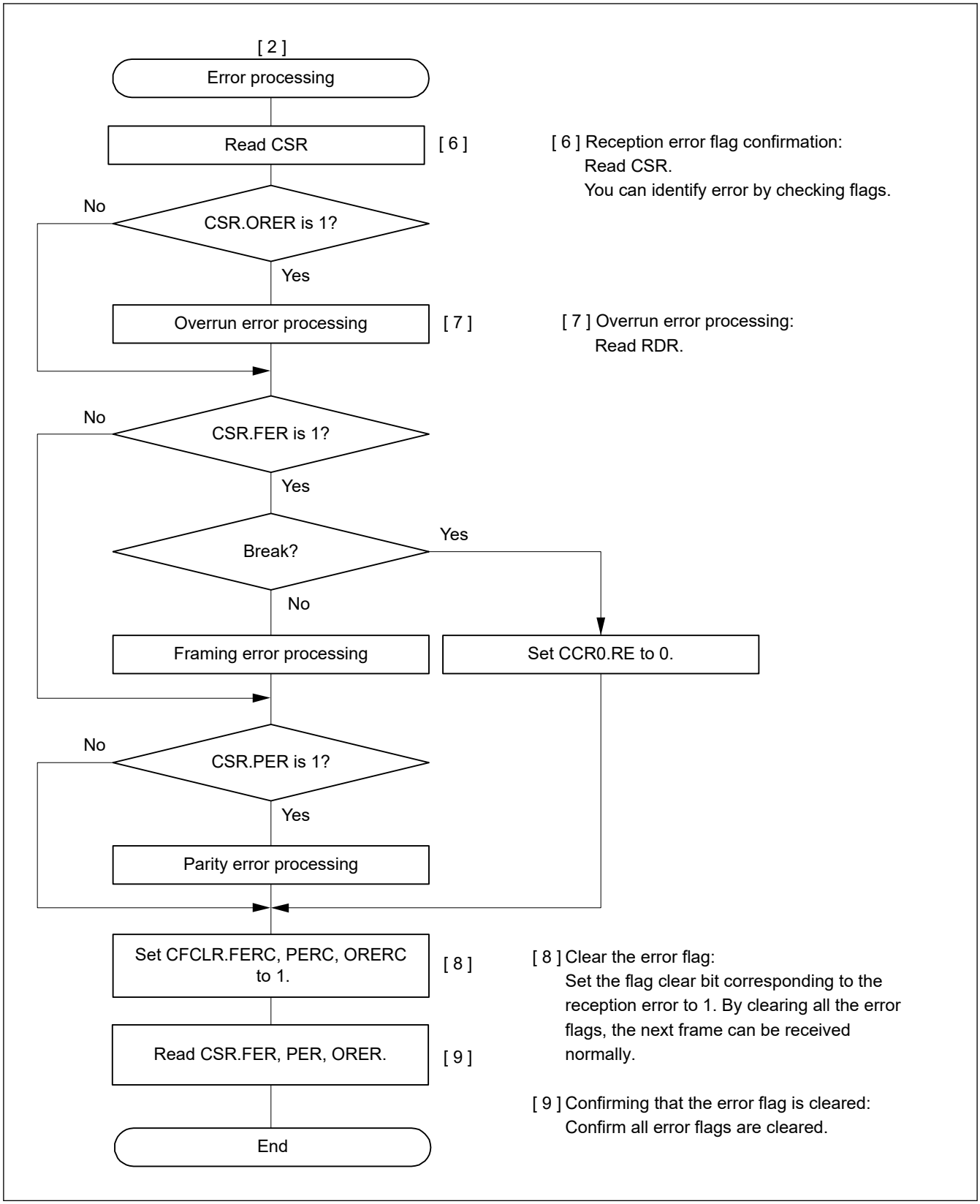


Figure 33.21 Example flowchart of serial reception in asynchronous mode (non-FIFO selected) (2)

(2) FIFO selected

Figure 33.22 shows an example of data format that is stored to receive FIFO (RDR) in asynchronous mode.

MPB (Receive FIFO (RDR) bit 9) is stored 0. Data is stored to receive FIFO (RDR) corresponded to data length. It is stored 0 for unused bits. If reading receive FIFO (RDR), SCI updates to next data which are FER, PER and receive data

(RDAT[8:0]) in receive FIFO. The flags which are BRK, RDRF, ORER and DR in receive FIFO, are always indicated to the flags corresponded to CSR register.

Data Length	Register Setting		Receive flag in RDR[31:0], MPB, RDAT[8:0]															
	CCR3.CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	–	–	–	FFER	FPER	DR	MPB	0	0	RDAT[6:0]						
8 bits	1	0	–	–	–	FFER	FPER	DR	MPB	0	RDAT[7:0]							
9 bits	0	x	–	–	–	FFER	FPER	DR	MPB	RDAT[8:0]								
	CCR3.CHR[1:0]		b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
7 bits	1	1	–	–	–	FER	PER	–	–	ORER	–	–	–	–	–	–	–	–
8 bits	1	0	–	–	–	FER	PER	–	–	ORER	–	–	–	–	–	–	–	–
9 bits	0	x	–	–	–	FER	PER	–	–	ORER	–	–	–	–	–	–	–	–

Note: The MPB flag, the RDR[9] bit, is always read as 0.
When a 7-bit data length is selected, the RDAT[8:7] bits are read as 00b.
When an 8-bit data length is selected, the RDAT[8] bit is read as 0.

Figure 33.22 Data format that is stored to receive FIFO (RDR) (FIFO selected)

Table 33.26 lists the states of the flags in CSR status register and receive data handling when a receive error is detected with FIFO selected. Figure 33.23 and Figure 33.24 show samples of flowcharts for serial data reception with FIFO selected.

In serial data reception, the SCI operates as described below.

1. When the value of CCR0.RE becomes 1, the output signal on the RTSn# pin goes to the low level in the case of RTS function use.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If there is no space in RDR, CSR.ORER is set to 1. If CCR0.RIE is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to receive FIFO (RDR).
4. If a parity error is detected, CSR.PER is set to 1 and receive data is transferred to receive FIFO (RDR). If the CCR0.RIE is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, CSR.FER is set to 1 and receive data is transferred to RDR. If CCR0.RIE is 1 at this time, an ERI interrupt request is generated.
6. After framing error is detected, when SCI detects that continuous receive data is 0 for 1 frame, reception stops.
7. When quantity of data stored in the receive FIFO data register (RDR) falls the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in asynchronous mode, FRSR.DR is set to 1. If RIE bit is set to 1, SCI occurs RXI interrupt request when FCR.DRES bit is 0 and SCI occurs ERI interrupt request when FCR.DRES = 1.
8. When reception finishes successfully, receive data is transferred to receive FIFO (RDR). RDRF is set to 1 when the quantity of receive data which is equal to or greater than the specified receive triggering number are stored in receive FIFO (RDR). If CCR0.RIE is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to receive FIFO (RDR) in this RXI interrupt handling routine before overrun error is occurred. Reading the received data that have been transferred to receive FIFO (RDR), and if it is less than RTS trigger number, causes the RTSn# pin to output the low level in the case of RTS function use.

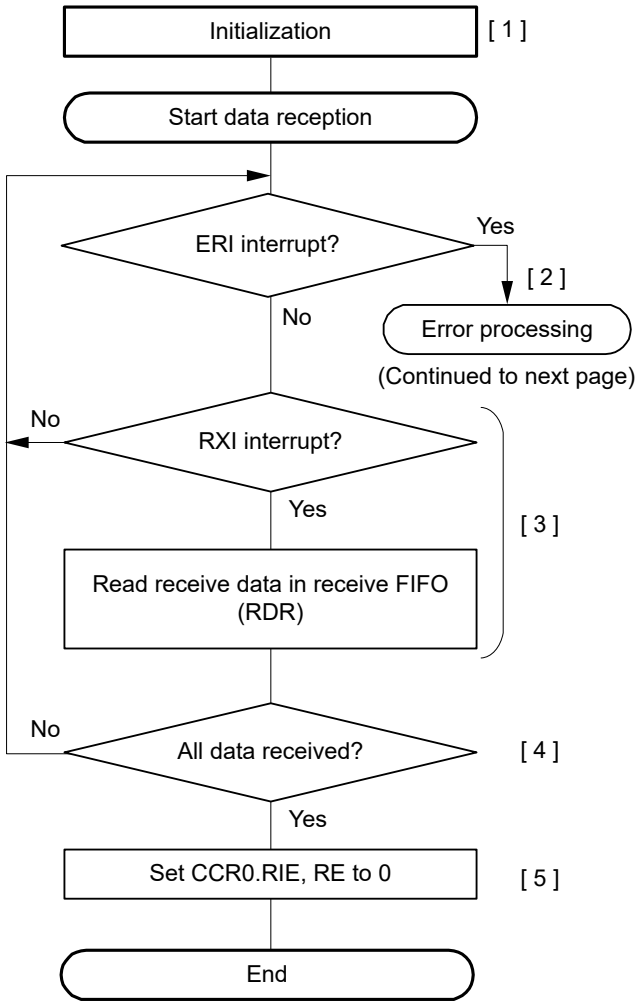
Table 33.26 Flags in the CSR status register and receive data handling (FIFO selected) (1 of 2)

CSR value			Receive FIFO (RDR)	
ORER	FER*1	PER*1	RDAT[8:0]	Receive Error Type
1	0	0	Lost	Overrun error
0	1	0	Transferred RDR	Framing error
0	0	1	Transferred RDR	Parity error

Table 33.26 Flags in the CSR status register and receive data handling (FIFO selected) (2 of 2)

CSR value			Receive FIFO (RDR)	
ORER	FER*1	PER*1	RDAT[8:0]	Receive Error Type
1	1	0	Lost	Overrun error + Framing error
1	0	1	Lost	Overrun error + Parity error
0	1	1	Transferred RDR	Framing error + Parity error
1	1	1	Lost	Overrun error + Framing error + Parity error

Note 1. This flag indicates whether there is an error in received data when reception is completed.



[1] Initialize SCI.
(Set CCR0.RE, RIE to 1.)

[2] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER, flags to 0. Reception cannot be resumed if ORER is set to 1. Reception can continue even if PER, FER, DR flags are 1.

[3] Read the receive data in the receive FIFO (RDR) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by FRSR.R[5:0].

[4] Serial reception continuation procedure:
To continue serial reception, before an overrun error occurs, please read data from the receive FIFO (RDR) in the RXI interrupt handling routine and clear CSR.RDRF flag and FRSR.DR flag to 0. The receive FIFO data can also be read by activating the DMA. The RDRF flag is cleared automatically in this case, so please do not operate the RDRF flag.

[5] Set CCR0.RIE, RE to 0.

Figure 33.23 Example flowchart of serial reception in asynchronous mode (FIFO selected) (1)

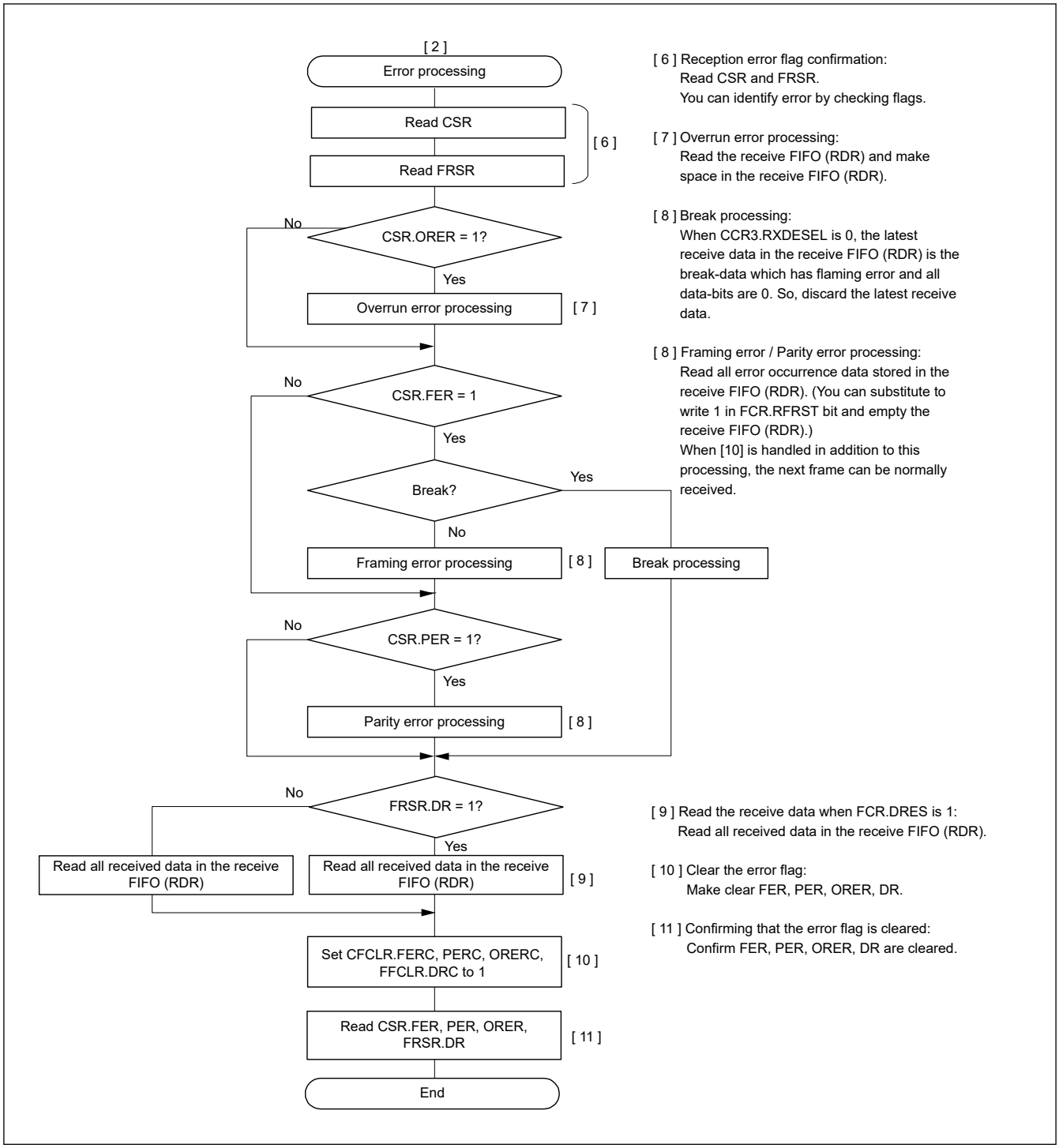


Figure 33.24 Example flowchart of serial reception in asynchronous mode (FIFO selected) (2)

33.4.10 The function of adjust receive sampling timing (Asynchronous Mode)

When there is the difference between the rising transfer time and the falling transfer time through a photo coupler, the receive sampling timing at middle of bit affects the reception margin. In this case, the receive sampling timing can adjust from the middle of bit to the optimum timing by using this function.

The receive sampling timing is adjusted from the middle of bit by following formula. And the adjustable direction is set by CCR4.AJD. You can select forward or backward from the middle of bit. When adjusting backward (CCR4.AJD = 0), substitute AJD = +1 and substitute AJD = -1 when adjusting forward (CCR4.AJD = 1).

Adjusted sampling timing = the middle of bit + AJD × (base clock × the setting value of CCR4.AST[2:0])

The setting timing is limited by base clock cycles per 1 bit. Please see Table 33.27 in detail.

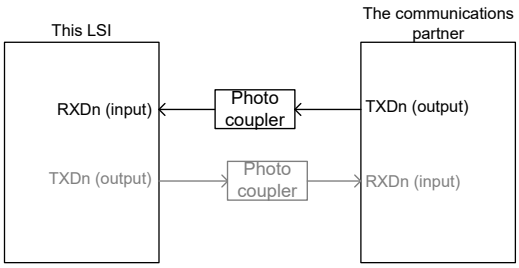
A reception movement image figure of the communication through a photo coupler with this function is shown in Figure 33.25 and Figure 33.26, the movement explanation of this function is shown in Figure 33.27.

Do not use this function when there is no difference between the rising transfer time and the falling transfer time, because there is a possibility of deteriorating the reception margin.

Table 33.27 The acceptable value of setting register (asynchronous mode using internal clock)

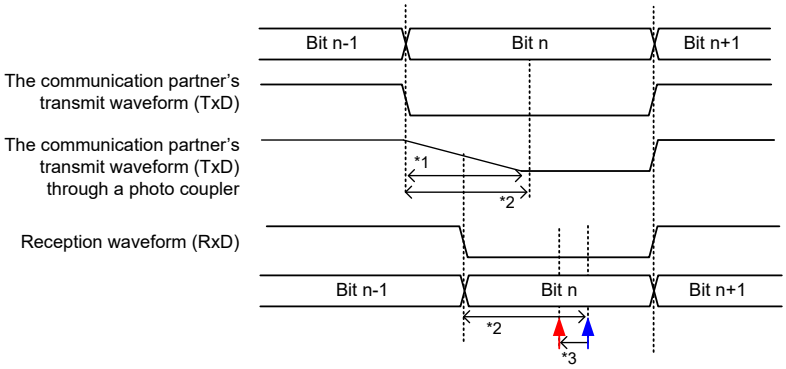
CCR2.ABCSE	CCR2.ABCS	The number of base cycles / 1 bit	The acceptable value	
			CCR4.AJD	CCR4.AST
1	x	6	0	000b to 010b*1
			1	
0	1	8	0	000b to 011b*1
			1	
0	0	16	0	000b to 111b
			1	

Note 1. When the value of CCR4.AST is over the acceptable value, sampling is done at default timing. (Adjustment of sampling is not done.)



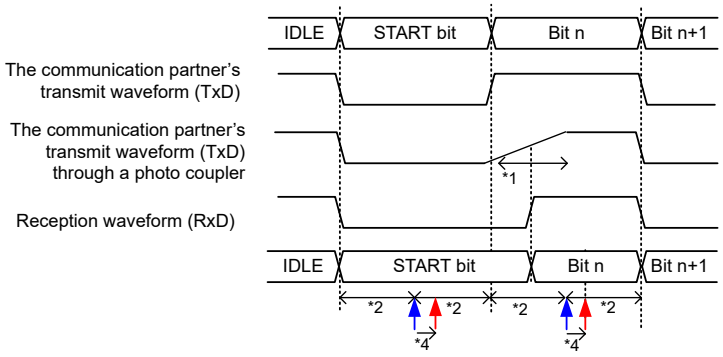
(a) In the case of the falling transfer time >> rising transfer time

The falling edge of reception waveform is made dull like following chart.
In this case, you can sampling at the middle of bit if you adjust the receive sampling timing to forward (AJD = 1b).



(b) In the case of the falling transfer time << rising transfer time

The rising edge of reception waveform is made dull like following chart.
So, the reception margin of communications partner will be bad. In this case, you can improve the reception margin if you adjust the receive sampling timing to back.



The receive sampling timing when unadjusted (middle of bit)

The adjusted receive sampling timing

Note 1. The dull period by a photo coupler

Note 2. Bit center timing at set communication rate

Note 3. When CCR4.AJD is 1, the receive sampling timing is shifted to forward by the setting value of CCR4.AST[2:0].

Note 4. When CCR4.AJD is 0, the Receive sampling timing is shifted to backward by the setting value of CCR4.AST[2:0].

Note: This waveform shows the operation image of adjustment receive sampling timing.

Figure 33.25 The reception movement image figure of the communication through a photo coupler

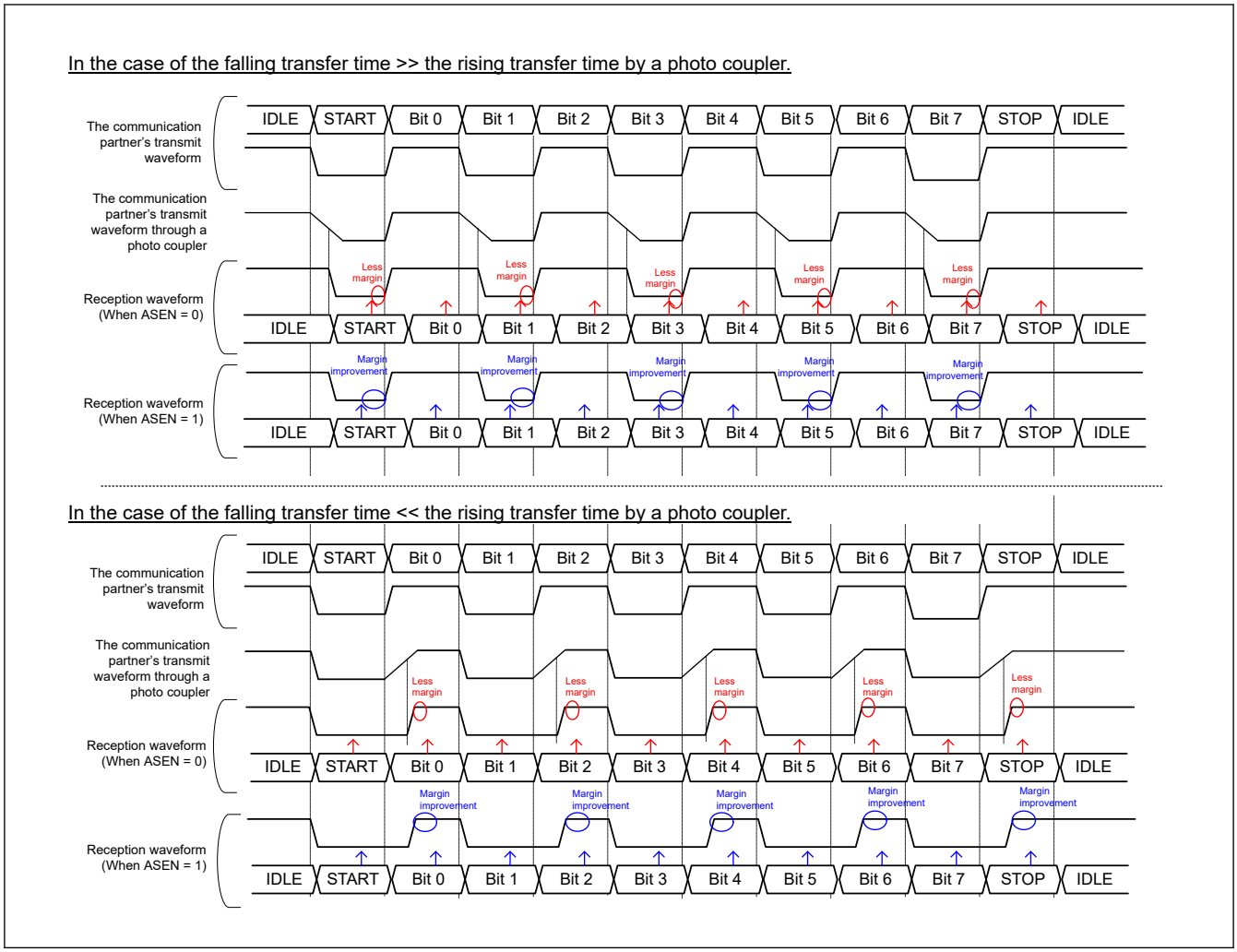


Figure 33.26 Improvement of reception margin when receive sampling timing adjustment function Image

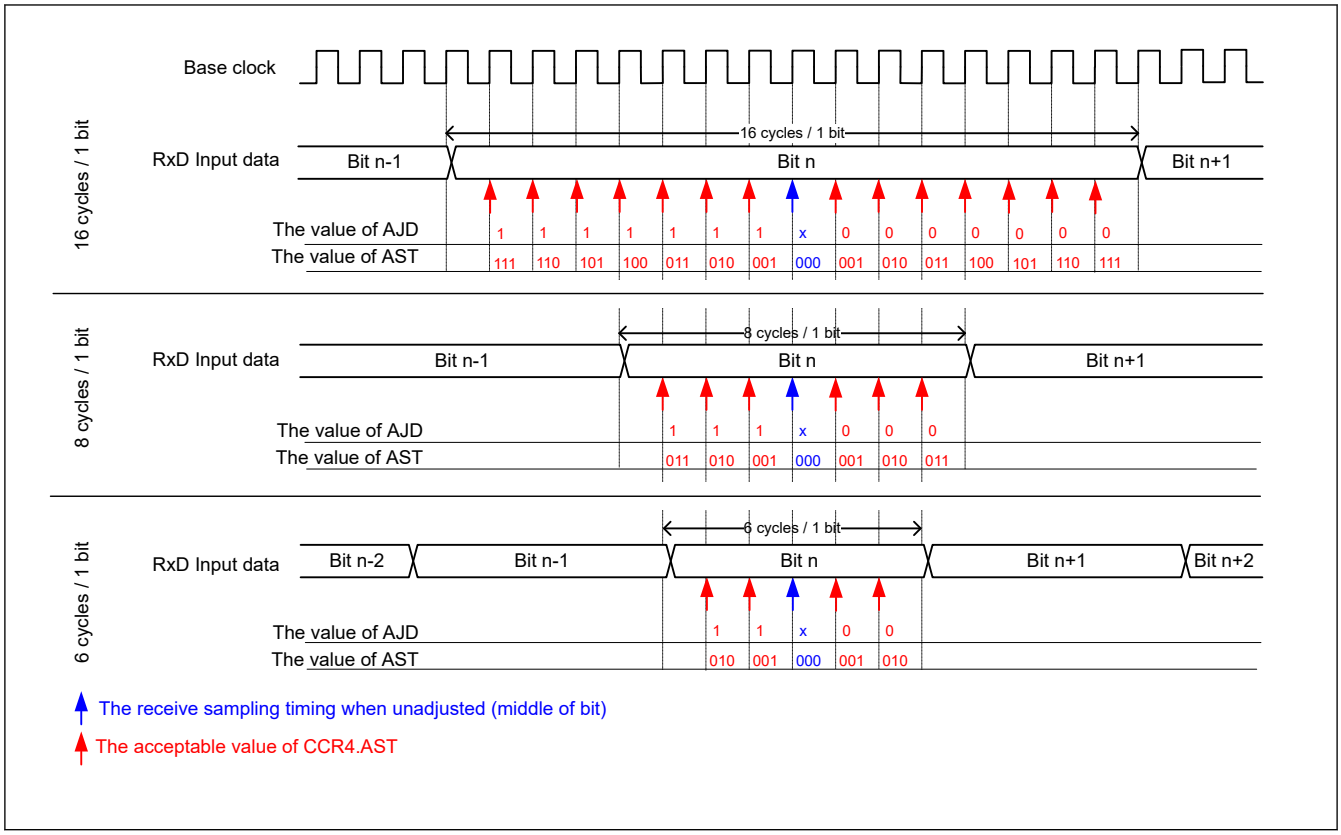


Figure 33.27 The adjustment operation explanation for the receive sampling timing (asynchronous mode using internal clock)

33.4.11 The function of adjust transmit timing (Asynchronous Mode)

In communication via a photo coupler or the like, when either the rising or falling transition time of the TxD output signal is long, then a communication partner receive dulled waveform. In this case, the reception margin may be affected.

In these cases, make a communication partner to be sampling at middle of bit using the function of adjust transmit timing. When CCR4.ATEN is 1, this function can adjust the edge timing at the timing calculated by the following formula for the edge set with CCR4.AET.

The adjustment edge timing = base clock × CCR4.ATT[2:0]

In addition, the upper limit of the adjustment edge timing is limited by setting the base clock cycles. For details, see Table 33.28.

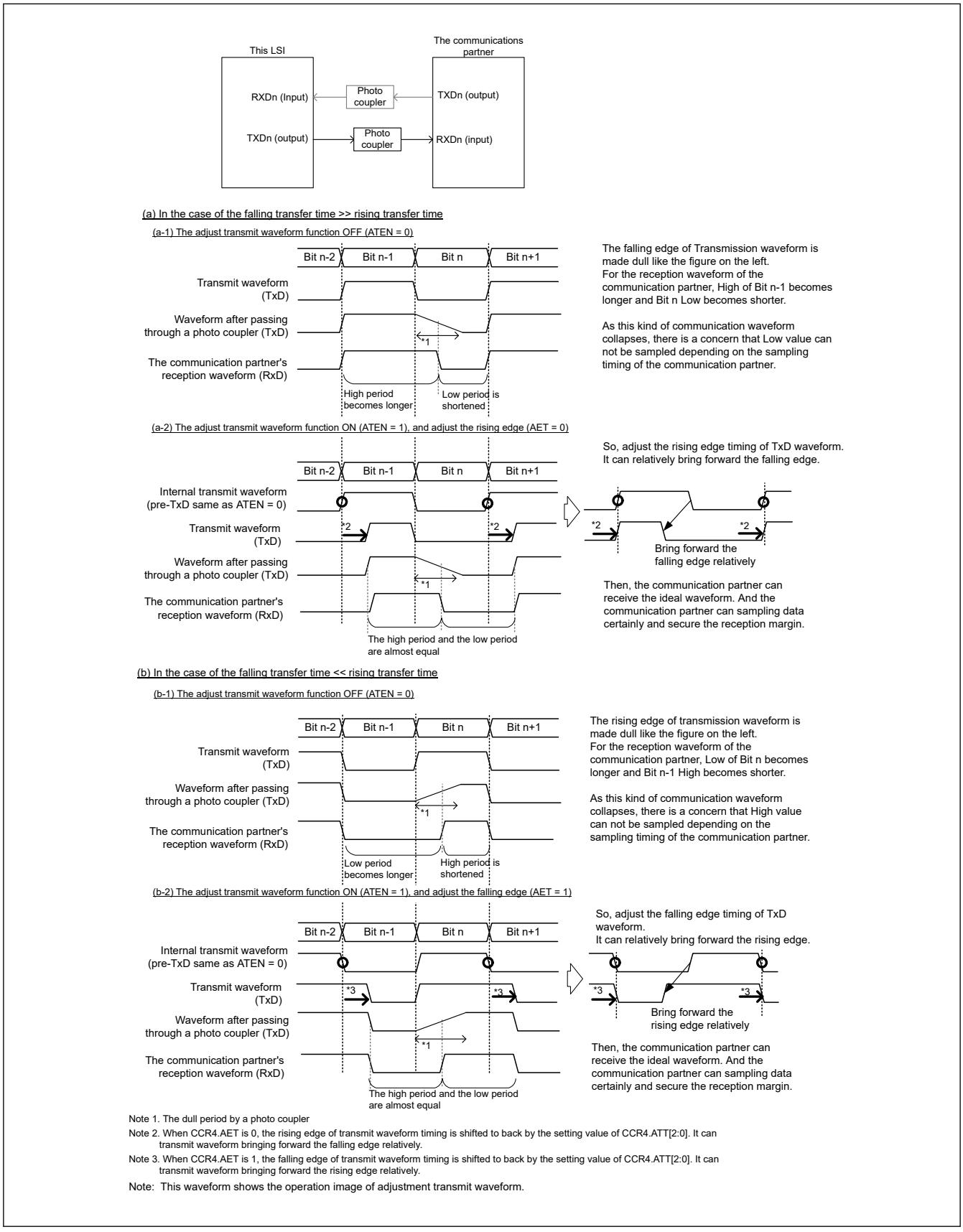
A transmission movement image figure of the communication through a photo coupler with this function is shown in Figure 33.28 and Figure 33.29, the movement explanation of this function is shown in Figure 33.30 and Figure 33.31.

Do not use this function when there is not the difference between the rising transfer time and the falling transfer time, there is a possibility of deteriorating the reception margin of a communication partner.

Table 33.28 The acceptable value of CCR4.AET and CCR4.ATT (asynchronous mode using internal clock)

ABCSE	ABCS	The number of base clock cycles / 1 bit	The acceptable value	
			AET	ATT[2:0]
1	x	6	0	000b to 101b
			1	
0	1	8	0	000b to 111b
			1	
0	0	16	0	000b to 111b
			1	

Note: When the value of CCR4.AET/ATT is out of the acceptable value, this module does not adjust transmit timing.



Note 1. The dull period by a photo coupler

Note 2. When CCR4.AET is 0, the rising edge of transmit waveform timing is shifted to back by the setting value of CCR4.ATT[2:0]. It can transmit waveform bringing forward the falling edge relatively.

Note 3. When CCR4.AET is 1, the falling edge of transmit waveform timing is shifted to back by the setting value of CCR4.ATT[2:0]. It can transmit waveform bringing forward the rising edge relatively.

Note: This waveform shows the operation image of adjustment transmit waveform.

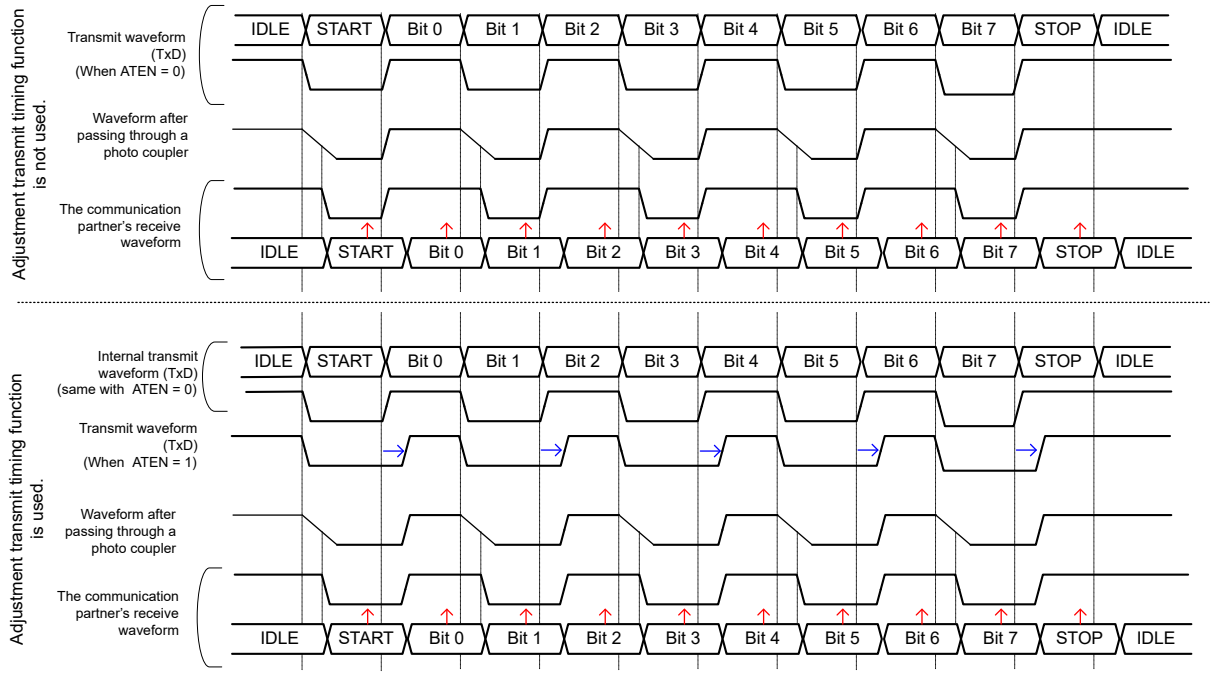
Figure 33.28 The transmission movement image figure of the communication through a photo coupler

The explanation of transmit waveforms of the communication through a photo coupler using adjust transmit timing function

When using the transmission timing adjustment function, adjust the edge timing of the transmission waveform and correct the reception waveform of the communication partner

The following example is 8 bit long data.

(a) In the case of the falling edge transfer time >> the rising transfer time



(b) In the case of the falling edge transfer time << the rising transfer time

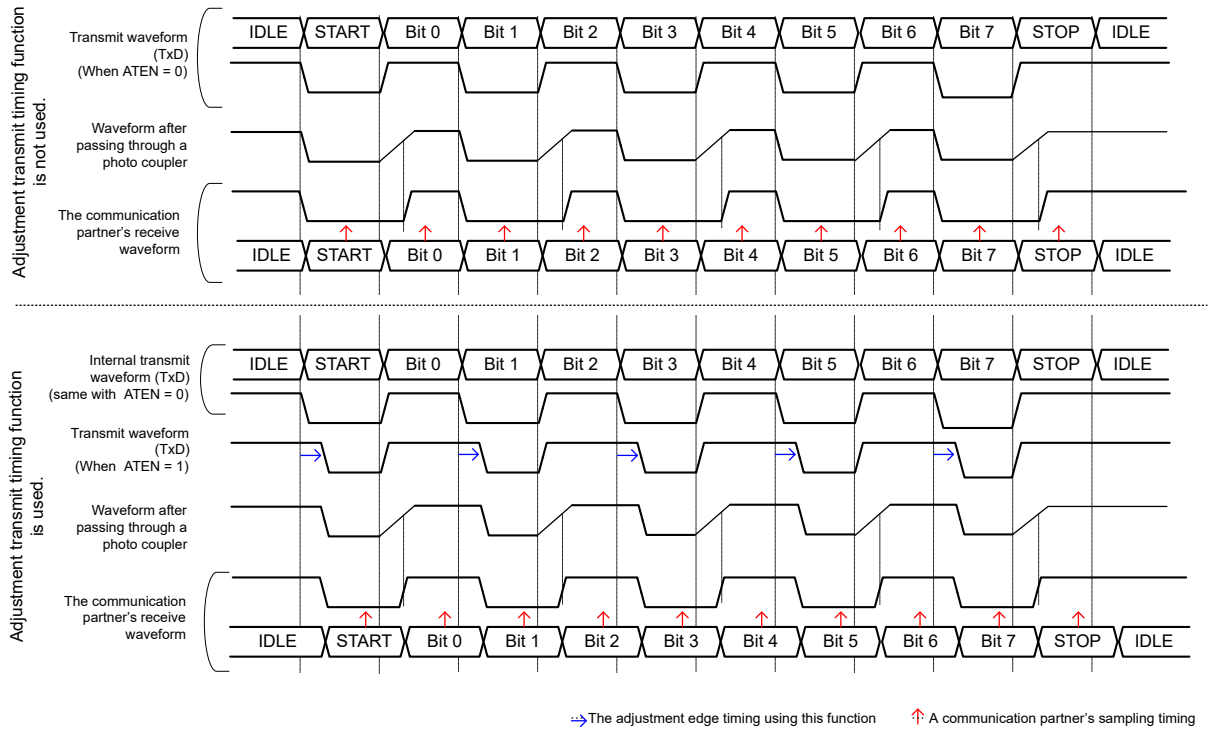


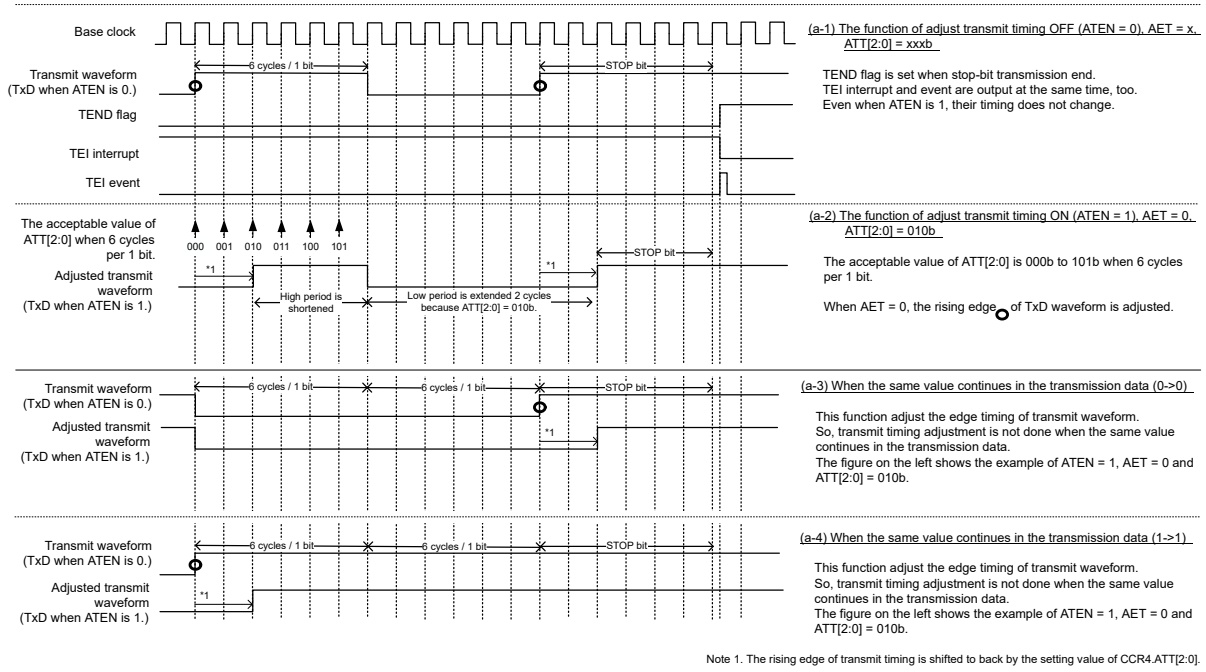
Figure 33.29 The explanation for the transmit waveform through a photo coupler

The operation explanation of adjustment the transmit timing

(a) In the case of the falling transfer time >> rising transfer time

In this case, the high period of a communication partner's reception waveform is made long, and the low period is made short.
Therefore, the LSI transmits the waveform with the falling edge relatively brought forward by adjusting the rising edge timing.
Then adjust value (ATT[2:0]) should be set to make equal the low-period/1 bit and high-period/1 bit for a communication partner.

This function's operation is explained as an example of 6 cycles/1 bit.



Note 1. The rising edge of transmit timing is shifted to back by the setting value of CCR4.ATT[2:0].

Figure 33.30 The adjustment operation explanation for the transmit timing when AET is 0

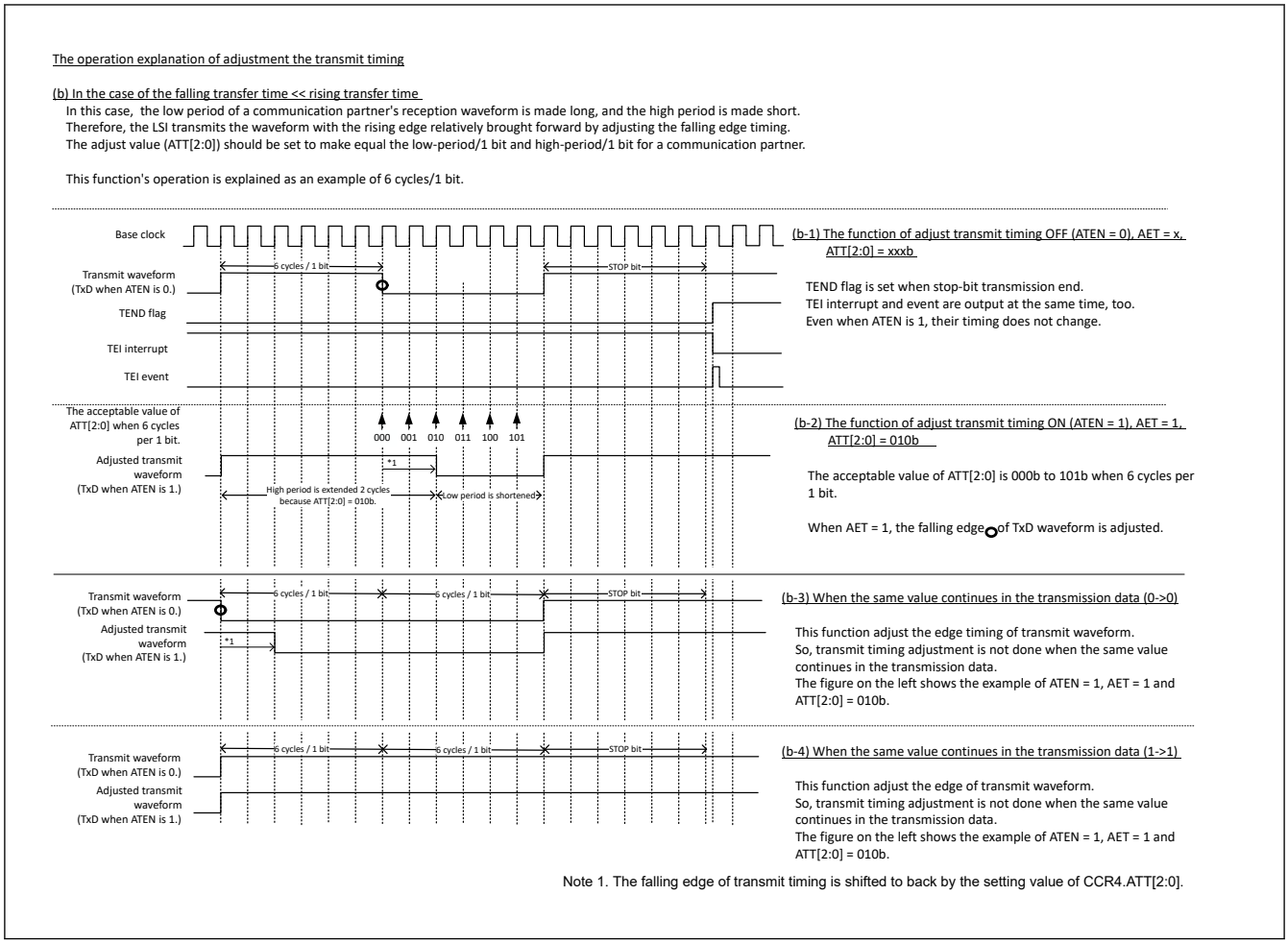


Figure 33.31 The adjustment operation explanation for the transmit timing when AET is 1

33.5 Multi-Processor Communication Function

Using the multi-processor communication functions enables to transmit and receive data by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle and when the multiprocessor bit is set to 0, it indicates the data transmission cycle. Figure 33.32 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. On receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two matches, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

RTS control cannot be used at the time of multi-processor communication function use, because this is a function corresponding to one-to-many communications.

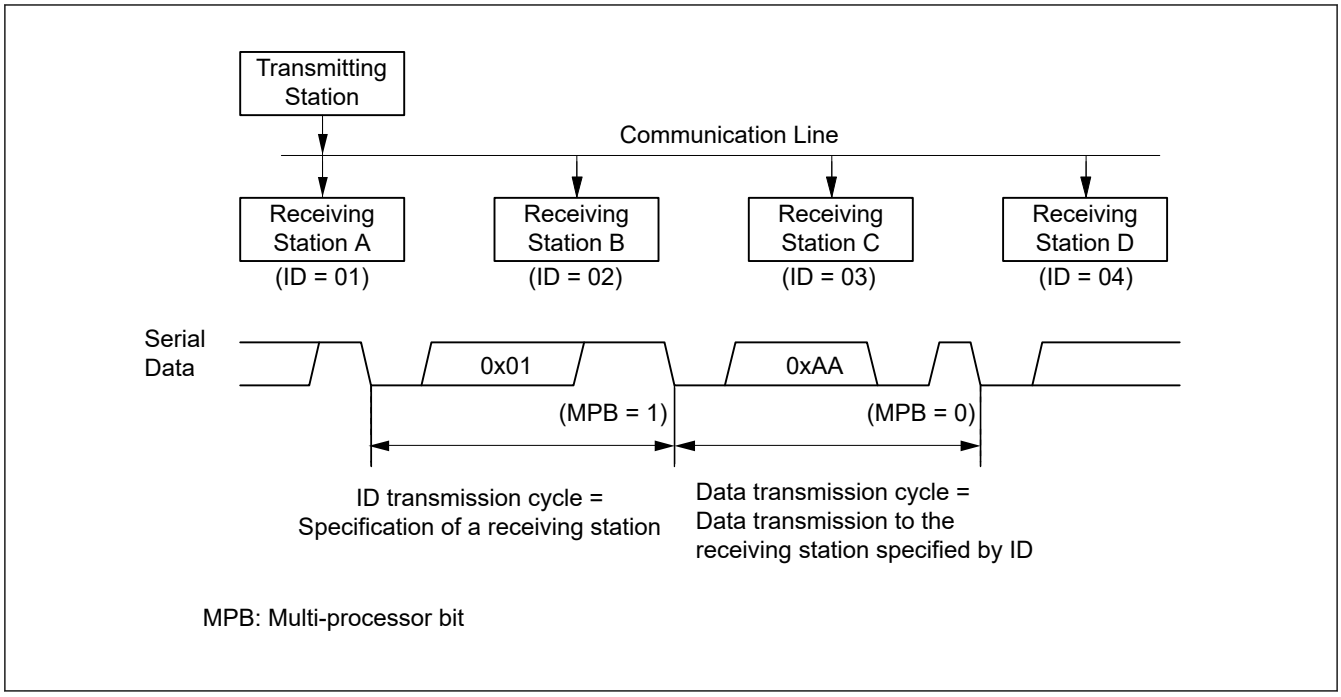


Figure 33.32 An example of communication using the multi-processor format (example of transmission of data 0xAA to receiving station A)

(1) Non-FIFO selected

For supporting this function, the SCI provides the MPIE bit in CCR0. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR, detection of a receive error, and setting the respective status flags RDRF, ORER, and FER in CSR are disabled until reception of data in which the multi-processor bit is set to 1.

On receiving a reception character in which the multi-processor bit is set to 1, the MPB bit in RDR is set to 1 and the MPIE bit in CCR0 is automatically cleared, thus returning to a non-multi-processor reception operation. At this time, an RXI interrupt is generated if the RIE bit in CCR0 is set.

When the multi-processor format is specified, specification of the parity bit is disabled.

Apart from this, there is no difference from the operation in the non-multi-processor asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the non-multi-processor asynchronous mode.

(2) FIFO selected

For transmission, it should write to TDR.MPBT (Multi-Processor Bit Transfer) which corresponds to transmit data in TDR.TDAT. For reception, Multi-processor bit that is a part of receive data, is stored to RDR.MPB, and receive data is stored to RDR.RDAT. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR.RDAT, detection of a receive error, and detection of DR, and setting the respective status flags RDRF, ORER, and FER in CSR are disabled until reception of data in which the multi-processor bit is set to 1.

On receiving a reception character in which the multi-processor bit is set to 1, the MPB bit in RDR is set to 1, and receive data is stored to receive FIFO (RDR.RDAT), and the MPIE bit in CCR0 is automatically cleared, thus returning to a non-multi-processor reception operation. At this time, an RXI interrupt is generated if the RIE bit in CCR0 is set.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the non-multi-processor asynchronous mode and non-FIFO selected.

33.5.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO selected

Figure 33.33 shows a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in TDR set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.

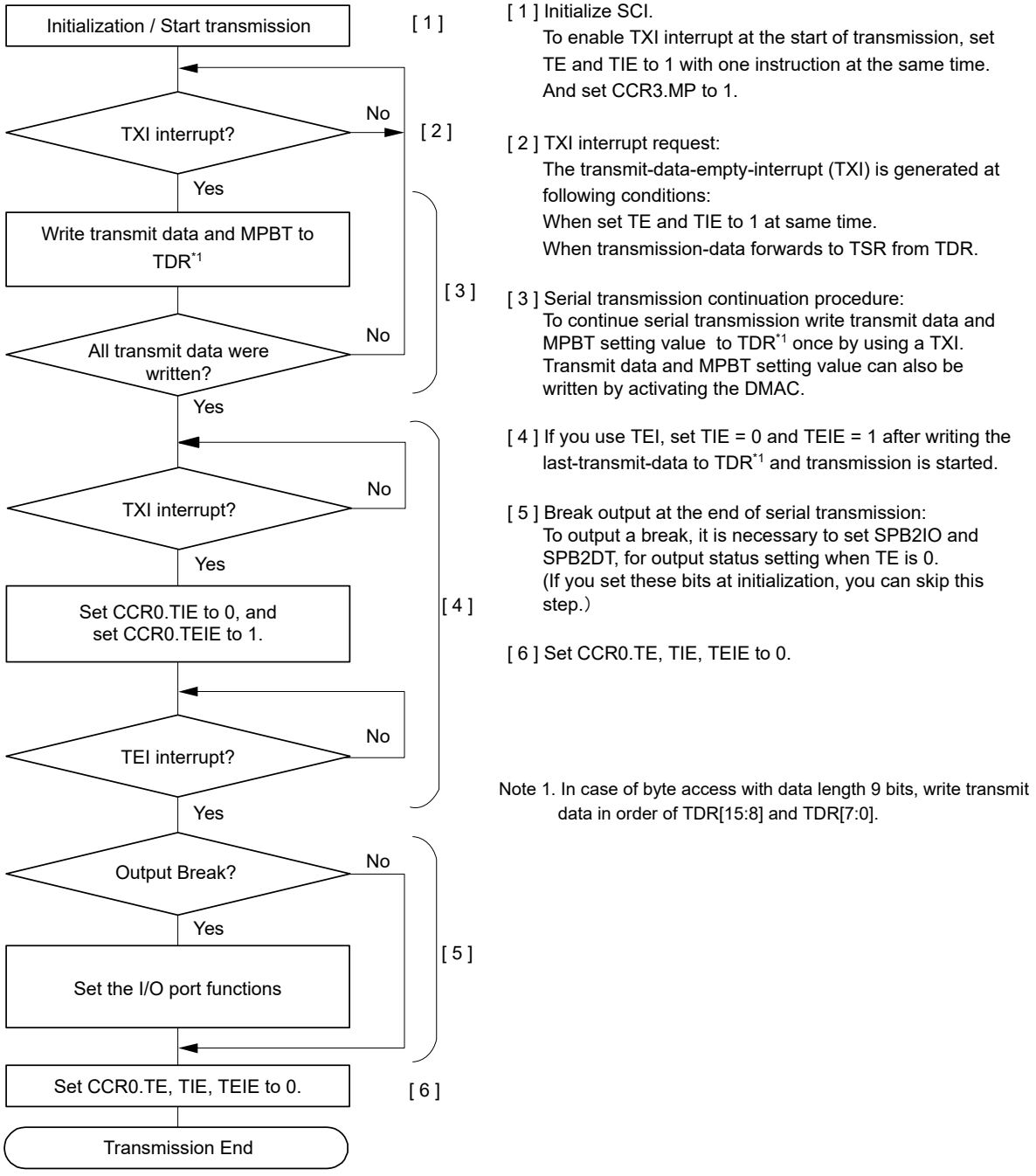


Figure 33.33 Example of multi-processor serial transmission flowchart (non-FIFO selected)

(2) FIFO selected

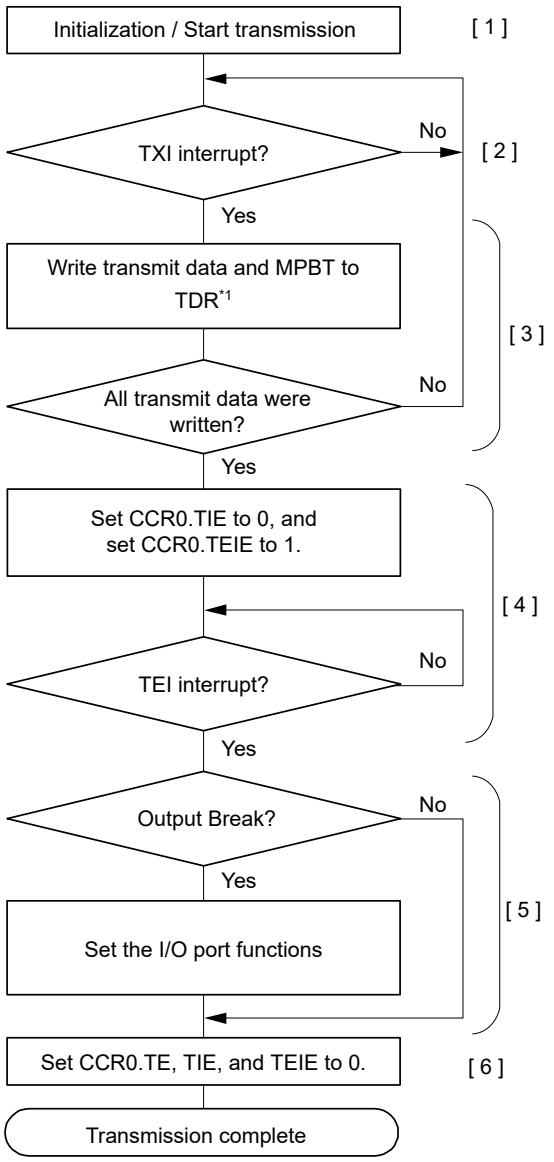
Figure 33.34 shows an example of data format that is written to transmit FIFO (TDR.TDAT) in multi-processor mode. Write MPBT in bit 9 of TDR. And write data to transmit FIFO (TDR.TDAT) corresponded to data length. It should write to 0 for unused bits.

Data Length	Register setting		Transmit data in TDR[15:0]															
	CCR3. CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	–	–	–	–	–	–	MPBT	–	–	TDAT[6:0]						
8 bits	1	0	–	–	–	–	–	–	MPBT	–	TDAT[7:0]							
9 bits	0	x	–	–	–	–	–	–	MPBT	TDAT[8:0]								

Note: The bits Indicated by "—" are not used. Therefore, those bits should be set to 0.

Figure 33.34 Data format in multi-processor mode that is written to transmit FIFO (TDR.TDAT)

Figure 33.35 shows a sample flowchart for multi-processor data transmission with FIFO selected. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in TDR set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode with FIFO selected.



[1] Initialize SCI.
To enable TXI interrupt at he start of transmission, set TE and TIE to 1 with one instructions at the same time. And set CCR3.FM and MP to 1.

[2] TXI interrupt request:
The transmit-data-empty-interrupt (TXI) is generated when transmission-data forwards to TSR from the transmit FIFO (TDR) and the number of transmit-data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[3] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data and MPBT setting value to the transmit FIFO (TDR) by using a TXI interrupt. Transmit data and MPBT setting value can also be written to the transmit FIFO (TDR) by activating the DMAC.

[4] If you use TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE, TIE, and TEIE to 0.

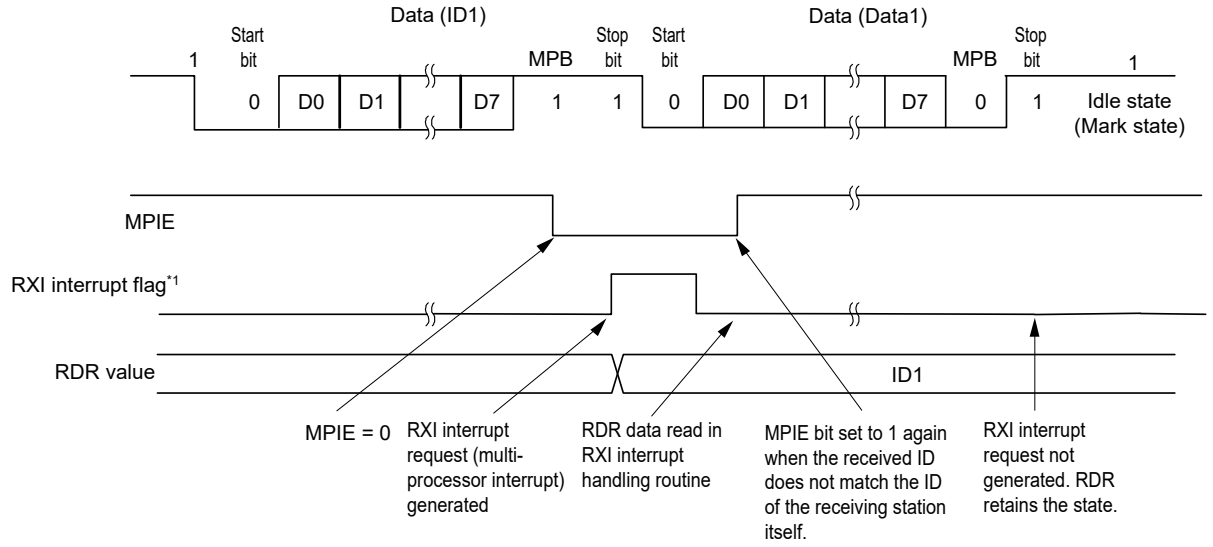
Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 33.35 Example of serial transmission flowchart in multi-processor mode (FIFO selected)

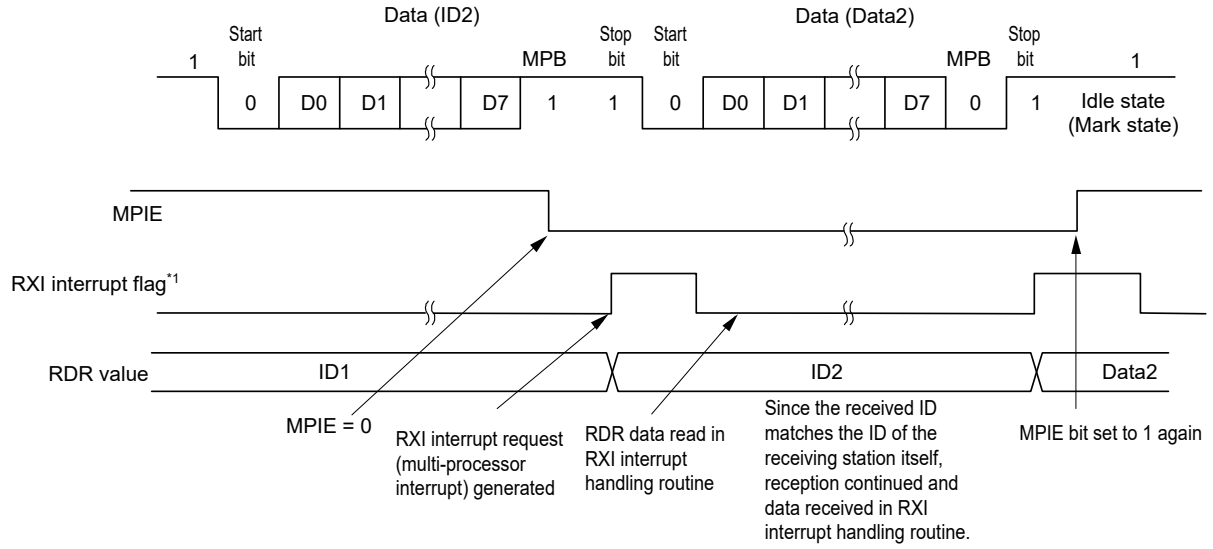
33.5.2 Multi-Processor Serial Data Reception

(1) Non-FIFO selected

Figure 33.37 and Figure 33.38 are sample flowcharts of multi-processor data reception. When the MPIE bit in CCR0 is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to RDR. At this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode. Figure 33.36 is the example of operation for reception.



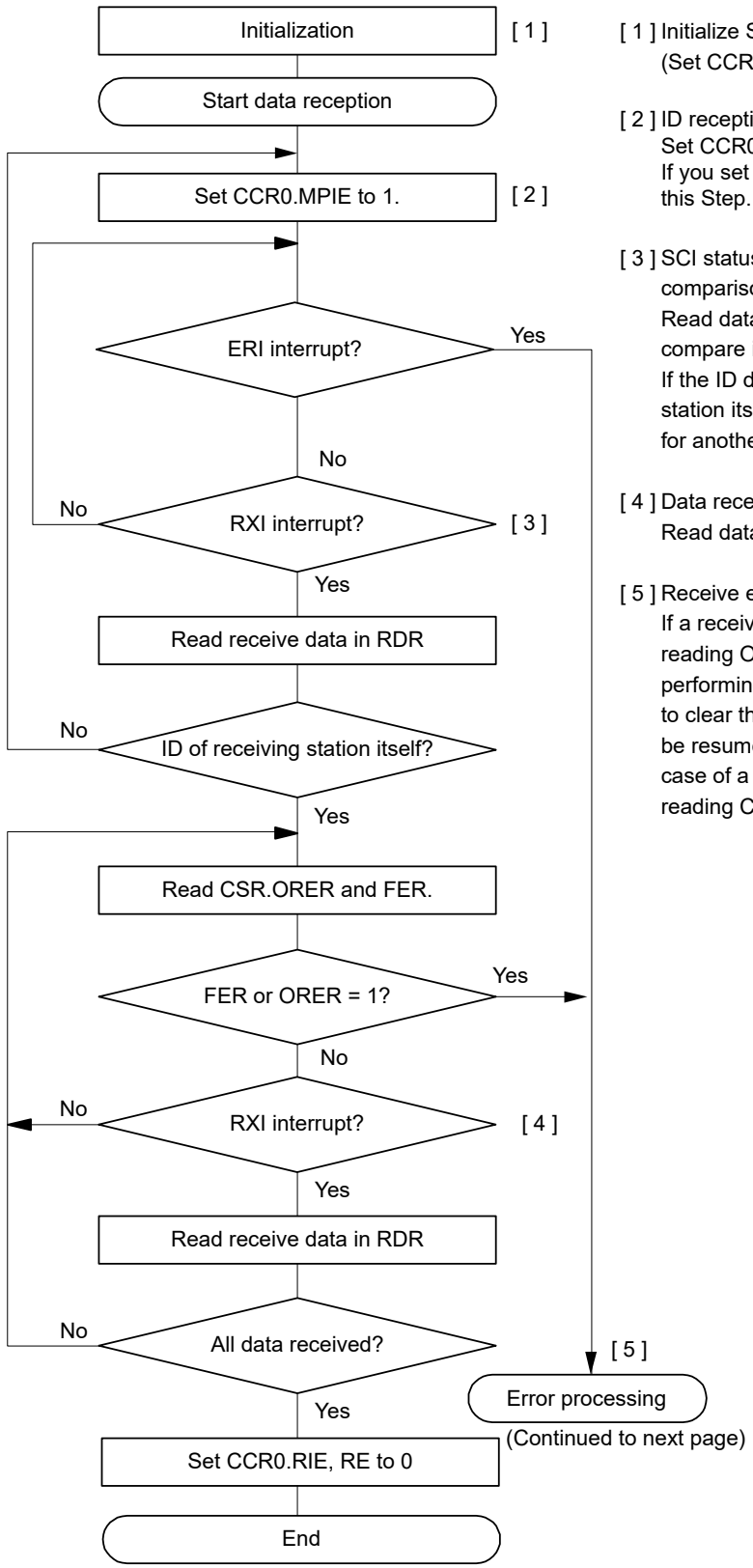
(a) When the received ID does not match the ID of the receiving station itself



(b) When the received ID matches the ID of the receiving station itself

Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.36 Example of SCI reception (8-bit data/multi-processor bit/1 stop bit)



[1] Initialize SCI.
(Set CCR0.RE, RIE to 1. And set CCR3.MP to 1.)

[2] ID reception cycle:
Set CCR0.MPIE to 1. and wait for ID reception.
If you set this bits at initialization ([1]), you can skip this Step.

[3] SCI status confirmation and reception and comparison of ID:
Read data in RDR at the first RXI interrupt and compare it with the ID of the receiving station itself.
If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another RXI interrupt request.

[4] Data reception at an RXI interrupt:
Read data in RDR once in the RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in CSR. After performing the appropriate error processing be sure to clear the ORER and FER flags. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading CSR.RXDMON bit.

Figure 33.37 Example of multi-processor serial reception flowchart (1) (non-FIFO selected)

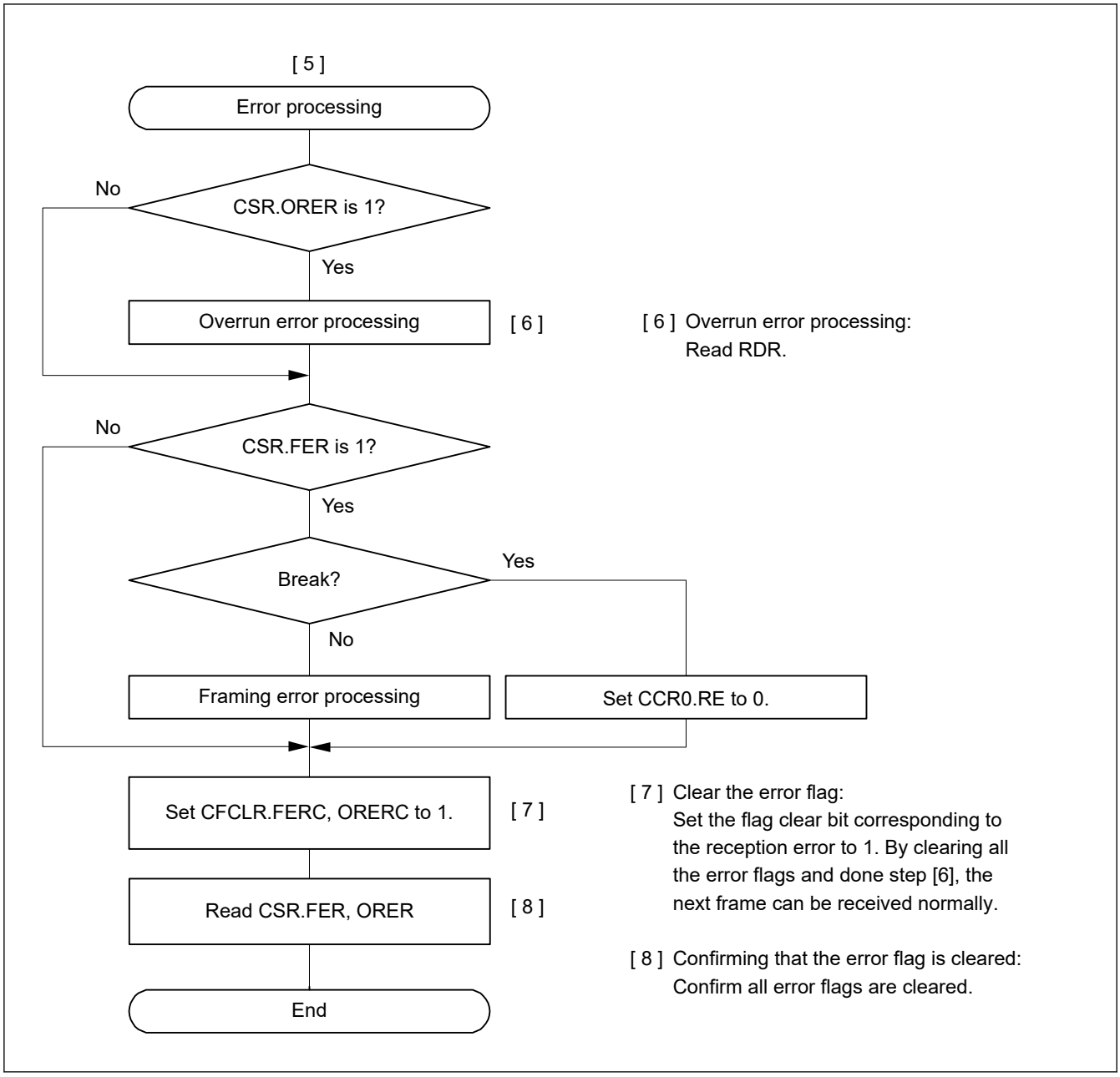


Figure 33.38 Example of multi-processor serial reception flowchart (2) (non-FIFO selected)

(2) FIFO selected

Figure 33.39 shows an example of data format that is stored to receive FIFO (RDR) in multi-processor mode. MPB is stored in bit 9 of RDR. 0 is stored to FPER and PER flag bit. Data is stored to receive FIFO (RDR) corresponded to data length. It is stored to 0 for unused bits. Reading the receive FIFO (RDR register) updates the FFER, FPER, MPB flags and receive data (RDAT[8:0]) in the receive FIFO (RDR register) with the next received data. The FER, PER, and ORER flags in the receive FIFO (RDR register) always reflect the status of the corresponding flags in the CSR and FRSR registers.

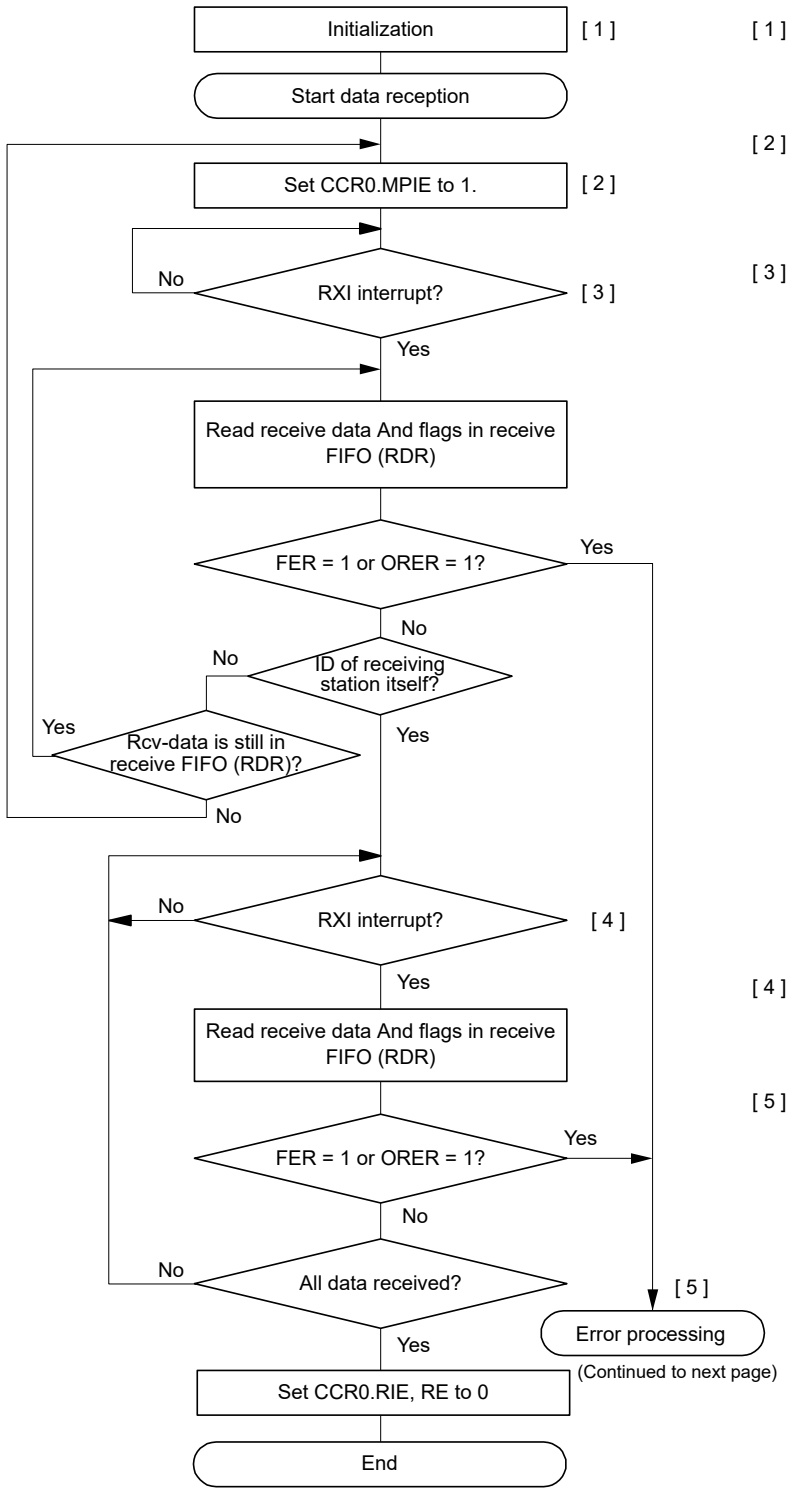
Data Length	Register Setting CCR3.CHR[1:0]		Receive data in RDR[31:0] b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	0	0	0	FFER	FPER	DR	MPB	0	0	RDAT[6:0]						
8 bits	1	0	0	0	0	FFER	FPER	DR	MPB	0	RDAT[7:0]							
9 bits	0	x	0	0	0	FFER	FPER	DR	MPB	RDAT[8:0]								
Data Length	CCR3.CHR[1:0]		b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
7 bits	1	1	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
8 bits	1	0	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
9 bits	0	x	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0

Note: When a 7-bit data length is selected, the RDAT[8:7] bits are always read as 00b.
When an 8-bit data length is selected, the RDAT[8] bit is always read as 0.

Figure 33.39 Data format in multi-processor mode that is stored to receive FIFO (FIFO selected)

Figure 33.40 shows a sample flowchart for multi-processor data reception at FIFO selected. When the MPIE bit in CCR0 is set to 1, reading the communication data is skipped until reception of the communication data in which the multiprocessor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data, MPB, and each errors are transferred to receive FIFO (RDR), and the MPIE bit in CCR0 is automatically cleared, thus returning to a normal reception operation. After a framing error occurred and CSR.FER flag is set to 1, but SCI continues data reception.

The other operations are the same as the operations in asynchronous mode at non-FIFO selected.



[1] Initialize SCI.
(Set CCR0.RE, RIE to 1.
And set CCR3.MP and FM to 1.)

[2] ID reception cycle:
Set CCR0.MPIE to 1. and wait for ID reception. If you set this bits at Initialization ([1]), you can skip this Step.

[3] SCI status confirmation and reception and comparison of ID:
The subsequent receive data including the data with MPB bit 1 is stored in receive FIFO (RDR). When the number of receive data stored in the receive FIFO (RDR register) reaches or exceeds the receive FIFO threshold, the CSR.RDRF flag is set to 1 and an RXI interrupt request is generated. When quantity of data stored in the receive FIFO data register (RDR) falls the specified receive triggering number and received data is equal to or greater than 1, and that no next data has been received yet after the elapse of 15 etus from the last stop bit, FRSR.DR is set to 1. Read data in receive FIFO (RDR) at the first RXI interrupt, and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, it reads until the data with MPB = 1, and compares next ID. If it is not a data with MPB = 1 in receive FIFO (RDR), it is set the MPIE bit to 1 again, and wait for another RXI interrupt request.

[4] Data reception at an RXI interrupt:
Read data in the receive FIFO (RDR) once in the RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in CSR. After performing the appropriate error processing be sure to clear the ORER and FER flags. In the case of a framing error, a break can be detected by reading the CSR.RXDMON bit.

Figure 33.40 Example flowchart of serial reception in multi-processor mode (1) (FIFO selected)

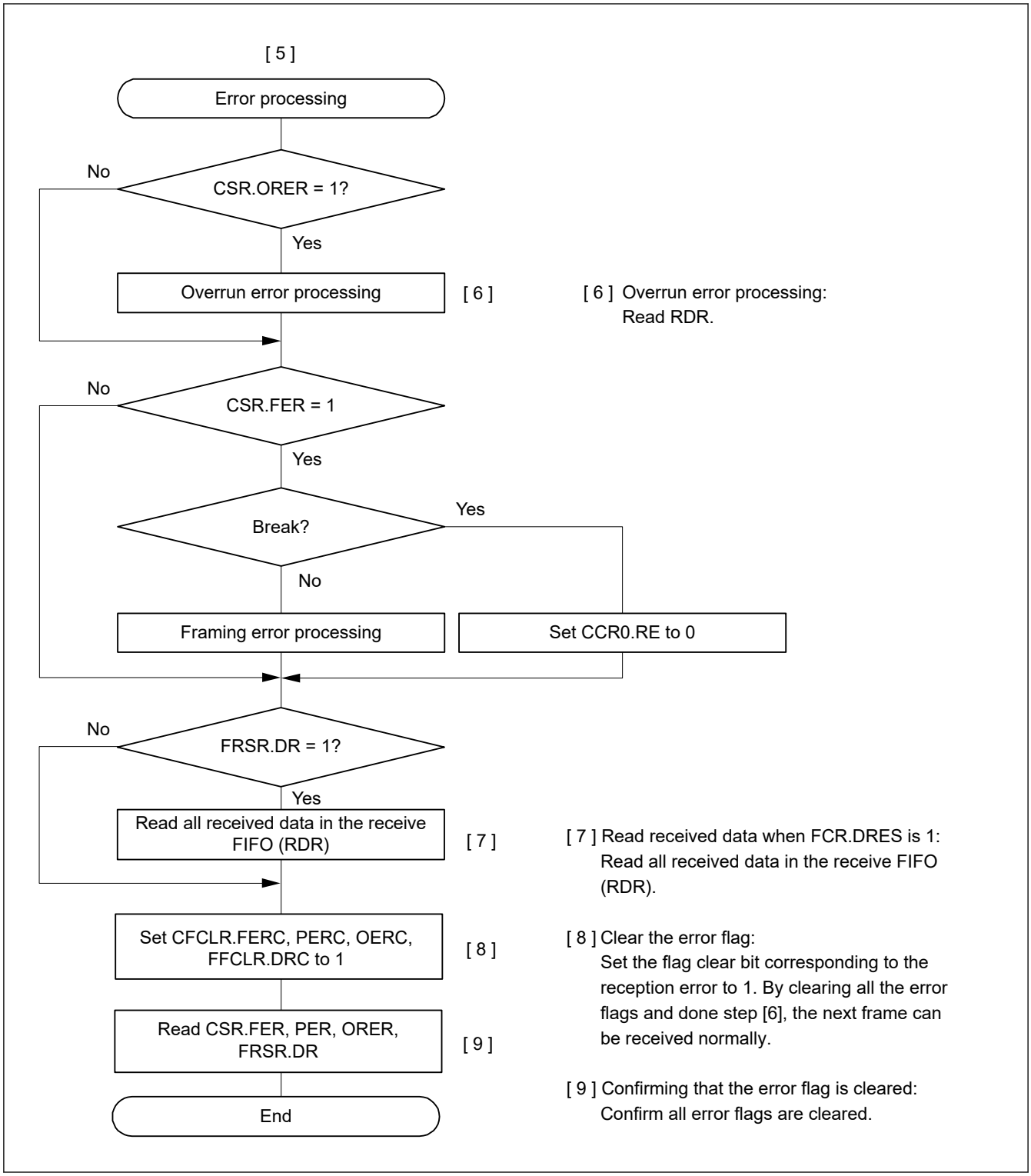


Figure 33.41 Example flowchart of serial reception in multi-processor mode (2) (FIFO selected)

33.6 Manchester Mode

In Manchester mode, the transmit or receive serial data is coded in Manchester encoding.

Figure 33.42 shows the conceptual image of Manchester encoding.

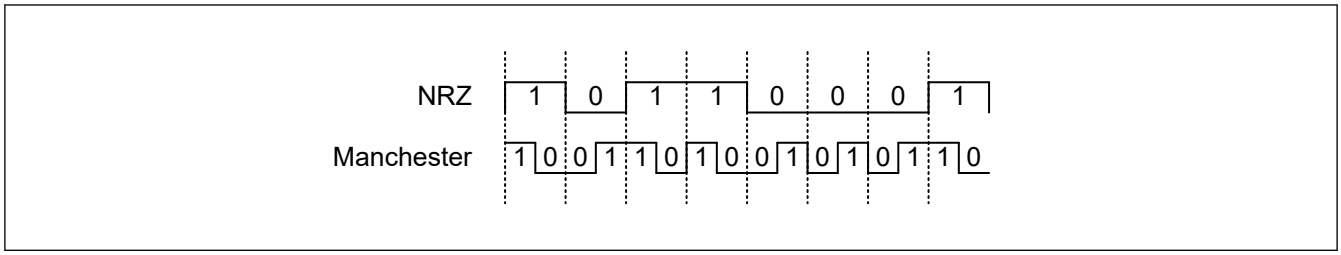


Figure 33.42 Example of Manchester encoding

In Manchester mode, a preface and a start bit area are added to the transmit data in the register to configure a transmit frame. For transmission, data is encoded in Manchester encoding. When data is received, frames having the same format as transmitted frames are detected and Manchester decoding is performed.

33.6.1 Frame Format

Figure 33.43 shows the frame format in Manchester mode. In the upper half of the figure, relevant setting registers are shown. The preface area and the data area are encoded in Manchester encoding.

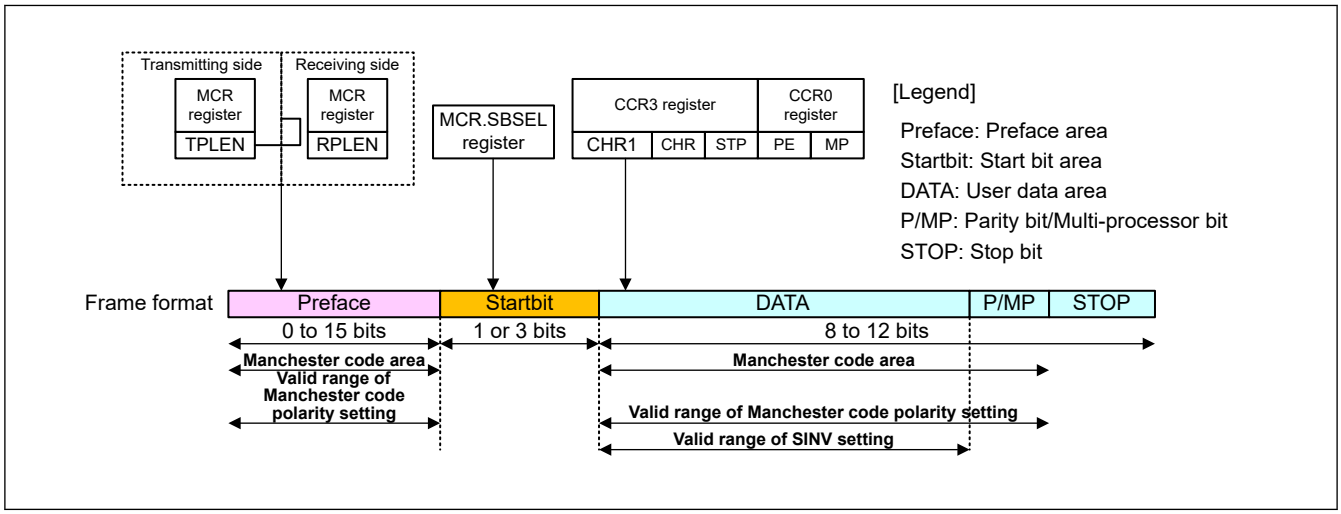


Figure 33.43 Frame format in Manchester mode

(1) Preface area

This is a fixed pattern area located at the beginning of each frame. Different registers are used to set the preface area for transmission and reception. The preface length is determined by setting MCR.TPLEN[3:0] for transmission. It is determined by setting MCR.RPLEN[3:0] for reception.

If it is set to 0, the transmit preface is disabled and is not added. If it is set to 1d to 15d, a preface whose length is determined by this setting is added. (For example, if it is set to 1d, a 1-bit preface is added. If it is set to 15d, a 15-bit preface is added.)

In addition, the preface pattern can be changed by setting, and it can be selected from four types of patterns by setting MCR.TPPAT[1:0] for transmission and MCR.RPPAT[1:0] for reception.

Figure 33.44 shows how the preface pattern is set. The preface area and the start bit area are added for each communication frame.

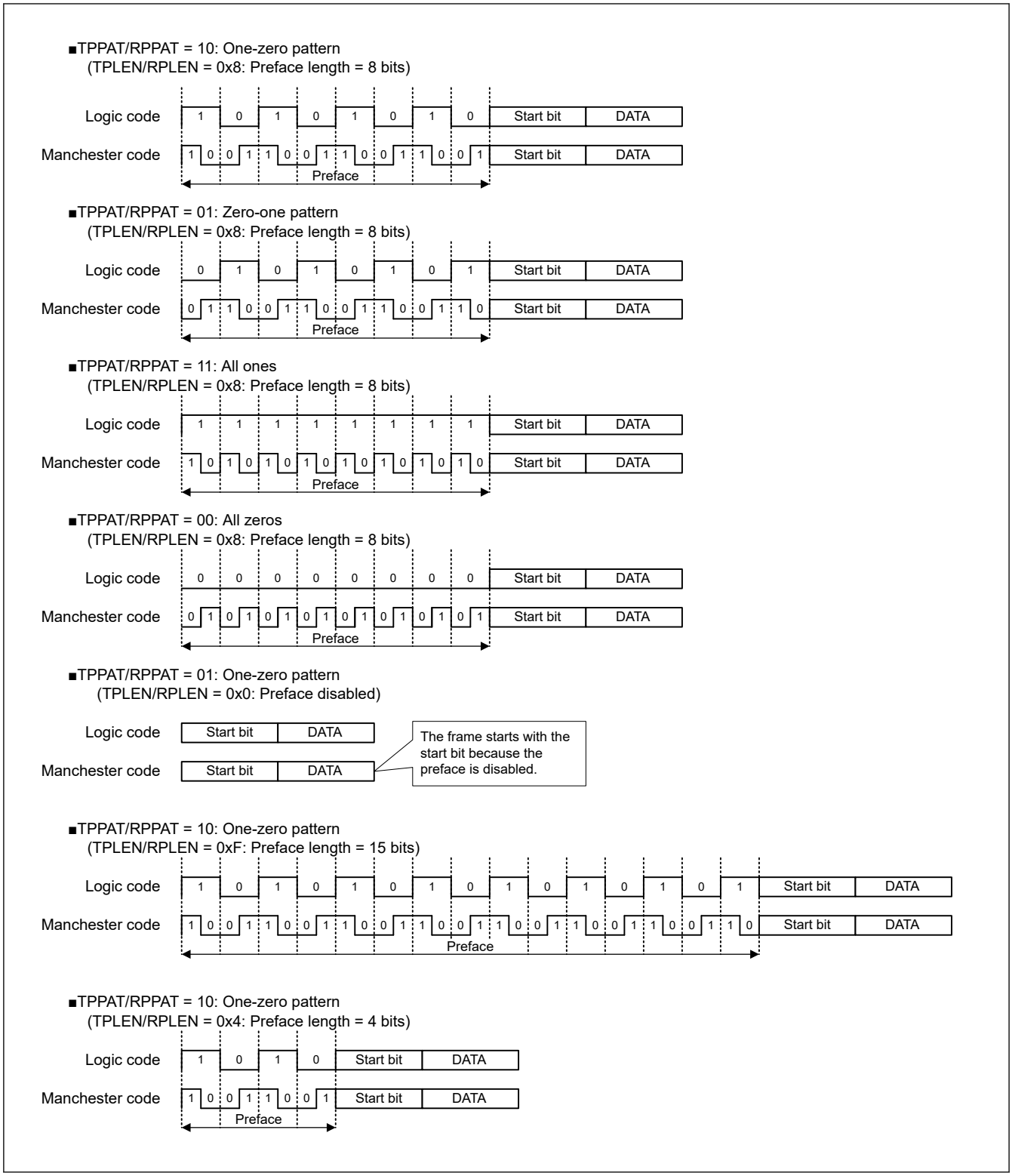


Figure 33.44 Preface pattern setting example

(2) Start bit area

This is an area indicating the start of valid data in a frame. It is added after the preface area. The start bit length is determined by MCR.SBSEL setting. When MCR.SBSEL = 0, the start bit length is 1 bit. When MCR.SBSEL = 1, the start bit length is 3 bits.

When MCR.SBSEL = 1, the SYNC type can be selected from command SYNC and data SYNC. Command SYNC means the three start bits are added as a one-to-zero transition. Data SYNC means the three start bits are added as a zero-to-one

transition. The SYNC type is determined by the MCR.SYNSEL, MCR.SYNVAL, and TDR.TSYNC settings. (When receiving, the received result is applied to MSR.RSYNC.)

When MCR.SBSEL = 0, the start bit is added as a zero-to-one or one-to-zero transition. The selection is determined by the MCR.SYNVAL setting.

The MCR.SYNSEL bit specifies the destination to be referred to when setting for transmission. When the MCR.SYNSEL bit is set to 1, the MCR.SYNVAL setting is referred to. When the MCR.SYNSEL bit is set to 0, the TDR.TSYNC setting is referred to.

Figure 33.45 shows the state of the start bit area according to the settings in the MCR.SYNSEL, MCR.SYNVAL, and TDR.TSYNC registers in the case of transmission. Figure 33.46 shows that in the case of reception. The start bit(s) is not affected by the MCR.TMPOL or MCR.RMPOL setting.

※ d.c. = don't care, Preface = preface area, START = start bit area, Data = user data area,
P = parity bit, MP = multi-processor bit, STOP = stop bit

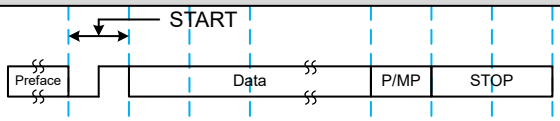
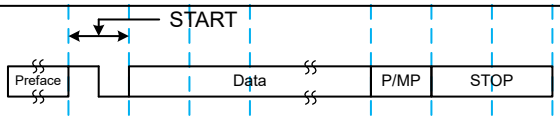
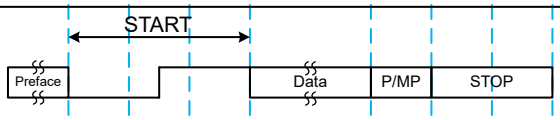
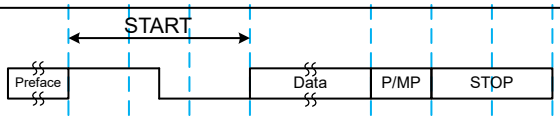
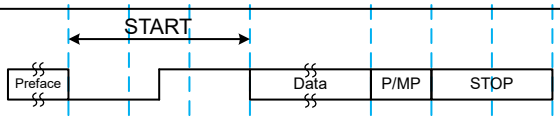
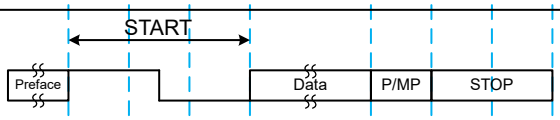
Register setting				Output signal	Output Start bit
MCR			TDR	Output TxD waveform	
SBSEL	SYNSEL	SYNVAL	TSYNC		
0	d.c.	0	d.c.		1 bit 0 to 1 transition
0	d.c.	1	d.c.		1 bit 1 to 0 transition
1	0	0	d.c.		3 bit 0 to 1 transition (DATA SYNC)
1	0	1	d.c.		3 bit 1 to 0 transition (COMMAND SYNC)
1	1	d.c.	0		3 bit 0 to 1 transition (DATA SYNC)
1	1	d.c.	1		3 bit 1 to 0 transition (COMMAND SYNC)

Figure 33.45 Preface pattern setting example

d.c. = don't care, Preface = Preface area, START = Start bit area, Data = Data area
P = Parity bit, MP = Multi-processor bit, STOP = Stop bit

* Data other than the start bit is assumed to be normal.

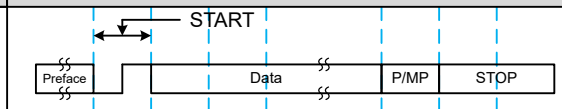
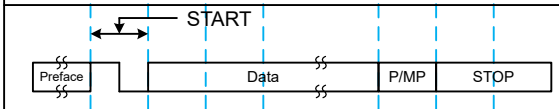
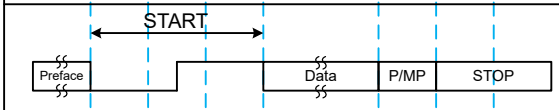
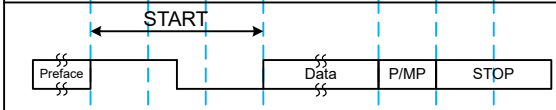
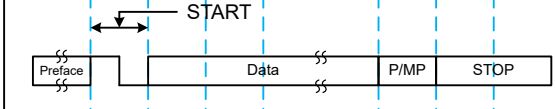
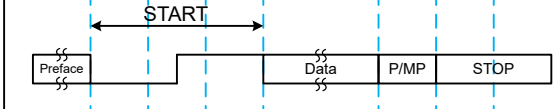
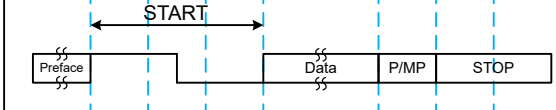
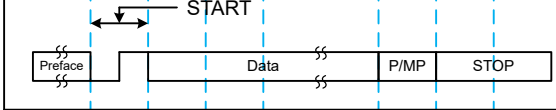
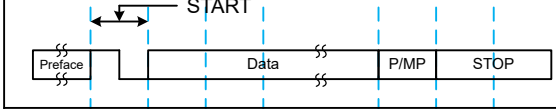
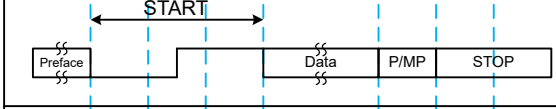
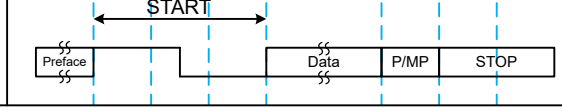
Register setting				Input signal	Start bit detection result*	Register indication
MCR			TDR	RxD input waveform		MSR.RSYNC
SBSEL	SYNSEL	SYNVAL	TSYNC			
0	d.c.	0	d.c.		Normal start bit (1 bit: 0-to-1 transition)	0
					Start bit error	0
					Start bit error	0
					Start bit error	0
0	d.c.	1	d.c.		Start bit error	0
					Normal start bit (1 bit: 1-to-0 transition)	0
					Start bit error	0
					Start bit error	0
1	d.c.	d.c.	d.c.		Start bit error	0
					Start bit error	0
					Data SYNC	0
					Command SYNC	1

Figure 33.46 Settings related to and judgment of the start bit area at reception

(3) DATA

Since the format of the data area is the same as that of the asynchronous mode, see section 33.4.1. Serial Data Transfer Format. As shown in Figure 33.43, Frame Format in Manchester Mode, the stop bit is not included in the Manchester encoding range.

33.6.2 Clock

As the transfer clock in Manchester mode, the clock generated by the on-chip baud rate generator is used by setting the CCR2.CKS[1:0] bit.

Also it is possible to set the oversampling (transfer rate of one-bit period) by CCR2.ABCS bit. When the CCR2.ABCS bit is set to 0, oversampling x16 is selected with the one-bit period being 16 cycles of the base clock. When the CCR2.ABCS bit is set to 1, oversampling x8 is selected with the one-bit period being 8 cycles of the base clock.

33.6.3 SCI Initialization of (Manchester Mode)

Before transferring data, write 0 to CCR0.TE and CCR0.RE (or write the initial value to CCR0), and initialize the SCI following the example of flowchart shown in Figure 33.47.

Be sure to write 0 to CCR0.TE and CCR0.RE before changing the operation mode or communication format.

Note that setting the CCR0.RE bit to 0 initializes none of the ORER, FER, PER, MER, RDRF, RDF, SYER, PFER, and SBER flags and the RDR registers.

Note also that switching the value of CCR0.TE from 1 to 0 or 0 to 1 when CCR0.TIE is 1 generates a TXI interrupt request.

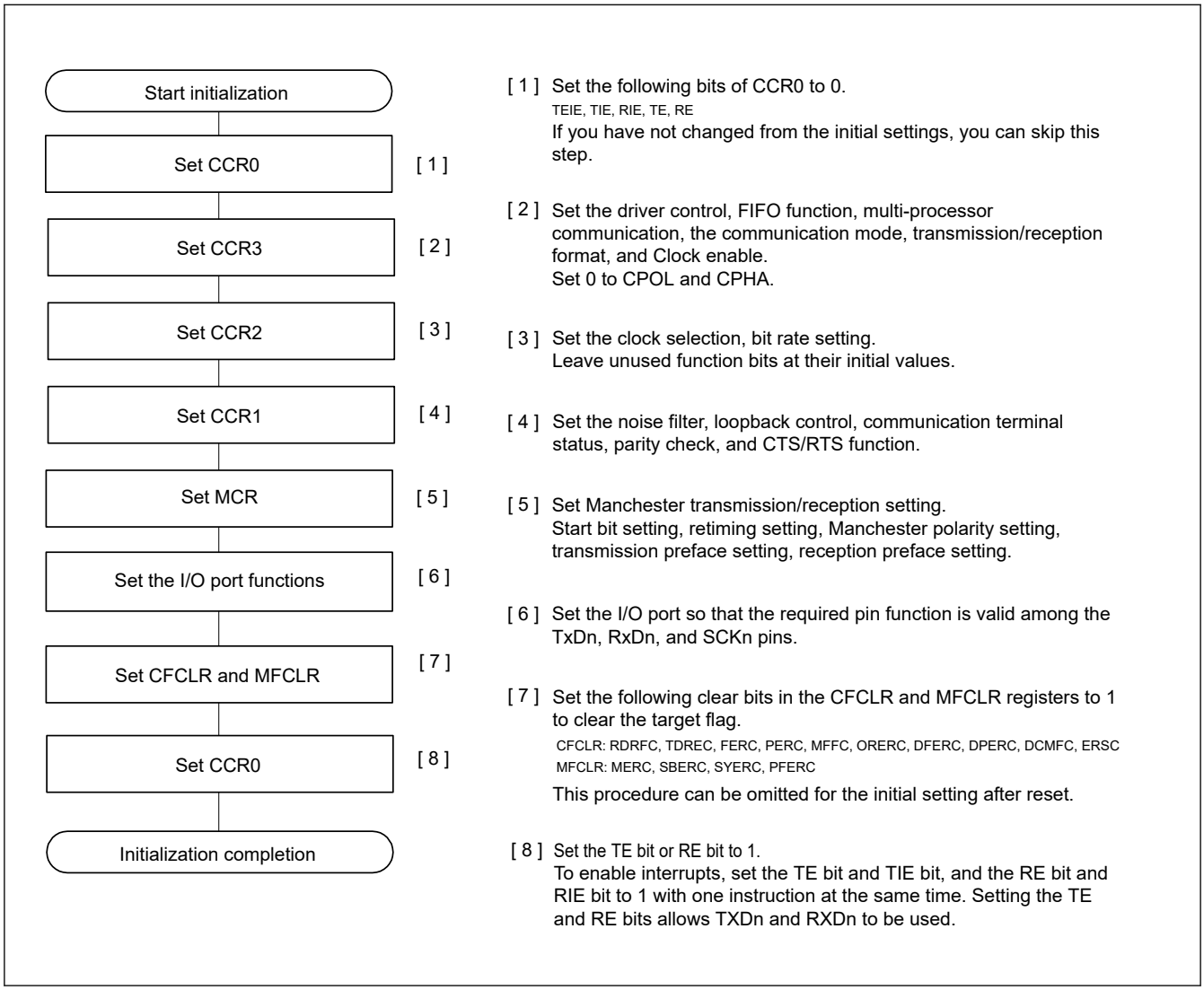


Figure 33.47 SCI initialization flow in Manchester mode

33.6.4 Double Speed Operation

When the ABCS bit in CCR2 is set to 1 and eight pulses of the base clock for a 1-bit period is selected, the SCI operates on the bit rate twice that of when ABCS is set to 0.

When the BGDM bit in CCR2 is set to 1, the cycle of the base clock is reduced to half and the SCI operates on the bit rate twice that of when BGDM is set to 0.

When the ABCS bit in CCR2 and the BGDM bit in CCR2 are set to 1, the SCI operates on the bit rate four times that of when the ABCS bit in CCR2 and the BGDM bit in CCR2 are set to 0.

33.6.5 CTS and RTS Functions

The CTS function controls transmission using the CTSn# pin input. Setting the CTSE bit in CCR1 to 1 enables the CTS function. The CTS/RTS pin can be set as a multiplexed pin which allows one pin to be used for either function, or as dedicated pins with each pin for a single function. Use the CTSPEN bit in CCR1 for this setting.

When the CTS function is enabled, reception starts only when the CTSn# pin is at the low level.

Even if the level of CTSn# pin goes High after transmission starts, does not affect transmission of the current frame, which continues.

The RTS function uses output on the CTSn# pin to request transmission. When SCI is ready to receive, it outputs a low level to the RTSn# pin, Conditions for output of the low level and high level are as follows:

[Conditions for low-level output]

When all conditions listed below are satisfied:

- The value of the CCR0.RE is 1.
- SCI is ready to receive.
- There is no received data yet to be read.
- All of the following flags are set to 0:
  CSR.ORER, FER, PER and MSR.MER, SBER (when SBEREN = 1), SYER (when SYEREN = 1), PFER (when PFEREN = 1)

[Conditions for high-level output]

When the conditions for low output are not satisfied.

33.6.6 Serial Data Transmission in Manchester Mode

The SCI encodes data in Manchester encoding and sends the resultant data in Manchester mode.

When the polarity setting (MCR.TMPOL) is set to 0, logic 0 is coded as a zero-to-one transition in Manchester code and logic 1 is coded as a one-to-zero transition in Manchester code.

When the polarity setting (MCR.TMPOL) is set to 1, logic 0 is coded as a one-to-zero transition in Manchester code and logic 1 is coded as a zero-to-one transition in Manchester code.

For this reason, a level transition occurs with the Manchester encoded data in the middle of individual logic data. (See Figure 33.42.)

The transmitter constructs transmit frames in a specific format by adding a preface area to data and setting the start bit(s) according to the polarity setting and sends resultant serial data.

For details on the frame format, see section 33.6.1. Frame Format.

Figure 33.48 shows the flowchart in transmission. At transmission starts, set the CCR0.TIE and CCR0.TE bits to 1 simultaneously with one instruction. Then, a TXI interrupt request is generated.

Figure 33.49, Figure 33.50, and Figure 33.51 show examples of the operation for serial transmission in Manchester mode.

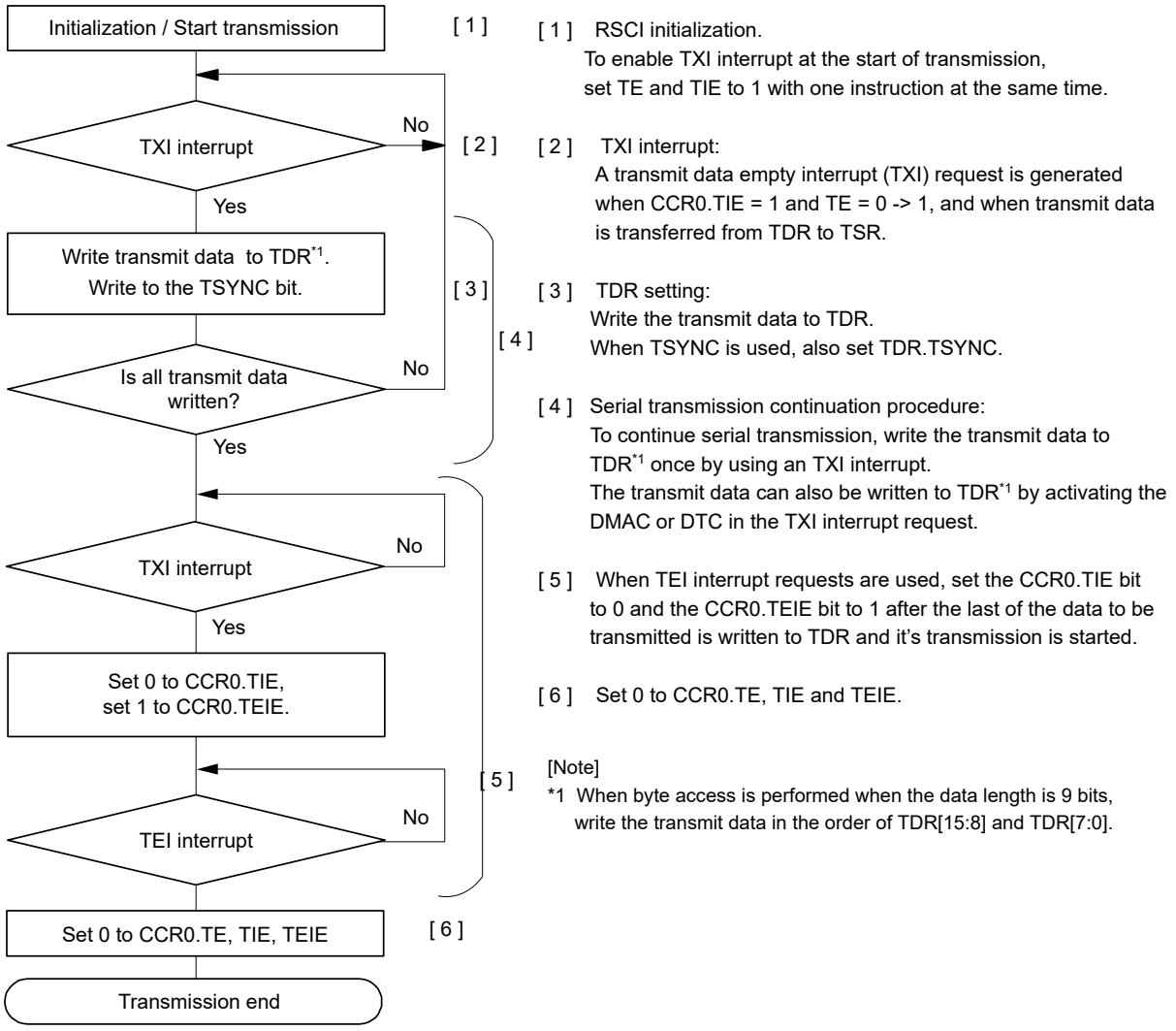


Figure 33.48 Example of serial transmission flowchart in Manchester mode

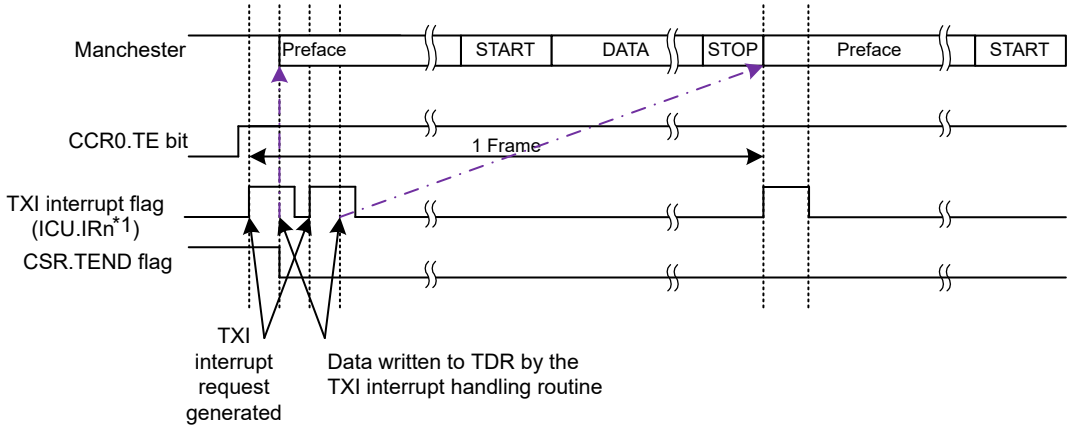


Figure 33.49 Example of Start-of-Transmission operation for serial transmission in Manchester mode (with Preface but without the CTS Function)

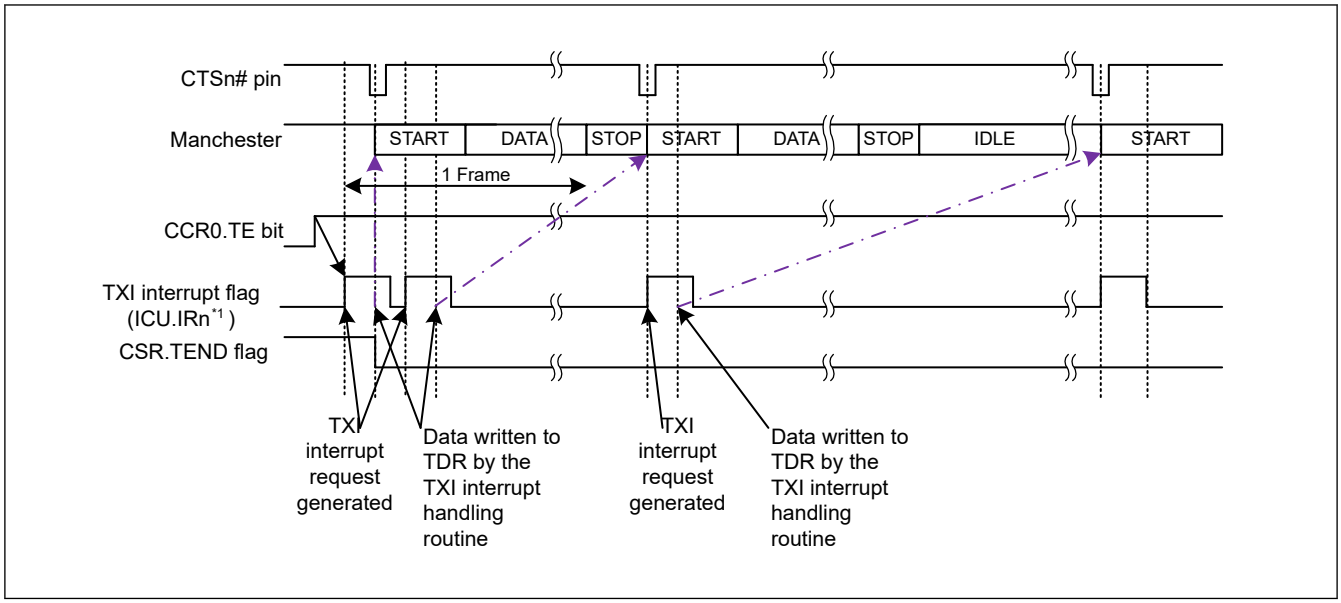


Figure 33.50 Example of Start-of-Transmission operation for serial transmission in Manchester mode (without Preface but with the CTS Function)

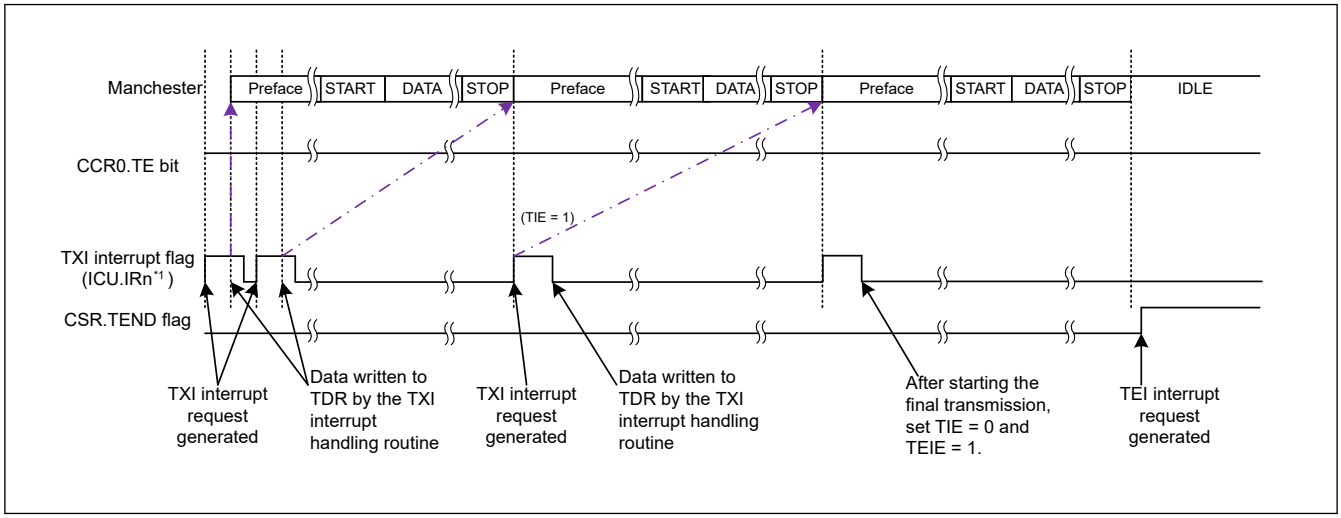


Figure 33.51 Example of End-of-Transmission operation for serial transmission in Manchester mode (with Preface but without the CTS Function)

33.6.7 Serial Data Reception in Manchester mode

In Manchester mode, the SCI operates on a base clock with a frequency of 16 times[*1] the bit rate. Reception starts by sampling the falling edges of received data at the base clock. As shown in Figure 33.52, reception starts at a falling edge of the received data and it continues if the received data keeps low for the duration of 1/4 bit. If the received data goes high within the duration of 1/4 bit, the SCI judges it as an error and waits for a falling edge again.

If a high level is expected in the first half of a bit in the received data, the SCI judges a low level that continues for one base clock cycle as an error and ignores the change to the low level.

Note 1. This is the case when CCR2.ABCS = 0. When CCR2.ABCS = 1, the SCI operates on a base clock with a frequency of 8 times the bit rate.

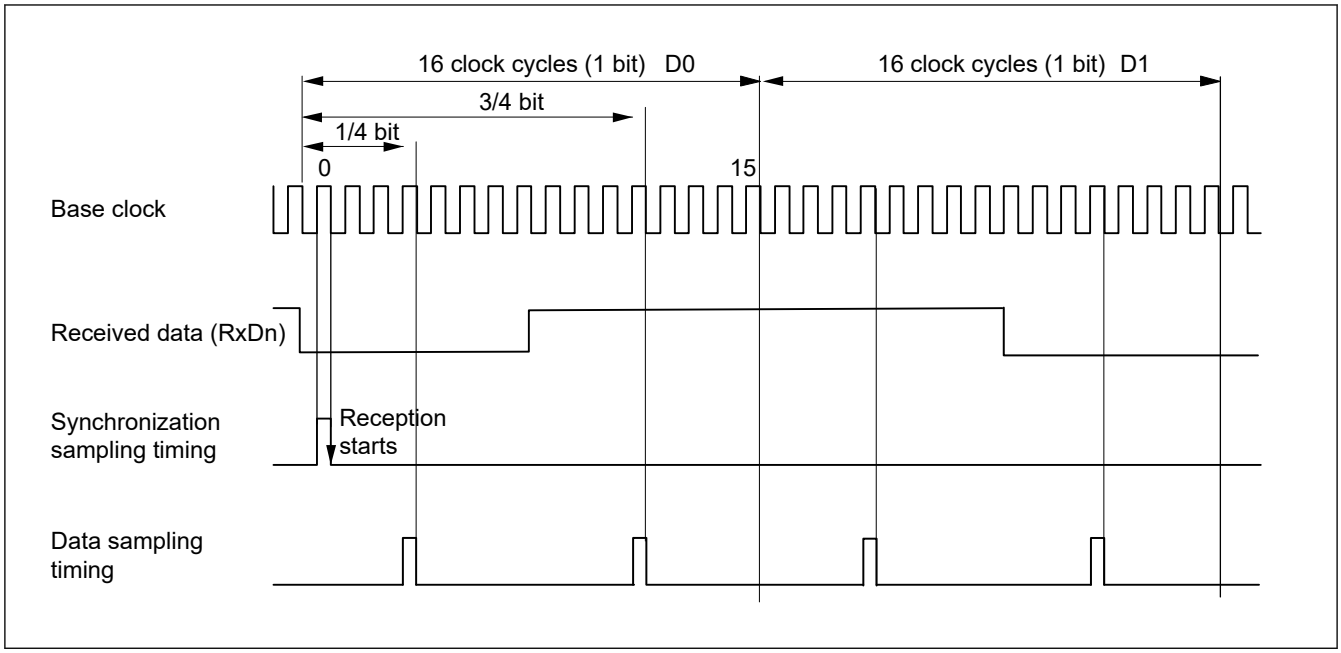


Figure 33.52 Data reception sampling timing in Manchester mode

In Manchester mode, data reception starts with detection of a preface and start bit area.

The SCI checks the input from the RxD pin to see whether a preface is added based on the value of MCR.RPLEN.

If the preface is disabled (MCR.RPLEN = 0), it moves on to the detection of a start bit area without detecting a preface.

When a preface is enabled, it identifies a preface pattern setting according to the set value in MCR.RPPAT, and compares it with the RxD input for a pattern match to detect a preface pattern.

Upon detection of a preface pattern match, it judges it as a normal preface and moves on to the detection of a start bit area.

If detecting a preface pattern mismatch or a Manchester code error in the preface area, it judges it as a preface error and asserts a preface error (PFER).

For start bit detection, the SCI selects an expected value based on the register settings (MCR.SBSEL and SYNVAL), compares it with the RxD input for a pattern match to detect a start bit area. Upon detection of a start bit pattern match, it judges it as a normal start bit area and moves on to the data processing.

Only when a preface and a start bit area are detected normally, it moves on to the next phase of data reception.

Upon detection of a start bit pattern mismatch, it asserts a start bit error flag (SBER).

In data processing, the SCI shifts the data by the expected received data length based on the register settings (CCR3.CHR[1:0]) through the RSR register. If two sampling points in a bit of the received data are identical, the SCI judges this as a Manchester code error. For details, see section 33.6.11. Errors in Manchester Mode (1) Manchester error.

When the parity function is disabled (CCR1.PE = 0), the SCI moves on to the next phase of stop bit detection. When the parity function is enabled (CCR1.PE = 1), the SCI performs parity checking. If detecting a parity error, it asserts a parity error flag (PER), and then moves on to stop-bit detection.

In stop bit detection, the SCI checks the following in the stop bit area of the received frame:

It has two sampling points in a bit. If both points are at the high level, the bit is recognized as a normal stop bit and the data is stored in the RDR register. At least one low-level point is judged as an abnormal stop bit, causing a framing error flag (FER) to be set. Even when an error is detected, the received data is stored in the RDR register as abnormal data.

Figure 33.53 shows an example of the operation for serial data reception in Manchester mode.

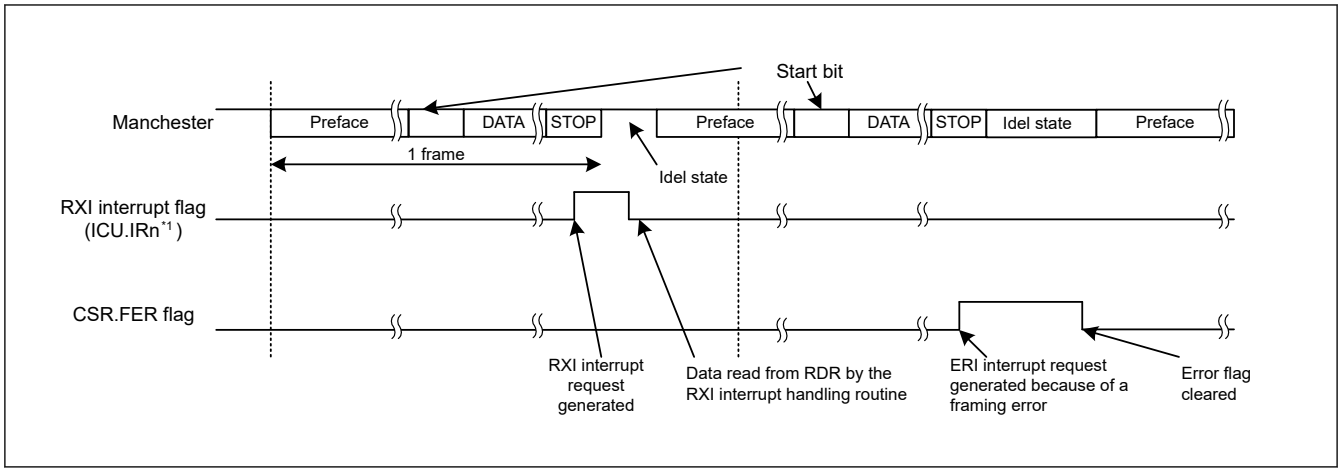


Figure 33.53 Example of operation for serial data reception in Manchester mode (with a Preface)

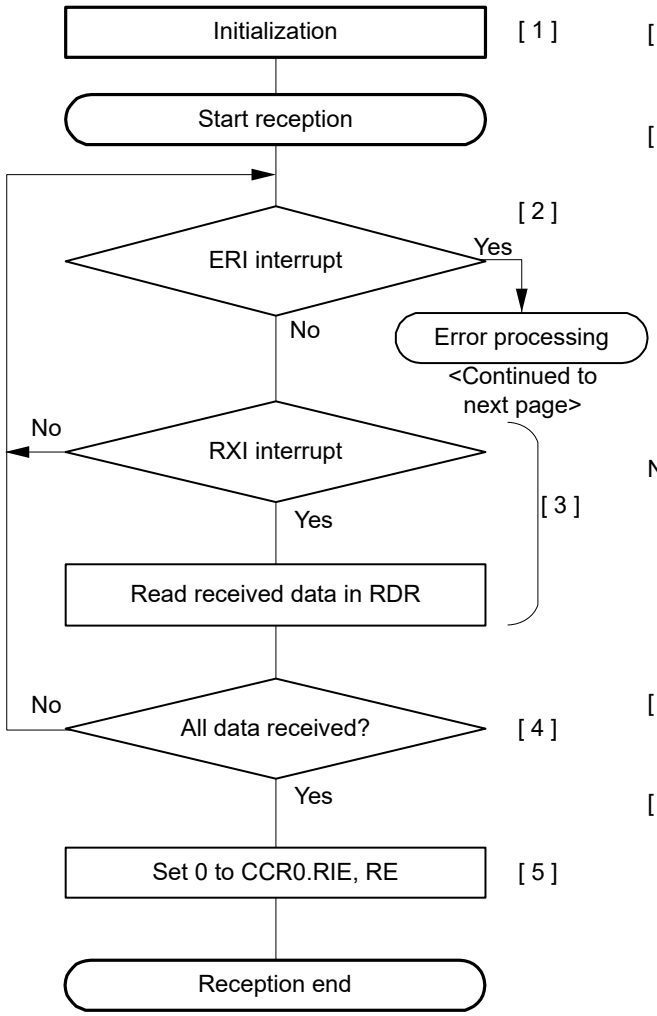
For the state of each status flag in the CSR register and RxD input processing when a receive error is detected, see section 33.6.11. Errors in Manchester Mode.

If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated.

Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, PER, MER, SYER*1, PFER*1, SBER*1 flags to 0 before resuming reception. Also, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the CCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.

Figure 33.54 and Figure 33.55 show examples of serial data reception flowchart in Manchester mode.

Note 1. Effective when the corresponding bit is enabled.



[1] RSCI initialization.
(Set CCR0.RE = 1 and RIE = 1)

[2] Receive error processing:
When a receive error occurs, RSCI generates an ERI interrupt. An error is identified by reading the ORER, PER, FER in CSR and MER, SBER, SYER, PFER in MSR. After performing the appropriate error processing, always set the ORER, PER, FER, MER, SBER, SYER, PFER flags to 0. Reception cannot be resumed if any of these flags is set to 1.

Note: Each of the PFER, SYER, SBER bits in MSR has an enable bit (PFEREN, SYEREN, and SBEREN in MECR). If the corresponding enable bit is set to 0 (disable), the flag value does not affect the communication. Perform error processing as required.

[3] Read the received data in RDR once in the RXI interrupt handling routine.

[4] Serial reception continuation procedure:
To continue serial reception, before the stop bit of a frame is received, read data from RDR in the RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC.

[5] Write 0 to CCR0.RIE and RE.

Figure 33.54 Example of serial data reception flowchart in Manchester mode (Normal reception)

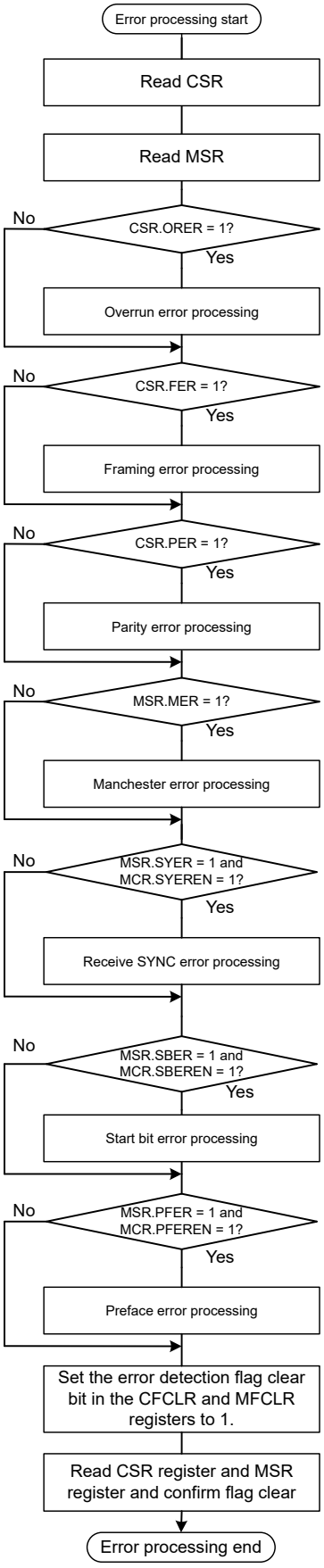


Figure 33.55 Example of serial reception flowchart in Manchester mode (Error processing)

33.6.8 Operation when Multi-processor bit is used

See section 33.5. Multi-Processor Communication Function (1) Non-FIFO selected for the operation in Manchester mode when using multi-processor mode because the operation is the same.

A preface and a start bit area are added to the frame format in Manchester mode.

See Figure 33.55 for error processing in Manchester mode for the reception flowchart. Refer to Table 33.31 for the operation status when detecting various errors.

33.6.9 Receive Retiming

This function corrects the timing for each central edge of the bit, taking advantage of the fact that each bit has an edge in the center in Manchester code.

The receive retiming function can be turned on or off by setting the ERTEN bit in the MCR register.

When the receive retiming function is turned off (MCR.ERTEN = 0), retiming is not performed, causing misalignment between the internal clock and the RxD input to be accumulated and the receive margin to be reduced.

When the receive retiming function is turned on (MCR.ERTEN = 1), retiming is performed for the preface area, the start bit area*1, and the data area (excluding the stop bit).

Note 1. Retiming is not performed for the start bit area if the preface length is 0 and the start bit length is 3.

As an example, the receive retiming when oversampling x16 is selected is shown below.

When detecting an RxD input edge two to four cycles before the expected receive cycle, the receive processing is shortened by one sampling CLK cycle.

When detecting a RxD input edge two to three cycles after the expected receive cycle, the receive processing is extended by one sampling CLK cycle.

(Even if the clock is misaligned with the data by more than two cycles, one cycle is corrected for each bit.)

Figure 33.56 shows the conceptual image of receive retiming range.

When detecting an edge in the tolerance area in the figure, data is received as is without making correction.

When detecting an edge in the SyncJump area in the figure, data is corrected for reception.

When detecting an edge in the SyncError area in the figure, data is received as abnormal data with no correction made.

For a Manchester code error (data matches at the 1/4-phase and 3/4-phase sampling points), the SCI reports a code error.

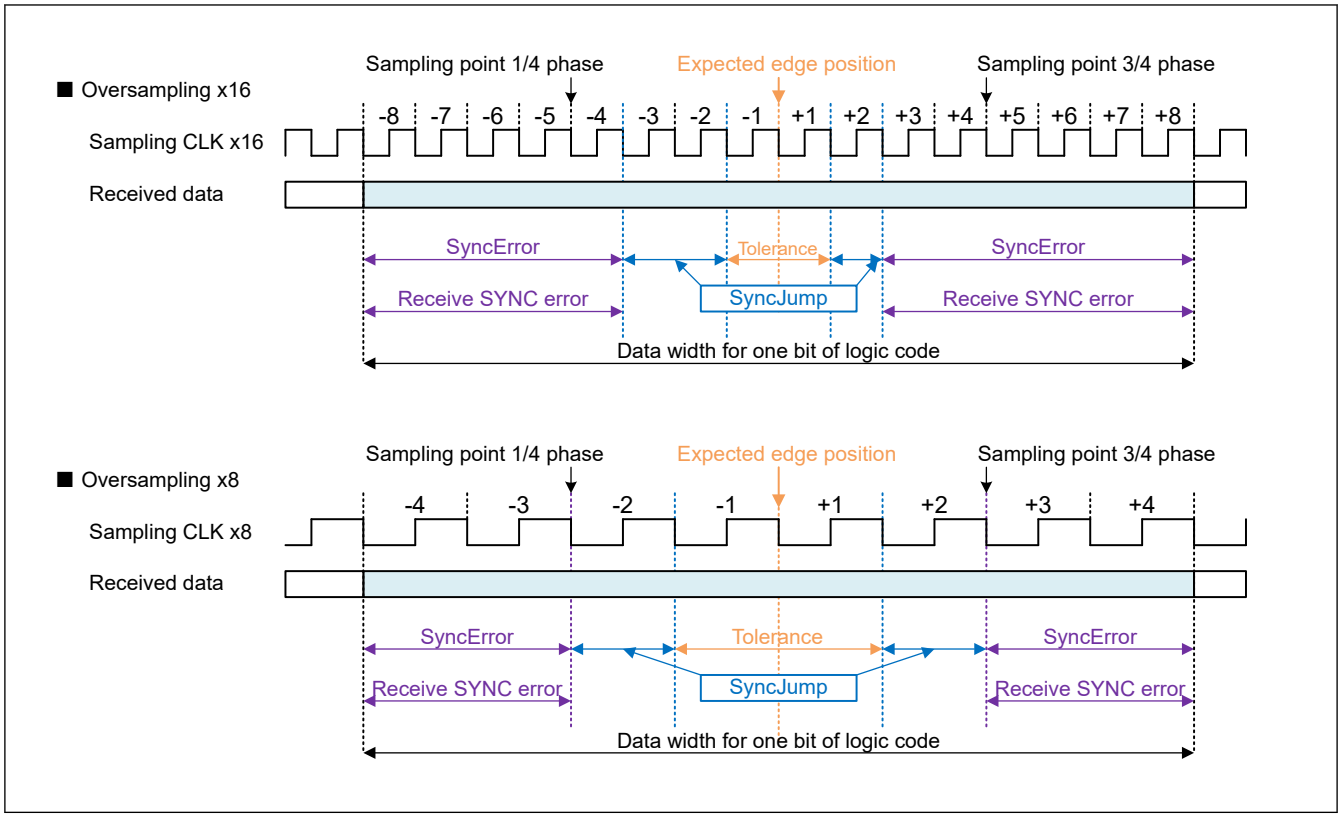


Figure 33.56 Conceptual image of reception retiming range

33.6.10 Polarity Setting for Manchester Code

The polarity of the Manchester code can be set with the Manchester Control Register (MCR).

It can be set separately for transmission and reception. Use the MCR.TMPOL bit to set the polarity for transmission and the MCR.RMPOL bit to set the polarity for reception.

The Manchester code polarity setting is valid for the preface area, the data area, and the parity or multi-processor area.

When the initial settings (TMPOL/RMPOL = 0) are used for the polarity of Manchester code, logic 0 is encoded as a zero-to-one transition in Manchester code and logic 1 is encoded as a one-to-zero transition in Manchester code.

If the settings are changed to TMPOL/RMPOL = 1, logic 0 is encoded as a one-to-zero transition in Manchester code and logic 1 is encoded as a zero-to-one transition in Manchester code. Figure 33.57 shows the conceptual image of the settings and operation.

Separately from the function above, the transmitted and received data in the data area can be inverted by the transmitted/received data inversion function (CCR3.SINV). Since the polarity of Manchester code (MCR.TMPOL/RMPOL) can be set separately from the transmitted/received data invert function (CCR3.SINV), if both are set to inversion (MCR.TMPOL/RMPOL = 1 and CCR3.SINV = 1), the transmitted and received data are set to initial state (inversion + inversion = normal).

The polarity of the start bit area can be set by a register different from the ones mentioned above.

Since a different register is used, the polarity of the start bit area is not affected by the polarity setting for Manchester code mentioned above.

For details on the setting for the start bit area, see section 33.6.1. Frame Format (2) Start bit area.

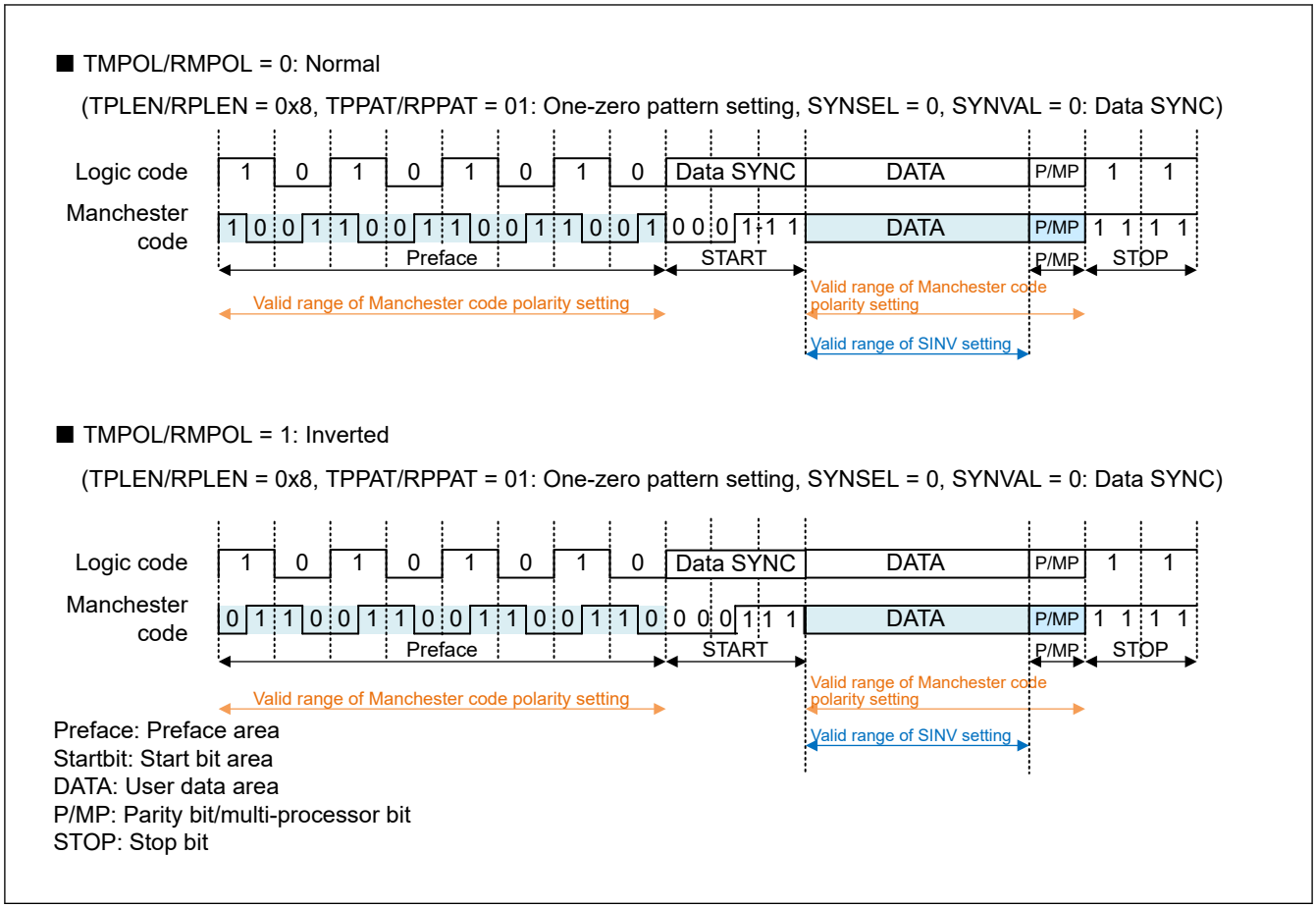


Figure 33.57 Valid range of the Manchester code polarity setting

33.6.11 Errors in Manchester Mode

There are the following errors in Manchester mode:

1. Parity error
2. Over run error
3. Framing error
4. Manchester error
5. Preface error
6. Start Bit error
7. Receive SYNC error

For errors 1 to 3, see section 33.4.9. Serial Data Reception (Asynchronous Mode) (1) Non-FIFO selected because they are the same as in asynchronous mode.

Each error is judged in each area, but they are reflected on flags and operations at the timing of 3/4-bit sampling of the STOP bit area. If a preface error or start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the previous information.

Table 33.29 lists the states of the serial status register when detecting errors and judgment about whether to store data in the RDR.

Table 33.30 lists the errors that can be detected in each area of a Manchester frame. If a Preface error or Start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the result of the previous frame reception. Also, if an error is detected in the previous frame, data will not be received, but errors in the preface area and start bit area will update that flag. Table 33.31 shows the flags and actions in this case.

(1) Manchester error

A Manchester error is generated when a Manchester code error is detected.

In Manchester code, there must be an edge (transition) in the center of the bit.

In the data area of a received frame (including the parity/multi-processor bit), the values of the 1/4-bit and 3/4-bit sampling points are checked in each received 1-bit data, and a Manchester code error is determined if these two values match.

If a Manchester code error is detected, the Manchester error flag (MSR.MER) is asserted.

If a Manchester error occurs, it is handled as an interrupt source and event source. If a Manchester error is detected, the next reception is not performed until the corresponding error flag is cleared.

(2) Preface error

A preface error is generated when the preface pattern does not match, or a Manchester code error is detected in the preface area. If a preface error is detected, the preface error flag (MSR.PFER) is asserted.

It is possible to set whether to use this error flag as an interrupt source with the setting of the MCR register.

When MCR.PFEREN = 1, a preface error is handled as an interrupt source or event source. If a preface error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.PFEREN = 0, a preface error is not handled as an interrupt source or event source, and the next reception is not halted. However, a preface error is notified to MSR.PFER.

(3) Start bit error

A start bit error is generated when a mismatch is detected between the start bit area in the received frame and the preset start bit pattern. Upon detection of a start bit error, a start bit error flag (MSR.SBER) is asserted.

It is possible to set whether to use the start bit error as an interrupt source with the setting of the MCR register.

When MCR.SBEREN = 1, a start bit error is handled as an interrupt source or event source. If a start bit error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.SBEREN = 0, a start bit error is not handled as an interrupt source or event source, and the next reception is not halted. However, a start bit error is notified to MSR.SBER.

(4) Receive SYNC error (SyncError)

When the receive retiming function described in section 33.6.9. Receive Retiming is enabled, the receive retiming operation is performed.

If no edges are detected within the receive retiming range (SyncError area in Figure 33.56) when receive timing operation is being performed, a receive SYNC error is generated. Upon detection of a receive SYNC error, a receive SYNC error flag (MSR.SYER) is asserted. In areas not subject to retiming, receive SYNC errors are not detected.

The preface area*1, the start bit area*1 *2, and the data area (excluding the stop bit) for which receive retiming operation is performed are checked.

It is possible to set whether to use the receive SYNC error as an interrupt source with the setting of the MCR register.

When MCR.SYEREN = 1, a receive SYNC error is handled as an interrupt source or event source. If a receive SYNC error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.SYEREN = 0, a receive SYNC error is not handled as an interrupt source or event source, and the next reception is not halted. However, a receive SYNC error is notified to MSR.SYER.

Note 1. In the case of a frame that starts with a pattern that expects the first half of the bit to be High, it is excluded from retiming.

Note 2. In the start bit area, when there is no preface length and 3 bit start bit is set, it is not subject to retiming. Also, the 1st bit and the 2nd bit in the start bit area when 3 bit start bit is set are not subject to retiming.

Table 33.29 Flags in the status register and receive data handling in Manchester (1 of 2)

Flags in the CSR register			Flags in the MSR register				Received data	Received error status (ERI interrupt/event generation)
ORER	FER	PER	MER	SBER*1	PFER*1	SYER		
0	0	0	0	0	0	0	transfer to RDR	No error

Table 33.29 Flags in the status register and receive data handling in Manchester (2 of 2)

Flags in the CSR register			Flags in the MSR register				Received data	Received error status (ERI interrupt/event generation)
ORER	FER	PER	MER	SBER*1	PFER*1	SYER		
0	1	0	0	0	0	0	transfer to RDR	Framing error
0	0	1	0	0	0	0	transfer to RDR	Parity error
0	1	1	0	0	0	0	transfer to RDR	Framing error + Parity error
0	0	0	1	0	0	0	transfer to RDR	Manchester error
0	1	0	1	0	0	0	transfer to RDR	Framing error + Manchester error
0	0	1	1	0	0	0	transfer to RDR	Parity error + Manchester error
0	1	1	1	0	0	0	transfer to RDR	Framing error + Parity error + Manchester error
1	0	0	0	0	0	0	Lost	Overrun error
1	1	0	0	0	0	0	Lost	Overrun error + Framing error
1	0	1	0	0	0	0	Lost	Overrun error + Parity error
1	1	1	0	0	0	0	Lost	Overrun error + Framing error + Parity error
1	0	0	1	0	0	0	Lost	Overrun error + Manchester error
1	1	0	1	0	0	0	Lost	Overrun error + Framing error + Manchester error
1	0	1	1	0	0	0	Lost	Overrun error + Parity error + Manchester error
1	1	1	1	0	0	0	Lost	Overrun error + Framing error + Parity error + Manchester error
0	Combination of above			0	0	1	transfer to RDR	Errors above + Receive SYNC error*2
1				0	0	1	Lost	Errors above + Receive SYNC error*2
hold	hold	hold	hold	0	1	0	Lost	Preface error*3
hold	hold	hold	hold	1	0	0	Lost	Start bit error*3
hold	hold	hold	hold	0	1	1	Lost	Preface error*3 + Receive SYNC error*2
hold	hold	hold	hold	1	0	1	Lost	Start bit error*3 + Receive SYNC error*2

Note 1. Start bit error and Preface error never become 1 at the same time.
Note 2. When MCR.SYEREN = 1, ERI interrupt/event is generated by SYER factor.
Note 3. If MCR.PFEREN = 1 or MCR.SBEREN = 1, an ERI interrupt/event is generated when the corresponding flag is set.

Table 33.30 Errors detectable in each area

	Preface error (PFER)	Start Bit error (SBER)	Manchester error (MER)	Receive SYNC error (SYER)	Parity error (PER)	Framing error (FER)
Preface area	✓	—	—*1	✓*2	—	—
Start Bit area	—	✓	—	✓*2	—	—
Data area	—	—	✓	✓	—	—
Parity area	—	—	✓	✓	✓	—
Multi-processor area	—	—	✓	✓	—	—
Stop Bit area	—	—	—	—	—	✓

Note: ✓: Detected, —: Not detected
Note 1. When an Manchester code error occurs in the preface area, it is defined as a preface error.
Note 2. It may not be subject to Receive SYNC error detection. For details, see (4) Receive SYNC error (SyncError).

Table 33.31 Operation status due to the presence/absence of error in the previous frame and operation status list when CCR0.MPIE = 0 in multi-processor mode (1 of 2)

Previous frame	Each area of the frame					PFEREN	SBEREN	SYEREN	Received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
No Error	PFER No SYER*1	No Error	Don't Care	Don't Care	Don't Care	0	Don't Care	Don't Care	Lost	set PFER*1	not output	not output
						1					output	output
	No Error	SBER No SYER*1	Don't Care	Don't Care	Don't Care	Don't Care	0	Don't Care	Lost	set SBER*1	not output	not output
							1				output	output
	SYER No PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
								1	Lost		output	output
	No Error	SYER No SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
								1	Lost		output	output
	No Error	No Error	SYER		No Error	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
								1			output	output
	No Error	No Error	MER		No Error	Don't Care	Don't Care	Don't Care	transfer to RDR	set MER	output	output
	No Error	No Error	Don't Care	PER	No Error	Don't Care	Don't Care	Don't Care	transfer to RDR	set PER	output	output
	No Error	No Error	Don't Care	Don't Care	FER	Don't Care	Don't Care	Don't Care	transfer to RDR	set FER	output	output
	There are some errors.				ORER	Don't Care	Don't Care	Don't Care	Lost	set some flags*2	output	output
	No Error	No Error	No Error	No Error	No Error ORER	Don't Care	Don't Care	Don't Care	Lost	set ORER	output	output

Table 33.31 Operation status due to the presence/absence of error in the previous frame and operation status list when CCR0.MPIE = 0 in multi-processor mode (2 of 2)

Previous frame	Each area of the frame					PFEREN	SBEREN	SYEREN	Received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
Some Error*3 *6	PFER No SYER*1	No Error	Don't Care	Don't Care	Don't Care	0	Don't Care	Don't Care	Lost	set PFER*1	Output*4	not output*5
						1						
	No Error	SBER No SYER*1	Don't Care	Don't Care	Don't Care	Don't Care	0	Don't Care		set SBER*1		
							1					
	SYER No PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0		set SYER		
								1				
	No Error	SYER No SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0		set SYER		
								1				
	No Error	No Error	SYER		No Error	Don't Care	Don't Care	0		set SYER		
								1				
	No Error	No Error	MER		No Error	Don't Care	Don't Care	Don't Care		don't set any flags		
	No Error	No Error	Don't Care	PER	No Error	Don't Care	Don't Care	Don't Care				
	No Error	No Error	Don't Care	Don't Care	FER	Don't Care	Don't Care	Don't Care				
	There are some errors. ORER					Don't Care	Don't Care	Don't Care				
	No Error	No Error	No Error	No Error	No Error ORER	Don't Care	Don't Care	Don't Care				

Note 1. If SYER is detected, the SYER flag is also set. Other operations are as shown in this table.
Note 2. Other detected error flags including ORER are also set.
Note 3. If all the error flags are cleared before the STOP bit is judged, the operation will be the same as the case where is no error in the previous frame of this table.
Note 4. Since the ERI interrupt request is level output, it remains active due to errors in the previous frame regardless of the presence or absence of errors in the relevant frame.
Note 5. Since the error cause is continuously detected, the ERI event is not newly output regardless of the presence or absence of errors in the relevant frame.
Note 6. For PFER, SBER and SYER, when each enable bit is set to disable, it is treated as no error.

Table 33.32 Operation when CCR0.MPIE = 1 in multi-processor mode

MPB*1	Each area of the frame					PFEREN	SBEREN	SYEREN	Received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
1	No Error No PFER SYER*3	No Error No SBER SYER*3	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	transfer to RDR	set some flags	output*2	output*2
			Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0				
								1	Lost	don't set any flags	not output	not output
	PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care				
	No Error	SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care					

Note 1. If the received MPB bit is 0, it is not received the frame, and the operation is the same as lost of the reception data of this table.
Note 2. If no error is detected, RXI interrupt request or event is output, and if it is detected, ERI interrupt request or event is output.
Note 3. When SYER is detected in the preface area or the start bit area, the behavior of handling as an error depending on the SYEREN bit changes.

33.7 Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function.

Smart card interface mode can be selected using the appropriate register.

33.7.1 Sample Connection

Figure 33.58 shows a sample connection between a smart card (IC card) and this LSI.

As in the figure, since this LSI communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor. Setting the TE and RE bits in CCR0 to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.

To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.

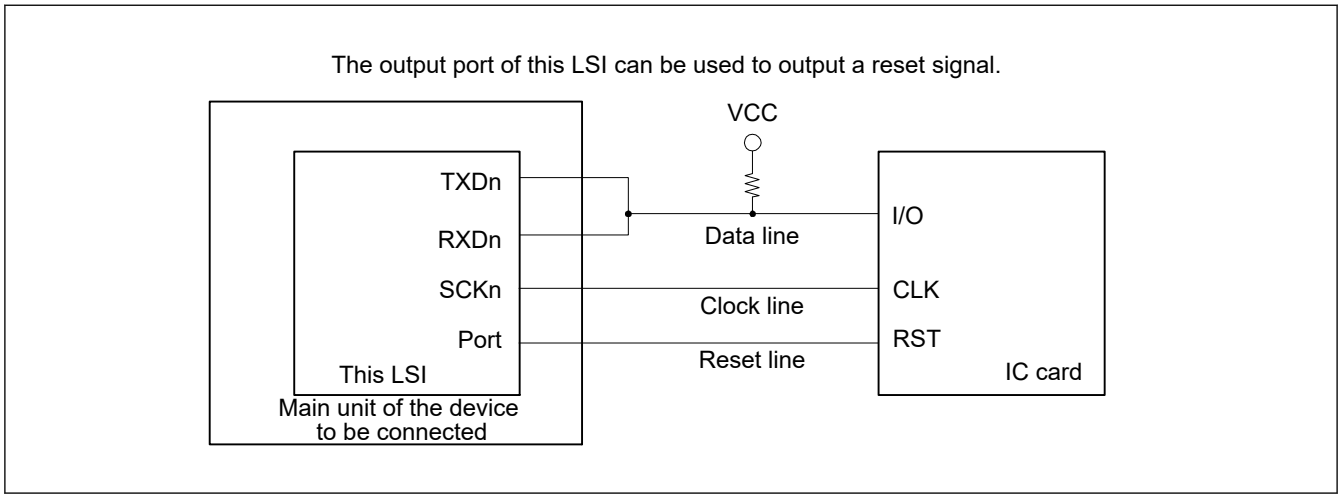


Figure 33.58 Sample connection with a smart card (IC card)

33.7.2 Data Format (Except in Block Transfer Mode)

Figure 33.59 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

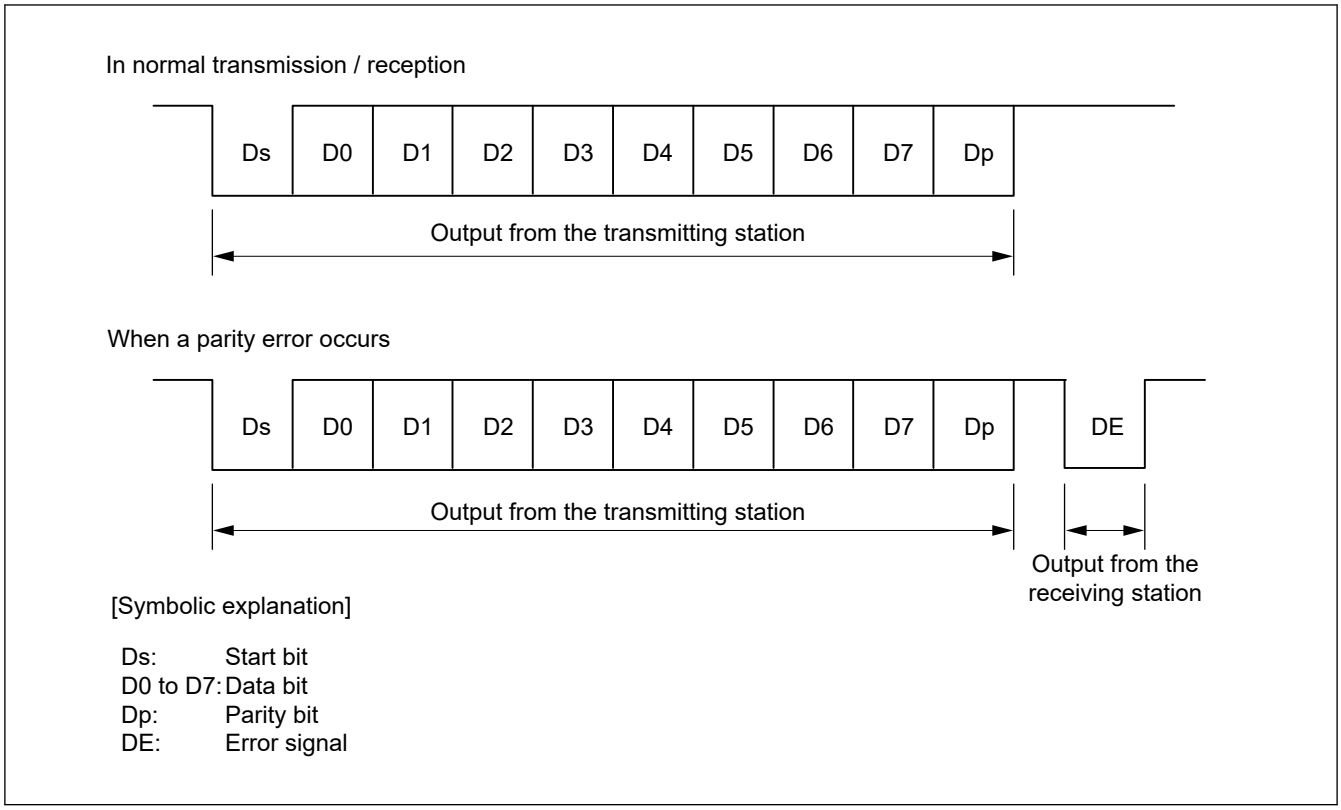


Figure 33.59 Data formats in smart card interface mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedure below.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB first as the start character, as shown in Figure 33.60. Therefore, data in the start character in the figure is 0x3B.

When using the direct convention type, write 0 to the CCR3.SINV bit and write 1 to the CCR3.LSBF bit. Write 0 to CCR1.PM in order to use even parity, which is prescribed by the smart card standard.

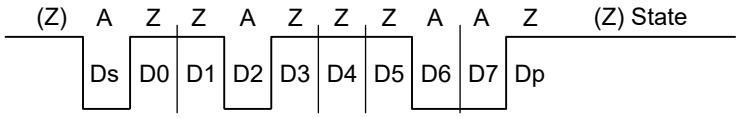


Figure 33.60 Direct convention type (CCR3.LSBF=1, CCR3.SINV=0, CCR1.PM=0)

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB first as the start character, as shown in Figure 33.61. Therefore, data in the start character in the figure is 0x3F.

When using the inverse convention type, write 1 to the CCR3.SINV bit and write 0 to the CCR3.LSBF bit. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of this LSI only inverts data bits D7 to D0, write 1 to CCR1.PM to invert the parity bit for both transmission and reception.

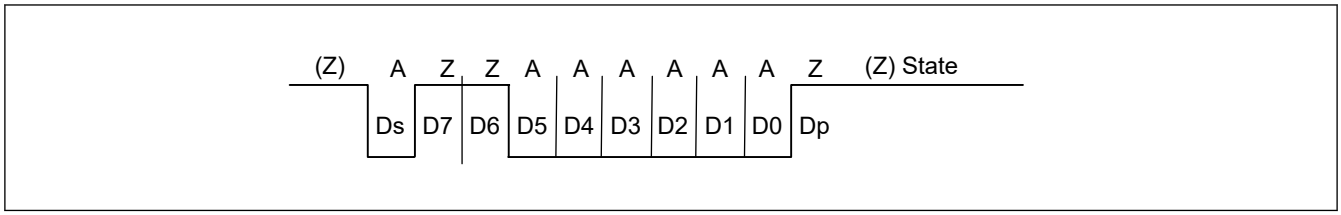


Figure 33.61 Inverse convention type (CCR3.LSBF=0, CCR3.SINV=1, CCR1.PM=1)

33.7.3 Block Transfer Mode

Block transfer mode is different from non-block-transfer-mode of smart card interface mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since CSR.PER is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, CSR.TEND is set 11.5 etu after transmission start.
- In block transfer mode, CSR.ERS flag indicates the error signal status as in non-block-transfer-mode of smart card interface mode, but the flag is always read as 0 because no error signal is transferred.

33.7.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the baud rate generator can be used as a transfer clock in smart card interface mode. In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the settings of CCR2.BCP[2:0]. For data reception, the falling edge of the start bit is sampled with the base clock to perform synchronization. Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 33.62. The reception margin here is determined by the following formula.

$$M = \left|(0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\%$$

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256)

D: Duty cycle of clock (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\% = 49.866\%$$

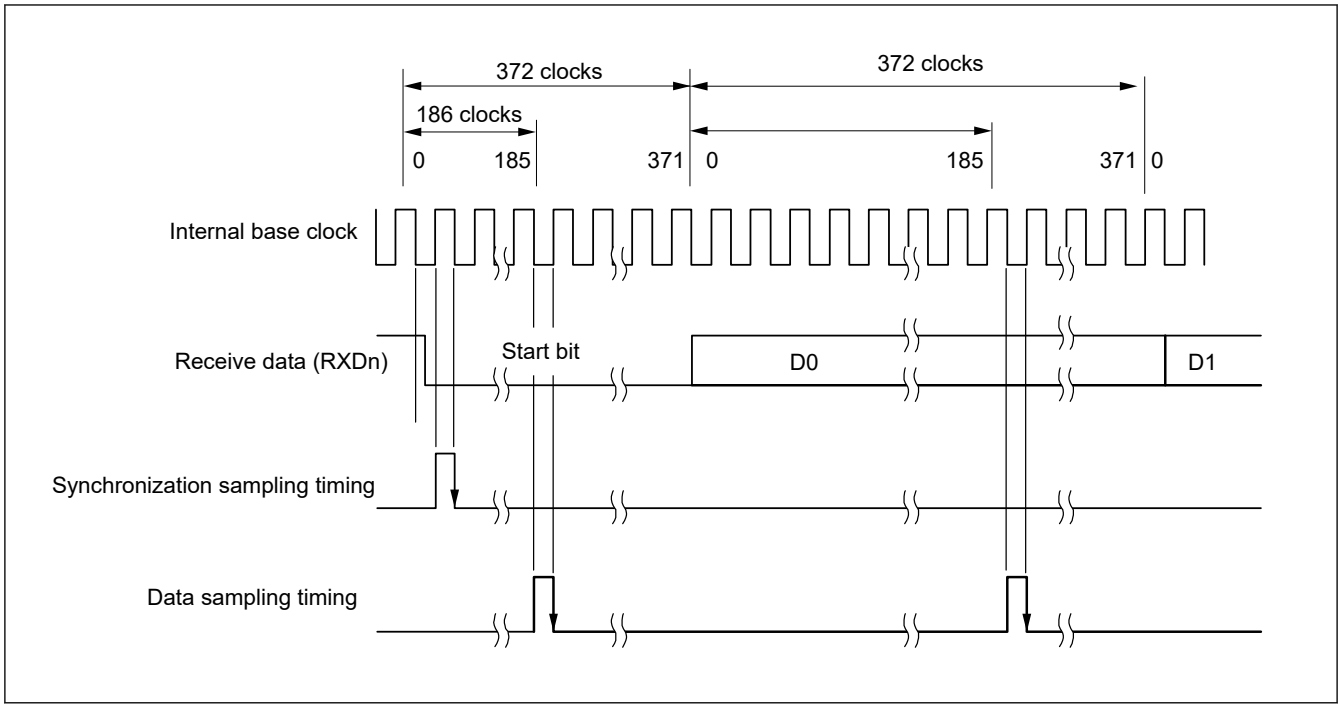


Figure 33.62 Receive data sampling timing in smart card interface mode (when clock frequency is 372 times the bit rate)

33.7.5 Initialization of SCI (Smart Card Interface Mode)

Before transmitting and receiving data, write 0 to CCR0.TE and CCR0.RE (Or write the initial value to CCR0). And initialize SCI following example flowchart in Figure 33.63.

Be sure to set the initial value in the TIE, RIE, TE, RE, and TEIE bits in CCR0 register before switching from transmission mode to reception mode and vice versa. Even if the RE bit is set to 0, the RDR register is not initialized. In transmission mode, set 1 to the TE bit and TIE bit simultaneously, then the TXI interrupt request is generated. To change reception mode to transmission mode, first check that reception has completed, and then initialize SCI. At the end of initialization, set TE = 1 and RE = 0. Reception completion can be verified by RXI request, CSR.ORER, or CSR.PER. To change transmission mode to reception mode, first check that transmission has completed, and then initialize SCI. At the end of initialization, set TE = 0 and RE = 1. Transmission completion can be verified by reading CSR.TEND.

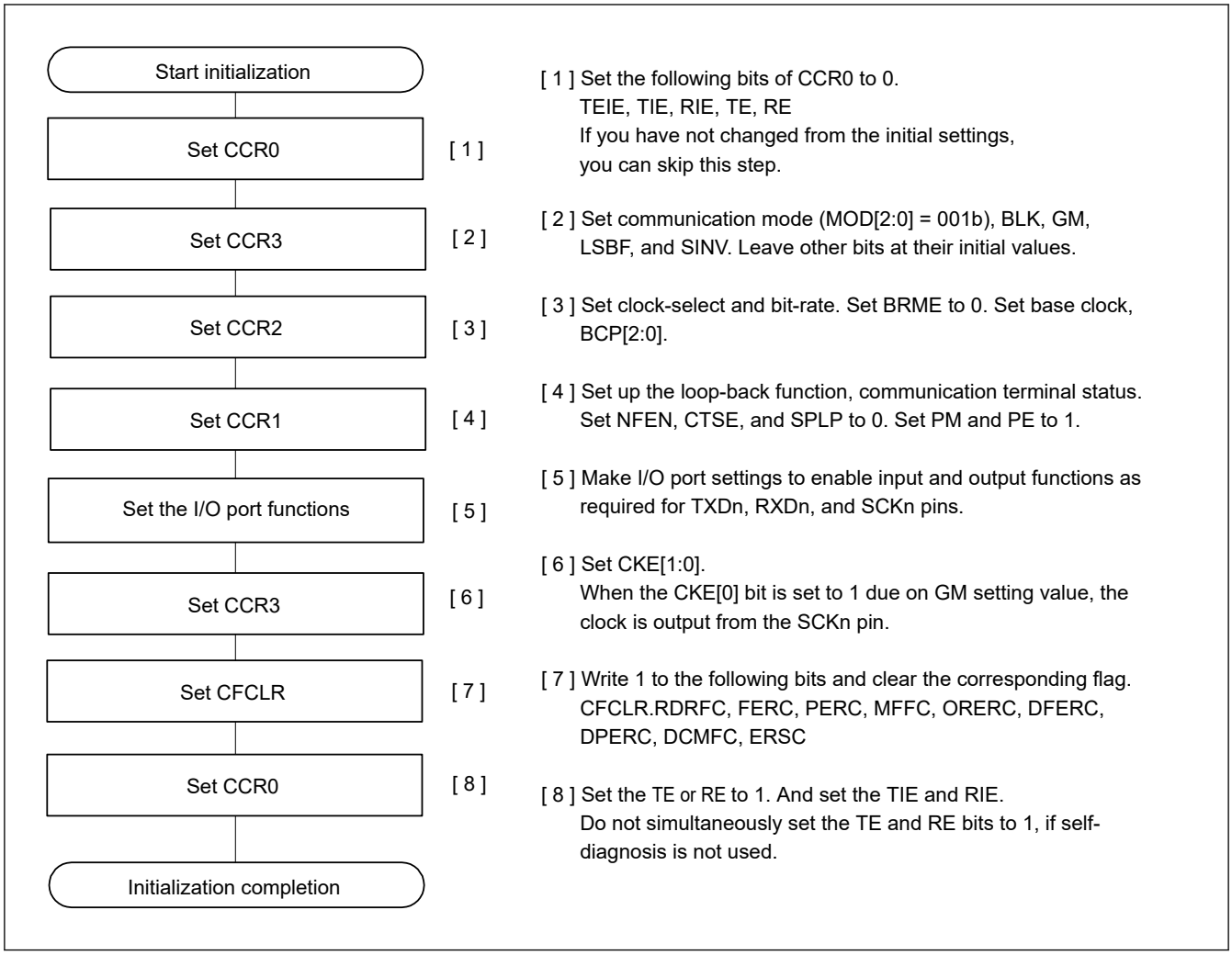


Figure 33.63 Example of SCI initialization flowchart (smart card interface mode)

Figure 33.64 is a timing chart when data transmission is performed by making transition to the Smart Card Interface mode according to the above flow chart. The figure shows the case when CCR3.GM bit is 0. As shown in the figure, when the pin function is set to the SCK pin, the SCK pin is high impedance because the CCR3.CKE[0] bit is 0. When the TXD pin is set, the TXD pin is high impedance because the CCR0.TE bit is 0. Start clock output to the SCK pin with the clock output setting CCR3.CKE[0] to 1, start data transmission by writing transmit data after setting CCR0.TE to 1.

In the smart card interface mode, even if not communicating at CCR0.TE = 0 and CCR0.RE = 0, the clock is continuously output if the clock output setting is used.

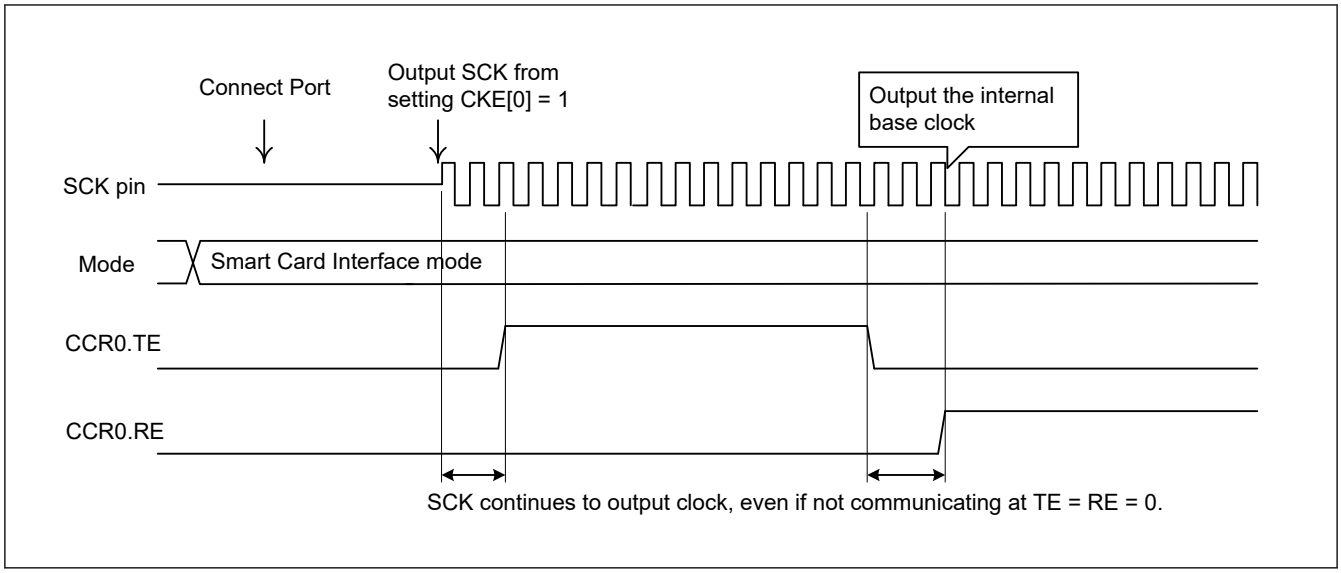
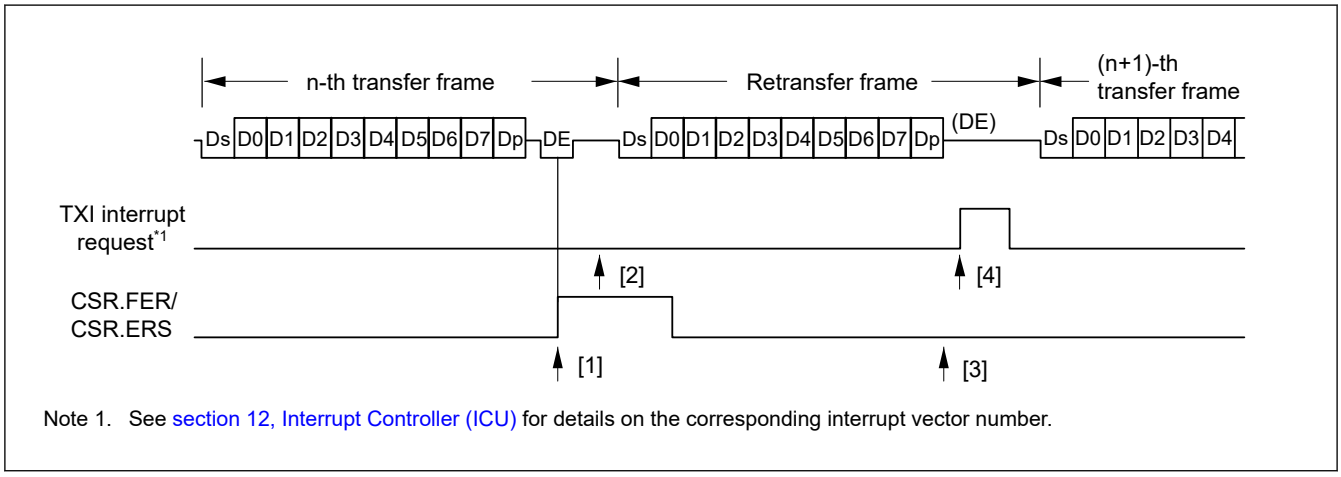


Figure 33.64 Example of timing chart of data transmission (smart card interface mode)

33.7.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 33.65 shows the data retransfer operation during transmission.

1. When an error signal from the receiver end is sampled after one-frame data has been transmitted, CSR.ERS is set to 1. If CCR0.RIE is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
2. For a frame in which an error signal is received, CSR.TEND is not set. Data is retransferred from TDR to TSR allowing data retransmission automatically.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI judges that transmission of one-frame data (including retransfer) has been completed, and the TEND flag is set. If CCR0.TIE is 1 at this time, a TXI interrupt request is generated. Writing transmit data to TDR starts transmission of the next data.



Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.65 Data retransfer operation in SCI transmission mode

Figure 33.67 shows a sample flowchart of serial transmission. All the processing steps are automatically performed by using a TXI interrupt request to activate the DMAC. When CSR.TEND is set to 1 in transmission, if the CCR0.TIE is 1, a TXI interrupt request is generated. The DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DMAC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DMAC transfers the data. If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DMAC is not activated. Therefore, the SCI and DMAC automatically

transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag to 0.

When transmitting/receiving data using the DMAC, be sure to make settings to enable the DMAC before making SCI settings. For DMAC settings, see section 15, DMA Controller (DMAC).

Note that CSR.TEND flag is set in different timings depending on the CCR3.GM bit setting. Figure 33.66 shows the TEND flag generation timing.

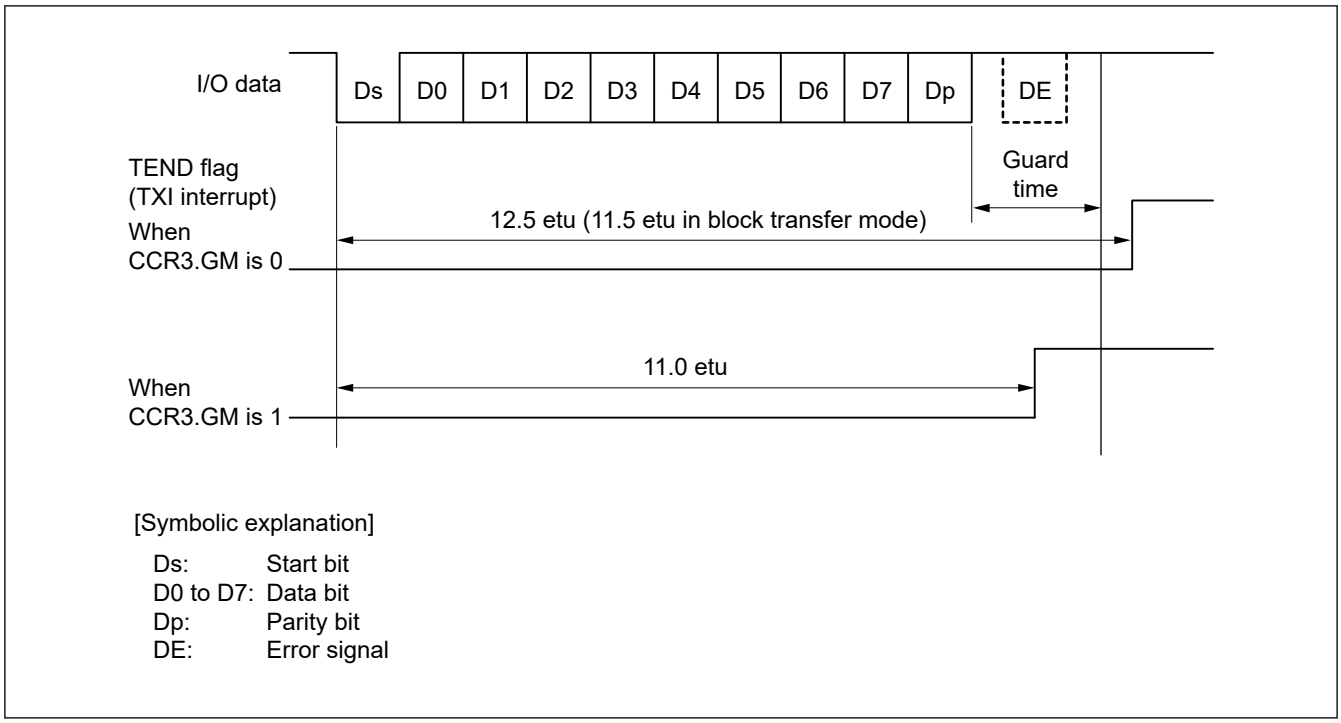


Figure 33.66 CSR.TEND flag generation timing during transmission

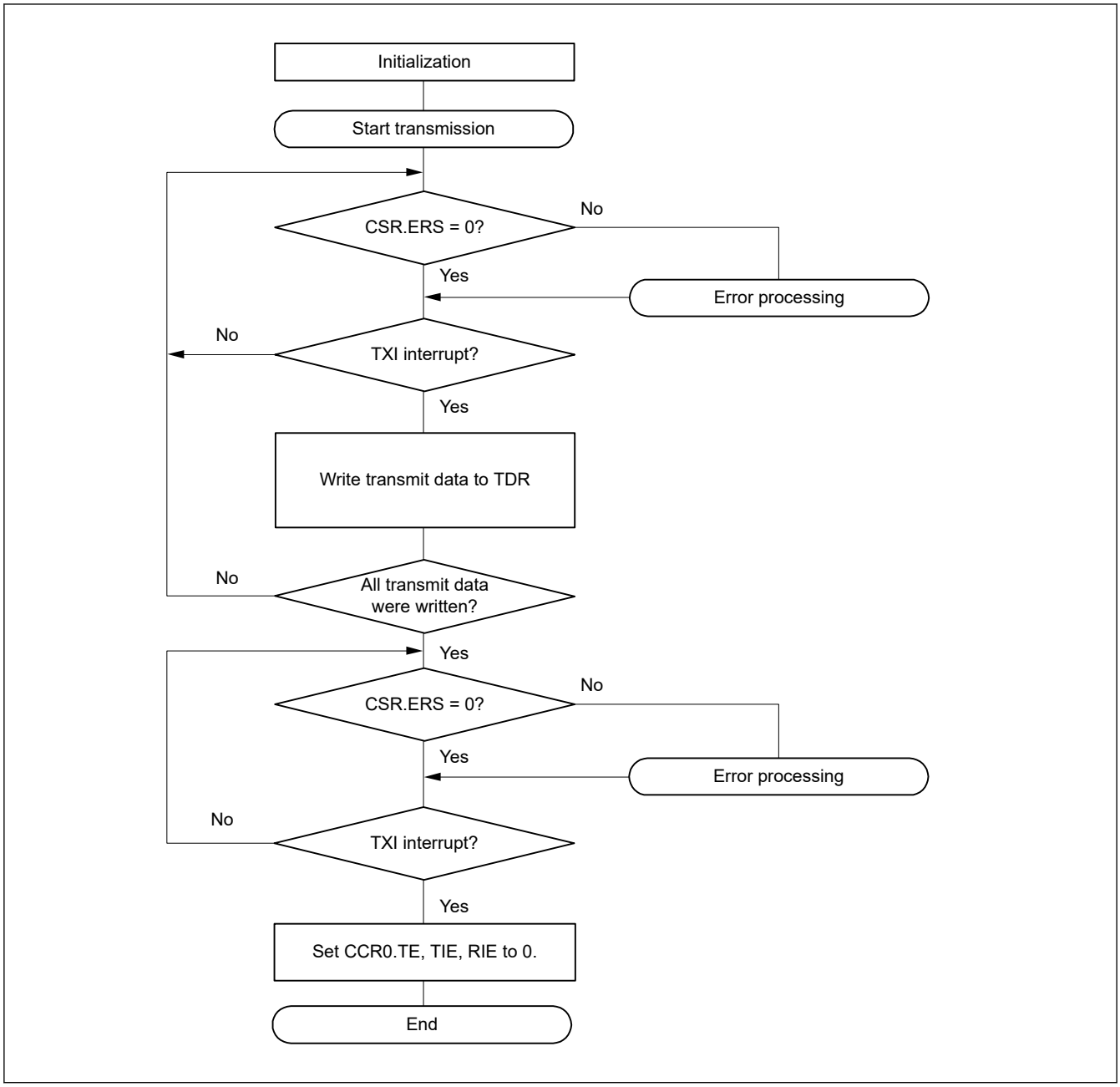
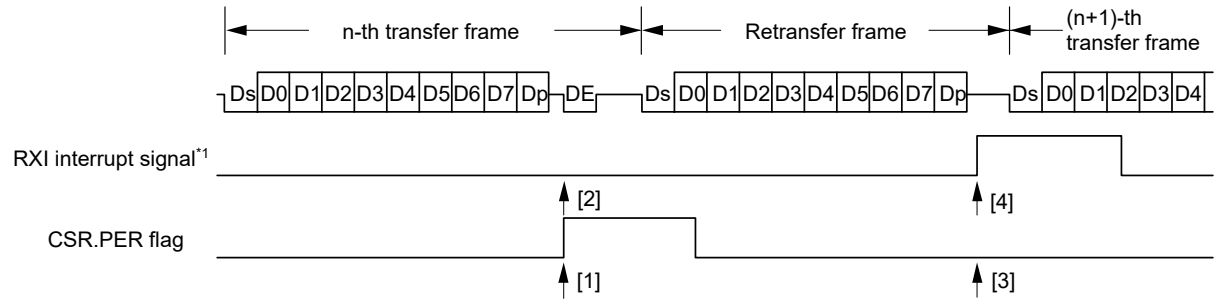


Figure 33.67 Sample smart card interface transmission flowchart

33.7.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is like that in non-smart card interface mode. Figure 33.68 shows the data retransfer operation in reception mode.

1. If a parity error is detected in receive data, the CSR.PER flag is set to 1. When the CCR0.RIE is 1 at this time, an ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no RXI interrupt is generated.
3. When no parity error is detected, the CSR.PER flag is not set to 1.
4. In this case, data is determined to have been received successfully. When the CCR0.RIE is 1, an RXI interrupt request is generated.



Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.68 Data retransfer operation in SCI reception mode (data retransfer operation during reception)

Figure 33.69 shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DMAC. In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DMAC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DMAC activation beforehand, allowing transfer of receive data. If an error occurs during reception and either the CSR.ORER or PER flag is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DMAC is transferred.

Even if a parity error occurs and the PER flag is set to 1 during reception, receive data is transferred to RDR, thus allowing the data to be read. When a reception is forcibly terminated by setting the CCR0.RE to 0 during operation, read the RDR register because the received data which has not yet been read may be left in RDR.

Note: For operations in block transfer mode, see section 33.4. Asynchronous Mode.

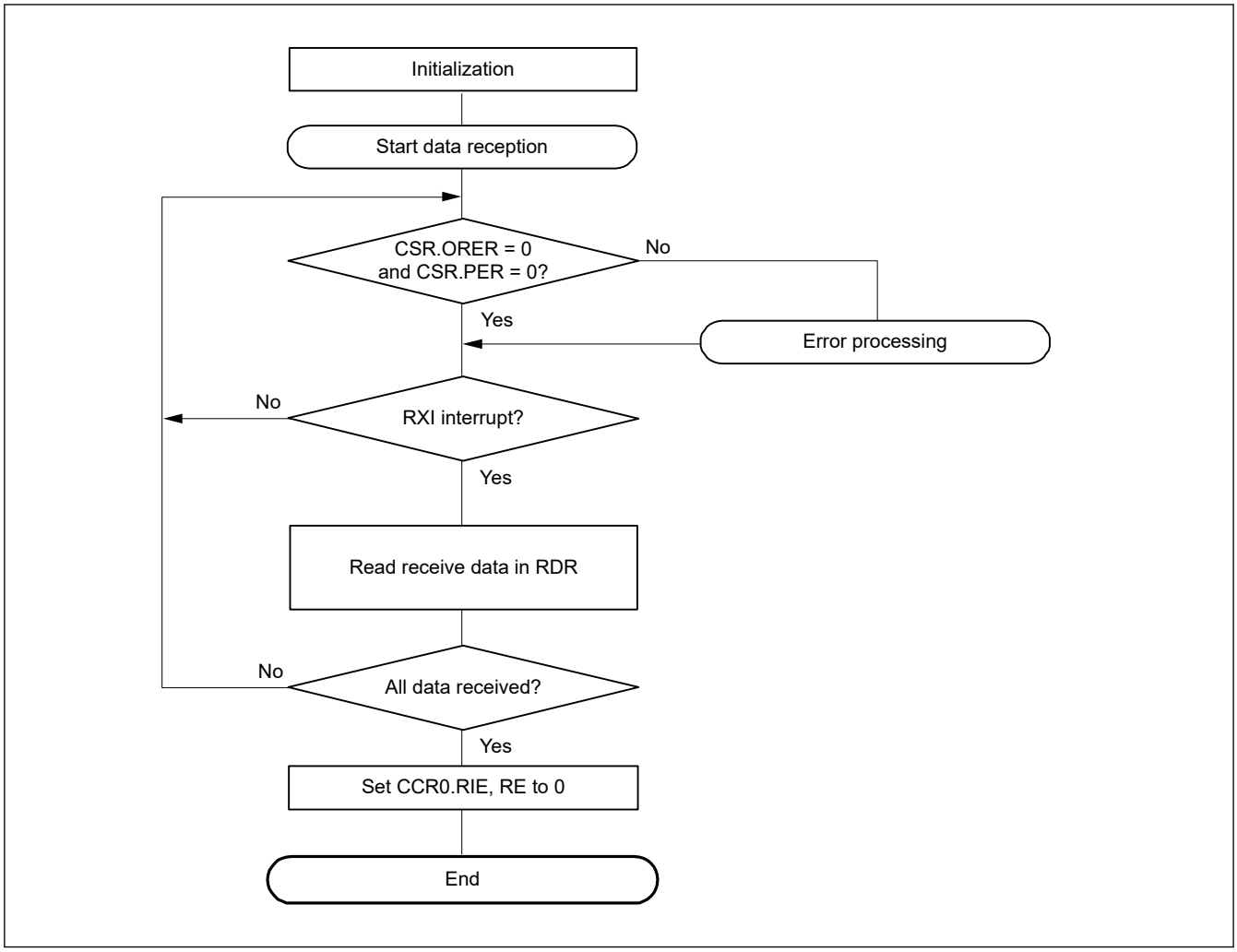


Figure 33.69 Sample smart card interface reception flowchart

33.7.8 Clock Output Control

When the CCR3.GM is set to 1, the clock output can be controlled by the CCR3.CKE[1:0]. Refer to the description of the CKE in section 33.3.8. CCR3 : Common Control Register 3. When setting the clock output, the base clock described in section 33.4.3. Clock is output, so the width of the clock pulse can be kept to the width specified by setting the bit rate. It is described in section 33.3.7. CCR2 : Common Control Register 2, the bit rate is set by CCR2.CKS[1:0], CCR2.BCP[2:0], and BRR[7:0].

Figure 33.70 shows the timing chart for explaining clock output control. This is an example when the CCR3.CKE[1] is set to 0 and the CCR3.CKE[0] is controlled.

When the CCR3.GM is 0, output control by the CCR3.CKE[0] is immediately reflected on the SCK pin, so there is a possibility that pulses with an unintended width may be output from the SCK pin.

When the CCR3.GM is 1, the output pulse control by the CCR3.CKE[0] controls the pulse width set to be based on the state of the base clock.

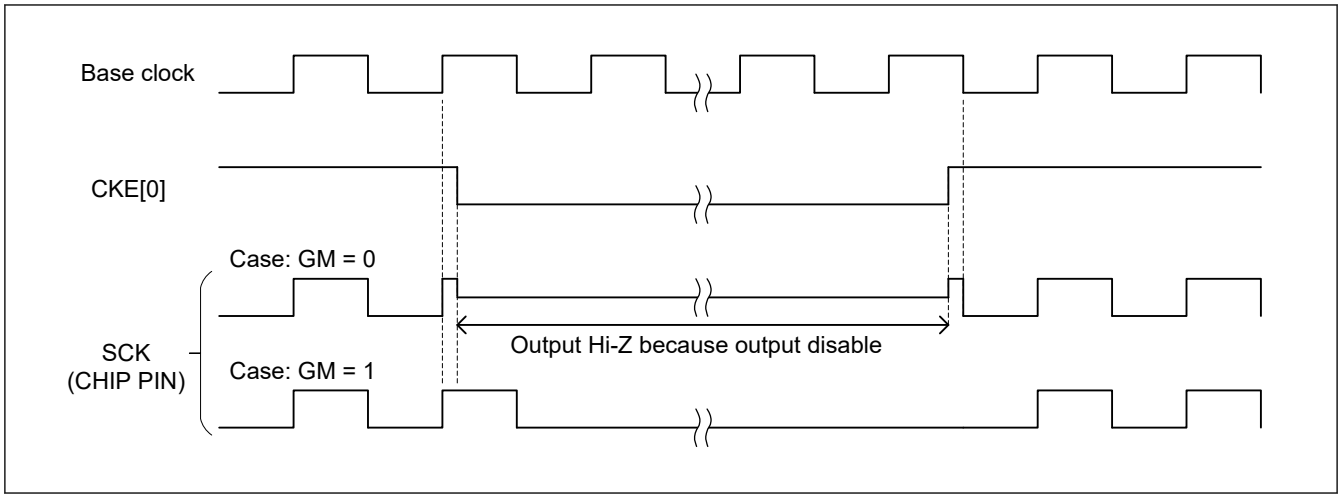


Figure 33.70 Clock control timing chart by CCR3.GM

33.8 Simple I2C mode

Simple I2C mode supports master operation only. Simple I2C bus format is composed of 8 data bits and an acknowledge bit. A slave-address frame follows start condition or restart condition. And the slave-address frame is used to specify a slave device as the partner by master device. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8-bit data of each frame is transmitted from MSB.

Figure 33.71 shows I2C bus format, and Figure 33.72 shows the timing of I2C.

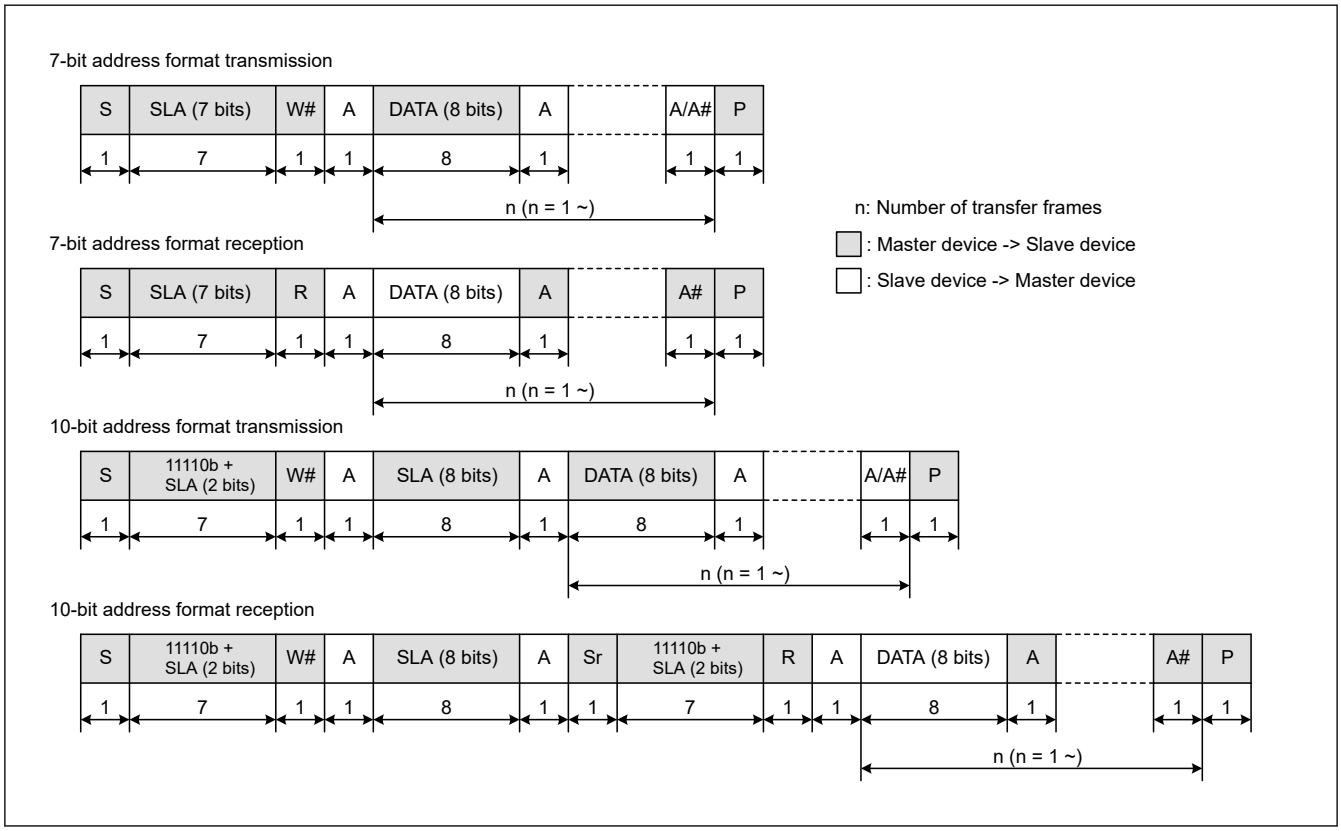


Figure 33.71 I2C bus format

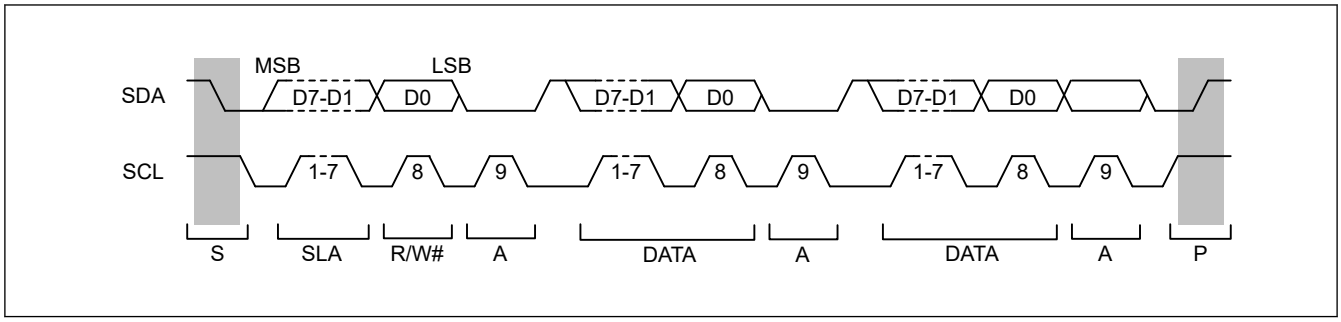


Figure 33.72 I2C bus timing (SLA = 7 bit)

[Symbol explanation]

S: Indicates a start condition. The master device changing the level on the SDAn line from the high to the low level when the SCLn line is at the high level.

SLA: Indicates a slave address. The master device selects a slave device.

R/W#: Indicates the direction of transfer (reception or transmission). When R/W# is high, the transfer direction is from the slave device to the master device. When R/W# is low, the transfer direction is from the master device to the slave device.

A/A#: Indicates an acknowledge. This is returned by the slave device for master transmission and by the master device for master reception. The low level indicates ACK and the high level indicates NACK.

Sr: Indicates a restart condition. The master device changing the level on the SDAn line from the high to the low level when the SCLn line is at the high level.

DATA: Indicates the received or transmitted data.

P: Indicates a stop condition. The master device changing the level on the SDAn line from the low to the high level when the SCLn line is at the high level.

33.8.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to ICR.IICSTAREQ causes the generation of a start condition. The following operations are done at the generation of a start condition.

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept in the released state.
- The hold time for the start condition is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The level on the SCLn line falls (from the high level to the low level), the ICR.IICSTAREQ is set (to 0), and a start condition generated STI interrupt (SCIn_TEI) is output.

Writing 1 to ICR.IICRSTAREQ causes the generation of a restart condition. The following operations are done at the generation of a restart condition.

- The SDAn line is released and the SCLn line is kept at the low level.
- The period at low level for the SCLn line is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The SCLn line is released (transition from the low to the high level).
- Once the high level on the SCLn line is detected, the setup time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The level on the SDAn line falls (from the high level to the low level).
- The hold time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The level on the SCLn line falls (from the high level to the low level), the ICR.IICRSTAREQ is set (to 0), and a restart condition generated STI interrupt (SCIn_TEI) is output.

Writing 1 to ICR.IICSTPREQ causes the generation of a stop condition. The following operations are done at the generation of a stop condition.

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept at the low level.
- The period at low level for the SCLn line is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The SCLn line is released (transition from the low to the high level).
- Once the high level on the SCLn line is detected, the setup time for the stop condition is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The SDAn is released (transition from the low to the high level), the ICR.IICSTPREQ is set (to 0), and a stop condition generated STI interrupt (SCIn_TEI) is output.

Figure 33.73 shows the timing of operations in the generation of start, restart and stop conditions.

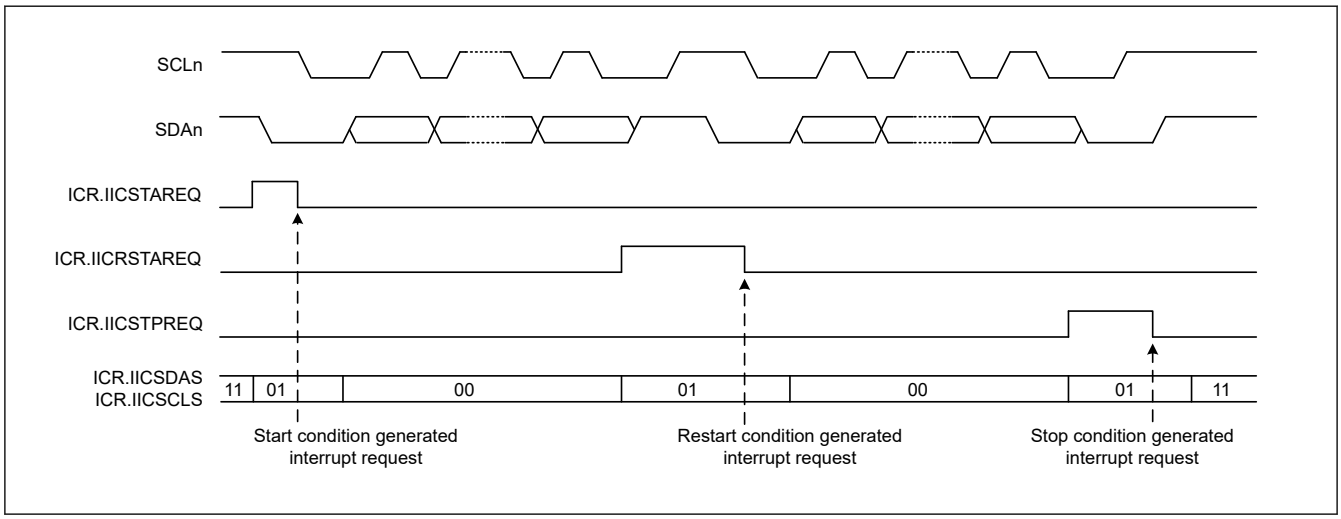


Figure 33.73 The timing of operations in the generation of start, restart and stop conditions

33.8.2 Clock Synchronization

The slave device of the communication partner may make SCLn line Low-level with a view to insert a wait. Setting the ICR.IICCSC to 1, applies control to obtain synchronization when the levels of the internal SCLn clock signal and the level being input on the SCLn pin differ.

When the ICR.IICCSC is set to 1, the level of the internal SCLn clock signal changes from low to high, counting to determine the period at high level is stopped while the low level is being input on the SCLn pin, and counting to determine the period at high level starts after the transition of the input on the SCLn pin to the high level. The interval until counting to determine the period at high level starts on the transition of the SCLn pin to the high level is the total time which contains the SCLn input delay, the noise filtering delay of the SCLn pin (2 or 3 cycles of the filtering clock), and the internal processing delay (1 or 2 cycles of PCLKSCIn). The period at high level of the internal SCLn clock is extended even if other devices are not placing the low level on the SCLn line.

If the ICR.IICCSC is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SCLn pin and the internal SCLn clock. If the ICR.IICCSC is 0, synchronization with the internal SCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a period of waiting into the interval until the transition of the internal SCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a period of waiting after the transition of the internal SCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the period of waiting, generation of the condition itself is not guaranteed.

Figure 33.74 shows an example of operations to synchronize the clocks.

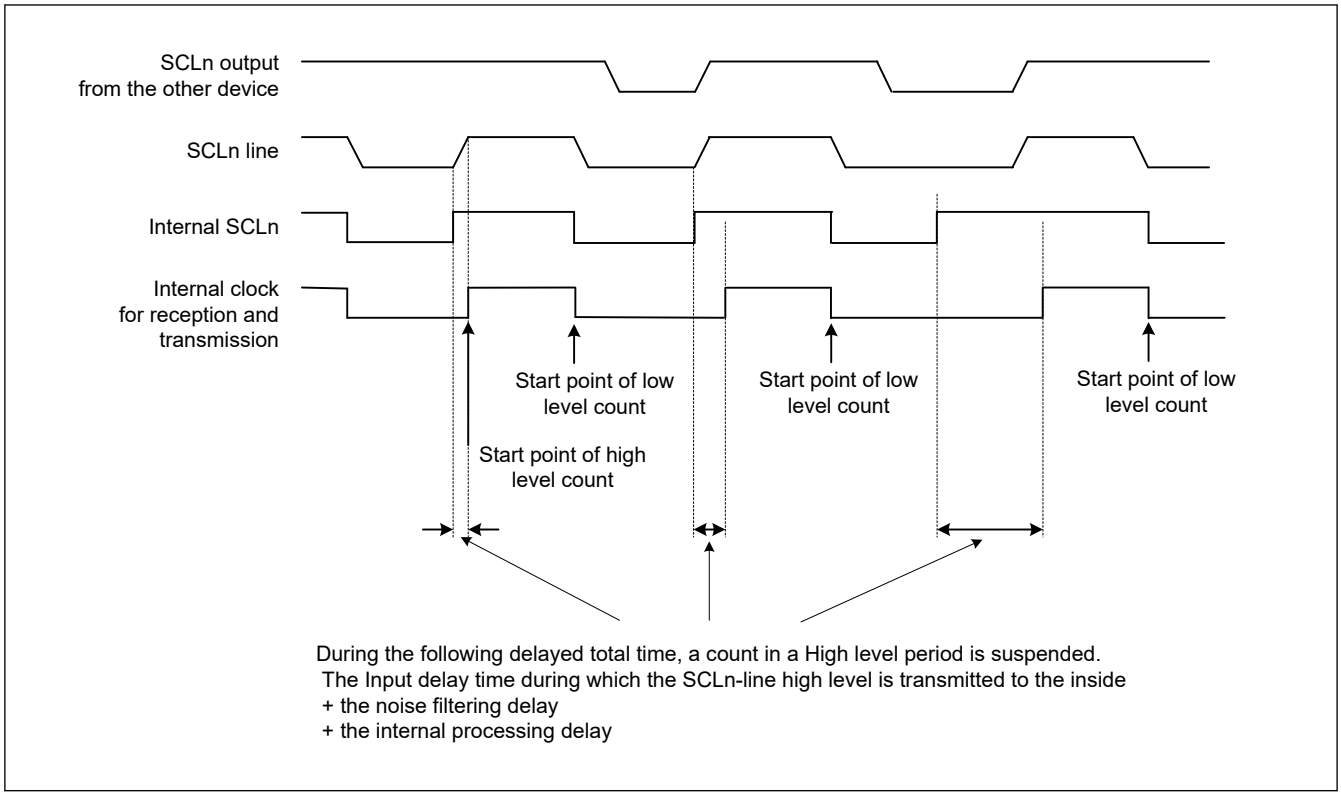


Figure 33.74 Example of operations for clock synchronization

33.8.3 SDA Output Delay

The ICR.IICDL[4:0] can be used to set a delay for output on the SDAn pin relative to falling edges of output on the SCLn pin. The delay-time settings are selectable from 0 to 31 cycles of the clock signal from the baud rate generator (The base is PCLKSCIn and selected the divided clock by the CCR2.CKS[1:0]). About Start/Restart/Stop conditions, 8-bit transmission data and acknowledge, the SDAn pin output can be delayed.

If the SDA output delay is shorter than the falling time of the SCLn output pin, the change of the SDAn output pin will start while the SCLn output pin level is falling, then there is a possibility of erroneous operation for slave devices. Ensure setting the SDA output delay greater than the SCLn maximum falling time (300 ns for I2C normal/fast mode).

Figure 33.75 shows the timing of SDA output delay.

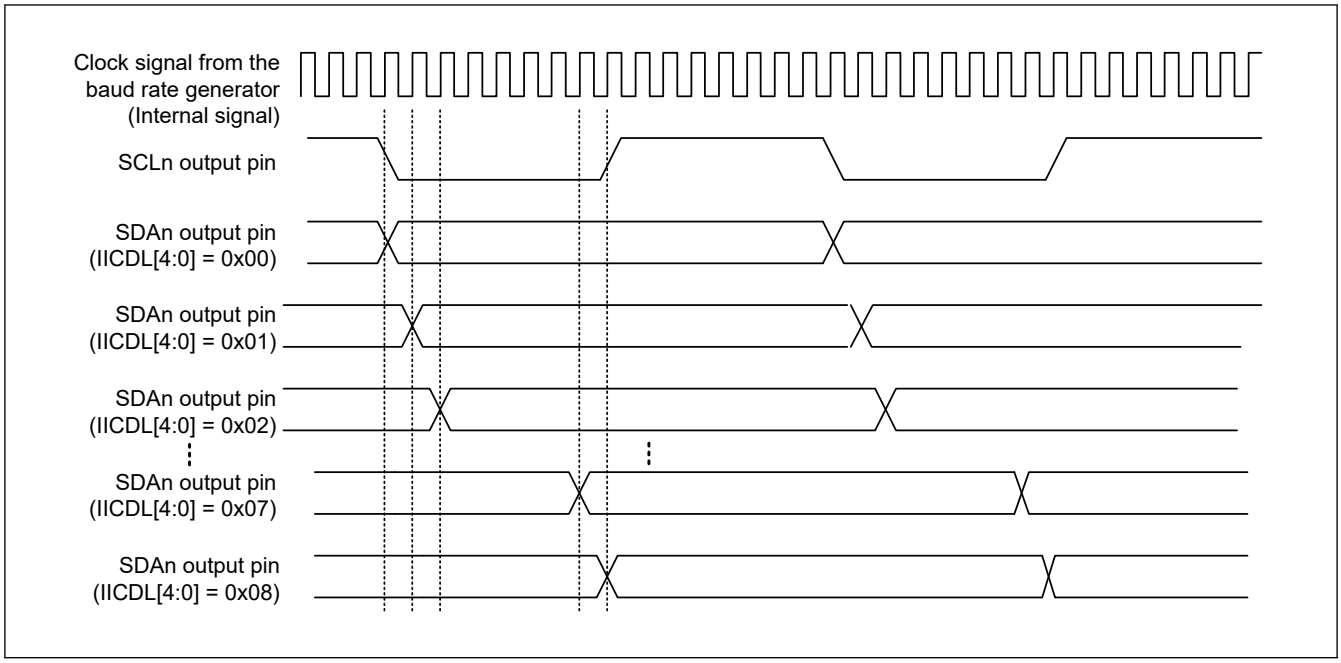
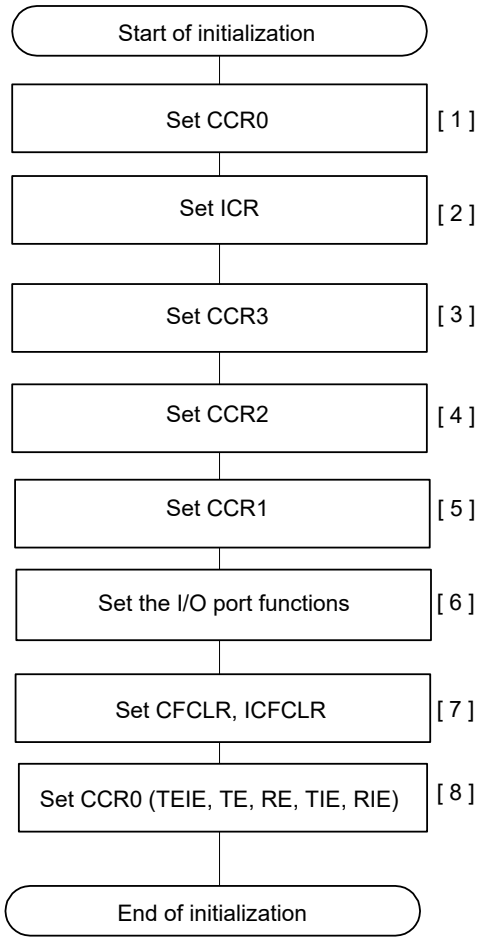


Figure 33.75 Timing of SDA output delay

33.8.4 SCI Initialization (Simple I2C mode)

Write initial value (all 0) to CCR0, then initialize SCI according to Figure 33.76.

When changing the operating mode, transfer format, and so on, be sure to set 0 to CCR0.TE and CCR0.RE before proceeding with the changes. (Or write initial value to CCR0 again.)



Note 1. If you do not use the bit rate modulation function, you do not need to set it.

Note: Set the CCR0.TE and RE bits to 0 or 1 at the same time.

[1] Set the following bits of CCR0 to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the IICSDAS[1:0] and IICSCLS[1:0] to 11b.
Set the IICDL[4:0] and IICINTM as required.
Set the IICACKT and the IICCSC bits to 1.

[3] Set the transmission/reception format as the communication mode (MOD[2:0] = 100b) and CCR3.LSBF = 0.

[4] Set the bit rate modulation function*1, the clock selection, and the bit rate.

[5] Set up the Noise-filter function.

[6] Set I/O port settings that allow use of the SCLn and SDAn pin functions.

[7] Write 1 to the following bits and clear the corresponding flag.
CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
ICFCLR.IICSTIFC

[8] Set the TE and RE to 1.
To enable interrupts, set the TE bit, TIE bit, TEIE bit, RE bit, and RIE bit to 1 with one instruction at the same time.
(for transmission and when the IICINTM bit is 1, clear the RIE bit.)
Setting the TE and RE bits to 1 makes the SCLn and SDAn pin functions available.

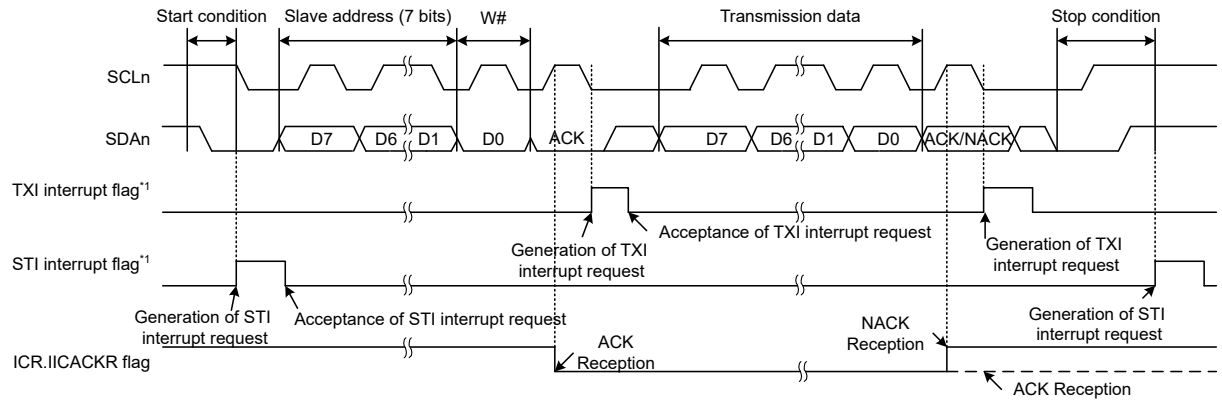
Figure 33.76 Example of the flow of SCI initialization (for simple I2C mode)

33.8.5 Operation in Master Transmission (Simple I2C mode)

Figure 33.77 and Figure 33.78 show examples of operations in master transmission, Figure 33.79, Figure 33.80, and Figure 33.81 show the example flowcharts. See Table 33.39 about the STI interrupt.

Figure 33.77 shows the operation example when ICR.IICINTM is 1 (Reception/Transmission interrupts are in use). In this case, you can start DMAC by TXI interrupt. However, if use DMAC, ACK/NACK can not be confirmed. So, if you want to confirm ACK/NACK, prepare transmit data by CPU. In simple I2C mode, TXI interrupt is generated when communication of one frame is completed. And it is not used RXI interrupt in master transmission, so the CCR0.RIE set to 0.

Figure 33.79 shows a flow chart in the case of ICR.IICINTM is 1 and address transmission by CPU and data transmission by DMAC. Figure 33.80 shows a flow chart of address and data transmission by CPU. When 10-bit slave addresses are in use, steps [3] and [4] are repeated twice.



Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

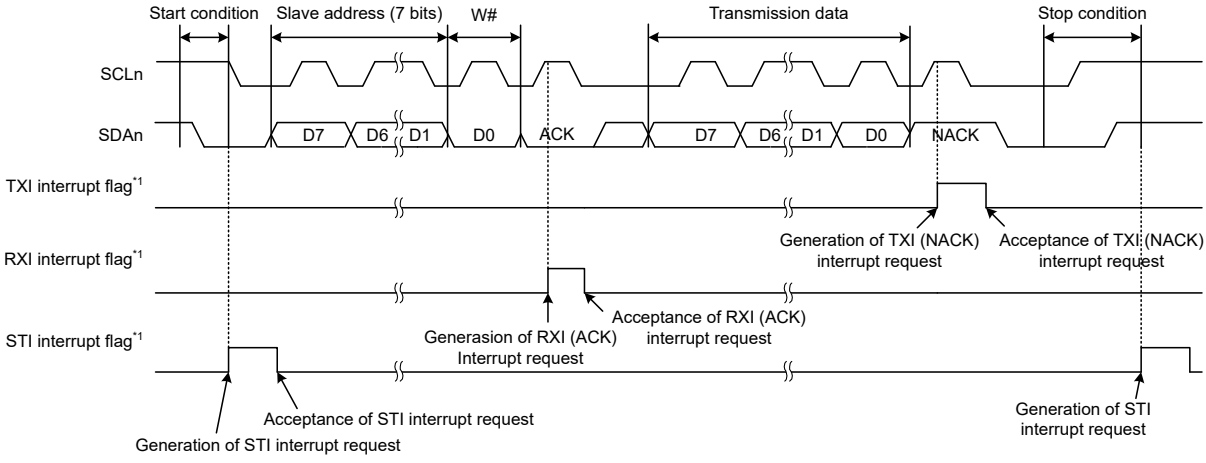
Figure 33.77 Example 1 of operations for master transmission in simple I2C bus mode (7-bit slave address, transmission and reception interrupt in use (ICR.IICINTM = 1))

Figure 33.78 shows an example of operations when ICR.IICINTM is 0 (ACK and NACK interrupt in use). In this case, DMAC is activated by the ACK (RXI) interrupt, and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK (TXI) interrupt as the trigger.

Figure 33.81 shows a flow chart of ICR.IICINTM is 0.

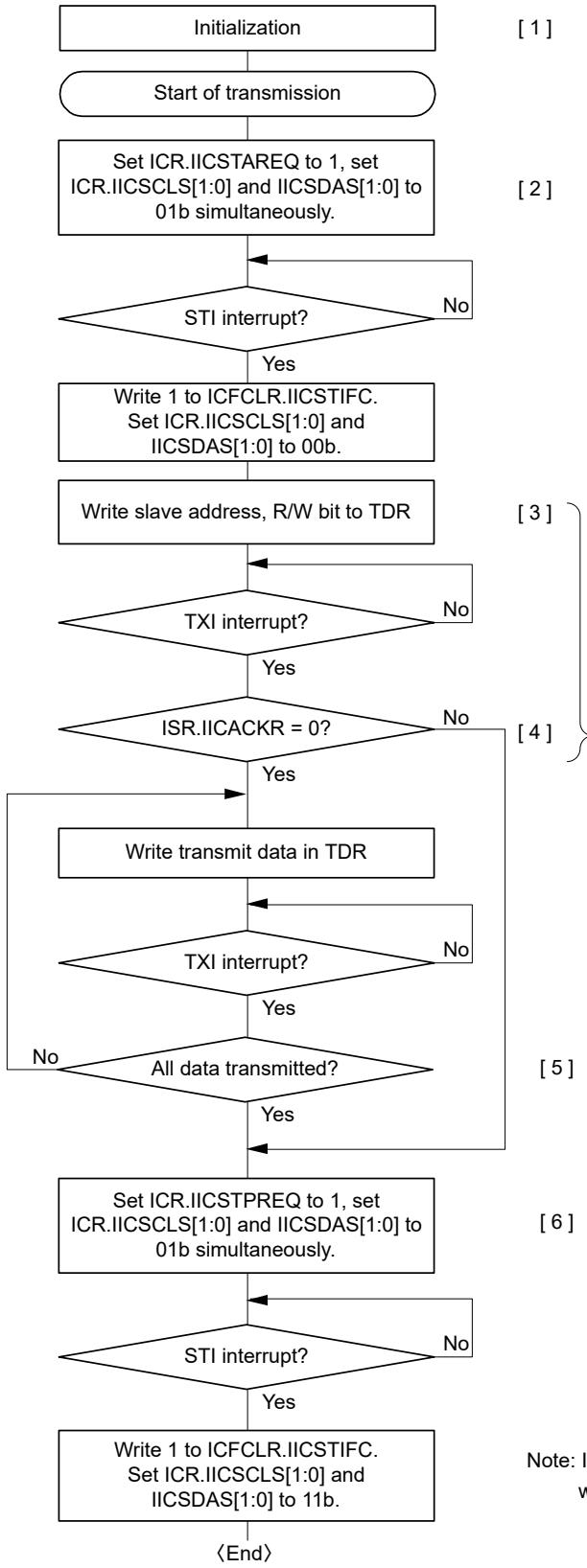
To resume communication after interrupting communication for some reason after writing transmit data to TDR, follow the procedure below.

1. Set the CCR0.TE and CCR0.RE bits to 0 to stop communication.
2. Set ICR.IICSCLS[1:0] and ICR.IICSDAS[1:0] to 11b, release the I2C bus, and clear various condition generation requests.
3. When CSR.RDRF = 1, the RDR register is read by dummy and the RDRF bit is set to 0.
4. Set the CCR0.TE and CCR0.RE bits to 1 and restart communication.



Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.78 Example 2 of operations for master transmission in simple I2C bus mode (7-bit slave address, ACK and NACK interrupt in use (ICR.IICINTM = 0))



[1] Initialization for simple I2C mode
For transmission, set the CCR0.RIE to 0.
(RXI and ERI interrupts requests are disabled)

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirm ACK response from the slave device:
Check the ISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

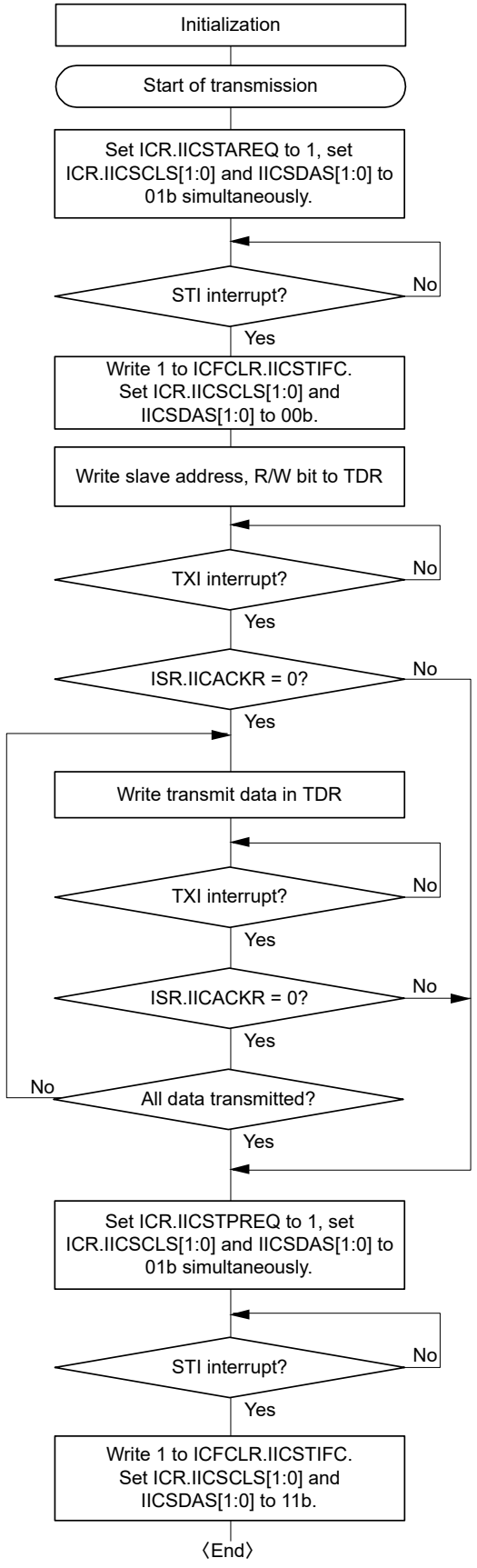
[5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. Except for the first data to be transmitted, a TXI interrupt request can activate the DMAC to handle writing of data to TDR.
But it cannot be confirmed status of ACK or NACK in this case.

[6] Generation of a stop condition.

If 10-bits slave address are in use, processing from [3] to [4] is repeated twice.

Note: In simple I2C mode, the TXI interrupt request is generated when communication is completed.

Figure 33.79 Example of the procedure for master transmission operations in simple I2C mode (when ICR.IICINTM is 1, and when confirming ACK / NACK by address transmission only.)



[1] [1] Initialization for simple I2C mode
For transmission, set the CCR0.RIE to 0.
(RXI and ERI interrupts requests are disabled)

[2] Generate a start condition.

[2] [3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirm ACK response from the slave device:
Check the ISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[3] [5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. If ACK or NACK is checked in every transmission, DMAC cannot be used.

[6] Generation of a stop condition.

[4] If 10-bits slave address are in use, processing from [3] to [4] is repeated twice.

[5]

[6]

Note: In simple I2C mode, the TXI interrupt request is generated when communication is completed.

Figure 33.80 Example of the procedure for master transmission operations in simple I2C mode (when ICR.IICINTM is 1, and when confirming ACK / NACK in all transmissions.)

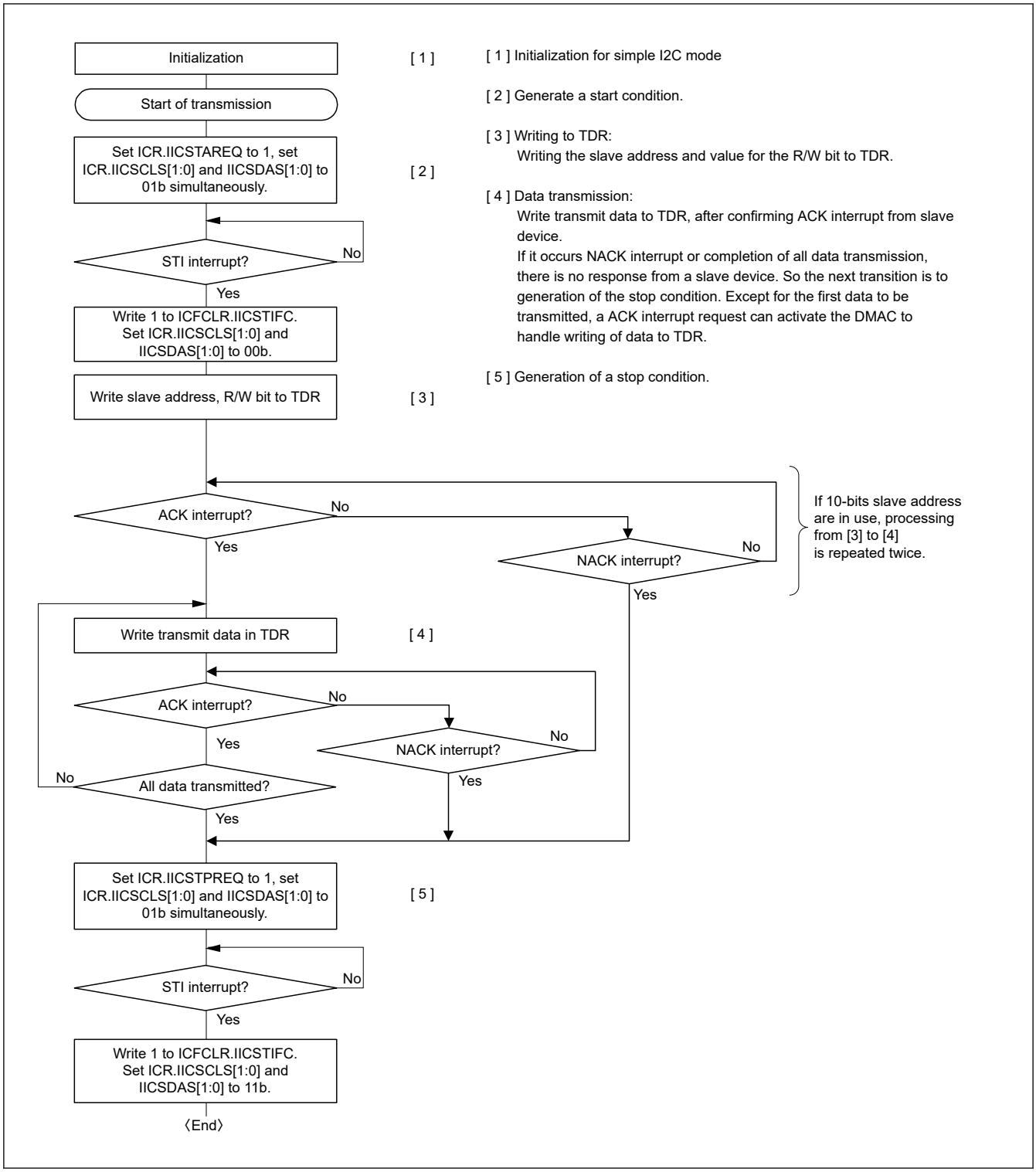
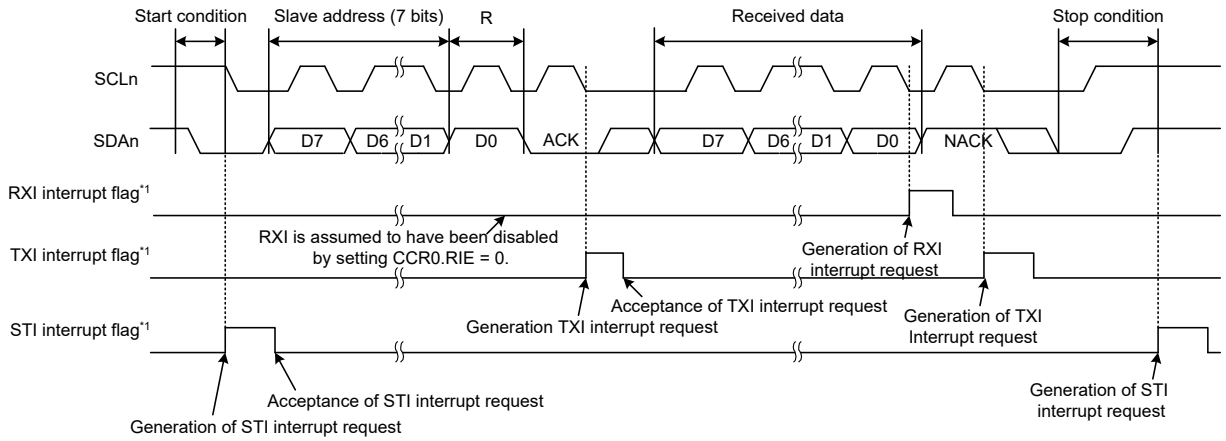


Figure 33.81 Example of the procedure for master transmission operations in simple I2C mode (when ICR.IICINTM is 0.)

33.8.6 Master Reception (Simple I2C mode)

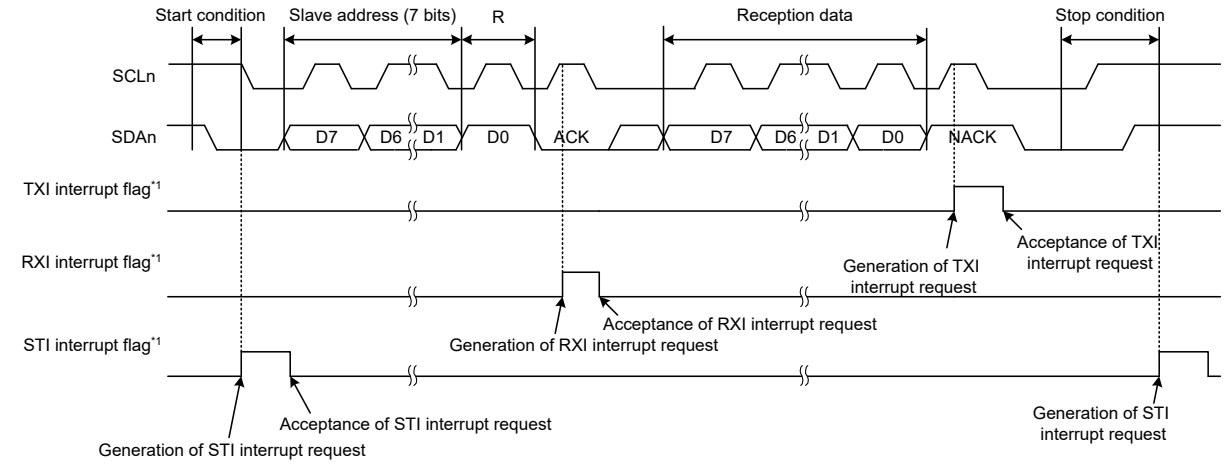
Figure 33.82 and Figure 33.83 show example of operations in simple I2C mode master reception. Figure 33.84 and Figure 33.85 show flowchart of the master reception. The value of the ICR.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and 0 (use ACK and NACK interrupts).

In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed.



Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.82 Example of operations for master reception in simple I2C bus mode (7-bit slave address, transmission and reception interrupt in use (ICR.IICINTM = 1))



Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.83 Example of operations for master reception in simple I2C bus mode (7-bit slave address, ACK and NACK interrupt in use (ICR.IICINTM = 0))

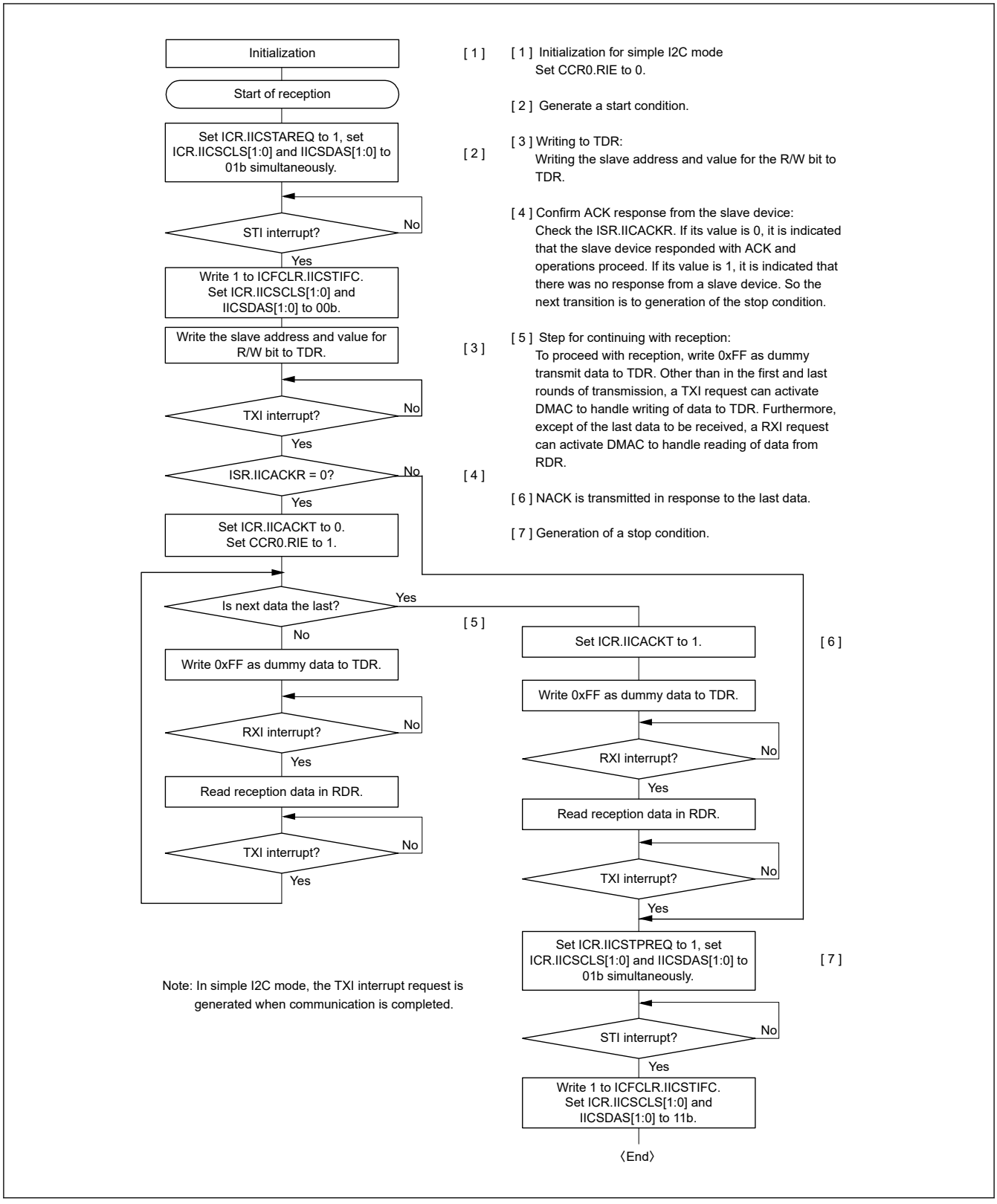


Figure 33.84 Example of the procedure for master reception operations in simple I2C mode (when ICR.IICINTM is 1, and transmission interrupts and reception interrupts are in use.)

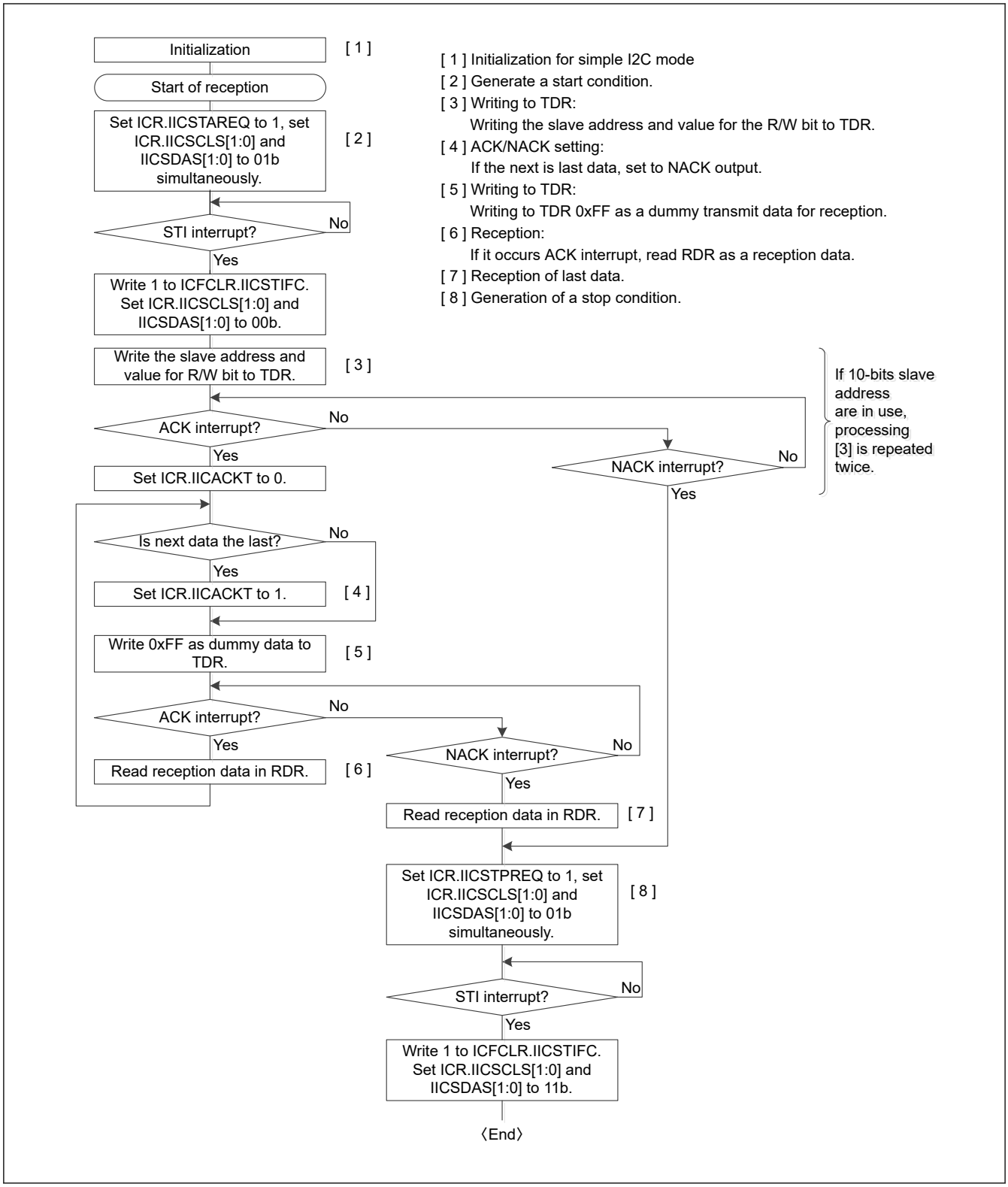


Figure 33.85 Example of the procedure for master reception operations in simple I2C mode (when ICR.IICINTM is 0, and ACK interrupts and NACK interrupts are in use.)

33.9 Clock Synchronous Mode

Figure 33.86 shows the communication data format of clock synchronous serial communication.

In clock synchronous mode, data is transmitted and received in synchronization with clock pulses. A communication data character consists of 8-bit data. In clock synchronous mode, no parity bit can be added. In data transmission when CPHA = 1 and CPOL = 1, the SCI outputs data from the falling edge of the sync clock until the next falling edge. In data reception,

data is read at the rising edges of the sync clock. After 8-bit data is output, the communication line holds the final-bit output state. In slave communication when CPHA = 1, however, the communication line holds the first-bit output state.

Because the SCI has an internal transmitter and a receiver independently, SCI enable full-duplex communication by sharing a communication clock of the transmitter and the receiver. Furthermore, because both the transmitter and the receiver have a double- buffer structure, continuous transmission and reception are possible by writing the next transmit data during transmission and reading the previous receive data during reception.

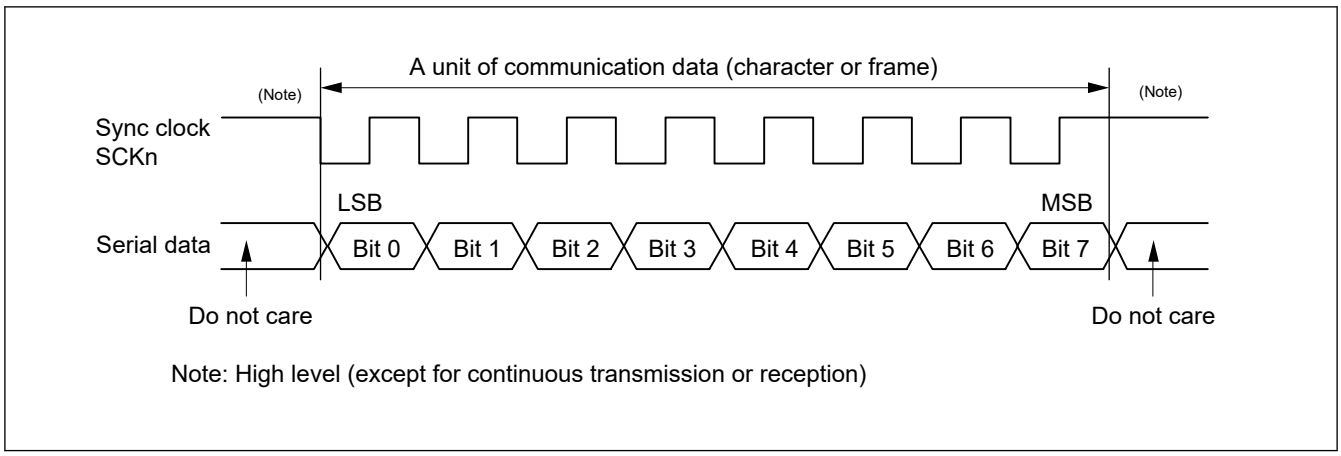


Figure 33.86 Data format of clock synchronous serial communication (LSB first, CPHA = 1, CPOL = 1)

33.9.1 Clock

(1) When the internal clock is selected

When the CCR3.CKE[1:0] bits are set to 00b or 01b (master mode), the internal clock generated by the baud rate generator can be selected and the sync clock is output from the SCKn pin. Eight pulses of the sync clock are output during single-character transmission/reception. The sync clock remains at a high level[*1] while no transmission or reception is performed. In transmission-only or transmission/reception, the sync clock is not output unless transmit data is prepared.

When the internal clock is selected, the clock with a delay from the SCKn signal is used for the master reception sampling clock. This ensures the data setup time and hold time for high-speed communication.

Note 1. When CCR3.CPOL = 1, the sync clock stops at a high level. When CCR3.CPOL = 0, the sync clock stops at a low level.

(2) When the external clock is selected

When the CCR3.CKE[1:0] bits are set to 10b or 11b (slave mode), data is transmitted and received using the external clock that is input from the SCKn pin.

33.9.2 CTS and RTS Functions

The CTS function performs transmission/reception and controls transmission start using the CTSn# pin input when the internal clock is selected. Setting the CCR1.CTSE bit to 1 enables the CTS function. Two settings are possible for the CTS/RTS pin: dual setting for using either function with a single pin, and dedicated setting for using each function simultaneously with two pins. The CCR1.CTSPEN register is used for these settings.

When the CTS function is enabled, transmission/reception and transmission start only when the CTSn# pin input level is low. Even if the CTSn# pin input becomes high level during transmission/reception or transmission operation, frames that are being transmitted/received or being transmitted are not affected and transmission/reception or transmission operation continues.

The RTS function makes a serial communication start request using the RTSn# pin output when the external sync clock is selected. When serial communication is enabled, the RTSn# pin outputs a low level. A low level and a high level are output under the following conditions.

[Conditions for low level]

(1) When a non-FIFO buffer is selected, all the following conditions are met.

- CCR0.RE bit or TE bit = 1
- Serial communication is enabled.
- No receive data is present before reading. (when CCR0.RE bit = 1)
- Data before transmission start is remaining in the TSR register. (when CCR0.TE bit = 1)
- CSR.ORER flag = 0

(2) When a FIFO buffer is selected, all the following conditions are met.

- CCR0.RE bit or TE bit = 1
- Serial communication is enabled
- The number of receive data stored in the receive FIFO (RDR register) is less than the value set in FCR.RSTRG[4:0]. (when CCR0.RE bit = 1)
- Unsent data is remaining in the transmit FIFO (TDR register). (when CCR0.TE bit = 1 and CCR3.CKE[1] bit = 0)
- Data before transmission start is remaining in the TSR register. (when CCR0.TE bit = 1 and CCR3.CKE[1] bit = 1)
- ORER (RDR.ORER) flag = 0

[Conditions for high level]

(1) When a non-FIFO buffer is selected

If conditions for low level are not met:

When CCR0.RE is set to 0 without reading the RDR register to terminate reception after reception is complete, the RTSn# pin output level remains high. At this time, write 0 to CCR0.RE.

(2) When a FIFO buffer is selected

If conditions for low level are not met:

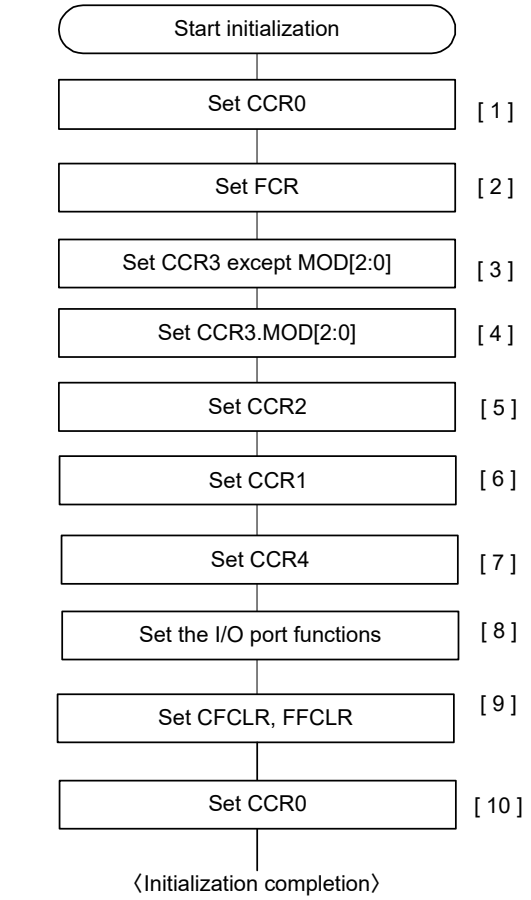
33.9.3 Initializing the SCI (Clock Synchronous Mode)

Before starting data transmission/reception, write 0 to CCR0.TE and CCR0.RE (or write initial values to the CCR0 register) and initialize the SCI according to the flowchart example in Figure 33.87.

Before changing operating mode or communication format, also be sure to write 0 to CCR0.TE and CCR0.RE.

Note that writing 0 to the RE bit does not initialize the ORER, FER, PER, and RDRF flags in CSR and the RDR register. Also note that writing 0 to the TE bit does not initialize the TEND flag when a FIFO buffer is selected. Attention is also needed for changing operating mode.

When CCR0.TIE = 1, note that setting the TE bit to 1 from 0 generates a TXI interrupt.



Note 1. In simultaneous transmit and receive operations, the TE and RE bits in CCR0 should both be to 0 or set to 1 simultaneously.

Note 2. Set CPOL and CPHA before setting the Communication mode.

Note 3. If you do not use the bit rate modulation function, you do not need to set it.

Note 4. If you use an external clock, you do not need to set it.

Note 5. When using the internal clock (master), the setting of reception only is prohibited.

[1] Set the following bits of CCR0 to 0[*1].
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the TFRST and RFRST to 1 to empty FIFO.
Set the TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits.

[3] Set CCR3 except of communication mode.
FIFO use / no-use
Transmission / reception format
Clock setting
Leave unused bits (CHR[1:0], STP, RXDESEL, MP, DE, ACS0,GM,BLK) at their initial values.

[4] Set communication mode (MOD[2:0] = 010b)[*2].

[5] Set up the bit-rate-modulation function[*3 *4], select clock, set bit rate[*4]. Leave unused bits (BCP[2:0], ABCS, ABCSE) at their initial values.

[6] Set up the Noise-filter function, the loop-back function, communication terminal status, the parity check function, and the CTS/RTS function.

[7] Set up the adjust sampling timing function.
Leave unused bits (CMPD[8:0], ATEN, ATT[2:0], AET) at their initial values.

[8] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, SCKn, CTS#, and RTSn# pins.

[9] Write 1 to the following bits and clear the corresponding flag.
CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
FFCLR.DRC

[10] Set the TE or RE bit to 1.[*1 *5] To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Figure 33.87 SCI initialization flowchart example (clock synchronous mode)

33.9.4 Serial Data Transmission (Clock Synchronous Mode)

(1) Non-FIFO selected

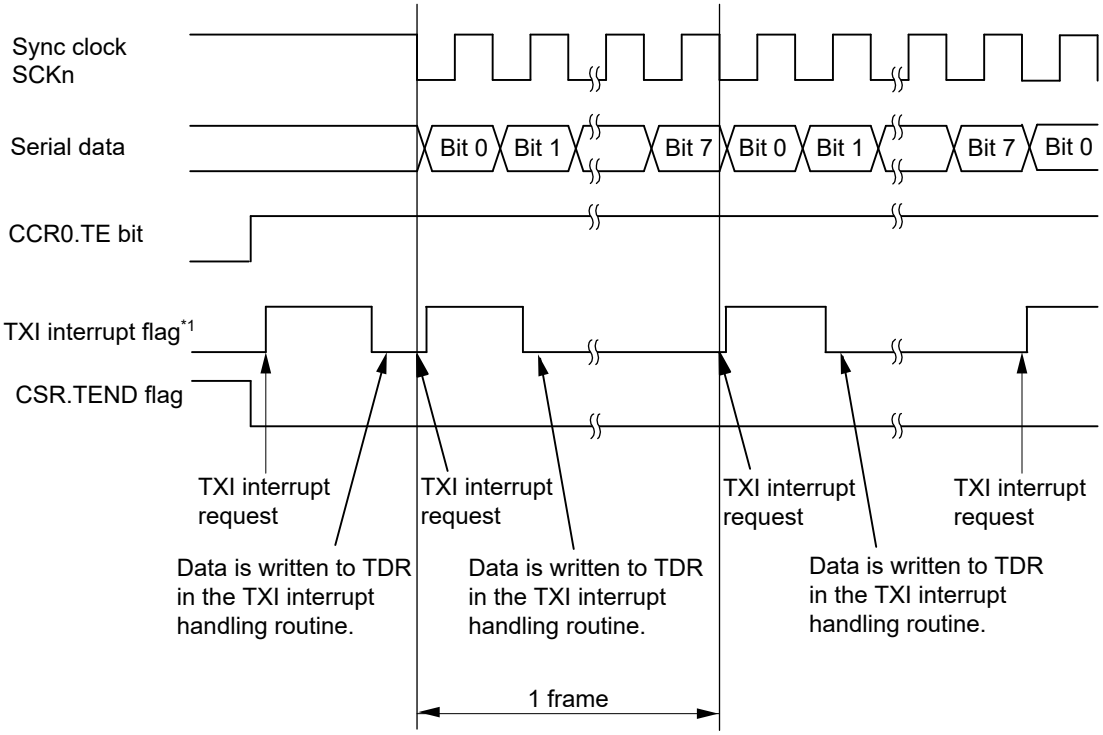
Figure 33.88, Figure 33.89, and Figure 33.90 show operation examples of serial transmission in clock synchronous mode. The SCI operates as follows during serial data transmission.

1. When data is written to the TDR register in the TXI interrupt routine, the SCI transfers the data from the TDR register to the TSR register. When starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a TXI interrupt request is generated.
2. The written data is transferred from the TDR register to the TSR register, which starts transmission. When the CCR0.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Writing the next transmit data to the TDR register before transmission of data transferred previously in the TXI interrupt routine is complete enables continuous transmission. When a TEI interrupt request is used, after the final transmit data is written to the TDR register in the TXI

interrupt request processing routine and the final data's transmission is started, set 0 to the CCR0.TIE bit and set 1 to the TEIE bit.

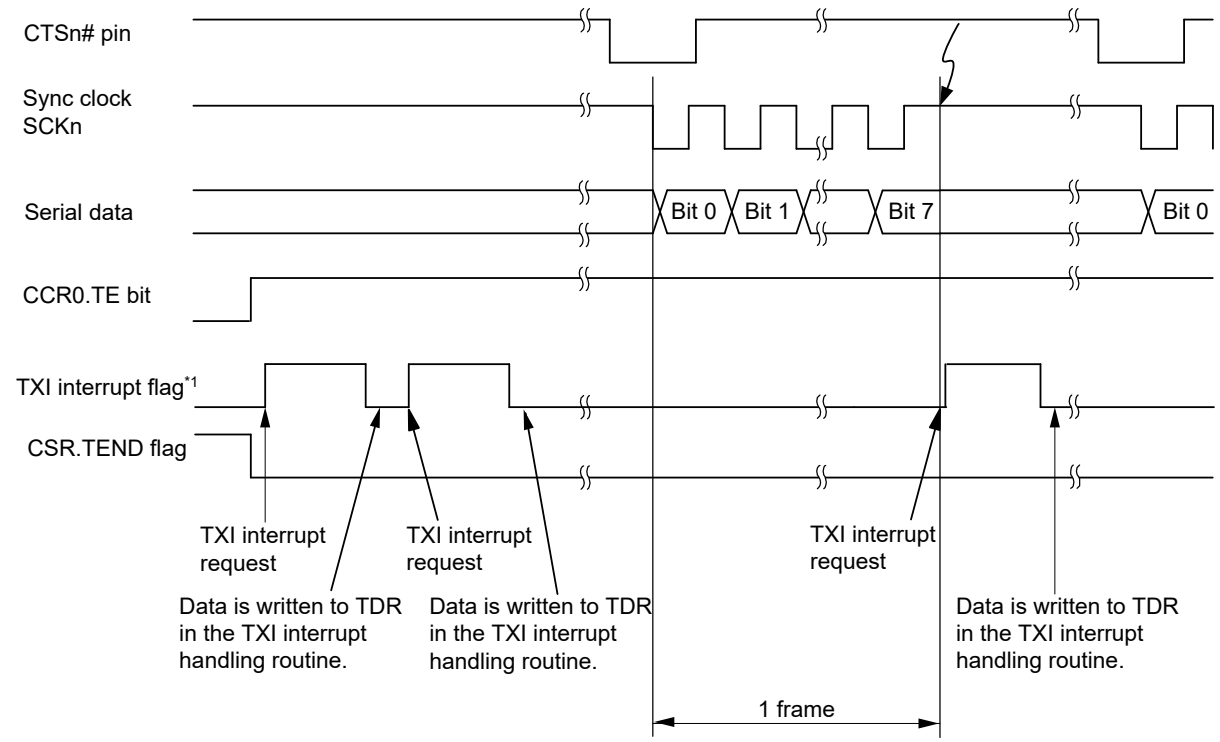
3. The TXDn pin outputs 8-bit data in synchronization with the output clock (in clock output mode) or with the input clock (when external clock is selected). When the CCR1.CTSE bit = 1 (CTS function enabled), the output clock starts after the CTS signal input becomes low level.
4. Update (data write) of the TDR register is checked at the final-bit transmission timing.
5. When the TDR register has been updated, data is transferred from the TDR register to the TSR register to start sending the next frame.
6. If the TDR register has not been updated, the CSR.TEND flag is set to 1 and the final-bit output state is retained. When the CCR0.TEIE bit is set to 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 33.91 shows an example of data transmission flowchart.



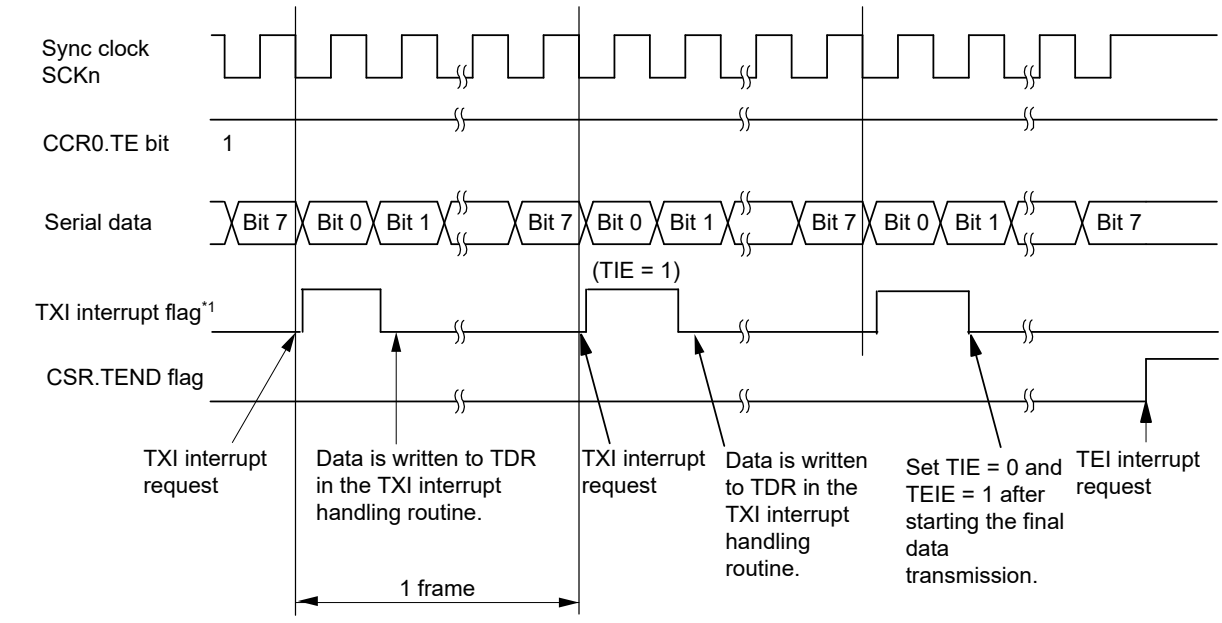
Note 1. For corresponding interrupt source numbers, see section 12, Interrupt Controller (ICU).

Figure 33.88 Serial transmission example in clock synchronous mode (1) (CTS function not used / transmission start / CPHA = 1, CPOL = 1)



Note 1. For corresponding interrupt source numbers, see section 12, Interrupt Controller (ICU).

Figure 33.89 Serial transmission example in clock synchronous mode (2) (CTS function used / transmission start / CPHA = 1, CPOL = 1)



Note 1. For corresponding interrupt source numbers, see section 12, Interrupt Controller (ICU).

Figure 33.90 Serial transmission example in clock synchronous mode (3) (during transmission to transmission end / CPHA = 1, CPOL = 1)

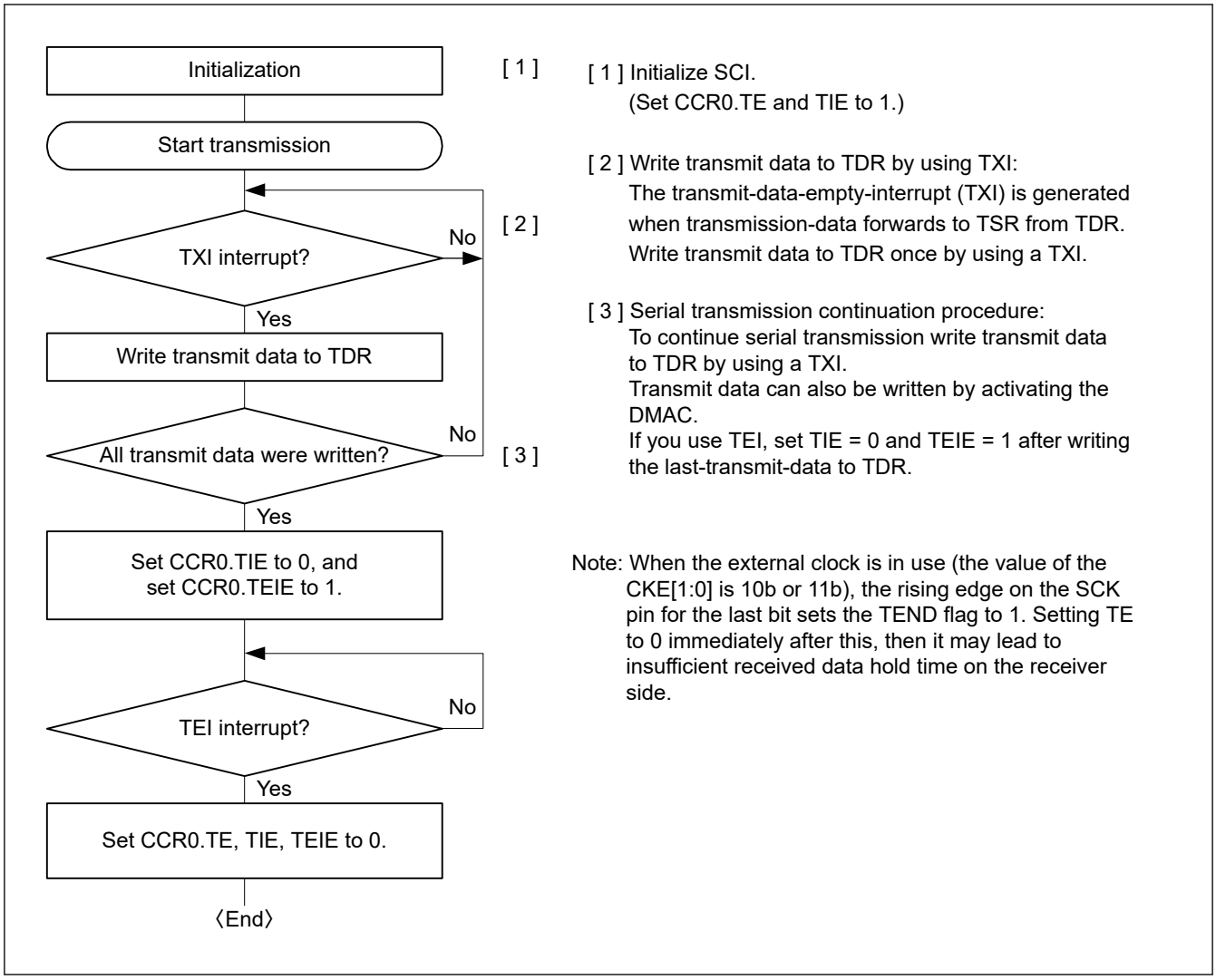


Figure 33.91 Serial transmission flowchart example in clock synchronous mode (non-FIFO selected)

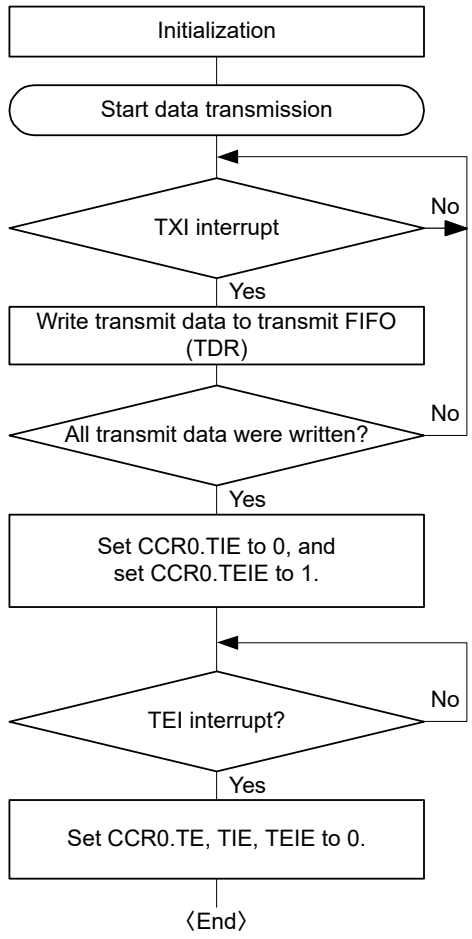
(2) FIFO selected

Figure 33.92 shows an example of flowchart of serial transmission (a FIFO buffer selected) in clock synchronous mode. The SCI operates as follows when serial data transmission.

1. When data is written to the transmit FIFO (TDR register) in the TXI interrupt routine, the SCI transfers the data from the transmit FIFO (TDR register) to the TSR register. The number of writable transmit data is [N_{FIFO} (= 16) - number of unsent transmit data stored in the transmit FIFO (TDR register)]. In addition, when starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a TXI interrupt request is generated.
2. Data is transferred from the transmit FIFO (TDR register) to the TSR register and transmission starts. When the number of data stored in the transmit FIFO (TDR register) is equal to or less than the threshold value of the transmit FIFO, the CSR.TDRE flag is set to 1. When the CCR0.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Writing the next transmit data to the transmit FIFO (TDR register) in the TXI interrupt routine before transmission of data written to the transmit FIFO (TDR register) is complete enables continuous transmission. When a TEI interrupt request is used, after the final transmit data is written to the transmit FIFO (TDR register) in the TXI interrupt request processing routine, set 0 to the CCR0.TIE bit and set 1 to the TEIE bit.
3. The TXDn pin outputs 8-bit data in synchronization with the output clock (in clock output mode) or with the input clock (when external clock is selected). When CCR1.CTSE = 1 (CTS function enabled), the output clock starts after the CTS signal input becomes low level.
4. The SCI checks whether unsent transmit data is remaining in the transmit FIFO (TDR register)[*1] at the final-bit transmission timing.

5. When data is remaining in the transmit FIFO (TDR register), the data is transferred from the transmit FIFO (TDR register) to the TSR register to start sending the next frame.
6. If no data is remaining in the transmit FIFO (TDR register), the CSR.TEND flag is set to 1 and the final-bit output state is retained. When the CCR0.TEIE bit is set to 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Note 1. The number of unsent transmit data stored in the TDR register (transmit FIFO) can be monitored by reading the FTSR.T[5:0] bits.



[1] [1] Initialize SCI.
(Set CCR0.TE and TIE to 1.)

[2] [2] Write transmit data to transmit FIFO (TDR) by using TXI:
The transmit-data-empty-interrupt (TXI) is generated when transmission-data forwards to TSR from the transmit FIFO (TDR) and the number of transmit-data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[3] [3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the transmit FIFO (TDR) by using a TXI interrupt and clear TDRE flag to 0.
The number of transmission data it's possible to write in is N_{FIFO} (= 16) - (the number of stored transmit FIFO data).
Transmit data can also be written to the transmit FIFO (TDR) by activating the DMAC. In this case, the TDRE flag is cleared automatically, so please do not operate the TDRE flag.
If you use TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR.

Note: When the external clock is in use (the value of the CKE[1:0] is 10b or 11b), the rising edge on the SCK pin for the last bit sets the TEND flag to 1. Setting TE to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

Figure 33.92 Serial transmission flowchart example in clock synchronous mode (FIFO buffer selected)

33.9.5 Serial Data Reception (Clock Synchronous Mode)

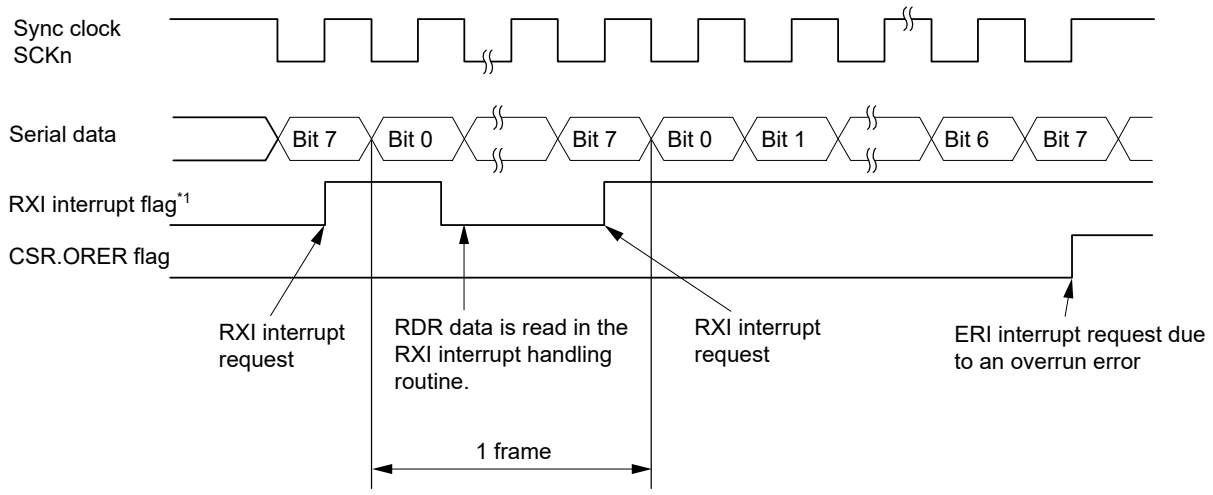
(1) Non-FIFO selected

Figure 33.93 and Figure 33.94 show operation examples of serial data reception in clock synchronous mode.

The SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

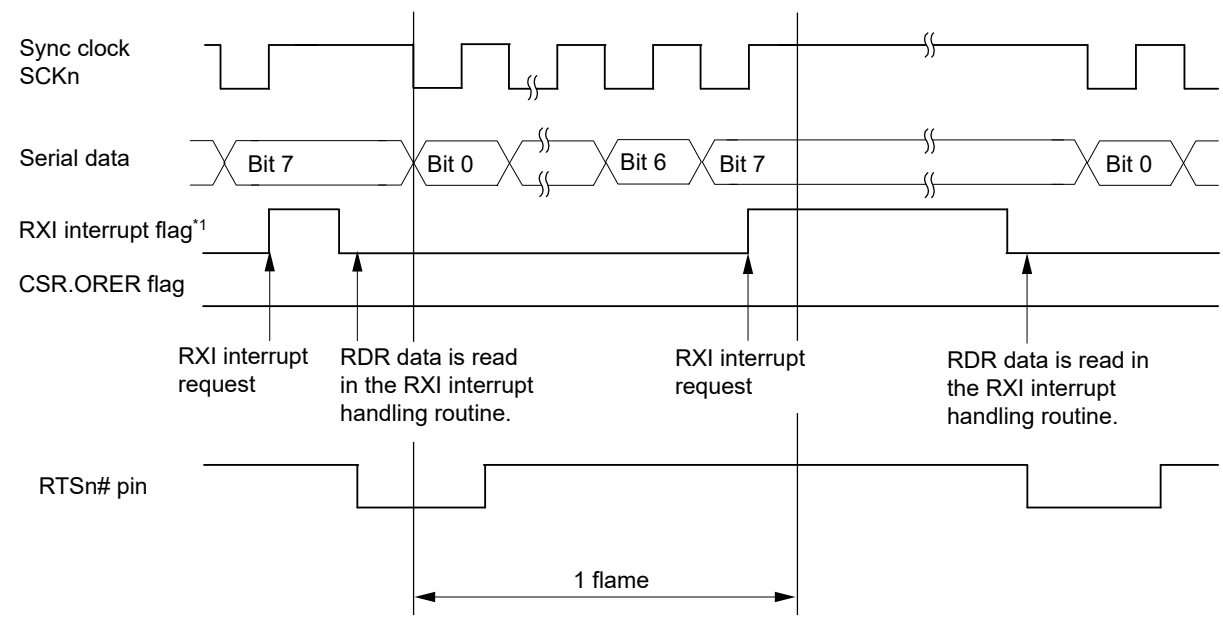
1. When the CCR0.RE bit is set to 1, the RTSn# pin output becomes low level (when the RTS function is used).
2. The SCI starts data reception in synchronization with input or output of the sync clock, and transfers receive data to the RSR register.
3. When an overrun error occurs, the CSR.ORER flag is set to 1. When the CCR0.RIE bit = 1 at this time, an ERI interrupt request is generated and the received data is not transferred to the RDR register.

4. When data is normally received, the received data is transferred to the RDR register. When the RIE bit = 1 at this time, an RXI interrupt request is generated. Reading the received data transferred to the RDR register in the RXI interrupt handling routine before the next data is completely received enables continuous reception. When the received data transferred to the RDR register is read, the RTSn# pin output becomes low level (when the RTS function is used). If you want to prevent the RTSn# pin output from turning low level after the final data is received, clear the CCR0.RE bit to 0 and then read the RDR register.



Note 1. For corresponding interrupt source numbers, see section 12, Interrupt Controller (ICU).

Figure 33.93 Serial data reception example in clock synchronous mode (1) (RTS function not used / CPHA = 1, CPOL = 1)



Note 1. For corresponding interrupt source numbers, see section 12, Interrupt Controller (ICU).

Figure 33.94 Serial data reception example in clock synchronous mode (2) (RTS function used / CPHA = 1, CPOL = 1)

While the reception error flag is set to 1, subsequent reception is disabled. Therefore, before continuing reception, be sure to clear the ORER flag in CSR to 0. Also be sure to read the RDR register in the overrun error processing. If the CCR0.RE bit

is set to 0 during reception to forcibly terminate the reception operation, unread receive data may be remaining in the RDR register. In this case, read the RDR register.

Figure 33.95 shows an example of serial data reception flowchart.

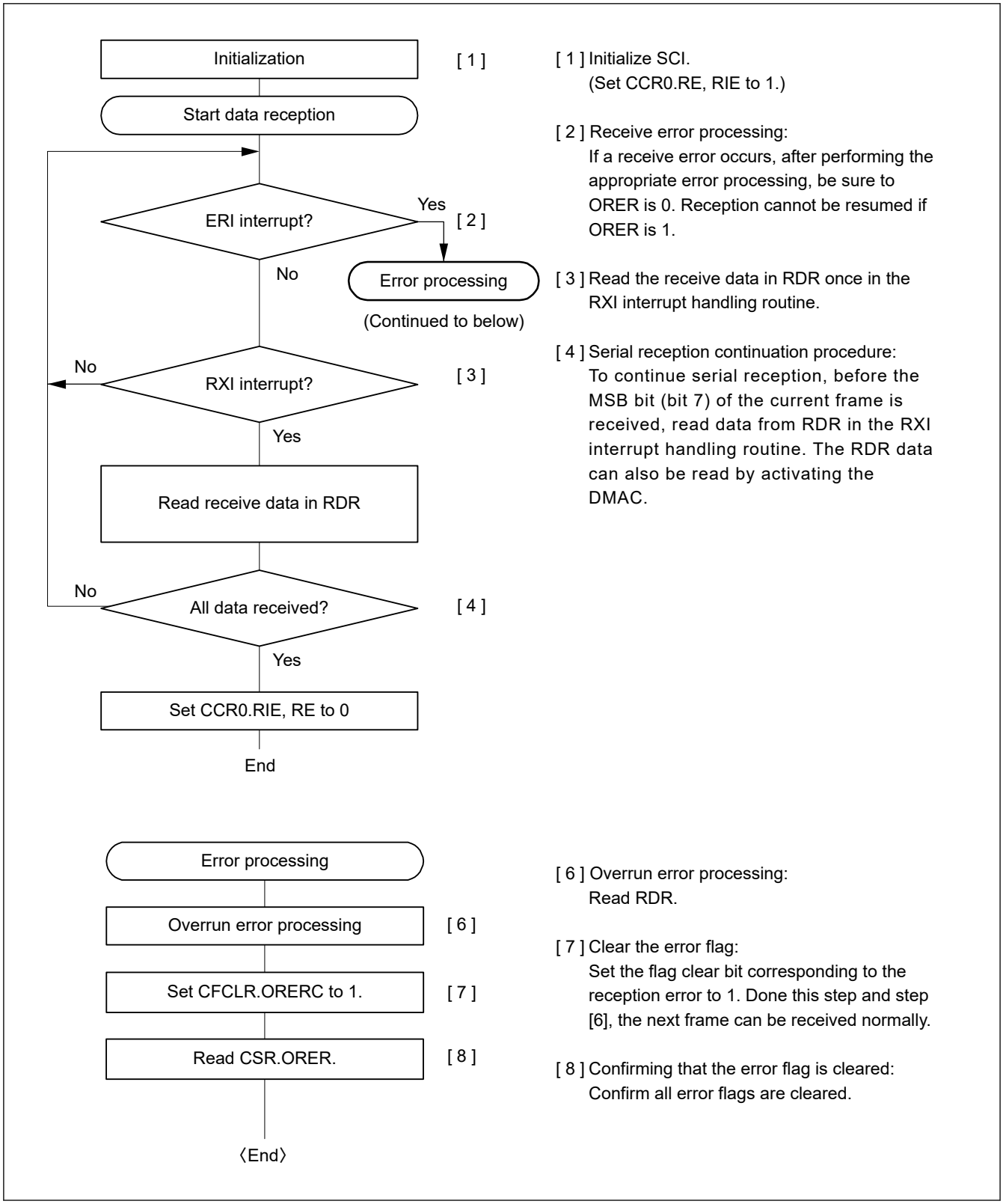


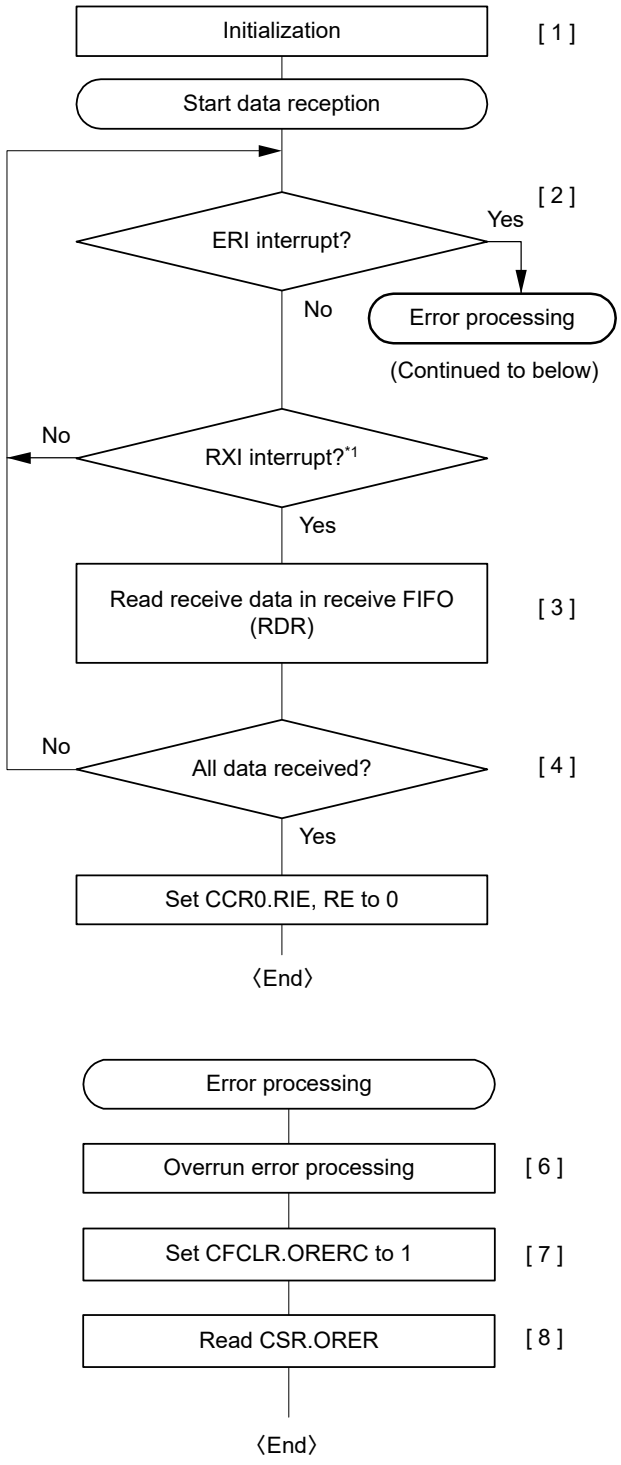
Figure 33.95 Sample flowchart for serial reception in clock synchronous mode (non-FIFO selected)

(2) FIFO selected

Figure 33.96 shows an example of serial data reception flowchart (a FIFO buffer selected) in clock synchronous mode. The SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the CCR0.RE bit is set to 1, the RTSn# pin output turns low level (when the RTS function is used).
2. The SCI starts receiving data in synchronization with input or output of the sync clock and transfers the received data to the receive FIFO (RDR register).
3. When an overrun error occurs, the CSR.ORER flag is set to 1. When the CCR0.RIE bit = 1 at this time, an ERI interrupt request is generated and the received data is not transferred to the receive FIFO (RDR register)*1.
4. When data is normally received, the received data is transferred to the receive FIFO (RDR register)*1. When the number of receive data stored in the receive FIFO (RDR register) is equal to or more than the threshold value of the receive FIFO, the CSR.RDRF flag is set to 1. When the RIE bit = 1 at this time, an RXI interrupt request is generated. Reading the received data transferred to the receive FIFO (RDR register) in the RXI interrupt handling routine before an overrun error occurs enables continuous reception. When the received data transferred to the receive FIFO (RDR register) is read and the number of data becomes lower than the RTS# output threshold value, the RTSn# pin output becomes low level (when the RTS function is used).

Note 1. The RDR.RDAT[8] bit is not used.



[1] Initialize SCI.
(Set CCR0.RE, RIE to 1.)

[2] Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to ORER is 0. Reception cannot be resumed if ORER is 1.

[3] Read the receive data in the receive FIFO (RDR) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by FRSR.R[5:0].

[4] Serial reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from RDR and clear RDRF flag to 0. The RDR data can also be read by activating the DMAC. In this case, the RDRF flag is cleared automatically, so please do not operate the RDRF flag.

Note 1. Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, return to step [2] and it will become an infinite loop, so set a timeout time separately using a timer etc.

[6] Overrun error processing:
Read RDR.

[7] Clear the error flag:
Set the flag clear bit corresponding to the reception error to 1. Done this step and step [6], the next frame can be received normally.

[8] Confirming that the error flag is cleared:
Confirm all error flags are cleared.

Figure 33.96 Example of serial reception in clock synchronous mode (FIFO selected)

33.9.6 Serial Data Transmission and Reception (Clock Synchronous Mode)

(1) Non-FIFO selected

Figure 33.97 shows an example of serial data concurrent transmission/reception flowchart in clock synchronous mode. After the SCI is initialized, perform the following procedure for serial data concurrent transmission/reception. When switching mode from transmission to concurrent transmission/reception, check that the CSR.TEND flag is set to 1 to ensure that the

SCI is in the transmission complete state. Then set CCR0.TE = 0 and RE = 0 and then set the TE, RE, TIE, and RIE bits in CCR0 to 1 simultaneously by a single instruction.

When switching mode from reception to concurrent transmission/reception, check that the SCI is in the reception complete state, and then set CCR0.TE = 0 and RE = 0. After that, check that the error flags (ORER, FER, and PER) in CSR are cleared to 0, and then set the TE, RE, TIE, and RIE bits in CCR0 to 1 simultaneously by a single instruction.

When the RTS function is used in the concurrent transmission/reception operation, if you want to prevent the RTSn# pin output from turning to low after the final data is received as in the reception operation, clear the RE and TE bits in CCR0 to 0 simultaneously, and then read the RDR register.

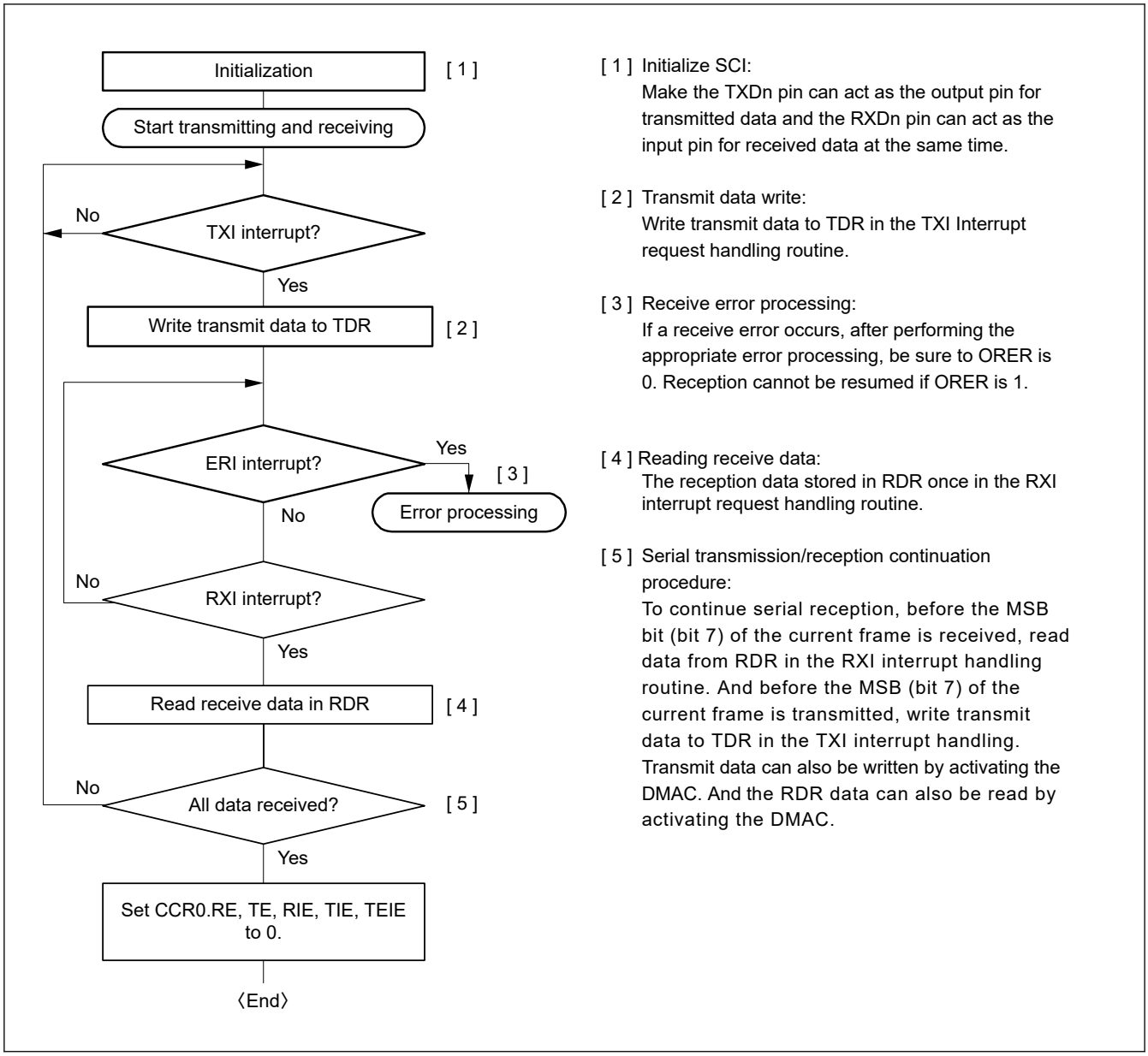


Figure 33.97 Serial data concurrent transmission/reception flowchart example in clock synchronous mode (non- FIFO selected)

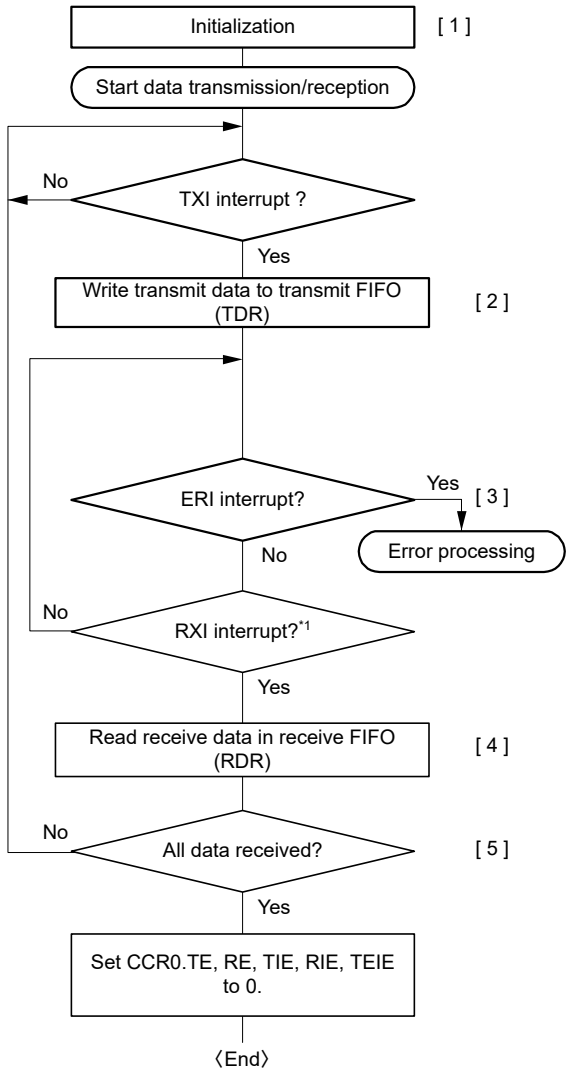
(2) FIFO selected

Figure 33.98 shows an example of serial data concurrent transmission/reception flowchart (a FIFO buffer selected) in clock synchronous mode.

After the SCI is initialized, perform the following procedure for serial data concurrent transmission/reception. When switching mode from transmission to concurrent transmission/reception, check that the CSR.TEND flag is set to 1 to ensure

that the SCI is in the transmission complete state. Then set CCR0.TE = 0 and RE = 0 and then set the TE, RE, TIE, and RIE bits in CCR0 to 1 simultaneously by a single instruction.

When switching mode from reception to concurrent transmission/reception, check that the SCI is in the reception complete state, and then set CCR0.TE = 0 and RE = 0. After that, check that the error flags (ORER, FER, and PER) in CSR are cleared to 0, and then set the TE, RE, TIE, and RIE bits in CCR0 to 1 simultaneously by a single instruction.



[1] Initialize SCI:
Make the TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to transmit FIFO (TDR) in the TXI Interrupt request handling routine.
The number of transmission data it's possible to write in is N_{FIFO} (= 16) - (the number of stored transmit FIFO data).

[3] Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to ORER is 0. Reception cannot be resumed if ORER is 1.

[4] Read receive data:
Read the receive data in the receive FIFO (RDR) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by FRSR.R[5:0].

[5] Serial transmission/reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from RDR and clear CSR.RDRF.
To continue serial transmission, write transmit data to the transmit FIFO (TDR) by using a TXI interrupt and clear TDRE flag to 0.
Transmit data can also be written to the transmit FIFO (TDR) by activating the DMAC. And The RDR data can also be read by activating the DMAC. In this case, the TDRE and RDRF are cleared automatically, so please do not operate the TDRE and RDRF.

Note 1. Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, return to step [2] and it will become an infinite loop, so set a timeout time separately using a timer etc.

Figure 33.98 Serial data concurrent transmission/reception flowchart example in clock synchronous mode (FIFO selected)

33.9.7 Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used

When the clock synchronous internal clock is used (master mode), MRCLK is used as a reception sampling clock.

This function adjusts the reception sampling timing by delaying MRCLK by 1 to 4 PCLKSCIn and adding a digital delay.

Setting the CCR4.ASEN bit to 1 enables this function. The delay value is set in CCR4.AST[1:0].

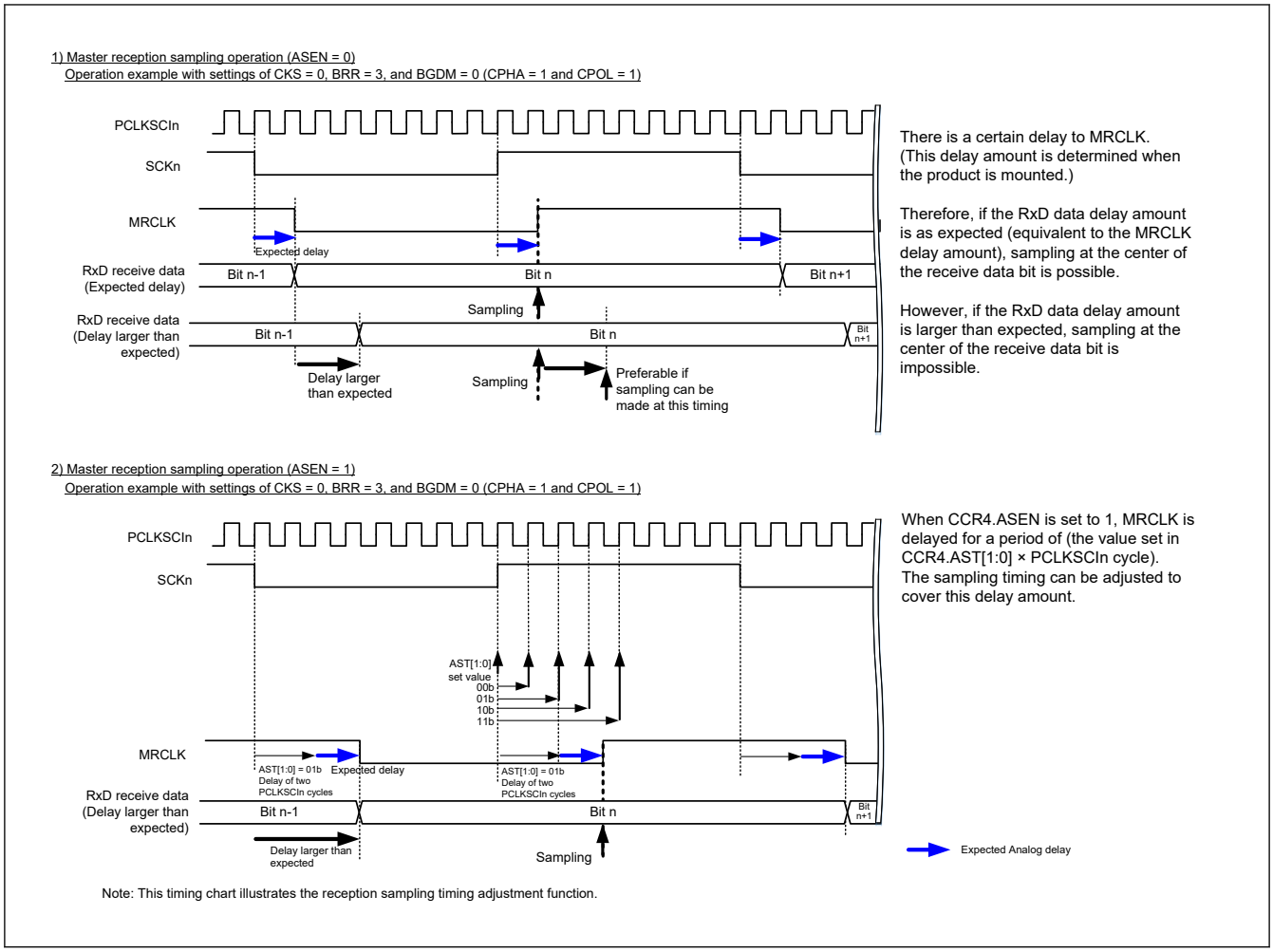


Figure 33.99 Reception sampling timing adjustment operation in clock synchronous mode (master)

33.10 Simple-SPI Mode

As an extended function of the SCI, the SCI supports Simple-SPI (4-wire serial bus) mode that allows communication from one or more master devices to multiple slave devices.

Simple-SPI mode is enabled by the Simple-SPI mode setting (CCR3.MOD[2:0] = 011b) and by setting the CCR0.SSE bit to 1. When using the 4-wire serial bus in master mode with a single master, the SS pin function on the master side is not required. Therefore, the CCR0.SSE bit is set to 0.

Figure 33.100 shows an example of Simple-SPI connection.

In simple-SPI mode, data is transmitted and received in synchronization with clock pulses in the same way as clock synchronous mode. A character of communication data consists of 8-bit data. No parity bit can be attached.

Because the SCI has an internal transmitter and a receiver independently, the transmitter and the receiver can share a clock to enable full-duplex communication. Furthermore, because both the transmitter and the receiver have a double-buffer structure, continuous transmission and reception are possible by writing the next transmit data during transmission and reading the previous receive data during reception.

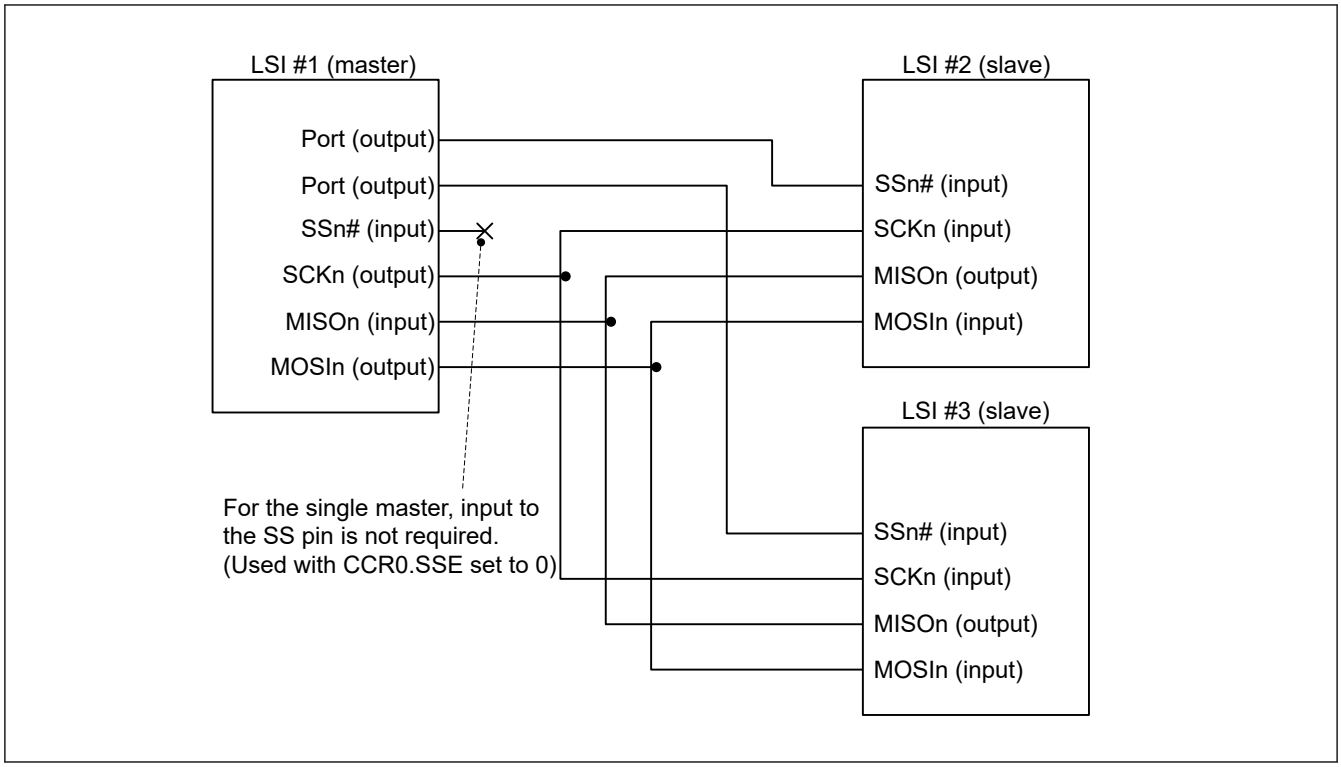


Figure 33.100 Simple-SPI connection example

33.10.1 Pin Status in Master and Slave Mode

In simple-SPI mode, input and output directions of each pin vary depending on master mode (CCR3.CKE[1:0] = 00b or 01b) and slave mode (CCR3.CKE[1:0] = 10b or 11b).

Table 33.33 shows the relationship among mode, SSn# pin input, and each pin state.

Table 33.33 Relationship among mode, SSn# pin input, and each pin state

Mode	SSn# pin input	TXDn (MOSIn) pin state	RXDn (MISOn) pin state	SCKn pin state
Master mode*1	High level (Communication enabled)	Transmit data output*2	Receive data input	Clock output*3
	Low level (Communication disabled)	High impedance	Receive data input (disabled)	High impedance
Slave mode	High level (Communication disabled)	Receive data input (disabled)	High impedance	Clock input (disabled)
	Low level (Communication enabled)	Receive data input	Transmit data output*2	Clock input

Note 1. When single master is selected (SSE bit = 0), communication is enabled (equivalent to the state when the SSn# pin input is high level) regardless of the SSn# pin input level. The SSn# pin is not used and is available for other purposes.
Note 2. High impedance when transmission is disabled (CCR0.TE bit = 0)
Note 3. High impedance when multi-master (SSE bit = 1) is selected and transmission/reception is disabled (CCR0.TE, RE bits = 00b)

33.10.2 SS Function in Master Mode

Setting the CCR3.CKE[1:0] bits to 00b or 01b enables master mode.

In single master mode (SSE bit = 0), the SSn# pin is not used and data transmission and reception are enabled regardless of the SSn# pin input level. The SSn# pin is available for other purposes.

When in multi-master mode (SSE bit = 1) and the SSn# pin input is high, the master outputs a clock from the SCKn pin and performs transmission and reception operations. And outputting clock indicates "There are no other masters" or "Another master is not performing reception or transmission". When the SSn# pin input level is low in multi-master mode (SSE bit = 1), this means that another master exists and it is performing data transmission/reception. At this time, the SCI makes the TXDn pin output and the SCKn pin output high impedance and does not start data transmission/reception. In

addition, the CSR.MFF bit is set to 1 as a mode fault error. In multi-master mode, read this flag bit to perform the error processing. If a mode fault error occurs during the transmission/reception operation, the SCKn pin and the TXDn pin output are made high impedance while the SSn# pin input level is low. When the SSn# pin input becomes high level, the SCKn pin outputs a clock signal and the TXDn pin outputs data. Even if the SCKn pin and the TXDn pin are in the high impedance state, internal transmission/reception operation continues, but it stops after transmission/reception of a single character is complete. In this case, any of TXI, RXI, and TEI interrupts occurs.

Control the SSn# pin output in master mode with a general-purpose port.

33.10.3 SS Function in Slave Mode

Setting the CCR3.CKE[1:0] bits to 10 or 11 enables slave mode.

When the SSn# pin input level is high, the RXDn pin output becomes high impedance and the clock input from the SCKn pin is ignored. When the SSn# pin input level is low, the clock input from the SCKn pin becomes effective, enabling data transmission and reception.

When the SSn# pin input changes from low to high level during the transmission/reception operation, the RXDn pin output is made high impedance and the transmission/reception operation is immediately suspended. If the transmission is in progress, the CSR.TEND flag will not be set, a transmit end interrupt will not be output, and an abnormal stop status will occur. So, do not negate the SSn# pin during slave transmission/reception. If an abnormal stop occurs, set CCR0.RE and CCR0.TE to 0 to stop transmission / reception. To resume transmission/reception, set CCR0.RE and CCR0.TE to 1 after at least PCLKSCIn × 3 cycles + PCLKM × 3 cycles.

33.10.4 Relationship between Clock and Transmit/Receive Data

The clock to be used for data transmission/reception is selectable from four types using the CCR3.CPOL and CPHA bits. Figure 33.101 shows the relationship between clock and transmit data/receive data. The same relationship applies to master mode and slave mode.

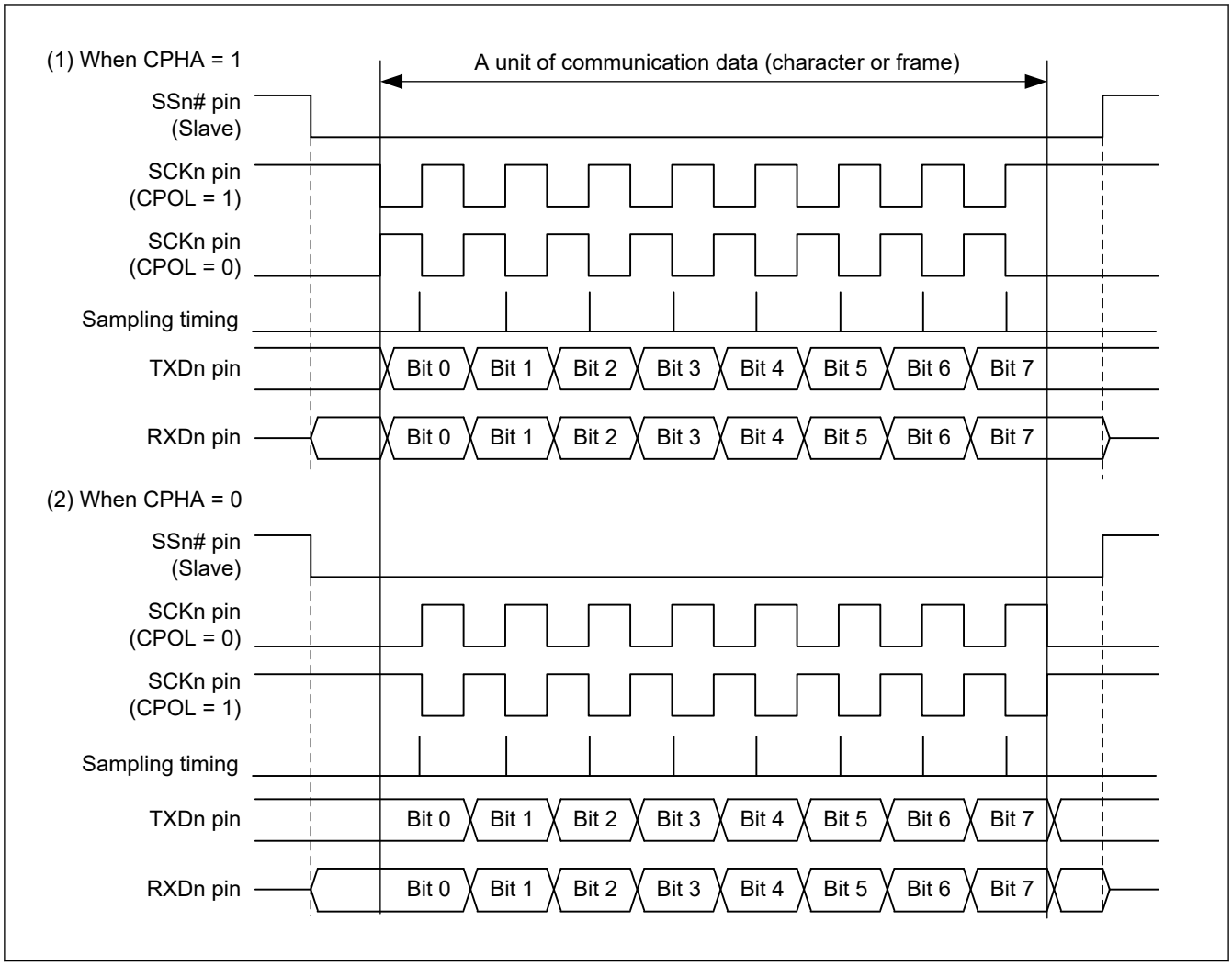


Figure 33.101 Relationship between clock and transmit data/receive data in simple-SPI mode

33.10.5 SCI Initialization in Simple-SPI Mode

The SCI can be initialized using the same initialization procedure as for clock synchronous mode (Figure 33.87). The master devices and slave devices use the same clock type selected by the CCR3.CPOL and CPHA bits.

Before performing initialization or changing operating mode or communication format, be sure to stop communication (CCR0.RE = 0 and CCR0.TE = 0).

Note that setting the RE bit to 0 does not initialize the ORER, FER, and PER flags in CSR and the RDR register.

Note that, when CCR0.TIE = 1, setting the TE bit to 1 from 0 generates a TXI interrupt.

33.10.6 Serial Data Transmission and Reception in Simple-SPI Mode

In master mode, set the SSn# pin of the destination slave device to low level before starting data transmission/reception and set to high level after the end of data transmission/reception. In multiple master operation with CCR0.SSE = 1 even in master mode, a mode fault error will occur if the SSn# pin goes low. Therefore, make sure that no mode fault error has occurred before starting communication, and start communication, and make sure that no mode fault error has occurred even after communication ends. If a mode fault error has occurred, communication may be incomplete, so measures such as retransmission are required. The other procedures are the same as in clock synchronous mode.

In slave mode, it operates according to the SSn# pin input level. Other steps are the same as those of clock synchronous mode.

33.10.7 Reception Sampling Timing Adjustment Function in Simple-SPI Mode with Internal Clock Used

The reception sampling timing adjustment function in simple SPI mode is the same as the reception sampling timing adjustment function in clock synchronous mode. For the description of operation, see section 33.9.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used.

33.11 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be corrected by evenly enabling the clocks of the number specified in the CCR2.MDDR[7:0] bits among 256 clock cycles of internal clocks which is selected by the CKS[1:0] bits in CCR2.

Figure 33.102 shows an example where the PCLKSCIn is selected by the CKS[1:0] bits in CCR2 and the BRR and MDDR are set to 0 and 160 respectively in asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256). The enable of internal clock has non-balanced. So, notice the pulse width of the internal base clock is caused bias.

Do not use this function in Clock Synchronous mode, Simple-SPI mode, Smart Card Interface mode, and Manchester mode.

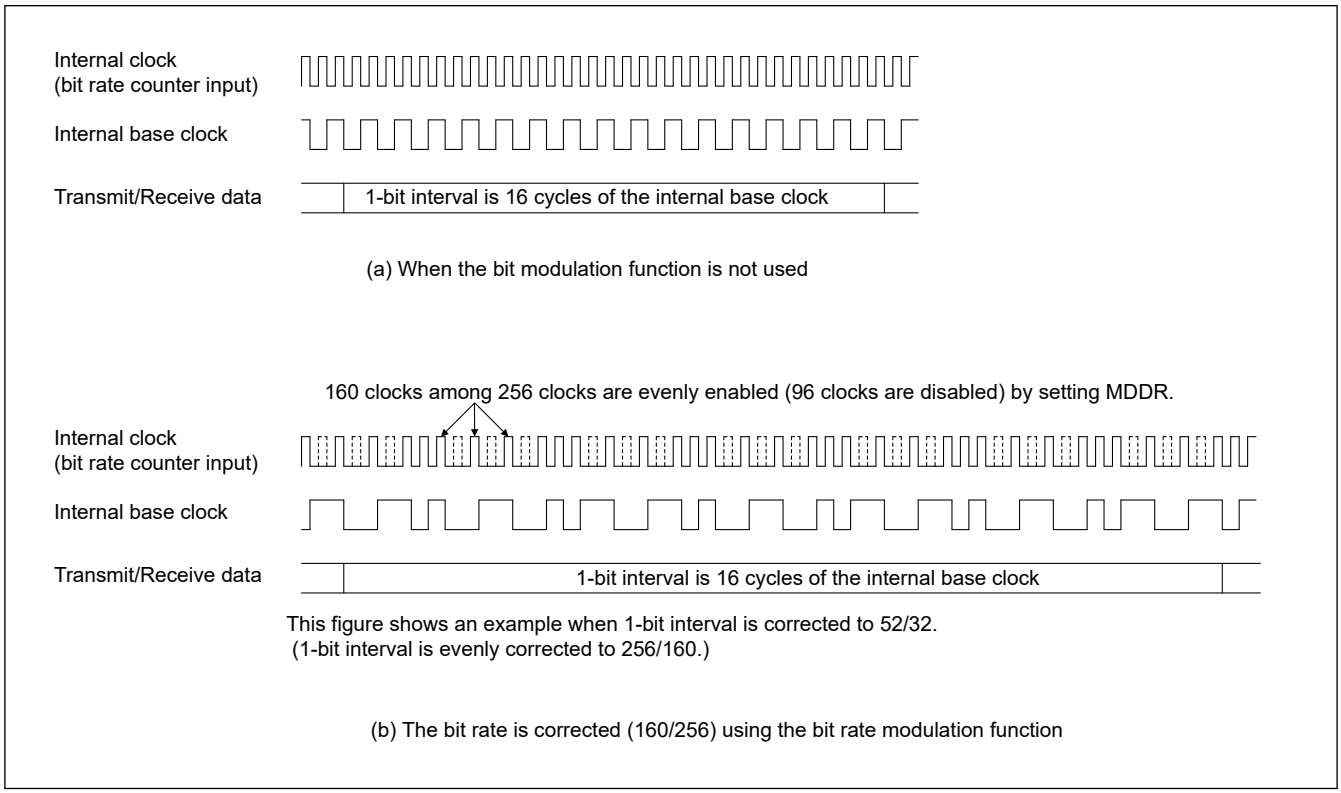


Figure 33.102 Example of internal base clock when bit rate modulation function is used

33.12 Noise Cancellation Function

Figure 33.103 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a two-stage flip-flop circuits and a match detection circuit. When the input signals of the noise filter and the output signals of the two-stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. (When the same level is retained for three cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for three cycles or shorter is considered as a noise, not as a receive signal.)

In asynchronous mode and Manchester mode, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The sampling period of the noise filter can be selected from the base clock period and the divided clock of the baud rate generator clock source by CCR1.NFCS[2:0].

(When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 0 and CCR2.ABCSE = 0, the cycle is 1/16 of a period 1 transfer bit.

When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 1 and CCR2.ABCSE = 0, the cycle is 1/8 of a period 1 transfer bit.

When CCR1.NFCS[2:0] = 000b, CCR2.ABCSE = 1, the cycle is 1/6 of a period 1 transfer bit.)

In simple I2C mode, the noise elimination function can be used for the input pins of SDAn and SCLn. The sampling period of the noise filter can be selected from the divided clock of the baud rate generator clock source by CCR1.NFCS[2:0].

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When CCR0.TE and CCR0.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, it is determined that a level match is detected and is conveyed as an internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive sampling cycles.

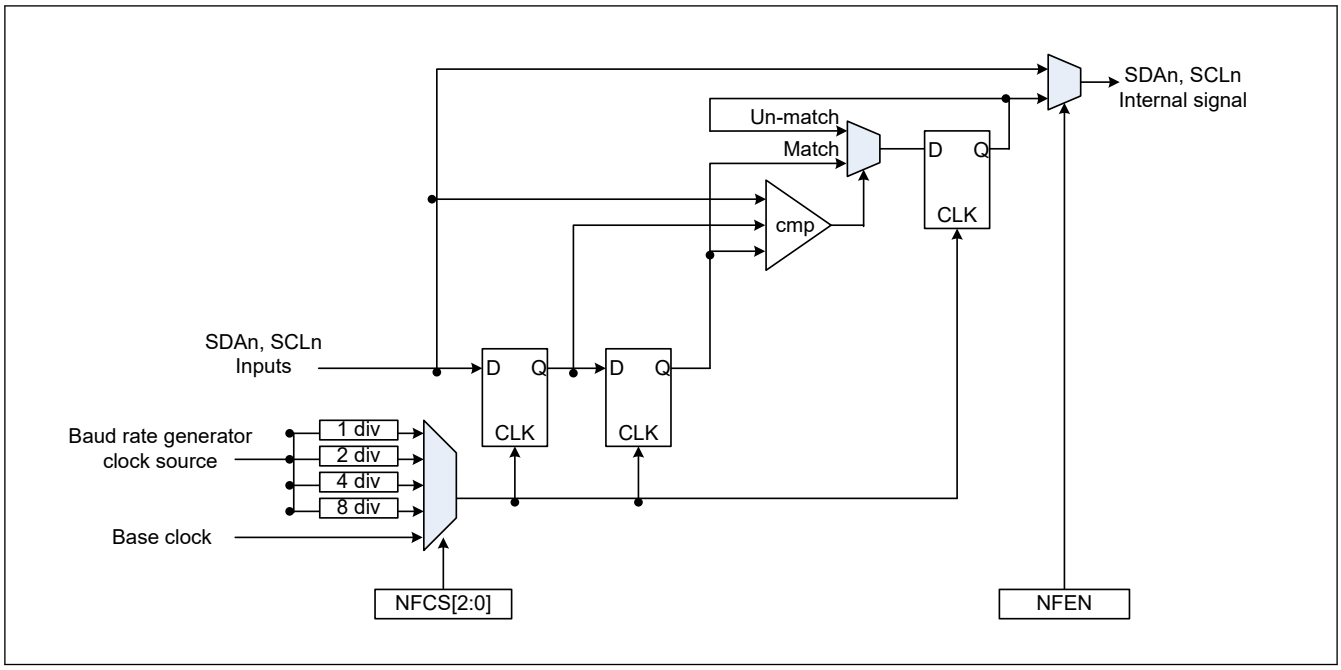


Figure 33.103 Block diagram of digital noise filter circuit

33.13 RS-485 Driver Control Function

Setting the DEN bit in the SCI Common Control Register 3 (CCR3) to 1 enables the RS-485 driver control function and generates a DE (Driver Enable) signal that enables the external transceiver transmission mode. The DE signal outputs a valid level for the period with assertion time and negation time added before and after data transmission. The DE signal valid level is set by the DEPOL bit in the driver control register (DCR).

The Assert time is the time from when the DE signal is valid until the start bit starts. Set by DEAST[4:0] of driver control register (DCR).

The Negate time is the time from the end of the last stop bit of the transmitted message to the invalidation of the DE signal. Set with DENGT[4:0] of the driver control register (DCR).

DEAST and DENGT are expressed in base clock units. For details, see section 33.3.13. DCR : Driver Control Register.

When this function is used (DEN = 1), the TEND set timing and TEI interrupt output timing are at the end of the driver negation time.

When transmission is completed and the next transmission data is not written before the DE signal is negated, the DE signal is negated once. If the timing for writing the next transmit data is not in time, assert the DE signal after negating it again, insert the set assert time, and transmit the next data. If you want to perform the next transmission with the DE signal asserted, write the next transmission data to the TDR quickly enough in consideration of the synchronization delay time of the register.

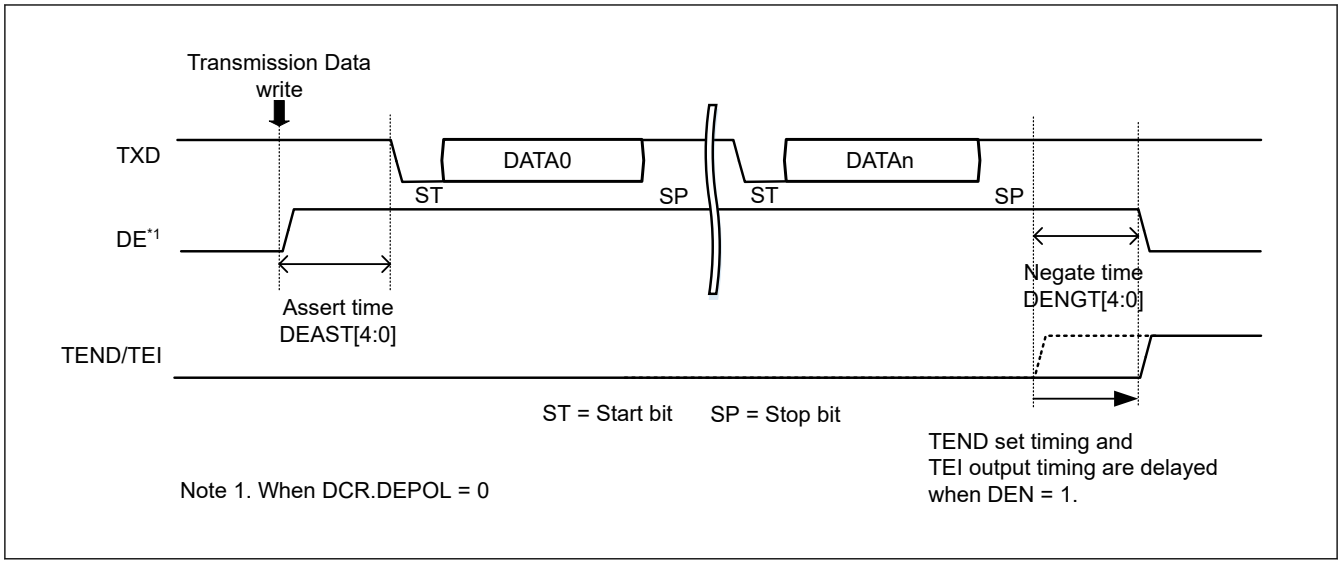


Figure 33.104 The image waveform for RS-485 driver control DE signal output

33.14 Loopback Function

The loopback function can be used in asynchronous mode with the internal clock, Manchester mode with the internal clock, and clock synchronous mode with the internal clock.

When 1 is written to the SPLP bit in the CCR1 register, SCI blocks the external input (RXD) path and connects the output path of the transmit data register and the input path of the receive data register.

When this function is used with TINV = 1, inversion of transmission data becomes reception data. However, this function can be used with TINV = 1 only when operating in clock synchronous mode internal clock.

Table 33.34 shows the relationship between the TINV and SPLP bit settings and the received data.

Table 33.34 TINV and SPLP bit settings and received data

TINV	SPLP	Receive data	Communication mode		
			Async internal clock	Manchester internal clock	Clock sync internal clock
x	0	Receive Data from RXD pin	Possible	Possible	Possible
0	1	Transmit Data	Possible	Possible	Possible
1	1	Inverted transmit data	Impossible	Impossible	Possible

Figure 33.105 shows the configuration of the shift register input/output path in loopback mode.

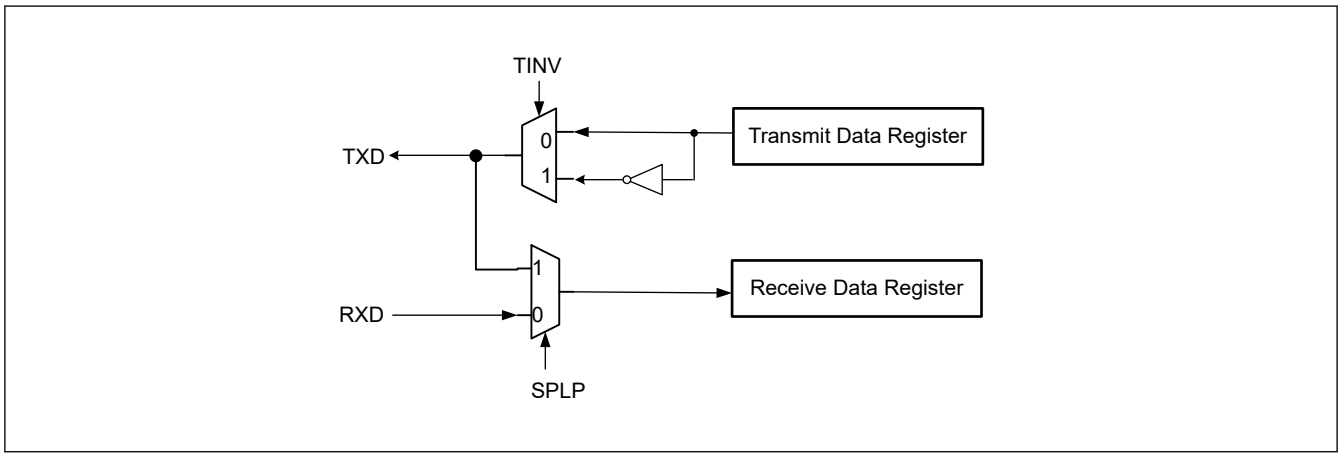


Figure 33.105 Shift register input/output configuration image in loopback mode

33.15 Half-Duplex communication Function

Do not use the half-duplex communication function in Simple-I2C mode, Simple-SPI mode, and Smart Card Interface mode.

In other communication modes, if the CCR1.SHARPS bit is set to 1, half-duplex communication using the TXD pin is possible. When half-duplex communication is used, transmission and reception must be performed exclusively. Transmission and reception settings (CCR0.TE = 1 and CCR0.RE = 1) is prohibited.

However, if half-duplex communication is performed as the master in clock synchronous mode, perform transmission/reception settings (CCR0.TE = 1 and CCR0.RE = 1) and perform dummy transmission. By dummy transmission (arbitrary transmission data is written to TDR), SCKn is output and reception is enabled. The dummy transmission data is discarded inside the module and is not actually transmitted.

During half-duplex communication, the only communication port terminal used is the TXD pin. Output when CCR0.TE = 1, input when CCR0.TE = 0.

33.16 Synchronizer Bypass Function

This module has a bus clock and the operation clock (PCLKSCIn). And these have each operating circuit. Therefore, there is a synchronization circuit for signal transfer between different clocks, and synchronization delay time is required for signal propagation between different clocks.

However, the synchronization circuit can be bypassed by the CCR3.BPEN bit only when the same clock is input to the bus clock and the operation clock. In this case, eliminates synchronization delay time and improves responsiveness. Figure 33.106 shows the image waveform of the bypass function.

This module also has a synchronization circuit between the communication clock (SCK) and the operation clock (PCLKSCIn), but this synchronization circuit can not be bypassed.

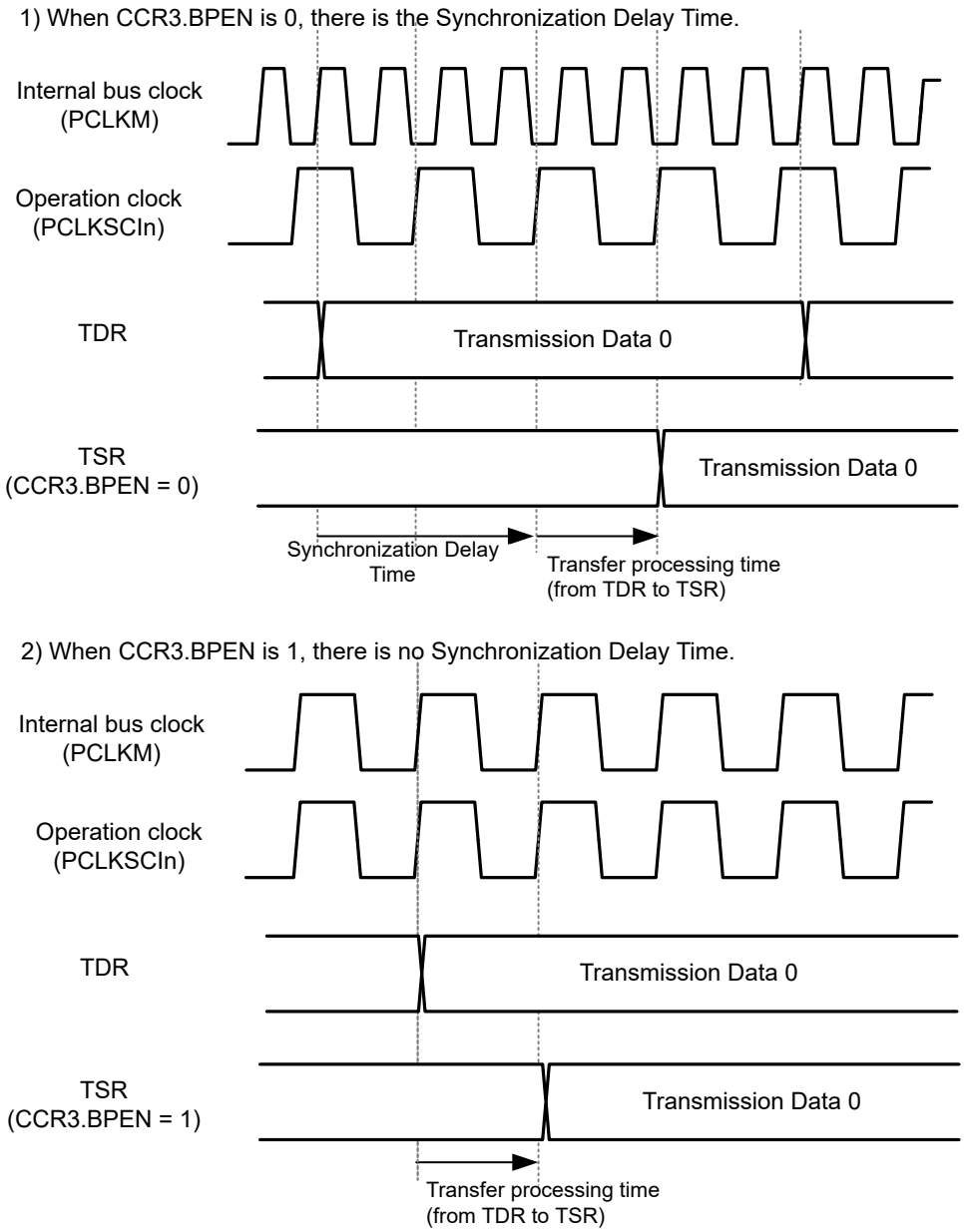


Figure 33.106 Image waveform of Synchronizer bypass function

33.17 Interrupt signal

Table 33.35 lists SCI interrupt signals.

When performing transmission and reception using DMAC, be sure to set DMAC first, and then set SCI. See section 15, DMA Controller (DMAC) for how to set DMAC.

Table 33.35 SCI interrupt sources (1 of 2)

Name	Interrupt sources	CPU0 GIC request	CPU1 GIC request	DMAC activation
SCIn_ERI	Error interrupt	Possible	Possible	Not possible
SCIn_RXI	Simple-I2C: Reception end interrupt Other mode: Receive data full interrupt	Possible	Possible	Possible*1

Table 33.35 SCI interrupt sources (2 of 2)

Name	Interrupt sources	CPU0 GIC request	CPU1 GIC request	DMAC activation
SCIn_TXI	Simple-I2C and Smart Card Interface: Transmit end interrupt Other mode: Transmit data empty interrupt	Possible	Possible	Possible*1
SCIn_TEI	Simple-I2C: Completion of generation of a start, restart, or stop condition (STI) Other mode: Transmit end interrupt	Possible	Possible	Not possible

Note: n = 0 to 5
Available interrupt sources change dependent on operation mode.
Note 1. DMAC activation is not available for n = 5.

33.17.1 Buffer Operations for TXI and RXI interrupts

The TXI and RXI interrupts have an interrupt buffer function. When the first interrupt request is generated during interrupt handling and the next interrupt request is generated (when the status flag of the Interrupt Controller Unit (ICU) is 1), the module does not output the interrupt request, and the SCI holds it internally. The interrupt that can be held is up to one.

33.17.2 Interrupt in Asynchronous Mode, Clock Synchronous Mode, and Simple-SPI Mode

A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled with the enable bits in CCR0 register.

(1) Non-FIFO selected

Table 33.36 lists interrupt sources in Asynchronous Mode, Clock Synchronous Mode, and Simple-SPI Mode with Non-FIFO selected.

If the CCR0.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR to the TSR. At the start of transmission, a TXI interrupt request can also be generated by using a single instruction to set the CCR0.TE and CCR0.TIE bit to 1 at the same time. A TXI interrupt request can activate the DMAC to handle data transfer.

A TXI interrupt request is not generated by setting the CCR0.TIE bit to 1 while the setting of the CCR0.TE bit is 1.*1

When the CCR0.TEIE bit is 1, the CSR.TEND flag becomes 1 and the TEI interrupt request is generated if the next data is not written to the TDR register by the timing to transmit the last bit of transmission data. In addition, the TEND flag holds 1 during the period from setting the CCR0.TE bit to 1 until writing transmit data to the TDR register, and if the TEIE bit is set to 1, a TEI interrupt request is generated.

Writing data to the TDR register clears the TEND flag and cancels the TEI interrupt request, but it takes time to cancel it.

If the CCR0.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR. An RXI interrupt request can activate the DMAC to handle data transfer.

Setting of any from among the ORER, FER, PER flags in the CSR to 1 while the CCR0.RIE bit is 1 leads to the generation of an ERI interrupt request.

An RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, PER) leads to discarding of the ERI interrupt request.

Note 1. To temporarily prohibit TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmit end interrupt, control prohibiting and permitting of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Table 33.36 SCI interrupt sources with Non-FIFO selected

Name	Interrupt source	Interrupt flag	Level/pulse output	Interrupt enable	DMAC activation	Priority
ERI	Receive error	ORER, FER, PER, DFER, DPER, MER, SYER (SYEREN = 1), PFER (PFEREN = 1), SBER (SBEREN = 1)	Level	RIE	Not possible	High
RXI	Receive data full	RDRF	Pulse	RIE	Possible	↑
	Address match	DCMF				
TXI	Transmit data empty	TDRE	Pulse	TIE	Possible	↑
	TE = 0 -> 1 detection					
TEI	Transmit end	TEND	Level	TEIE	Not possible	Low

(2) FIFO selected

Table 33.37 lists interrupt sources in Asynchronous Mode, Clock Synchronous Mode, and Simple-SPI Mode with FIFO selected.

If the CCR0.TIE bit is 1, a TXI interrupt request is generated when the stored number of data in transmit FIFO becomes the threshold value indicated in FCR.TTRG or below. A TXI interrupt request can also be generated by using a single instruction to set the CCR0.TE and CCR0.TIE bit to 1 at the same time. A TXI interrupt request is not generated by setting the CCR0.TIE bit to 1 while the setting of the CCR0.TE bit is 1. If CCR0.TEIE bit is 1, when the next data is not being written in transmit FIFO by the timing to which the last bit of the transmission data is sent, CSR.TEND flag will be 1 and TEI interrupt request is generated.

If the CCR0.RIE bit is 1, RXI interrupt request is generated when the stored number of data in receive FIFO becomes the threshold value indicated in FCR.RTRG or above. When FCR.RTRG is set to 0, RXI interrupt request is occurred if the quantity of data in receive FIFO is greater than or equal to 1.

If the CCR0.RIE bit is 1, when CSR.ORER flag is set to 1 or the data a framing error or a parity error generated is stored in receive FIFO, ERI interrupt request is generated.

When the number of data stored in a receive FIFO at this time is a threshold value or above, RXI interrupt request is also generated. The ERI interrupt request can be canceled by clearing all flags (CSR.ORER, FER, and PER).

Table 33.37 SCI interrupt sources with FIFO selected

Name	Interrupt source	Interrupt flag	Level/pulse output	Interrupt enable	DMAC activation	Priority
ERI	Receive error	ORER, FER, PER, DFER, DPER	Level	RIE	Not possible	High
		DR (FCR.DRES = 1)				
RXI	Receive FIFO data full	RDRF	Pulse	RIE	Possible	↑
	Receive data ready	DR (FCR.DRES = 0)				
	Address match	DCMF				
TXI	Transmit FIFO data empty	TDRE	Pulse	TIE	Possible	↑
	TE = 0 -> 1 detection					
TEI	Transmit end	TEND	Level	TEIE	Not possible	Low

33.17.3 Interrupt in Smart Card Interface Mode

Table 33.38 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 33.38 SCI interrupt sources in smart card interface mode

Name	Interrupt source	Interrupt flag	Level/pulse output	Interrupt enable	DMAC activation	Priority
ERI	Receive error or error signal detection	ORER, PER, ERS	Level	RIE	Not possible	High
RXI	Receive data full	RDRF	Pulse	RIE	Possible	↑
TXI	Transmit end	TEND	Pulse	TIE	Possible	Low
	When set TE = 0 -> 1					

Data transmission/reception using the DMAC is also possible in smart card interface mode. In transmission operation, when the TEND flag in CSR is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DMAC activation. The TEND flag is automatically set to 0 when the DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DMAC is not activated. Therefore, the SCI and DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the ERS flag in CSR is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the RIE bit in CCR0 to 1 to enable an ERI interrupt request to be generated at error occurrence.

In reception operation, an RXI interrupt request is generated when receive data is set to RDR. This RXI interrupt request activates the DMAC allowing transfer of receive data if the RXI request is specified beforehand as a source of DMAC activation. If an error occurs, the error flag is set. Therefore, the DMAC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

When transmitting/receiving data using the DMAC, be sure to make settings to enable the DMAC before making SCI settings. For DMAC settings, see section 15, DMA Controller (DMAC).

33.17.4 Interrupts in Simple-I2C Mode

Table 33.39 lists SCI interrupts in simple-I2C mode.

The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DMAC can also be used to handle transfer in simple I2C mode.

When the value of the IICINTM bit in ICR is 1, a RXI request will be generated on the falling edge of the SCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DMAC beforehand, the RXI request will activate the DMAC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DMAC beforehand, the TXI request will activate the DMAC to handle transfer of the transmit data. (In this case, ACK/NACK judging are impossible.)

When the value of the IICINTM bit in ICR is 0, SCI operates as follows.

RXI request (ACK detection) is generated if the input on the SDAn pin is at the low level on the rising edge of the SCLn signal for the ninth bit (acknowledge bit).

TXI request (NACK detection) is generated if the input on the SDAn pin is at the high level on the rising edge of the SCLn signal for the ninth bit (acknowledge bit).

If the RXI has been set up as an activating request for the DMAC beforehand, the RXI request will activate the DMAC to handle transfer of the received data. Also, if the DMAC is used for data transfer in reception or transmission, be sure to set up and enable the DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in ICR are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 33.39 SCI interrupt sources in Simple-I2C mode

Name	Interrupt source		Interrupt flag	Level/ pulse output	Interrupt enable	DMAC activation	Priority
	IICINTM = 1	IICINTM = 0					
RXI	Reception end	—	—	Pulse	RIE	Possible*1	High
	—	ACK detection	—			Possible	↑
TXI	Transmit end	—	—	Pulse	TIE	Possible*1	↑
	—	NACK detection	—			Possible	↑
STI	Completion of generation of a start, restart, or stop condition		IICSTIF	Level	TEIE	Not Possible	Low

Note 1. If the DMAC is being used, you can not confirm whether ACK or NACK.

33.18 Usage Notes

33.18.1 SCI Operations during Low Power Consumption State

(1) Transmission

Before using the power consumption reduction function to reduce SCI's power consumption, please do the following to confirming transmission end (CSR.TEND = 1):

- Set the output pin state after transmission operation is stopped by CCR1.SPB2DT, SPB2IO.
- Stop the transmission (CCR0.TIE = 0, TE = 0, TEIE = 0).

When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same transmission mode after cancellation of the low power consumption state, set the TE bit to 1, read CSR, and write data to TDR sequentially to start data transmission. To transmit data with a different transmission mode, initialize the SCI first.

To start transmission using the DMAC after cancellation of the low power consumption state, set the TE and TIE bit to 1. The TXI interrupt flag is set to 1 and transmission starts using the DMAC.

Figure 33.107 shows a sample flowchart for transition to the low power consumption state during transmission. Figure 33.108 and Figure 33.109 show the port pin states during transition to the low power consumption state.

(2) Reception

(a) When Address match function is non used as condition of resumption (wake-up)

Before specifying the module stop state or making a transition to the low power consumption state, stop the receive operations (CCR0.RE = 0). If transition is made during data reception, the data being received will be invalid.

Figure 33.110 shows a sample flowchart for reception to software standby mode during reception.

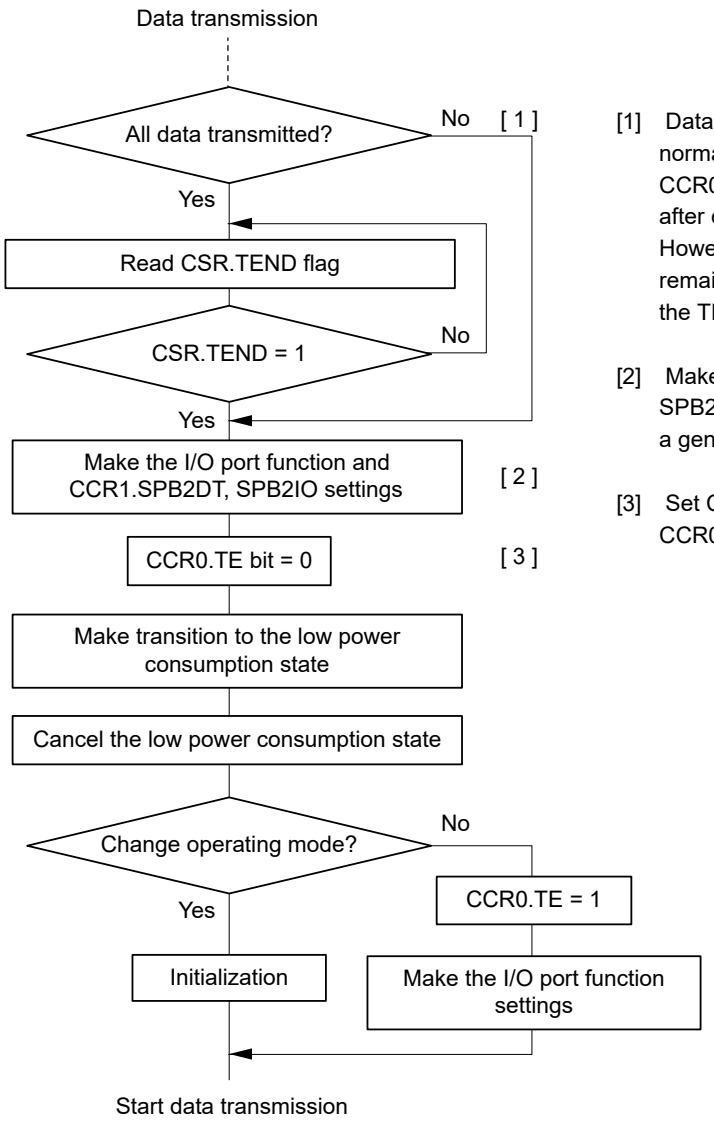
(b) When Address match function is used as condition of resumption (wake-up)

When using the power consumption reduction function to reduce SCI's power consumption, please do the following:

- Set the operation mode of after released.
- Set the compare data to CCR4.CMPD and set 1 to CCR0.DCME.
- Set the receive operation (CCR0.RE = 1).

When SCI transfers to the low power consumption mode if the reception data pin (RXD) is Low level, set CCR3.RXDESEL = 0. When setting it as CCR3.RXDESEL = 1, there is a possibility that a start bit (fall edge of RXDn pin) can not be detected at the time of low power consumption mode release.

Figure 33.111 shows a sample flowchart for reception to the low power consumption state during reception.



[1] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting the CCR0.TE to 1, reading CSR, and writing data to TDR after canceling the low power consumption state. However, if the DMAC has been activated, the data remaining in the DMAC will be transmitted when both the TE and TIE bits in CCR0 are set to 1.

[2] Make the I/O port function and CCR1.SPB2DT, SPB2IO settings to switch the TXDn pin to operate as a general I/O port.

[3] Set CCR0.TE bit to 0. Set also 0 to CCR0.TIE and CCR0.TEIE if you set these bits were 1.

Figure 33.107 Example of flowchart for transition to the low power consumption state during transmission

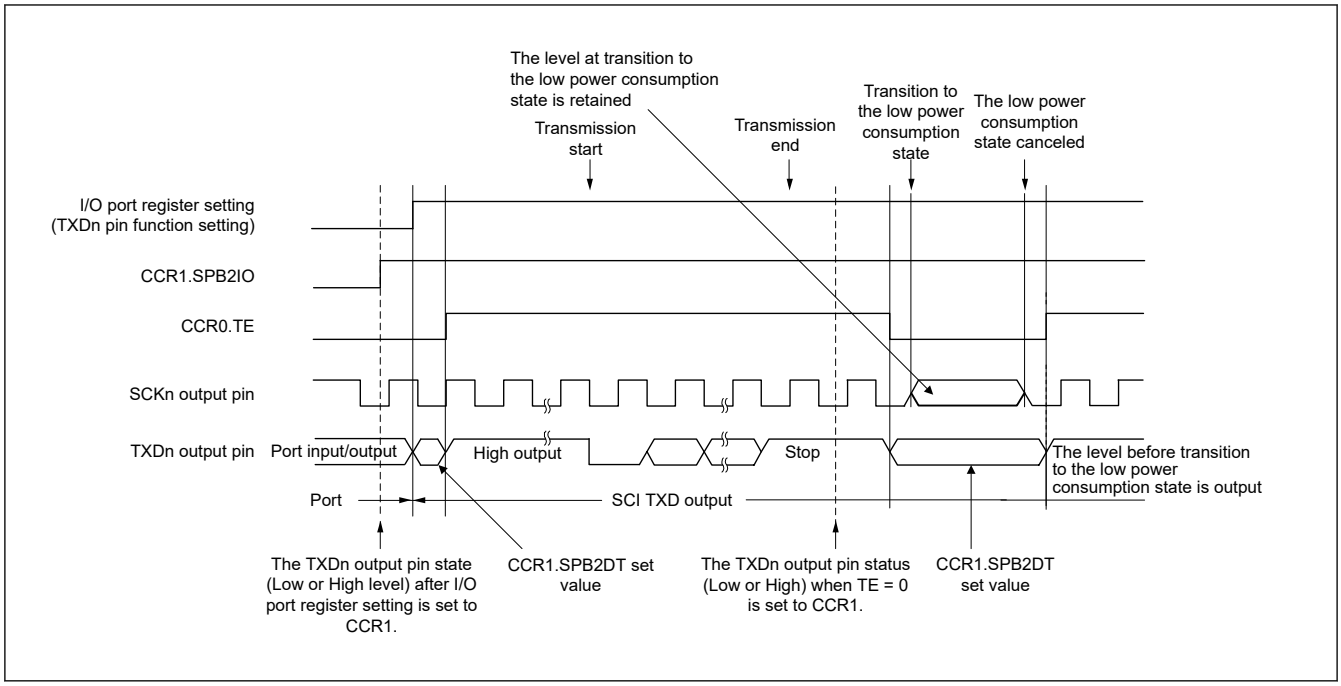


Figure 33.108 Port pin states during transition to the low power consumption state (internal clock, asynchronous transmission)

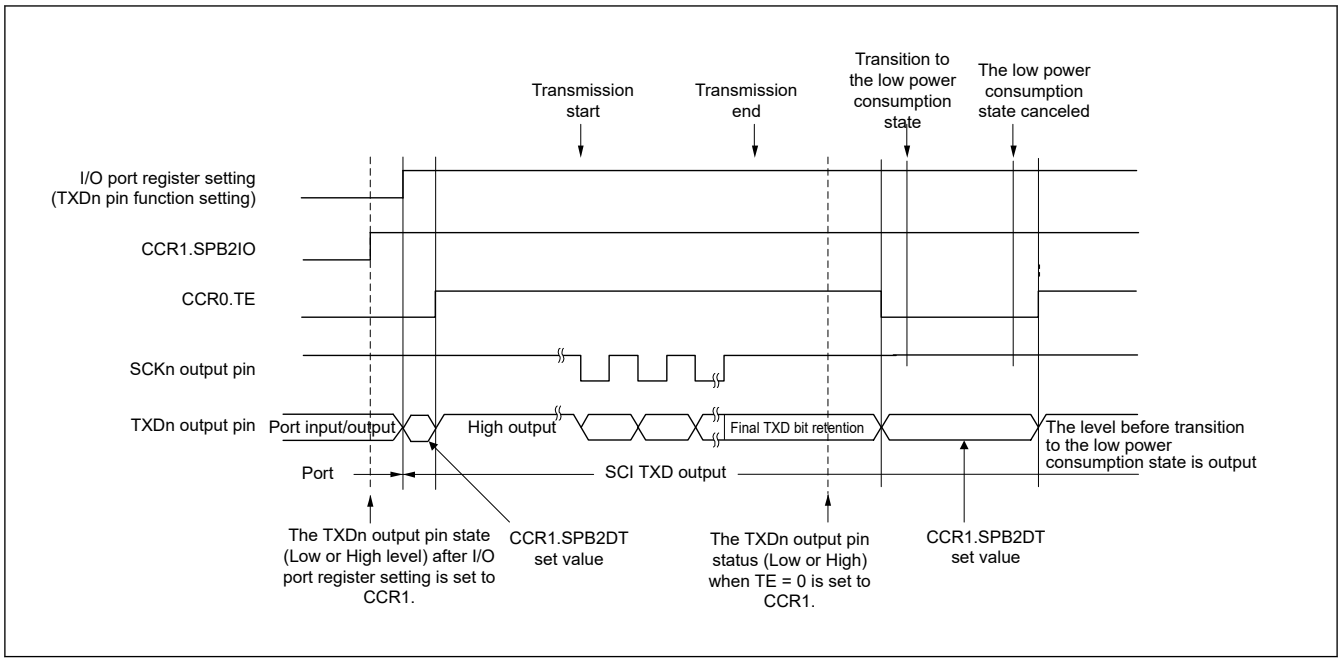


Figure 33.109 Port pin states during transition to the low power consumption state (internal clock, clock synchronous transmission)

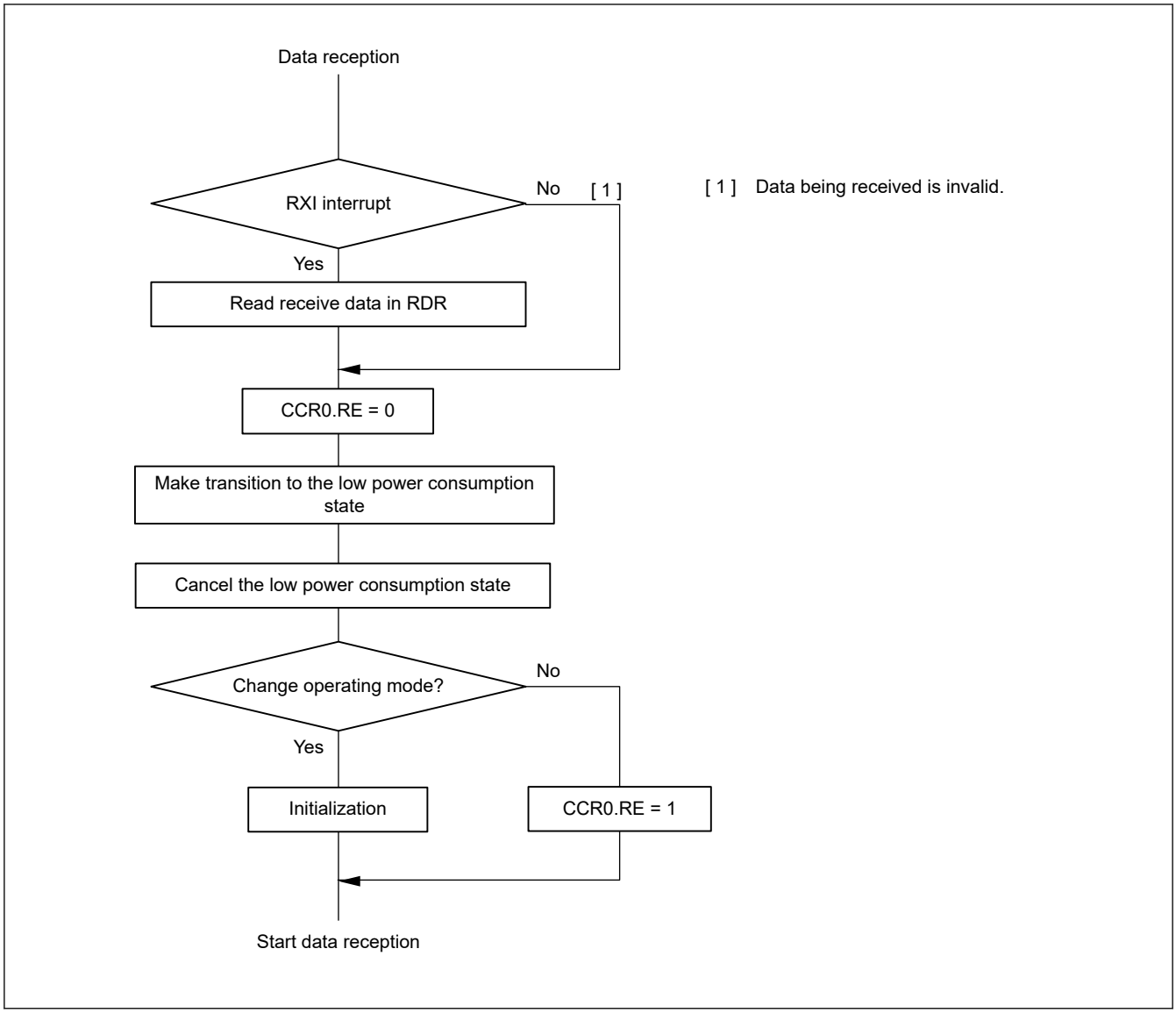


Figure 33.110 Example of flowchart for reception to the low power consumption state during reception

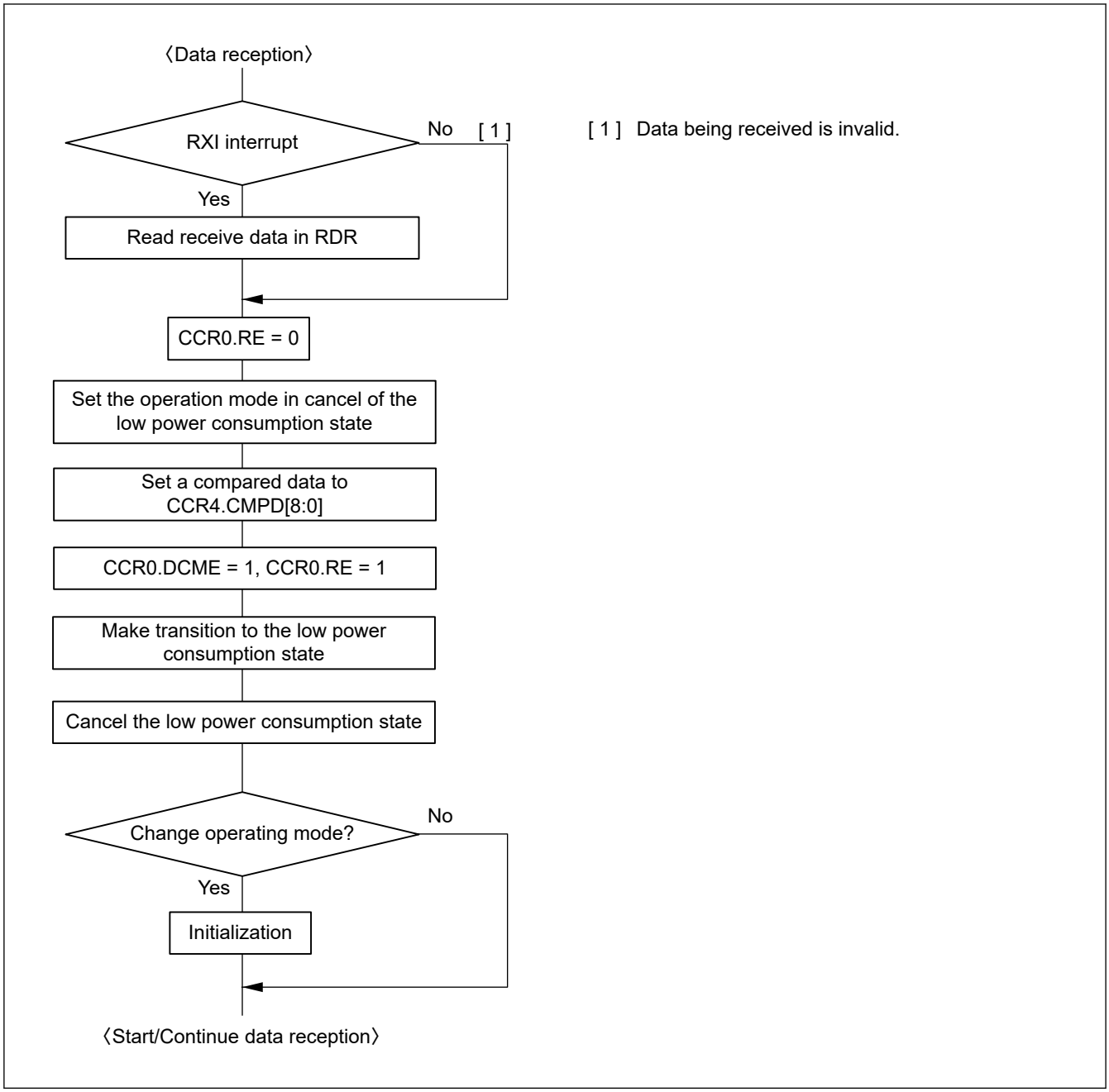


Figure 33.111 Example of flowchart for reception to the low power consumption state during reception with address match

33.18.2 Break Detection and Processing

(1) Non-FIFO selected

When a framing error is detected, a break can be detected by reading CSR.RXDMON bit value. In a break, the input from the RXDn pin becomes all 0s, and so the CSR.FER flag is set to 1 (framing error has occurred), and the CSR.PER flag may also be set to 1 (parity error has occurred). When the CCR3.RXDESEL bit is 0, the SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the CCR3.RXDESEL bit is 1, the SCI sets the CSR.FER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the CSR.FER flag is set to 0 at this time, the CSR.FER flag retains 0 during the break. When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit at the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

(2) FIFO selected

After a framing error is detected, when SCI detects that continuous receive data is 0 for 1 frame, the data stored into the receive FIFO (RDR) and reception stops. When a framing error is detected, a break can be detected by reading CSR.RXDMON bit value. After the RxD signal is in the mark state and it has finished the break, a stock of reception data to the receive FIFO (RDR) resumes.

33.18.3 Mark State and Production of Breaks

When CCR0.TE = 0 (serial transmission is disabled), the state of the TXDn pin can be set by CCR1.SPB2IO bit and CCR1.SPB2DT bit. Using this, it's possible to do a TXDn pin in the mark state and send a breakout. When you want to make a communication-line in the mark state (the state of 1) until the CCR0.TE bit is set to 1 (serial transmission is enabled), execute the following operations. First, set output level High by the CCR1.SPB2IO and CCR1.SPB2DT bits. Next, it's changed to a TXDn pin by I/O port function. On the other hand, if you want to send a break when sending data, set the CCR0.TE bit to 0 after setting the SPB2IO and SPB2DT bits in the CCR1 register to low level output. Setting the TE bit to 0 initializes the transmitter regardless of the current transmission status.

33.18.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission can be started by writing transmit-data to TDR even if CSR.ORER is 1. However, reception can not be started. Note also that the receive error flags cannot be set to 0 even if the CCR0.RE is set to 0 (serial reception is disabled).

33.18.5 Writing Data to TDR (Transmit FIFO)

(1) Non-FIFO selected

Data can be written to TDR anytime when CCR0.TE = 1. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. If you use DMAC, be sure to write transmit data to TDR in the TXI interrupt request handling routine.

(2) FIFO selected

Data can be written to transmit FIFO (TDR) when TE is 1. Check the number of writable data with the FTSR.T[4:0] bits.

33.18.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

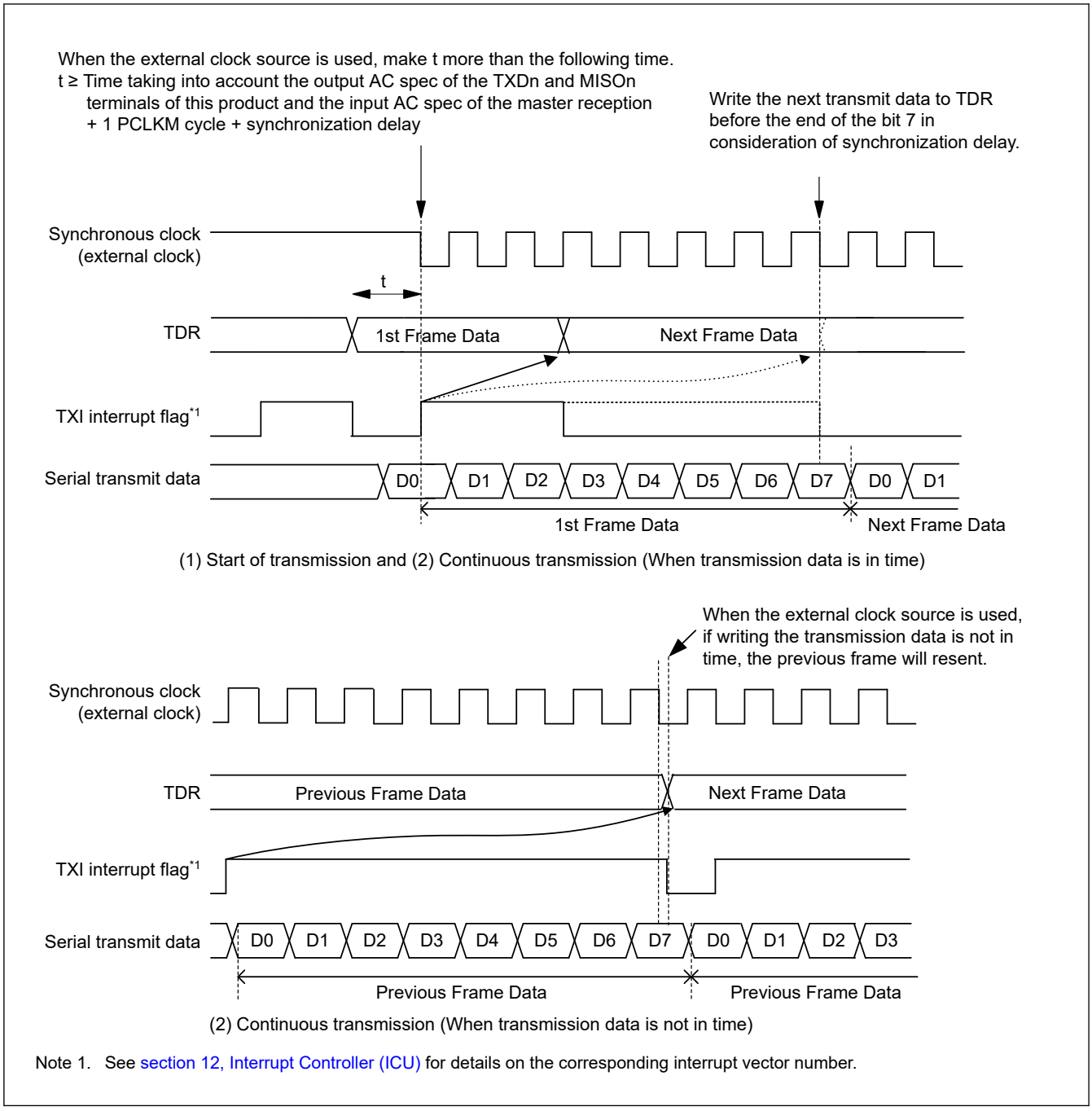
Update TDR by the CPU or DMAC and wait at least the following time until the start of the external clock input: (See Figure 33.112.)

Time taking into account the output AC specification of the TXDn and MISOn pins of this product and the input AC specification of the master reception + 1 PCLKM cycle + synchronization delay (2 to 3 PCLKSCIn).

(2) Continuous transmission

Write the next transmit data to TDR before the falling edge[*1] of the transmit clock for bit 7. Please write the transmit data to TDR in consideration of synchronization delay [2 to 3 PCLKSCIn]. If the transmit data can not be written in time, the previous frame data is resent. (See Figure 33.112.)

Note 1. When CCR3.CPOL = 1 and CCR3.CPHA = 0, or CCR3.CPOL = 0 and CCR3.CPHA = 1. In the case of CCR3.CPOL = 0 and CCR3.CPHA = 0, or CCR3.CPOL = 1 and CCR3.CPHA = 1, it is the rising edge.



Note 1. See section 12, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 33.112 Restrictions on use of external clock in clock synchronous transmission

33.18.7 Notes on Starting Transfer

At the point where transfer starts, when the interrupt status flag in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the CCR0.TE or CCR0.RE bit to 1).

For details on the interrupt status flag, see section 12, Interrupt Controller (ICU).

- Confirm that transfer has stopped (the setting of the CCR0.TE or CCR0.RE bits is 0).
- Set the corresponding interrupt enable bit (CCR0.TIE, CCR0.TEIE, or CCR0.RIE) to 0.
- Read the corresponding interrupt enable bit (CCR0.TIE, CCR0.TEIE, or CCR0.RIE bit) to check that it has actually become 0.
- Set the interrupt status flag in the interrupt controller to 0.

33.18.8 Limitations on Simple SPI Mode

(1) Master Mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set by the CCR3.CPHA and CPOL bits when CCR0.SSE = 1. This prevents the clock line from being placed in the high-impedance state when the CCR0.TE bit is set to 0 or unexpected edges from being generated on the clock line when the CCR0.TE bit is changed from 0 to 1.
  In a single-master mode, pull up or pull down the clock line is not necessary because the clock line does not become high-impedance state when CCR0.SSE = 0 and CCR0.TE = 0.
- In the case of the setting for clock delay (CCR3.CPHA = 0), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 33.113. If the TE and RE bits become 0 at this time before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Furthermore, an RXI interrupt may lead to the input signal on the SSn# pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master mode, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode fault error occurs as the current character is being transferred. And stopping supply of the clock signal to the connected slave. Remake the settings for the connected slave to avoid misaligned bits when transfer is restarted.

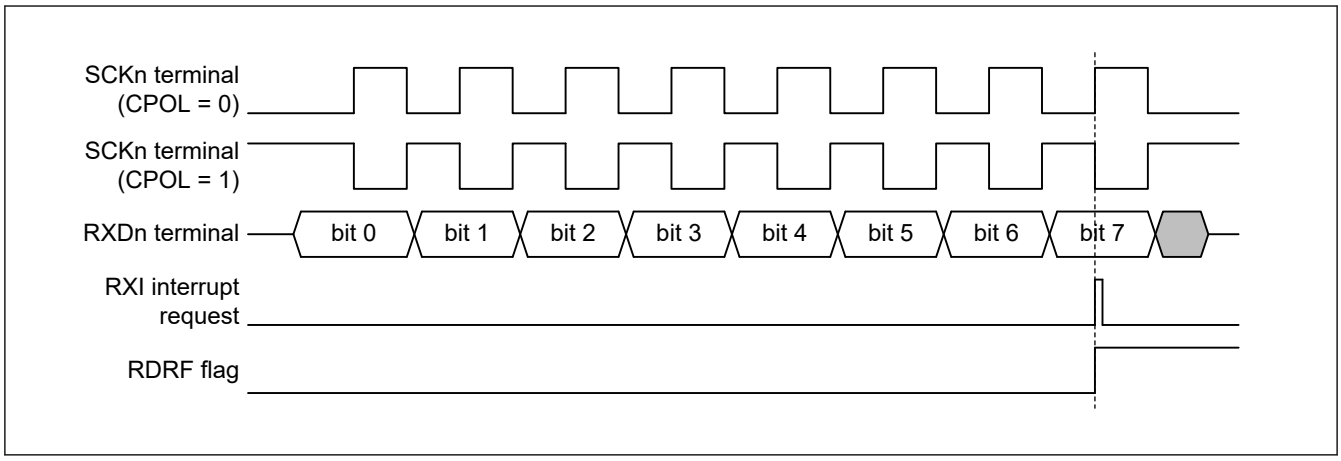


Figure 33.113 Timing of the RXI interrupt in simple SPI mode (with clock delay)

(2) Slave Mode

- It takes "1 PCLKM + synchronization delay time + data output delay time (AC spec)" from writing the transmit data to the TDR register until the data is output to the MSIOn pin. Take these into account when starting external clock input.
- Provide an external clock signal to the master the same as the data length for transfer.
- Secure the SS input setup time (AC spec) from the SS# low-level input to the start of external clock input.
- Control the input on the SSn# pin before the start and after the end of data transfer.
- When the level being input on the SSn# pin is to be changed from low to high while the current character is being transferred, the transmission and reception are stopped immediately. Set the TE and RE bits in the CCR0 to 0 and, after remaking the settings, restart transfer of the first byte.

33.18.9 Notes on Transmit Enable bit (TE)

In initial register value, when CCR0.TE = 0, and the terminal function is TXDn, the pin outputs high impedance.

So please make sure that the TXDn line will not be high impedance by the following one of ways.

1. The pull-up resistance is connected to the TXDn line.
2. Before CCR0.TE bit is 0, the function of the terminal is changed to general-purpose input port or output port. And after CCR0.TE bit is 1, the function of the pin is changed to TXDn.

3. In asynchronous mode, you can set CCR1 and decided level of TXDn pin during TE is 0.

In the simple SPI mode slave operation, the MSIOn pin operates in the same way as the above TXDn pin, so please deal with 1 or 2 in the same way. (3 can not be used.)

33.18.10 Notes on RS-485 Driver Control function

- RS-485 Driver control function is valid only in asynchronous mode.
- When RS-485 Driver control function is active (CCR3.DEN = 1), the TEND set timing/TEI output timing changes as follows. Wait for the TEI interrupt and set the TE bit in CCR0 to 0.
  When RS-485 Driver control function is inactive: When STOP bit output is completed.
  When RS-485 Driver control function is active: At the end of DE negation time.

33.18.11 Notes on Loopback function

The Loopback function is valid in asynchronous mode with internal clock, Manchester mode with internal clock, and clock synchronous mode with internal clock.

33.18.12 Notes Regarding Register Access When Operation Clock (PCLKSCIn) Is Slower Than Bus Clock (PCLKM)

If the operating clock (PCLKSCIn) is slower than the bus clock (PCLKM), the interval for writing access to the same register should be at least PCLKSCIn × 3 cycles + PCLKM × 3 cycles. Synchronization delay time and signal processing time are required to ensure that the setting value of the first write access is propagated to the subsequent logic circuit and the circuit is controlled correctly.

33.18.13 Notes on interrupting operation

If 0 is written to CCR0.RE during data reception and the reception operation is interrupted, there is a possibility of an invalid state, so please do not use the received data (RDR register stored value) and the flag value of each status register. To interrupt the reception operation, stop the interrupt or event link reception side and then write 0 to the CCR0.RE bit.

In addition, when 0 is written to each operation enable bit (CCR0.RE, CCR0.TE) and the operation is interrupted, the following time is required to safely stop the internal circuit. During this time, access to this register is prohibited.

PCLKSCIn × 3 cycles + PCLKM × 3 cycles

33.18.14 Notes on communication mode transition

When changing the communication mode setting bits CCR3.MOD[2:0] and switching the communication mode, set CCR0.RE = 0 and CCR0.TE = 0 to stop the communication operation. In order to transition the internal circuit state safely, the register access interval should be at least PCLKSCIn × 3 cycles + PCLKM × 3 cycles. During this time, write access to this register is prohibited. As an example, the following describes the transition from Asynchronous mode to Clock-synchronous mode.

1. Confirm stop of reception and transmission at asynchronous mode (CCR3.MOD[2:0] = 000b).
2. Write 0 to CCR0.TE and CCR0.RE, stop communication.
3. Wait "PCLKSCIn × 3 cycles + PCLKM × 3 cycles". During this time, access to this register is prohibited.
4. Write 010b to CCR3.MOD[2:0].
5. Wait "PCLKSCIn × 3 cycles + PCLKM × 3 cycles". During this time, access to this register is prohibited.

34. I²C Bus Interface (IIC)

34.1 Overview

The LSI has a two-channel I²C Bus Interface (IIC) module that conforms with and provides a subset of the NXP I²C bus (inter-integrated circuit bus) interface functions. Table 34.1 lists the IIC specifications, Figure 34.1 shows a block diagram, and Figure 34.2 shows an example of I/O pin connections to external circuits, with an I²C bus configuration. Table 34.2 lists the I/O pins.

Table 34.1 IIC specifications (1 of 2)

Parameter	Specifications
Number of channels	3 channels (ch0 to ch1 in NONSAFETY and ch2 in SAFETY)
Communications format	• I²C bus format or SMBus format. • Master or slave mode selectable. • Automatic securing of the setup times, hold times, and bus-free times for the transfer rate.
Transfer rate	Fast-mode supported, up to 400 kbps.
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%.
Issuing and detecting conditions	• Start, restart, and stop conditions are automatically generated. • Start conditions (including restart conditions) and stop conditions are detectable.
Slave address	• Configurable for up to three different slave addresses. • 7-bit and 10-bit address formats supported, including simultaneous use. • General call addresses, device ID addresses, and SMBus host addresses detectable.
Acknowledgment	• For transmission, automatic loading of the acknowledge bit. Transfer of the next transmit data can be automatically suspended on detection of a not-acknowledge bit. • For reception, automatic transmission of the acknowledge bit. If a wait between the 8th and 9th clock cycles is selected, software can control the value in the acknowledge field in response to the received value.
Wait function	During reception, the following wait periods are available by holding the SCL clock low: • Waiting between the 8th and 9th clock cycles. • Waiting between the 9th clock cycle and the 1st clock cycle of the next transfer.
SDA output delay function	Output timing of transmitted data, including the acknowledge bit, can be delayed.
Arbitration	• For multi-master operation: – SCL clock synchronization is possible when conflict occurs with the SCL signal from another master. – When issuing the start condition creates conflict on the bus, loss of arbitration is detected by testing for mismatching between the internal signal for the SDA line and the level on the SDA line. – In master operation, loss of arbitration is detected by testing for mismatching between the signal on the SDA line and the internal signal for the SDA line. • Loss of arbitration because the start condition occurs while the bus is busy is detectable, to prevent the issuing of double start conditions. • Loss of arbitration is detectable on transfer of a not-acknowledge bit because the internal signal for the SDA line and the level on the SDA line do not match. • Loss of arbitration because mismatching of internal and line levels for data is detectable in slave transmission.
Timeout function	Internal detection of long-interval stops of the SCL clock.
Noise cancellation	• Digital noise filters for both the SCL and SDA signals. • Programmable window for noise cancellation by the filters.
Interrupt sources	• Transfer error or event generation: arbitration detection, NACK, timeout, start or restart condition, or stop condition. • Receive data full, including matching with a slave address. • Transmit data empty, including matching with a slave address. • Transmit end.
Module-stop function	Module-stop state can be set to reduce power consumption.

Table 34.1 IIC specifications (2 of 2)

Parameter	Specifications
IIC operating modes	• Master transmit • Master receive • Slave transmit • Slave receive

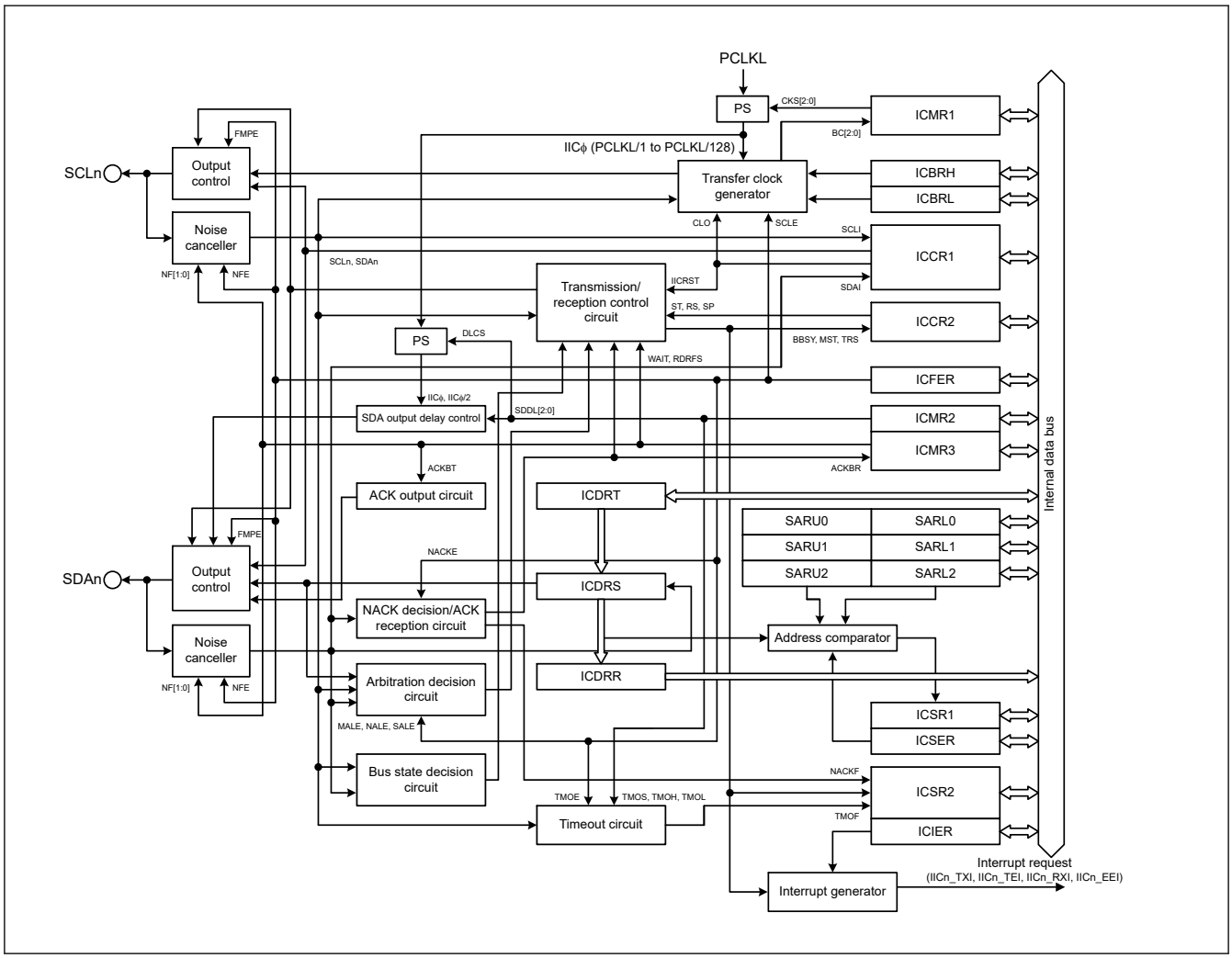


Figure 34.1 IIC block diagram

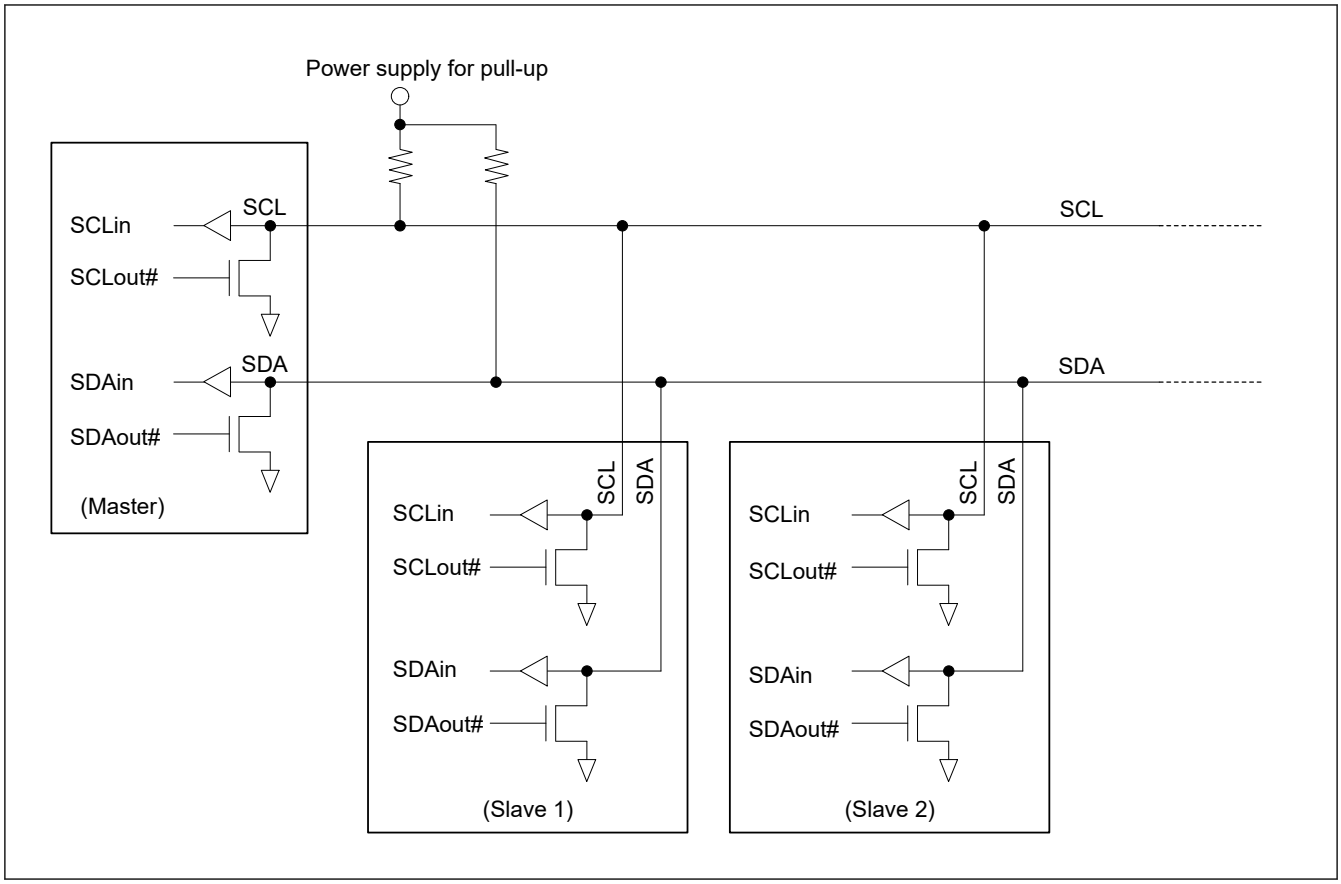


Figure 34.2 I/O pin connection to an external circuit (I^2C bus configuration example)

The input level of the signals for IIC is CMOS. Change to TTL is not supported.

Table 34.2 IIC I/O pins

Channel	Pin name*1	I/O	Function
IIC0	IIC_SCL0	I/O	IIC0 serial clock I/O pin
	IIC_SDA0	I/O	IIC0 serial data I/O pin
IIC1	IIC_SCL1	I/O	IIC1 serial clock I/O pin
	IIC_SDA1	I/O	IIC1 serial data I/O pin
IIC2	IIC_SCL2	I/O	IIC2 serial clock I/O pin
	IIC_SDA2	I/O	IIC2 serial data I/O pin

Note 1. For convention, pin names within this chapter are written without "IIC_" notation except this table.

Table 34.3 IIC interrupt sources

Name	Interrupt sources
IICn_EEI	Transfer error or event generation
IICn_RXI	Receive data full
IICn_TXI	Transmit data empty
IICn_TEI	Transmit end

Note: n = 0 to 2

34.2 Register Map

Table 34.4 IIC register map

Address	Register symbol	Register name	Write protection
0x8008_8000 + 0x0400 × n (n = 0, 1) 0x8100_8000 (n = 2)	ICCR1	I2C Bus Control Register 1	—
0x8008_8001 + 0x0400 × n (n = 0, 1) 0x8100_8001 (n = 2)	ICCR2	I2C Bus Control Register 2	—
0x8008_8002 + 0x0400 × n (n = 0, 1) 0x8100_8002 (n = 2)	ICMR1	I2C Bus Mode Register 1	—
0x8008_8003 + 0x0400 × n (n = 0, 1) 0x8100_8003 (n = 2)	ICMR2	I2C Bus Mode Register 2	—
0x8008_8004 + 0x0400 × n (n = 0, 1) 0x8100_8004 (n = 2)	ICMR3	I2C Bus Mode Register 3	—
0x8008_8005 + 0x0400 × n (n = 0, 1) 0x8100_8005 (n = 2)	ICFER	I2C Bus Function Enable Register	—
0x8008_8006 + 0x0400 × n (n = 0, 1) 0x8100_8006 (n = 2)	ICSER	I2C Bus Status Enable Register	—
0x8008_8007 + 0x0400 × n (n = 0, 1) 0x8100_8007 (n = 2)	ICIER	I2C Bus Interrupt Enable Register	—
0x8008_8008 + 0x0400 × n (n = 0, 1) 0x8100_8008 (n = 2)	ICSR1	I2C Bus Status Register 1	—
0x8008_8009 + 0x0400 × n (n = 0, 1) 0x8100_8009 (n = 2)	ICSR2	I2C Bus Status Register 2	—
0x8008_800A + 0x0400 × n + 0x02 × y (n = 0, 1) 0x8100_800A + 0x02 × y (n = 2)	SARLy	Slave Address Register L y (y = 0 to 2)	—
0x8008_8009 + 0x0400 × n (n = 0, 1) 0x8100_8009 (n = 2)	SARUy	Slave Address Register U y (y = 0 to 2)	—
0x8008_8010 + 0x0400 × n (n = 0, 1) 0x8100_8010 (n = 2)	ICBRL	I2C Bus Bit Rate Low-Level Register	—
0x8008_8011 + 0x0400 × n (n = 0, 1) 0x8100_8011 (n = 2)	ICBRH	I2C Bus Bit Rate High-Level Register	—
0x8008_8012 + 0x0400 × n (n = 0, 1) 0x8100_8012 (n = 2)	ICDRT	I2C Bus Transmit Data Register	—
0x8008_8013 + 0x0400 × n (n = 0, 1) 0x8100_8013 (n = 2)	ICDRR	I2C Bus Receive Data Register	—
—	ICDRS	I2C Bus Shift Register	—

Table 34.5 IIC related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0 (NONSAFETY)	—	MSTPCRB.MSTPCRB00	SLVACCCTL1.IIC0_SL
1 (NONSAFETY)	—	MSTPCRB.MSTPCRB01	SLVACCCTL1.IIC1_SL
2 (SAFETY)	—	MSTPCRG.MSTPCRG01	SLVACCCTL2.IIC2_SL

34.3 Register Descriptions

34.3.1 ICCR1 : I2C Bus Control Register 1

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Function	R/W
0	SDAI	SDA Line Monitor 0: SDAn line is low 1: SDAn line is high	R
1	SCLI	SCL Line Monitor 0: SCLn line is low 1: SCLn line is high	R
2	SDAO	SDA Output Control/Monitor 0: Read: IIC drives SDAn pin low Write: IIC drives SDAn pin low 1: Read: IIC releases SDAn pin Write: IIC releases SDAn pin	R/W
3	SCLO	SCL Output Control/Monitor Use an external pull-up resistor to drive the signal high. 0: Read: IIC drives SCLn pin low Write: IIC drives SCLn pin low 1: Read: IIC releases SCLn pin Write: IIC releases SCLn pin	R/W
4	SOWP	SCLO/SDAO Write Protect 0: Write enable SCLO and SDAO bits 1: Write protect SCLO and SDAO bits	W
5	CLO	Extra SCL Clock Cycle Output This bit clears automatically after one clock cycle is output. 0: Do not output extra SCL clock cycle (default) 1: Output extra SCL clock cycle	R/W
6	IICRST	IIC-Bus Interface Internal Reset This setting clears the bit counter and the SCLn/SDAn output latch. 0: Release IIC reset or internal reset 1: Initiate IIC reset or internal reset	R/W
7	ICE	IIC-Bus Interface Enable Used in combination with the IICRST bit to select either IIC or internal reset. 0: Disable (SCLn and SDAn pins in inactive state) 1: Enable (SCLn and SDAn pins in active state)	R/W

SDAO bit (SDA Output Control/Monitor) and SCLO bit (SCL Output Control/Monitor)

The SDAO and SCLO bits directly control the SDAn and SCLn signals output from the IIC.

When writing to these bits, also write 0 to the SOWP bit. Setting these bits results in input to the IIC by the input buffer.

When slave mode is selected, a start condition might be detected and the bus might be released, depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, or during transmission or reception.

Operation after rewriting under the specified conditions is not guaranteed. When reading these bits, the state of signals output from the IIC can be read.

CLO bit (Extra SCL Clock Cycle Output)

The CLO bit allows the output of an extra SCL clock cycle for debugging or error processing. Normally, set this bit to 0.

Setting this bit to 1 in a normal communication state causes a communication error. For details on this function, see section 34.12.2. Extra SCL Clock Cycle Output Function.

IICRST bit (IIC-Bus Interface Internal Reset)

The IICRST bit initiates an internal state reset of the IIC. Setting this bit to 1 initiates an IIC reset or internal reset.

Whether an IIC reset or internal reset is initiated is determined by setting this bit in combination with the ICE bit.

Table 34.6 lists the IIC resets.

The IIC reset initializes all registers except ICCR1.ICE and ICCR1.IICRST bits, and internal states of the IIC. In addition to the internal states of the IIC, the internal reset initializes the following:

- Bit counter (ICMR1.BC[2:0] bits)
- I^2C Bus Shift Register (ICDRS)
- I^2C Bus Status Registers (ICSR1 and ICSR2)
- SDAO and SCLO Output Control/Monitor (ICCR1.SCLO and ICCR1.SDAO bits)
- I^2C Bus Control Register 2 (except ICCR2.BBSY bit)

For the reset conditions of each register, see section 34.15. Resets, Registers, and Function States When Issuing Each Condition.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the IIC without initializing the port settings and the control and setting registers of the IIC when the bus or IIC hangs up because of a communication error. If the IIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCLn pin and SDAn pin at high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up that occurs during communication with the master device in slave mode, the slave and master devices might enter different states, because the bit counter information differs. For this reason, do not initiate an internal reset in slave mode. Initiate recovery processing from the master device. If an internal reset is necessary because the IIC hangs with the SCLn line in a low level output state in slave mode, initiate an internal reset, and then issue a restart condition from the master device, or issue a stop condition and resume communication from the start condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start or restart condition from the master device, synchronization is lost because the master and slave devices operate asynchronously.

Table 34.6 IIC resets

IICRST	ICE	State	Specifications
1	0	IIC reset	Resets all registers except ICCR1.ICE and ICCR1.IICRST bits, and the internal states of the IIC
	1	Internal reset	Resets the following: • ICMR1.BC[2:0] bits • ICCR1, ICSR2, ICDRS registers • SDAO and SCLO Output Control/Monitor (ICCR1.SCLO and ICCR1.SDAO bits) • I^2C Bus Control Register 2 (except ICCR2.BBSY bit) • Internal states of the IIC.

ICE bit (IIC-Bus Interface Enable)

The ICE bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 34.6 for the reset types.

Set the ICE bit to 1 when using the IIC. The SCLn and SDAn pins are placed in the active state when the ICE bit is set to 1. Set the ICE bit to 0 when the IIC is not used. The SCLn and SDAn pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCLn or SDAn pin to the IIC when setting up the pin function control. Slave address comparison is performed if the pins are assigned to the IIC.

34.3.2 ICCR2 : I²C Bus Control Register 2

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ST	Start Condition Issuance Request 0: Do not issue a start condition request 1: Issue a start condition request	R/W
2	RS	Restart Condition Issuance Request 0: Do not issue a restart condition request 1: Issue a restart condition request	R/W
3	SP	Stop Condition Issuance Request 0: Do not issue a stop condition request 1: Issue a stop condition request	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TRS	Transmit/Receive Mode 0: Receive mode 1: Transmit mode	R/W*1
6	MST	Master/Slave Mode 0: Slave mode 1: Master mode	R/W*1
7	BBSY	Bus Busy Detection Flag 0: I²C bus released (bus free state) 1: I²C bus occupied (bus busy state)	R

Note 1. The MST and TRS bits can be written to when the ICMR1.MTWP bit is set to 1.

ST bit (Start Condition Issuance Request)

The ST bit requests transition to master mode and triggers a start condition.

When this bit is set to 1, a start condition is issued when the BBSY flag is set to 0 (bus free state). For details on issuing a start condition, see section 34.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit.

[Clearing conditions]

- When 0 is written to the ST bit.
- When a start condition is issued (a start condition is detected).
- When the AL (arbitration-lost) flag in ICSR2 is set to 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free state). Arbitration might be lost if the ST bit is set to 1 (start condition request) when the BBSY flag is 1 (bus busy state).

RS bit (Restart Condition Issuance Request)

The RS bit requests that a restart condition be issued in master mode.

When this bit is set to 1 to request a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on issuing a restart condition, see section 34.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the RS bit.
- When a restart condition is issued (a start condition is detected).
- When the AL (arbitration-lost) flag in ICSR2 is set to 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Do not set the RS bit to 1 while issuing a stop condition.

Note: If 1 (restart condition request) is written to the RS bit in slave mode, the restart condition is not issued, but the RS bit remains set to 1. If the operating mode changes to master mode without the bit being cleared, a restart condition might be issued.

SP bit (Stop Condition Issuance Request)

The SP bit requests that a stop condition be issued in master mode.

When this bit is set to 1, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see section 34.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the SP bit.
- When a stop condition is issued (a stop condition is detected).
- When the AL (arbitration-lost) flag in ICSR2 is set to 1.
- When a start condition and a restart condition are detected.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Writing to the SP bit is not possible while the BBSY flag is 0 (bus free state).

Note: Do not set the SP bit to 1 while a restart condition is being issued.

TRS bit (Transmit/Receive Mode)

The TRS bit indicates transmit or receive mode.

The IIC is in receive mode when the TRS bit is 0 and in transmit mode when the bit is 1. The combination of the TRS bit and the MST bit indicates the operating mode of the IIC.

The value of the TRS bit automatically changes to 1 for transmit mode or 0 for receive mode when a start condition is issued or detected and the R/W# bit is set. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1).
- When a restart condition is issued normally because of a restart condition request (when a restart condition is detected with the RS bit set to 1).
- When the R/W# bit added to the slave address is set to 0 in master mode.
- When the address received in slave mode matches the address enabled in ICSER, with the R/W# bit set to 1.
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected.
- When the AL (arbitration-lost) flag in ICSR2 is set to 1.
- In master mode, on the reception of a slave address to which a R/W# bit with the value 1 is appended.
- In slave mode, on a match between the received address and the address enabled in ICSER when the value of the received R/W# bit is 0, including when the received address is the general call address.
- In slave mode, when a restart condition is detected (a start condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0).
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

MST bit (Master/Slave Mode)

The MST bit indicates master or slave mode. The IIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. The combination of the MST bit and the TRS bit indicates the operating mode of the IIC.

The value of the MST bit automatically changes to 1 for master mode or 0 for slave mode when a start condition is issued, or when a stop condition is issued or detected. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1).
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected.
- When the AL (arbitration-lost) flag in ICSR2 is set to 1.
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

BBSY bit (Bus Busy Detection Flag)

The BBSY flag indicates whether the I^2C bus is occupied (bus busy state) or released (bus free state).

The flag is set to 1 when the SDAn line changes from high to low when the SCLn line is high, assuming that a start condition was issued.

The flag is set to 0 when the SDAn line changes from low to high with the SCLn line high, if the bus free time (ICBRL setting) start condition is not detected, assuming that a stop condition was issued.

[Setting condition]

- When a start condition is detected.

[Clearing conditions]

- When the bus free time (ICBRL setting) start condition is not detected after detecting a stop condition.
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (IIC reset).

34.3.3 ICMR1 : I²C Bus Mode Register 1

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	BC[2:0]	Bit Counter 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
3	BCWP	BC Write Protect 0: Write enable BC[2:0] bits 1: Write protect BC[2:0] bits	W*1
6:4	CKS[2:0]	Internal Reference Clock Select Select the internal reference clock source (IICΦ) for the IIC. 0 0 0: PCLKL clock 0 0 1: PCLKL/2 clock 0 1 0: PCLKL/4 clock 0 1 1: PCLKL/8 clock 1 0 0: PCLKL/16 clock 1 0 1: PCLKL/32 clock 1 1 0: PCLKL/64 clock 1 1 1: PCLKL/128 clock	R/W
7	MTWP	MST/TRS Write Protect 0: Write protect MST and TRS bits in ICCR2 1: Write enable MST and TRS bits in ICCR2	R/W

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] bits (Bit Counter)

The BC[2:0] bits function as a counter that indicates the number of bits remaining to be transferred on detection of a rising edge on the SCLn line. Although BC[2:0] are read/write bits, it is not required to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one, for an additional acknowledge bit, between transferred frames when the SCLn line is at a low level.

The value in the BC[2:0] bits return to 000b at the end of a data transfer, including the acknowledge bit, or when a start or restart condition is detected.

34.3.4 ICMR2 : I²C Bus Mode Register 2

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
0	TMOS	Timeout Detection Time Select 0: Select long mode 1: Select short mode	R/W
1	TMOL	Timeout L Count Control 0: Disable count while SCLn line is low 1: Enable count while SCLn line is low	R/W
2	TMOH	Timeout H Count Control 0: Disable count while SCLn line is high 1: Enable count while SCLn line is high	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	SDDL[2:0]	SDA Output Delay Counter 0 0 0: No output delay 0 0 1: 1 IICΦ cycle (When ICMR2.DLCS = 0 (IICΦ)) 1 or 2 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 0 1 0: 2 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 3 or 4 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 0 1 1: 3 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 5 or 6 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 1 0 0: 4 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 7 or 8 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 1 0 1: 5 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 9 or 10 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 1 1 0: 6 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 11 or 12 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 1 1 1: 7 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 13 or 14 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2))	R/W
7	DLCS	SDA Output Delay Clock Source Select 0: Select internal reference clock (IICΦ) as clock source for SDA output delay counter 1: Select internal reference clock divided by 2 (IICΦ/2) as clock source for SDA output delay counter*1	R/W

Note 1. The setting DLCS = 1 (IICΦ/2) is only valid when SCL is low. When SCL is high, the DLCS = 1 setting becomes invalid and the clock source becomes the internal reference clock (IICΦ).

TMOS bit (Timeout Detection Time Select)

The TMOS bit selects long or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE = 1).

When this bit is set to 0, long mode is selected. When it is set to 1, short mode is selected.

In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit counter. While the SCLn line is in the state that enables this counter as specified in the TMOH and TMOL bits, the counter counts up in synchronization with the internal reference clock (IICΦ) as a count source.

For details on this function, see section 34.12.1. Timeout Function.

TMOL bit (Timeout L Count Control)

The TMOLbit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held low and the timeout function is enabled (ICFER.TMOE = 1).

TMOH bit (Timeout H Count Control)

The TMOH bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held high and the timeout function is enabled (ICFER.TMOE = 1).

SDDL[2:0] bits (SDA Output Delay Counter)

The SDDL[2:0] bits can be used to delay the SDA output. This counter works with the clock source selected in the DLCS bit. This setting can be used for all types of SDA output, including the transmission of the acknowledge bit.

Set the SDA output delay to meet the I^2C bus standard for the data enable time/acknowledge enable time,*1 or the SMBus standard, within [data hold time (300 ns or more + the SCL-clock low-level period) - the data setup time (250 ns)]. If a

value outside the standard is set, communication between the devices might malfunction or falsely indicate a start or stop condition, depending on the bus state.

For details on this function, see section 34.6. SDA Output Delay Function.

Note 1. Data enable time/acknowledge enable time

3,450 ns for up to 100 kbps: Standard-mode (Sm)

900 ns for up to 400 kbps: Fast-mode (Fm)

34.3.5 ICMR3 : I2C Bus Mode Register 3

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	NF[1:0]	Noise Filter Stage Select 0 0: Filter out noise of up to 1 IICΦ cycle (single-stage filter) 0 1: Filter out noise of up to 2 IICΦ cycles (2-stage filter) 1 0: Filter out noise of up to 3 IICΦ cycles (3-stage filter) 1 1: Filter out noise of up to 4 IICΦ cycles (4-stage filter)	R/W
2	ACKBR	Receive Acknowledge 0: 0 received as the acknowledge bit (ACK reception) 1: 1 received as the acknowledge bit (NACK reception)	R
3	ACKBT	Transmit Acknowledge 0: Send 0 as the acknowledge bit (ACK transmission) 1: Send 1 as the acknowledge bit (NACK transmission)	R/W*1
4	ACKWP	ACKBT Write Protect 0: Write protect ACKBT bit 1: Write enable ACKBT bit	R/W*1
5	RDRFS	RDRF Flag Set Timing Select Low-hold is released by writing to ACKBT. 0: Set the RDRF flag on the rising edge of the 9th SCL clock cycle. The SCLn line is not held low on the falling edge of the 8th clock cycle. 1: Set the RDRF flag on the rising edge of the 8th SCL clock cycle. The SCLn line is held low on the falling edge of the 8th clock cycle.	R/W*2
6	WAIT	WAIT Low-hold is released by reading ICDRR. 0: No wait. SCLn is not held low during the period between 9th clock cycle and 1st clock cycle. 1: Wait. SCLn is held low during the period between 9th clock cycle and 1st clock cycle.	R/W*2
7	SMBS	SMBus/IIC-Bus Select 0: Select I2C bus 1: Select SMBus	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If the application writes 1 to both the ACKWP and ACKBT bits at the same time, the ACKBT bit is not set to 1.

Note 2. The WAIT and RDRFS bits are only valid in receive mode (invalid in transmit mode).

NF[1:0] bits (Noise Filter Stage Select)

The NF[1:0] bits select the number of stages in the digital noise filter. For details on this function, see section 34.7. Digital Noise Filter Circuits.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCLn line high-level period or low-level period. If the noise range is set to a value of [SCL clock width: high-level period or low-level period,

whichever is shorter] - [1.5 internal reference clock (IICΦ) cycles] or more, the SCL clock is regarded as noise by the noise filter function of the IIC, which might prevent the IIC from operating normally.

ACKBR bit (Receive Acknowledge)

The ACKBR bit stores the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1.
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKBT bit (Transmit Acknowledge)

The ACKBT bit sets the acknowledge bit to be sent in receive mode.

[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1.

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1.
- When stop condition issuance is detected with the SP bit in ICCR2 set to 1.
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKWP bit (ACKBT Write Protect)

The ACKWP bit controls write enabling of the ACKBT bit.

RDRFS bit (RDRF Flag Set Timing Select)

The RDRFS bit selects the RDRF flag set timing in receive mode and also selects whether to hold the SCLn line low on the falling edge of the 8th SCL clock cycle.

When the RDRFS bit is 0, the SCLn line is not held low on the falling edge of the 8th SCL clock cycle, and the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 on the rising edge of the 8th SCL clock cycle, and the SCLn line is held low on the falling edge of the 8th SCL clock cycle. The low-hold of the SCLn line is released by a write to the ACKBT bit.

After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1), based on the receive data.

WAIT bit (WAIT)

The WAIT bit controls whether to forcefully hold the period between the 9th SCL clock cycle and the 1st SCL clock cycle low until the receive data buffer (ICDRR) is completely read each time a single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without holding the period between the 9th and the 1st SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCLn line is held low from the falling edge of the 9th clock cycle until the ICDRR value is read each time a single-byte data is received. This enables the receive operation in byte units.

Note: When the value of the WAIT bit is to be read, be sure to first read the ICDRR.

SMBS bit (SMBus/IIC-Bus Select)

Setting the SMBS bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

34.3.6 ICFER : I²C Bus Function Enable Register

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Function	R/W
0	TMOE	Timeout Function Enable 0: Disable 1: Enable	R/W
1	MALE	Master Arbitration-Lost Detection Enable 0: Disable the arbitration-lost detection function and disable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost 1: Enable the arbitration-lost detection function and enable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost	R/W
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
3	SALE	Slave Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
4	NACKE	NACK Reception Transfer Suspension Enable 0: Do not suspend transfer operation during NACK reception (disable transfer suspension) 1: Suspend transfer operation during NACK reception (enable transfer suspension)	R/W
5	NFE	Digital Noise Filter Circuit Enable 0: Do not use the digital noise filter circuit 1: Use the digital noise filter circuit	R/W
6	SCLE	SCL Synchronous Circuit Enable 0: Do not use the SCL synchronous circuit 1: Use the SCL synchronous circuit	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

TMOE bit (Timeout Function Enable)

The TMOE bit enables or disables the timeout function. For details on this function, see section 34.12.1. Timeout Function.

MALE bit (Master Arbitration-Lost Detection Enable)

The MALE bit specifies whether to use the arbitration-lost detection function in master mode. For normal operation, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

The NALE bit specifies whether to cause loss of arbitration when ACK is detected during the transmission of NACK in receive mode, for example, when slaves with the same address exist on the bus, or when two or more masters select the same slave device simultaneously with a different number of receive bytes.

SALE bit (Slave Arbitration-Lost Detection Enable)

The SALE bit specifies whether to cause loss of arbitration when a value different from the value being transmitted is detected on the bus in slave transmit mode, for example, when slaves with the same address exist on the bus, or when a mismatch with the transmit data occurs because of noise.

NACKE bit (NACK Reception Transfer Suspension Enable)

The NACKE bit specifies whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. For normal operation, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended. When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.

For details, see section 34.9.2. NACK Reception Transfer Suspension Function.

SCLE bit (SCL Synchronous Circuit Enable)

The SCLE bit specifies whether to synchronize the SCL clock with the SCL input clock. For normal operation, set this bit to 1.

When the SCLE bit is set to 0 (no SCL synchronous circuit used), the IIC does not synchronize the SCL clock with the SCL input clock. With this setting, the IIC outputs the SCL clock with the transfer rate set in ICBRH and ICBRL, regardless of the SCLn line state. For this reason, if the bus load of the I^2C bus line is much larger than the specification value, or if the SCL clock output overlaps in multiple masters, a short-cycle SCL clock that does not meet the specification might be output. When no SCL synchronous circuit is used, it also affects the issuance of the start, restart, and stop conditions, and the continuous output of extra SCL clock cycles.

Do not set this bit to 0 except when checking the output of the set transfer rate.

34.3.7 ICSER : I^2C Bus Status Enable Register

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Function	R/W
0	SAR0E	Slave Address Register 0 Enable 0: Disable slave address in SARL0 and SARU0 1: Enable slave address in SARL0 and SARU0	R/W
1	SAR1E	Slave Address Register 1 Enable 0: Disable slave address in SARL1 and SARU1 1: Enable slave address in SARL1 and SARU1	R/W
2	SAR2E	Slave Address Register 2 Enable 0: Disable slave address in SARL2 and SARU2 1: Enable slave address in SARL2 and SARU2	R/W
3	GCAE	General Call Address Enable 0: Disable general call address detection 1: Enable general call address detection	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIDE	Device ID Address Detection Enable 0: Disable device ID address detection 1: Enable device ID address detection	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HOAE	Host Address Enable 0: Disable host address detection 1: Enable host address detection	R/W

SARyE bit (Slave Address Register y Enable) (y = 0 to 2)

The SARyE bit enables or disables the received slave address and the slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address. When this bit is set to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE bit (General Call Address Enable)

The GCAE bit specifies whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the IIC recognizes the received slave address as the general call address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2), and performs the data receive operation. When this bit is set to 0, the received slave address is ignored even if it matches the

general call address.

DIDE bit (Device ID Address Detection Enable)

The DIDE bit specifies whether to recognize and execute the device ID address when a device ID (1111 100b) is received in the first frame after a start or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the IIC recognizes that the device ID address was received. When the next R/W# bit is 0 (W), the IIC recognizes the second and the subsequent frames as slave addresses and continues the receive operation. When this bit is set to 0, the IIC ignores the received first frame even if it matches the device ID address, and recognizes the first frame as a normal slave address.

For details on the device ID function, see section 34.8.3. Device ID Address Detection.

HOAE bit (Host Address Enable)

The HOAE bit specifies whether to ignore the received host address (0001 000b) when the SMBS bit in ICMR3 is 1.

When this bit is set to 1 while the SMBS bit in ICMR3 is 1, if the received slave address matches the host address, the IIC recognizes the received slave address as the host address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2), and performs the receive operation.

When the SMBS bit in ICMR3 or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

34.3.8 ICIER : I2C Bus Interrupt Enable Register

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOIE	Timeout Interrupt Request Enable 0: Disable timeout interrupt IICn_EEI (TMOF) request 1: Enable timeout interrupt IICn_EEI (TMOF) request	R/W
1	ALIE	Arbitration-Lost Interrupt Request Enable 0: Disable arbitration-lost interrupt IICn_EEI (AL) request 1: Enable arbitration-lost interrupt IICn_EEI (AL) request	R/W
2	STIE	Start Condition Detection Interrupt Request Enable 0: Disable start condition detection interrupt IICn_EEI (START) request 1: Enable start condition detection interrupt IICn_EEI (START) request	R/W
3	SPIE	Stop Condition Detection Interrupt Request Enable 0: Disable stop condition detection interrupt IICn_EEI (STOP) request 1: Enable stop condition detection interrupt IICn_EEI (STOP) request	R/W

Bit	Symbol	Function	R/W
4	NAKIE	NACK Reception Interrupt Request Enable 0: Disable NACK reception interrupt IICn_EEI (NACKF) request 1: Enable NACK reception interrupt IICn_EEI (NACKF) request	R/W
5	RIE	Receive Data Full Interrupt Request Enable 0: Disable receive data full interrupt (IICn_RXI) request 1: Enable receive data full interrupt (IICn_RXI) request	R/W
6	TEIE	Transmit End Interrupt Request Enable 0: Disable transmit end interrupt (IICn_TEI) request 1: Enable transmit end interrupt (IICn_TEI) request	R/W
7	TIE	Transmit Data Empty Interrupt Request Enable 0: Disable transmit data empty interrupt (IICn_TXI) request 1: Enable transmit data empty interrupt (IICn_TXI) request	R/W

TMOIE bit (Timeout Interrupt Request Enable)

The TMOIE bit enables or disables timeout interrupt IICn_EEI (TMOF) requests when the TMOF flag in ICSR2 is 1. To cancel a IICn_EEI (TMOF) interrupt request, set the TMOF flag or the TMOIE bit to 0.

ALIE bit (Arbitration-Lost Interrupt Request Enable)

The ALIE bit enables or disables arbitration-lost interrupt IICn_EEI (AL) requests when the AL flag in ICSR2 is 1. To cancel an IICn_EEI (AL) interrupt request, set the AL flag or the ALIE bit to 0.

STIE bit (Start Condition Detection Interrupt Request Enable)

The STIE bit enables or disables start condition detection interrupt IICn_EEI (START) requests when the START flag in ICSR2 is 1. To cancel an IICn_EEI (START) interrupt request, set the START flag or the STIE bit to 0.

SPIE bit (Stop Condition Detection Interrupt Request Enable)

The SPIE bit enables or disables stop condition detection interrupt IICn_EEI (STOP) requests when the STOP flag in ICSR2 is 1. To cancel an IICn_EEI (STOP) interrupt request, set the STOP flag or the SPIE bit to 0.

NAKIE bit (NACK Reception Interrupt Request Enable)

The NAKIE bit enables or disables NACK reception interrupt IICn_EEI (NACKF) requests when the NACKF flag in ICSR2 is 1. To cancel an IICn_EEI (NACKF) interrupt request, set the NACKF flag or the NAKIE bit to 0.

RIE bit (Receive Data Full Interrupt Request Enable)

The RIE bit enables or disables receive data full interrupt (IICn_RXI) requests when the RDRF flag in ICSR2 is 1.

TEIE bit (Transmit End Interrupt Request Enable)

The TEIE bit enables or disables transmit end interrupt (IICn_TEI) requests when the TEND flag in ICSR2 is 1. To cancel an IICn_TEI interrupt request, set the TEND flag or the TEIE bit to 0.

TIE bit (Transmit Data Empty Interrupt Request Enable)

The TIE bit enables or disables transmit data empty interrupt (IICn_TXI) requests when the TDRE flag in ICSR2 is 1.

34.3.9 ICSR1 : I2C Bus Status Register 1

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 not detected 1: Slave address 0 detected	R/W*1
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 not detected 1: Slave address 1 detected	R/W*1
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 not detected 1: Slave address 2 detected	R/W*1
3	GCA	General Call Address Detection Flag 0: General call address not detected 1: General call address detected	R/W*1
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DID	Device ID Address Detection Flag This bit is set to 1 when the first frame received immediately after a start condition is detected, matches a value of (device ID (1111 100b) + 0 [W]). 0: Device ID command not detected 1: Device ID command detected	R/W*1
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HOA	Host Address Detection Flag This bit is set to 1 when the received slave address matches the host address (0001 000b). 0: Host address not detected 1: Host address detected	R/W*1

Note 1. Only 0 can be written to clear the flag.

AASy bit (Slave Address y Detection Flag) (y = 0 to 2)

The AASy flag indicates whether slave address y was detected.

[Setting conditions]

For 7-bit address format (SARUy.FS = 0):

- When the received slave address matches the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

For 10-bit address format: (SARUy.FS = 1):

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address matches the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy flag after reading AASy = 1
- When a stop condition is detected.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

For 7-bit address format (SARUy.FS = 0):

- When the received slave address does not match the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.

For 10-bit address format (SARUy.FS = 1):

- When the received slave address does not match a value of 11110b + SVA[1:0] in SARUy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.

- When the received slave address matches a value of 11110b + SVA[1:0] in SARUy, and the subsequent address does not match the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.

GCA bit (General Call Address Detection Flag)

The GCA flag indicates whether the general call address was detected.

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled).
  The GCA flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA = 1.
- When a stop condition is detected.
- When the received slave address does not match the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled).
  The GCA flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

DID bit (Device ID Address Detection Flag)

The DID flag indicates whether the device ID address was detected.

[Setting condition]

- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), with the DIDE bit in ICSER set to 1 (device ID address detection enabled).
  The DID flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID = 1.
- When a stop condition is detected.
- When the first frame received immediately after a start or restart condition is detected does not match a value of the device ID (1111 100b), with the DIDE bit in ICSER set to 1 (device ID address detection enabled).
  The DID flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.
- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), and the second frame does not match any slave address from 0 to 2, with the DIDE bit in ICSER set to 1 (device ID address detection enabled).
  The DID flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

HOA bit (Host Address Detection Flag)

The HOA flag indicates whether the host address was detected.

[Setting condition]

- When the received slave address matches the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled).
  The HOA flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA = 1.
- When a stop condition is detected.
- When the received slave address does not match the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled).
  The HOA flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.

- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

34.3.10 ICSR2 : I2C Bus Status Register 2

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOF	Timeout Detection Flag 0: Timeout not detected 1: Timeout detected.	R/W*1
1	AL	Arbitration-Lost Flag 0: Arbitration not lost 1: Arbitration lost	R/W*1
2	START	Start Condition Detection Flag 0: Start condition not detected 1: Start condition detected	R/W*1
3	STOP	Stop Condition Detection Flag 0: Stop condition not detected 1: Stop condition detected	R/W*1
4	NACKF	NACK Detection Flag 0: NACK not detected 1: NACK detected	R/W*1
5	RDRF	Receive Data Full Flag 0: ICDRR contains no receive data 1: ICDRR contains receive data	R/W*1
6	TEND	Transmit End Flag 0: Data being transmitted 1: Data transmit complete	R/W*1
7	TDRE	Transmit Data Empty Flag 0: ICDRT contains transmit data 1: ICDRT contains no transmit data	R

Note 1. Only 0 can be written, to clear the flag.

TMOF bit (Timeout Detection Flag)

The TMOF flag is set to 1 when IIC detects a timeout because the SCLn line state remains unchanged for the set period.

[Setting condition]

- When the SCLn line state remains unchanged for the period specified in the ICMR2.TMOH, TMOL, and TMOS bits while the ICFER.TMOE bit is 1 (timeout function enabled) in master or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF flag after reading TMOF = 1.
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

AL bit (Arbitration-Lost Flag)

The AL flag indicates that the bus mastership is lost in arbitration because of a bus conflict or because of some other reason, when a start condition was issued, or an address and data was transmitted.

The IIC monitors the level on the SDAn line during transmission. If the level on the line does not match the value of the bit being output, the IIC sets the AL flag to 1 to indicate that the bus is occupied by another device. The IIC can also set the flag to indicate the detection of arbitration loss during NACK transmission in master mode or during data transmission in slave mode.

[Setting conditions]

When master arbitration-lost detection is enabled (ICFER.MALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock except for the ACK period during data transmission in master transmit mode.
- When a start condition is detected while the ST bit in ICCR2 is 1 (start condition requested) or the internal SDA output state does not match the SDAn line level.
- When the ST bit in ICCR2 is 1 (start condition requested), with the BBSY flag in ICCR2 set to 1.

When NACK arbitration-lost detection is enabled (ICFER.NALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled (ICFER.SALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock, except for the ACK period during data transmission in slave transmit mode.

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Table 34.7 Relationship between arbitration-lost generation sources and arbitration-lost enable functions

ICFER			ICSR2		
MALE	NALE	SALE	AL	Error	Arbitration-lost generation source
1	×	×	1	Start condition issuance error	When internal SDA output state does not match SDAn line level when a start condition is detected, while the ST bit in ICCR2 is 1.
					When ST in ICCR2 is set to 1 while BBSY in ICCR2 is 1.
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode.
×	1	×	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master or slave receive mode.
×	×	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode.

Note: ×: Don't care

START bit (Start Condition Detection Flag)

The START flag indicates whether a start or restart condition is detected.

[Setting condition]

- When a start or restart condition is detected.

[Clearing conditions]

- When 0 is written to the START flag after reading START = 1.
- When a stop condition is detected.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

STOP bit (Stop Condition Detection Flag)

The STOP flag indicates whether a stop condition is detected.

[Setting condition]

- When a stop condition is detected.

[Clearing conditions]

- When 0 is written to the STOP flag after reading STOP = 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

NACKF bit (NACK Detection Flag)

The NACKF flag indicates whether a NACK was detected.

[Setting condition]

- When acknowledge is not received (NACK received) from the receive device in transmit mode, with the NACKE bit in ICFER set to 1 (transfer suspension enabled).

[Clearing conditions]

- When 0 is written to the NACKF flag after reading NACKF = 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1, the IIC suspends data transmission and reception. Writing to ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit or receive operation. To restart data transmission or reception, set the NACKF flag to 0.

RDRF bit (Receive Data Full Flag)

The RDRF flag indicates whether the IDCRR contains receive data.

[Setting conditions]

- When receive data is transferred from ICDRS to ICDRR.
  The RDRF flag is set to 1 on the rising edge of the 8th or 9th SCL clock cycle (selected in the RDRFS bit in ICMR3).
- When the received slave address matches after a start (or restart) condition is detected with the TRS bit in ICCR2 set to 0.

[Clearing conditions]

- When 0 is written to the RDRF flag after reading RDRF = 1.
- When data is read from ICDRR.
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

TEND bit (Transmit End Flag)

The TEND flag indicates whether data transmission is still being transmitted or is complete.

[Setting condition]

- On the rising edge of the 9th SCL clock cycle while the TDRE flag is 1.

[Clearing conditions]

- When 0 is written to the TEND flag after reading TEND = 1.
- When data is written to ICDRT.
- When a stop condition is detected.
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

TDRE bit (Transmit Data Empty Flag)

The TDRE flag indicates whether the ICDRT contains transmit data.

[Setting conditions]

- When data is transferred from ICDRT to ICDRS and ICDRT becomes empty.
- When the TRS bit in ICCR2 is set to 1.

- When the received slave address matches while the TRS bit is 1.

[Clearing conditions]

- When data is written to ICDRT.
- When the TRS bit in ICCR2 is set to 0.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1 while the NACKE bit in ICFER is 1, the IIC suspends data transmission and reception. In this case, if the TDRE flag is 0 (next transmit data written), data is transferred to the ICDRS register and the ICDRT register becomes empty on the rising edge of the 9th clock cycle, but the TDRE flag is not set to 1.

34.3.11 SARLy : Slave Address Register L y (y = 0 to 2)

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x0A + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SVA[6:0]							SVA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SVA0	10-bit Address LSB Slave address setting	R/W
7:1	SVA[6:0]	7-bit Address/10-bit Address Lower Bits Slave address setting	R/W

SVA0 bit (10-bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), the SVA0 bit functions as the LSB of a 10-bit address and is combined with the SVA[6:0] bits to form the lower 8 bits of a 10-bit address.

This bit is valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in this bit is ignored.

SVA[6:0] bits (7-bit Address/10-bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS = 0), the SVA[6:0] bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits combined with the SVA0 bit to form the lower 8 bits of a 10-bit address. When the SARyE bit in ICSER is 0, the setting in these bits is ignored.

34.3.12 SARUy : Slave Address Register U y (y = 0 to 2)

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x0B + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SVA[1:0]		FS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FS	7-bit/10-bit Address Format Select 0: Select 7-bit address format 1: Select 10-bit address format	R/W

Bit	Symbol	Function	R/W
2:1	SVA[1:0]	10-bit Address Upper Bits Slave address setting	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

FS bit (7-bit/10-bit Address Format Select)

The FS bit selects a 7-bit or 10-bit address format for slave address y (in SARLy and SARUy).

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the SVA[1:0] and SVA0 settings in SARLy are ignored.

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the SVA[1:0] and SARLy settings are valid.

When the SARyE bit in ICSER is 0 (SARLy and SARUy disabled), the SARUy.FS setting is invalid.

SVA[1:0] bits (10-bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), the SVA[1:0] bits function as the upper 2 bits of a 10-bit address.

These bits are valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in these bits is ignored.

34.3.13 ICBRL : I2C Bus Bit Rate Low-Level Register

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	BRL[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	BRL[4:0]	Bit Rate Low-Level Period Low-level period setting of SCL clock	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

ICBRL is a 5-bit register that sets the low-level period of the SCL clock. ICBRL also works to generate the data setup time for the automatic SCL low-hold operation (see section 34.9. Automatic Low-Hold Function for SCL). When the IIC is used only in slave mode, this register must be set to a value longer than the data setup time.*1

ICBRL counts the low-level period with the internal reference clock source (IICΦ) specified in the CKS[2:0] bits in ICMR1.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set the ICBRL register to a value at least one greater than the number of stages in the noise filter. For this number, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns for up to 100 kbps: Standard-mode (Sm)
100 ns for up to 400 kbps: Fast-mode (Fm)

34.3.14 ICBRH : I²C Bus Bit Rate High-Level Register

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	BRH[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	BRH[4:0]	Bit Rate High-Level Period High-level period setting of SCL clock	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

ICBRH is a 5-bit register that sets the high-level period of the SCL clock. ICBRH is valid in master mode. If the IIC is used only in slave mode, no setting is required in this register.

ICBRH counts the high-level period with the internal reference clock source (IICΦ) specified in the CKS[2:0] bits in ICMR1.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set the ICBRH register to a value at least one greater than the number of stages in the noise filter. For this number, see the description of the ICMR3.NF[1:0] bits.

The IIC transfer rate and the SCL clock duty are calculated using the following expressions (1) to (5):

(1) ICFER.SCLE = 0
Transfer rate = 1/{[(BRH + 1) + (BRL + 1)]/IICΦ*1 + tr*2 + tf*2}
Duty cycle = {tr + [(BRH + 1)/IICΦ]}/{tr + tf + [(BRH + 1) + (BRL + 1)]/IICΦ}

(2) ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] = 000b (IICΦ = PCLKL)
Transfer rate = 1/{[(BRH + 3) + (BRL+ 3)]/IICΦ + tr + tf}
Duty cycle = {tr + [(BRH + 3)/IICΦ]}/{tr + tf + [(BRH + 3) + (BRL + 3)]/IICΦ}

(3) ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] = 000b (IICΦ = PCLKL)
Transfer rate = 1/{[(BRH + 3 + nf*3) + (BRL + 3 + nf)]/IICΦ + tr + tf}
Duty cycle = {tr + [(BRH + 3 + nf)/IICΦ]}/{tr + tf + [(BRH + 3 + nf) + (BRL + 3 + nf)]/IICΦ}

(4) ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] ≠ 000b
Transfer rate = 1/{[(BRH + 2) + (BRL + 2)]/IICΦ + tr + tf}
Duty cycle = {tr + [(BRH + 2)/IICΦ]}/{tr + tf + [(BRH + 2) + (BRL + 2)]/IICΦ}

(5) ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] ≠ 000b
Transfer rate = 1/{[(BRH + 2 + nf) + (BRL + 2 + nf)]/IICΦ + tr + tf}
Duty cycle = {tr + [(BRH + 2 + nf)/IICΦ]}/{tr + tf + [(BRH + 2 + nf) + (BRL + 2 + nf)]/IICΦ}

Note 1. IICΦ = PCLKL × Division ratio.
Note 2. The SCLn line rise time (tr) and SCLn line fall time (tf) depend on the total bus line capacitance (Cb) and the pull-up resistor (Rp). For details, see the I²C bus standard from NXP Semiconductors.
Note 3. nf = Number of digital noise filter stages selected in the ICMR3.NF bit.

Table 34.8 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 0

Transfer rate (kbps)	CKS[2:0]	BRH[4:0]	BRL[4:0]	PCLKL (MHz)	NF[1:0]	Computation expression
100	100b	16 (0x10)	16 (0x10)	62.5	—	(1)
400	001b	29 (0x1D)	29 (0x1D)	62.5	—	(1)

Table 34.9 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 1 and NFE = 0

Transfer rate (kbps)	CKS[2:0]	BRH[4:0]	BRL[4:0]	PCLKL (MHz)	NF[1:0]	Computation expression
100	100b	15 (0x0F)	15 (0x0F)	62.5	—	(4)
400	001b	28 (0x1C)	28 (0x1C)	62.5	—	(4)

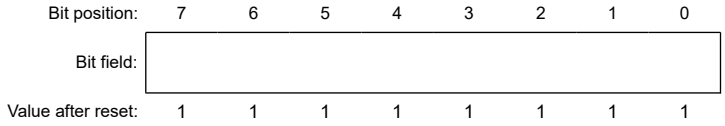
Table 34.10 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 1 and NFE = 1

Transfer rate (kbps)	CKS[2:0]	BRH[4:0]	BRL[4:0]	PCLKL (MHz)	NF[1:0]	Computation expression
100	100b	14 (0x0E)	14 (0x0E)	62.5	01b	(5)
400	001b	27 (0x1B)	27 (0x1B)	62.5	01b	(5)

34.3.15 ICDRT : I2C Bus Transmit Data Register

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x12



Bit	Symbol	Function	R/W
7:0	n/a	Transmit Data	R/W

When ICDRT detects a space in the I2C Bus Shift Register (ICDRS), it transfers the transmit data that is written to ICDRT to ICDRS and starts transmitting data in transmit mode.

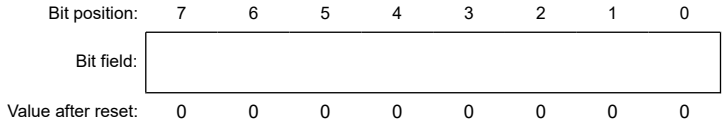
The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data is written to ICDRT while the ICDRS data is being transmitted.

ICDRT can always be read from and written to. Write transmit data to ICDRT when a transmit data empty interrupt (IICn_TXI) request is generated.

34.3.16 ICDRR : I2C Bus Receive Data Register

Base address: IICn = 0x8008_8000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x13



Bit	Symbol	Function	R/W
7:0	n/a	Receive Data	R

When 1 byte of data is received, the received data is transferred from the I2C Bus Shift Register (ICDRS) to ICDRR to enable the next data to be received.

The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data is read from ICDRR while ICDRS is receiving data. ICDRR cannot be written to. Read data from ICDRR when a receive data full interrupt (IICn_RXI) request is generated.

If ICDRR receives the next receive data before the current data is read from ICDRR while the RDRF flag in ICSR2 is 1, the IIC automatically holds the SCL clock low for 1 cycle before the RDRF flag is set to 1 again.

34.3.17 ICDRS : I²C Bus Shift Register

Base address: n/a

Offset address: n/a

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	—	—	—	—	—	—	—	—

ICDRS is an 8-bit shift register to transmit and receive data.

During transmission, transmit data is transferred from ICDRT to ICDRS and is sent from the SDAn pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data is received. ICDRS cannot be accessed directly.

34.4 Operation

34.4.1 Communication Data Format

The I²C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start or restart condition is an address frame that specifies a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

Figure 34.3 shows the I²C bus format, and Figure 34.4 shows the I²C bus timing.

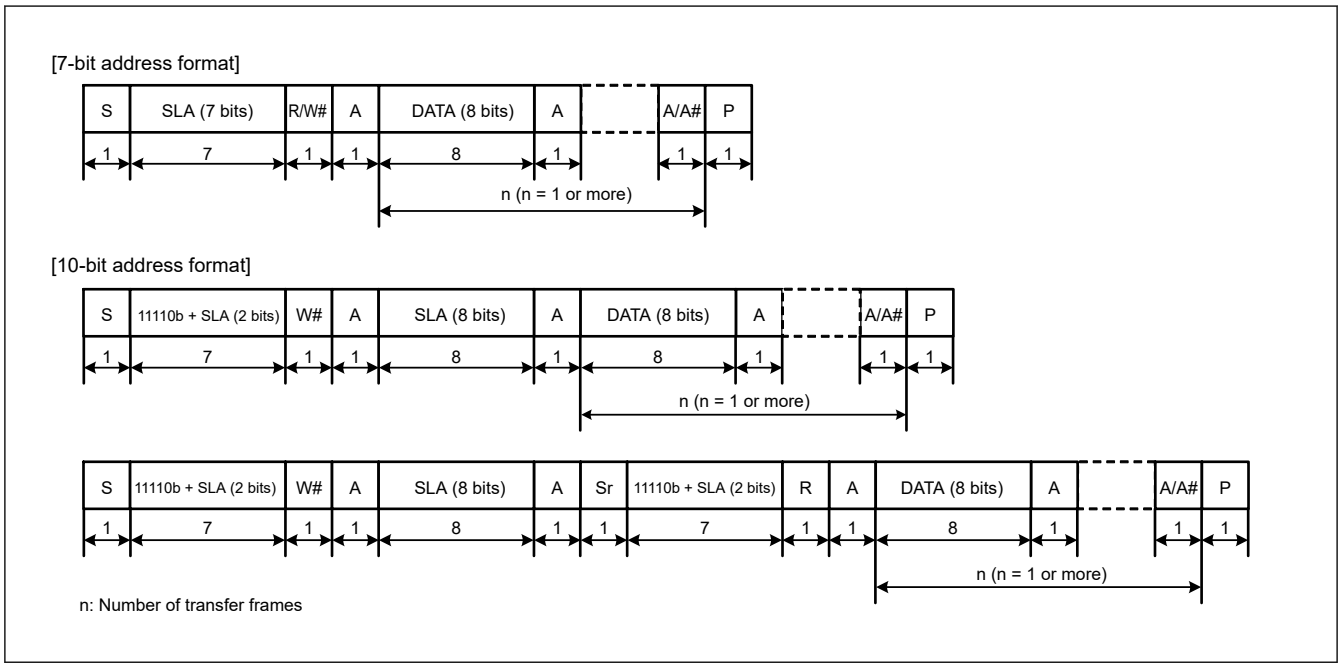


Figure 34.3 I²C bus format

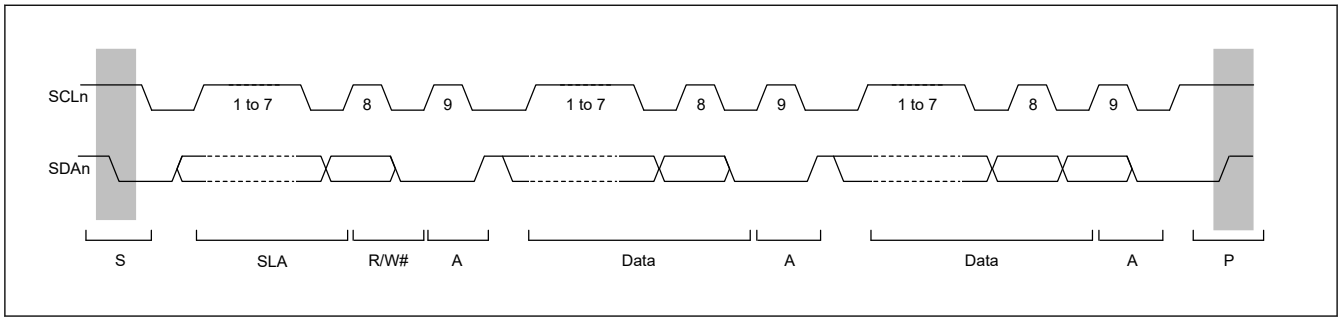


Figure 34.4 I²C bus timing when the SLA setting = 7 bits

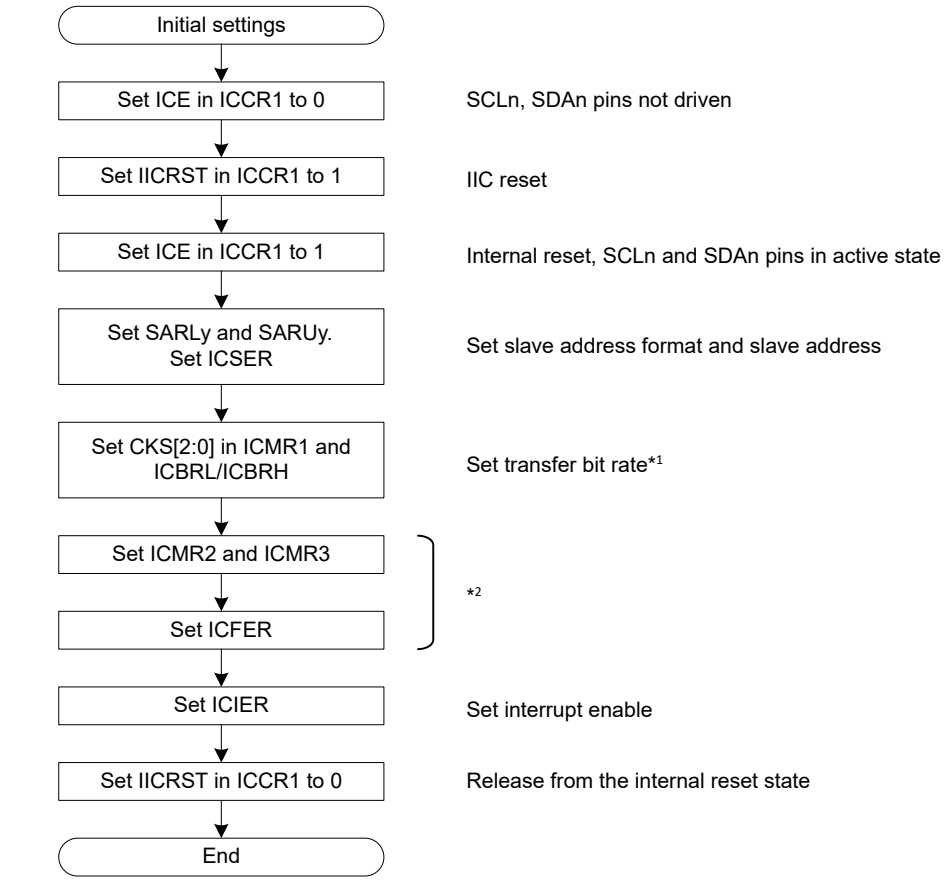
S:	Start condition. The master device drives the SDAn line low from high while the SCLn line is high.
SLA:	Slave address, by which the master device selects a slave device.
R/W#:	Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.
A:	Acknowledge. The receive device drives the SDAn line low. In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.
A#:	Not Acknowledge. The receive device drives the SDAn line high.
Sr:	Restart condition. The master device drives the SDAn line low from the high level after the setup time has elapsed with the SCLn line high.
DATA:	Transmitted or received data
P:	Stop condition. The master device drives the SDAn line high from low while the SCLn line is high.

34.4.2 Initial Settings

Before starting data transmission and reception, initialize the IIC using the procedure shown in Figure 34.5.

1. Set the ICCR1.ICE bit set to 0 to set the SCLn and SDAn pins to the inactive state.
2. Set the ICCR1.IICRST bit to 1 to initiate IIC reset.
3. Set the ICCR1.ICE bit to 1 to initiate internal reset, which initializes the flags and the internal state of the ICSR1 register.
4. Set the SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL registers (y = 0 to 2), and set the other registers as required. For the initial IIC settings, see Figure 34.5.
5. When the required register settings are complete, set the ICCR1.IICRST bit to 0, to release the IIC reset.

Note: This procedure is not necessary if the IIC initialization is already complete.



y = 0 to 2

Note 1. When the IIC is used only in slave mode, set the ICBRL register to a value longer than the data setup time.
Note 2. Set these registers as required.

Figure 34.5 Example IIC initialization flow

34.4.3 Master Transmit Operation

In master transmit operation, the IIC outputs the SCL clock and transmit data signals as the master device, and the slave device returns the acknowledgments. Figure 34.6 shows an example of master transmission, and Figure 34.7 to Figure 34.9 show the operation timing in master transmission.

To set up and perform master transmission:

1. To initialize the IIC, follow the procedure in section 34.4.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and START flags in ICSR2 automatically set to 1 and the ST bit is automatically set to 0. If the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that the start condition requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically sets to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag is automatically set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again sets to 1. After the byte containing the slave address and R/W# bit is transmitted, the value of the TRS bit automatically updates to select master transmit or master receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 0, the IIC continues in master transmit mode.

If the ICSR2.NACKF flag is 1, indicating that no slave device recognized the address or that there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
To transmit data with an address in the 10-bit format, start by writing 11110b, the 2 upper bits of the slave address, and W to ICDRT as the first address transmission. For the second address transmission, write the 8 lower bits of the slave address to ICDRT.

4. After confirming that the TDRE flag in ICSR2 is 1, write the transmit data to the ICDRT register. The IIC automatically holds the SCLn line low until the transmit data is ready or a stop condition is issued.
5. After all bytes of transmit data are written to the ICDRT register, wait until the value in the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition requested). On receiving a stop condition request, the IIC issues the stop condition.
6. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. In addition, the IIC automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
7. After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

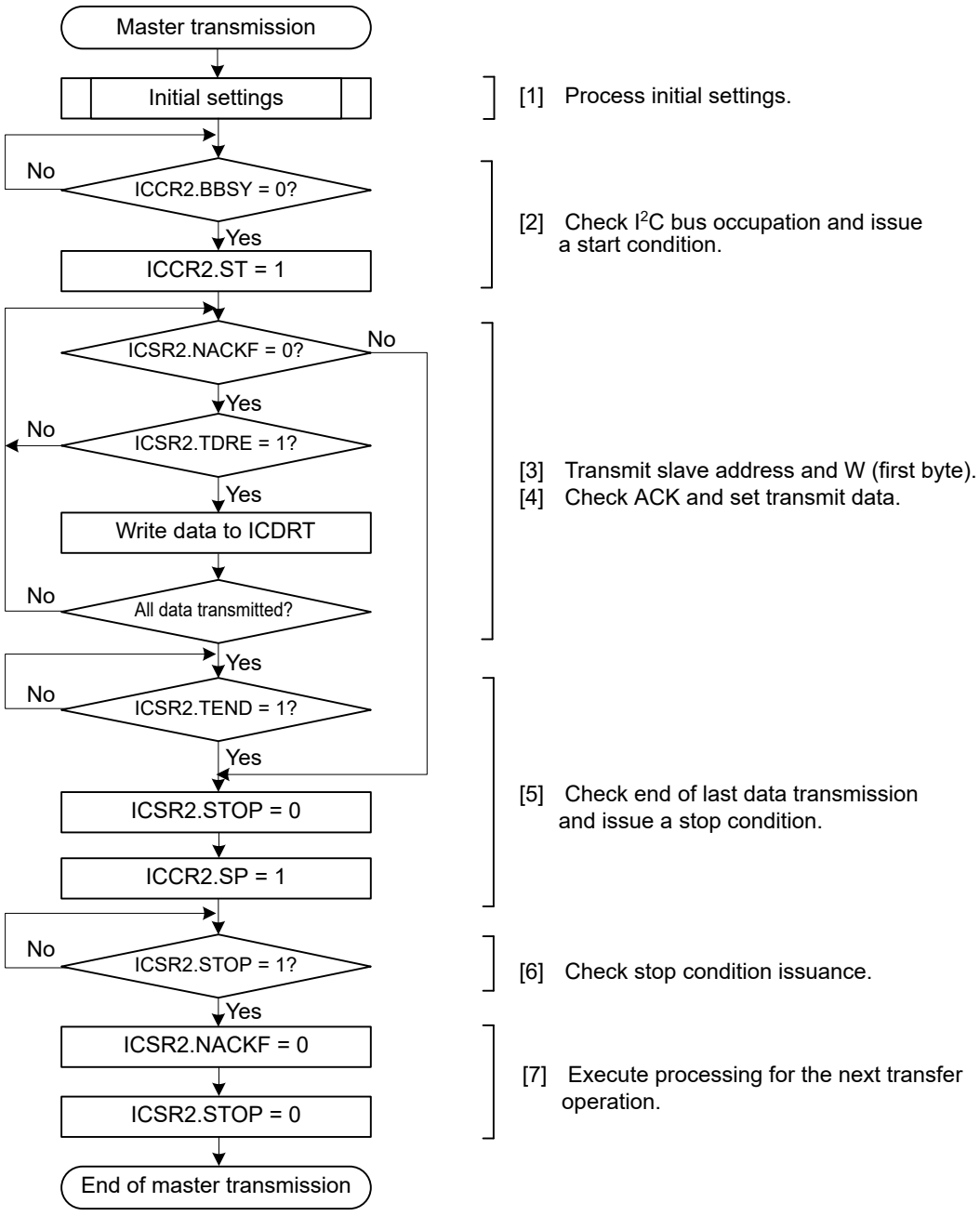


Figure 34.6 Example master transmission flow

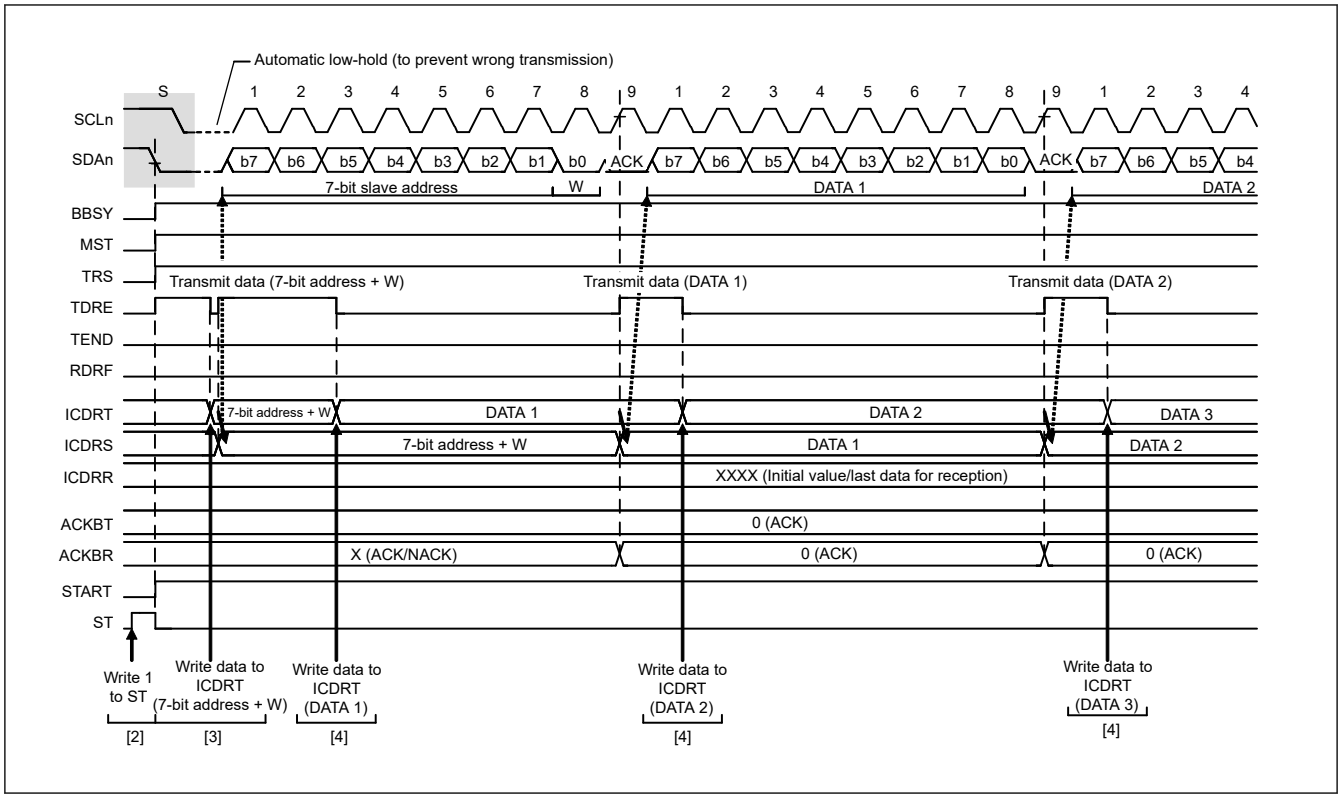


Figure 34.7 Master transmit operation timing (1) with 7-bit address format

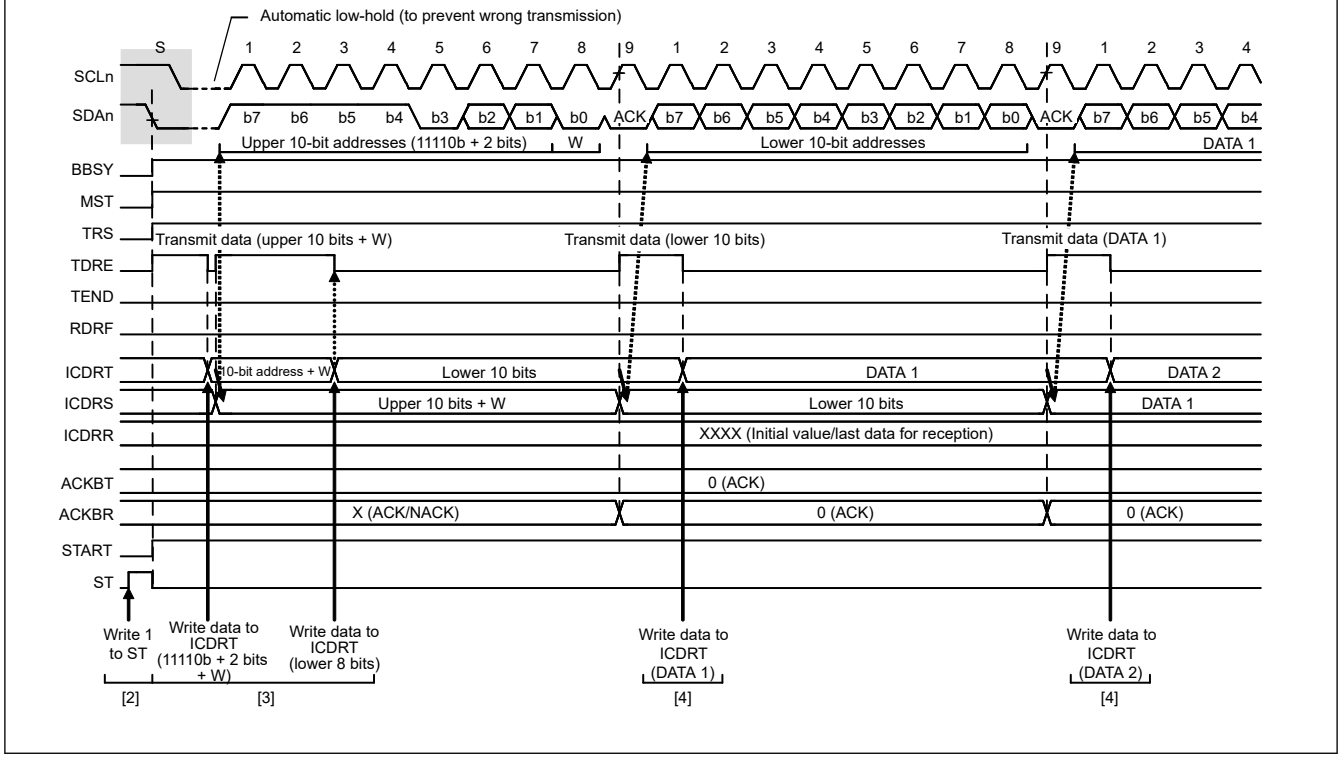


Figure 34.8 Master transmit operation timing (2) with 10-bit address format

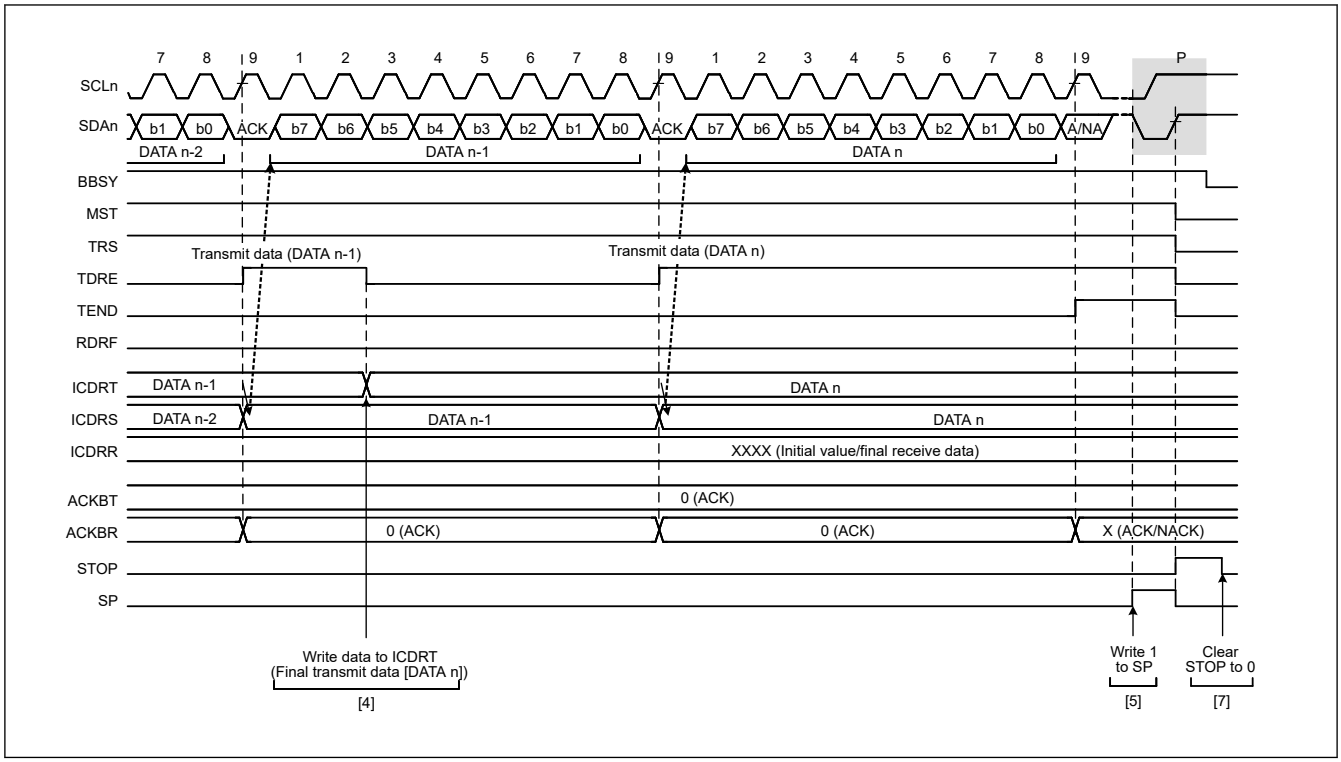


Figure 34.9 Master transmit operation timing (3)

34.4.4 Master Receive Operation

In master receive operation, the IIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because the IIC must start by sending a slave address to the associated slave device, the slave address part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 34.10 and Figure 34.11 show examples of master reception (7-bit address format), and Figure 34.12 to Figure 34.14 show the operation timing in master reception.

To set up and perform master reception:

1. To initialize the IIC, follow the procedure in section 34.4.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 to request the issue of a start condition. On receiving the request, the IIC issues a start condition. When the IIC detects the start condition, the BBSY and START flags in ICSR2 automatically set to 1, and the ST bit automatically sets to 0. If the start condition is detected and the levels for the SDA output and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also is automatically set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag is automatically set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. When the byte containing the slave address and R/W# bit is transmitted, the value of the ICCR2.TRS bit automatically updates to select transmit or receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 1, the TRS bit is set to 0 on the rising edge of the 9th cycle of the SCL clock, placing the IIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag is automatically set to 1.
   If the ICSR2.NACKF flag is 1, indicating that no slave device recognized the address or that there is an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
   For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmit 11110b, the two upper bits of the slave address, and the R bit to place the IIC in master receive mode.
4. Dummy read ICDRR after confirming that the RDRF flag in ICSR2 is 1. Doing so causes the IIC to start output of the SCL clock and start data reception.

5. After 1 byte of data is received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the 8th or 9th cycle of the SCL clock, as selected in the RDRFS bit in ICMR3. Reading ICDRR produces the received data, and automatically sets the RDRF flag to 0. The value of the acknowledgment field received during the 9th cycle of the SCL clock is returned as the value set in the ICMR3.ACKBT bit. If the next byte to be received is the second-to-last byte, set the ICMR3.WAIT bit to 1 for wait insertion before reading ICDRR, containing the second-to-last byte. In addition to enabling NACK output, even when interrupts or other operations result in delays in setting the ICMR3.ACKBT bit to 1 (NACK) in step (6), this fixes the SCLn line to the low level on the rising edge of the 9th clock cycle in reception of the last byte, which enables the issuing of a stop condition.
6. When the ICMR3.RDRFS bit is 0, and the slave device must be notified that it is to end transfer for data reception after transfer of the next and final byte, set the ICMR3.ACKBT bit to 1 (NACK).
7. After reading the second-to-last byte from the ICDRR register, if the value of the ICSR2.RDRF flag is 1, write 1 to the SP bit in ICCR2 (stop condition requested), and then read the last byte from ICDRR. When ICDRR is read, the IIC is released from the wait state and issues the stop condition after low-level output in the 9th clock cycle is complete or the SCLn line is released from the low-hold state.
8. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, detection of the stop condition sets the ICSR2.STOP flag to 1.
9. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and ICSR2.STOP flags to 0 for the next transfer operation.

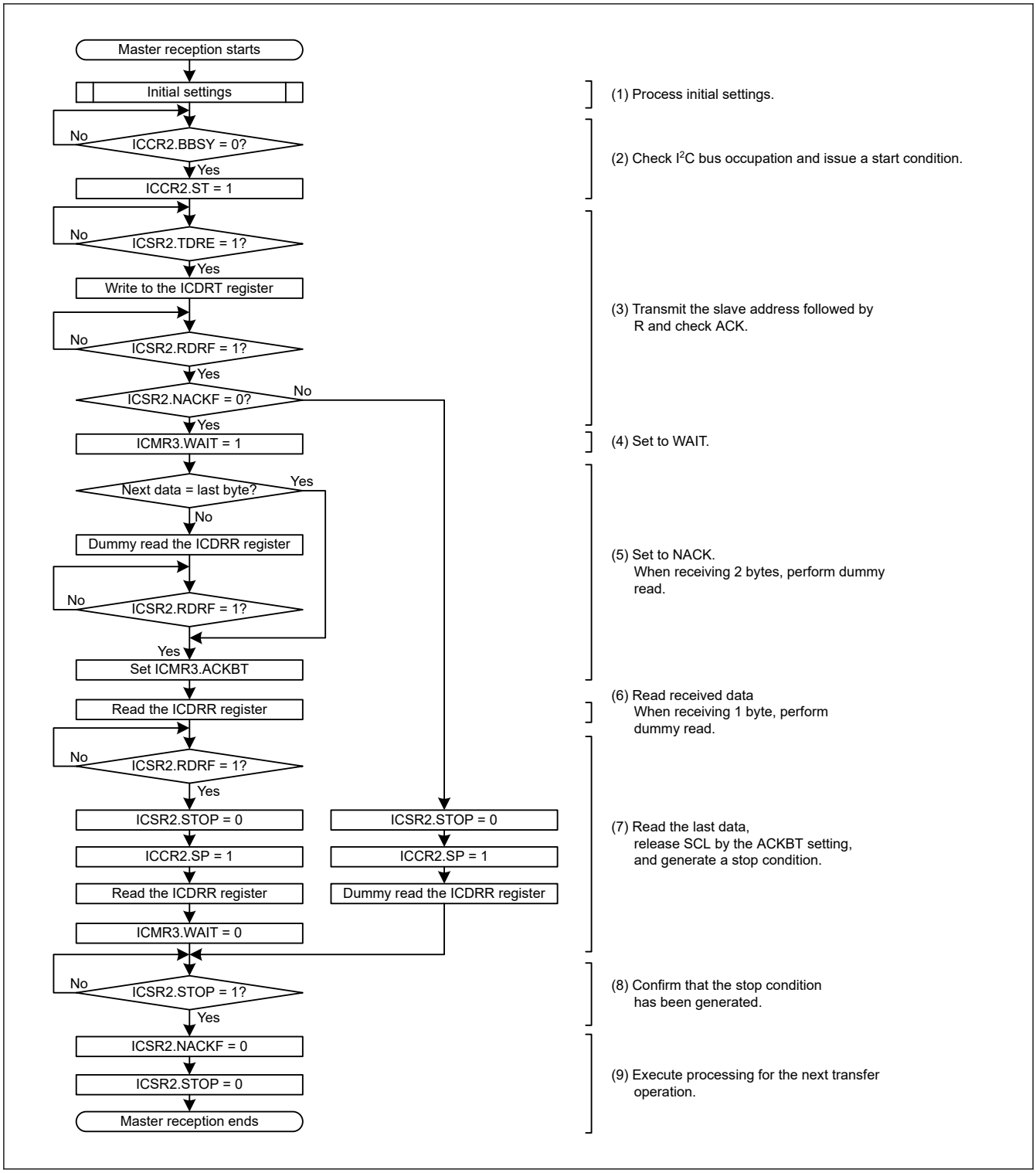


Figure 34.10 Example master reception flow with 7-bit address format and 1 or 2 bytes

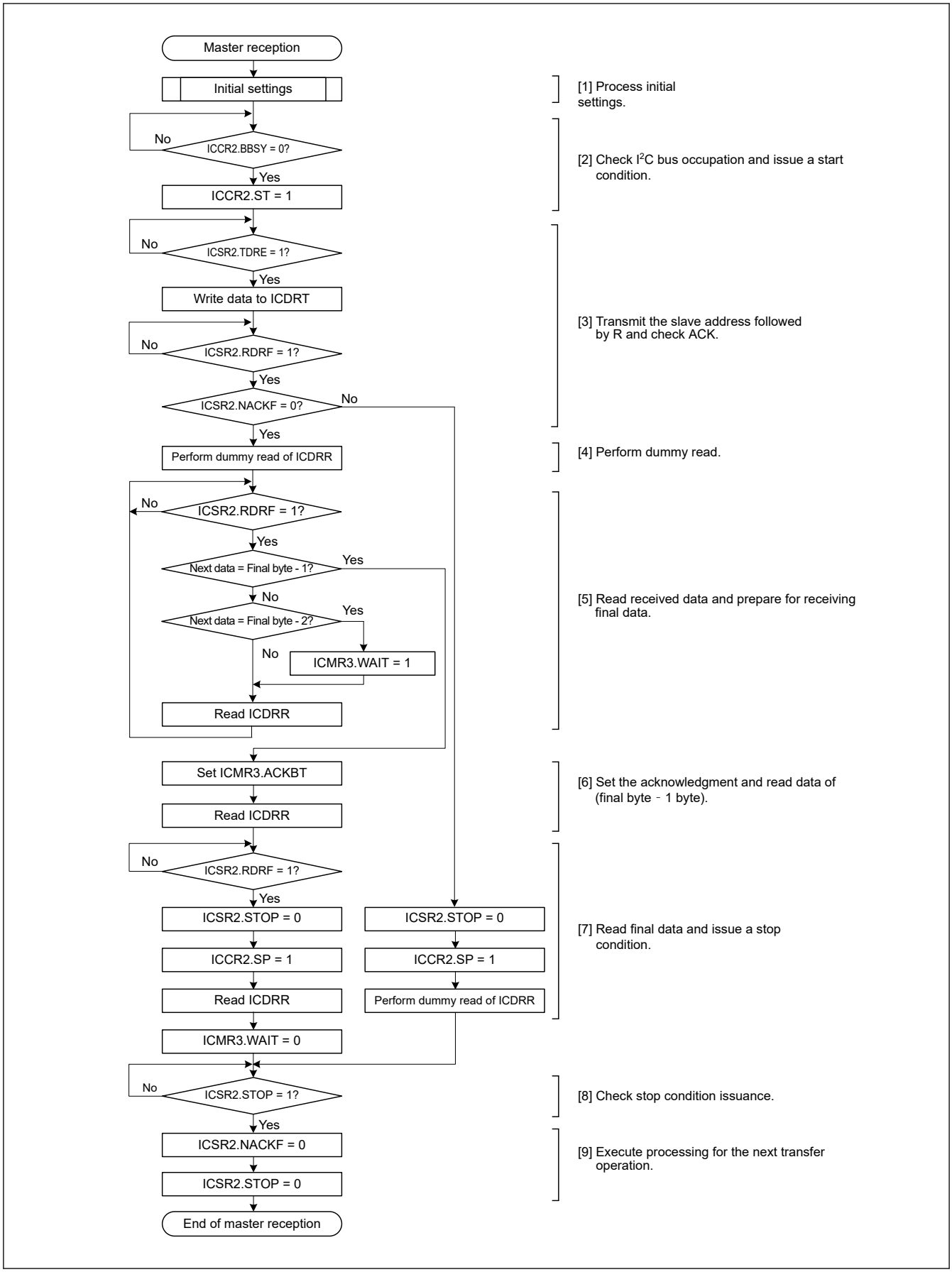


Figure 34.11 Example master reception flow with 7-bit address format and 3 or more bytes

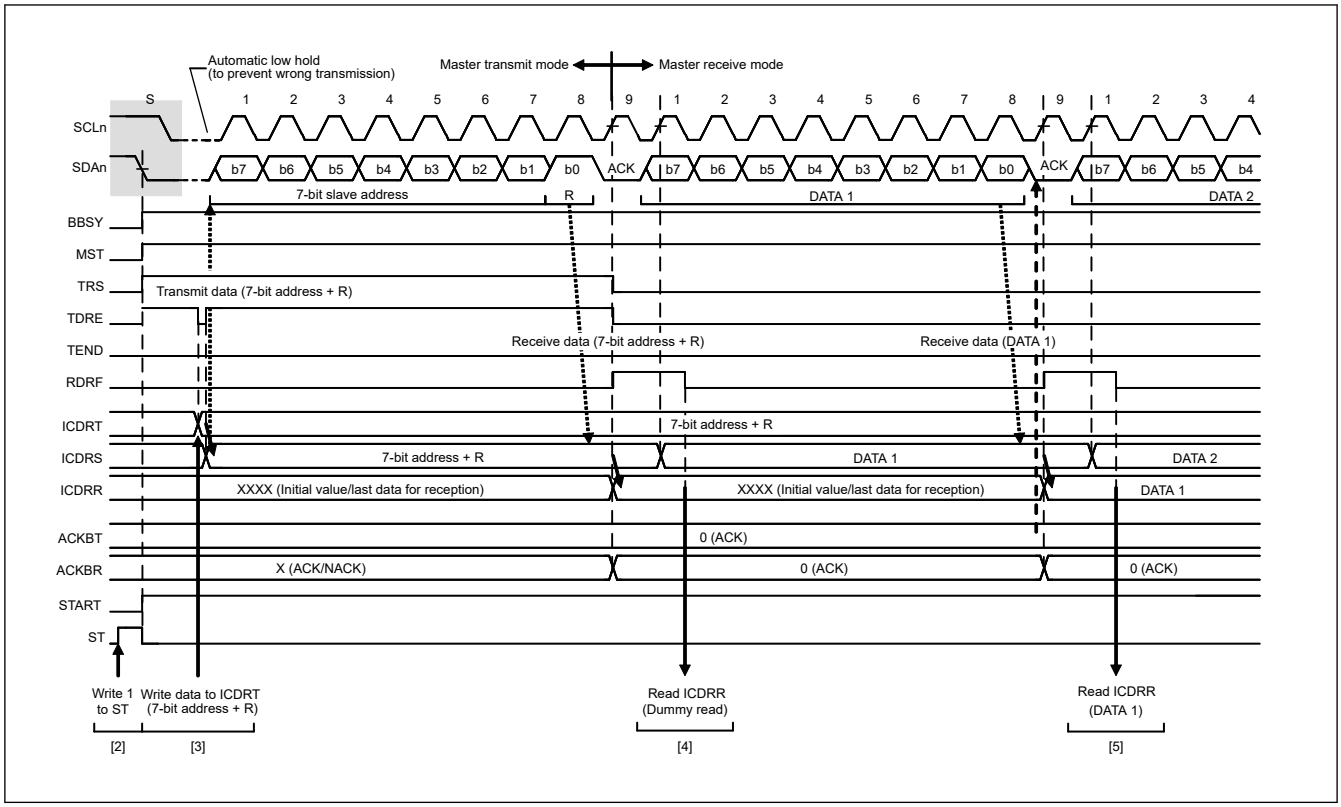


Figure 34.12 Master receive operation timing (1) with 7-bit address format, when RDRFS = 0

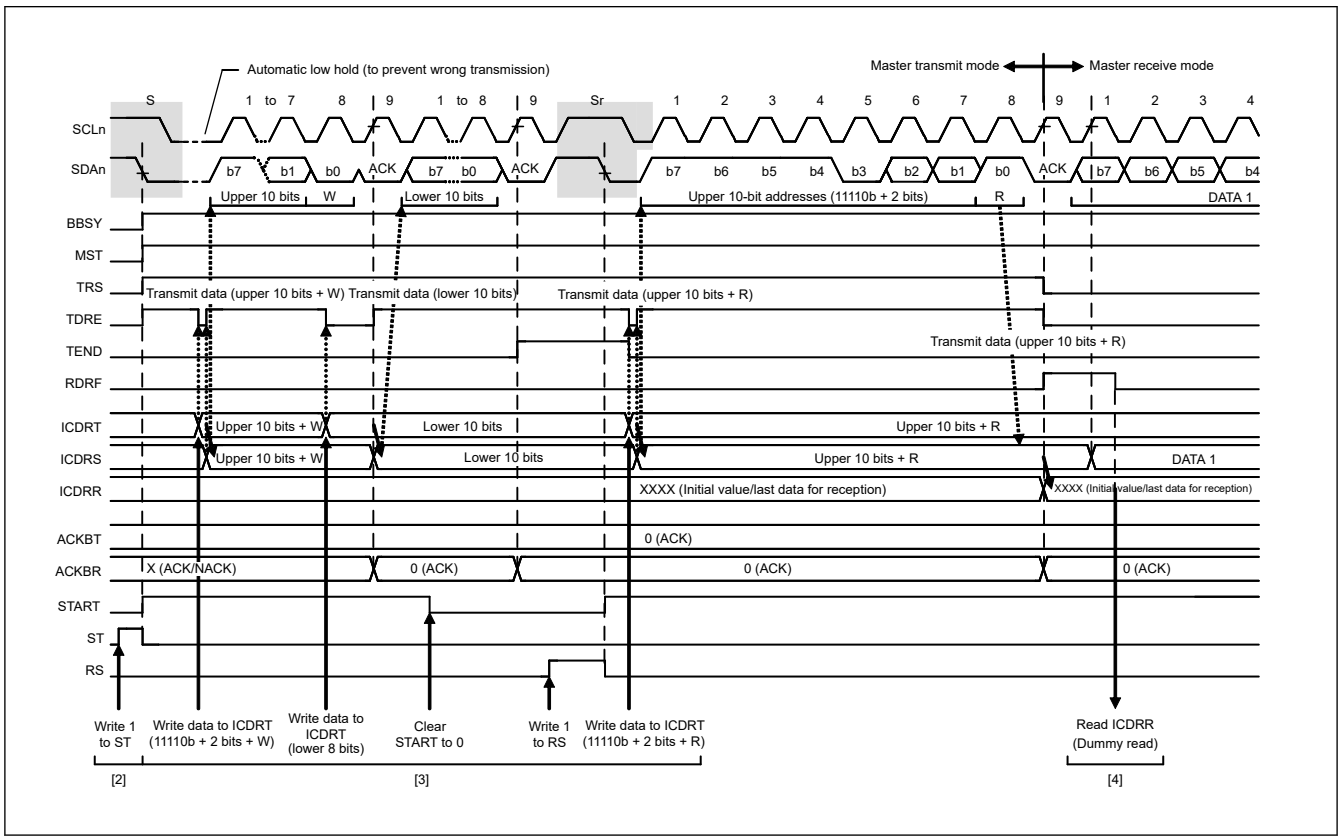


Figure 34.13 Master receive operation timing (2) with 10-bit address format, when RDRFS = 0

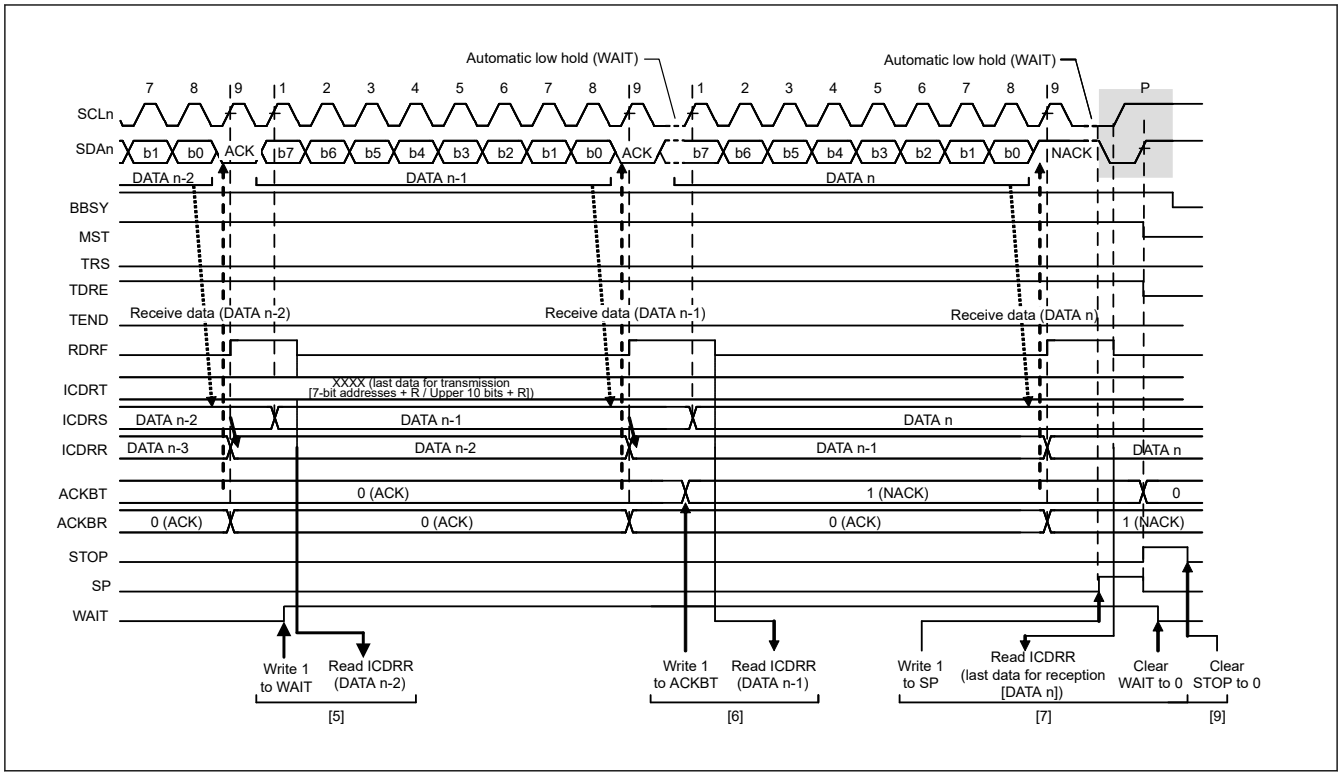


Figure 34.14 Master receive operation timing (3) when RDRFS = 0

34.4.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the IIC transmits data as a slave device, and the master device returns the acknowledgments.

Figure 34.15 shows an example of slave transmission, and Figure 34.16 and Figure 34.17 show the operation timing in slave transmission.

To set up and perform slave transmission:

1. To initialize the IIC, follow the procedure in section 34.4.2. Initial Settings.
   After the initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) flags to 1 on the rising edge of the 9th cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the 9th cycle of the SCL clock. If the value of the received R/W# bit is 1, the IIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.
3. Check that the ICSR2.TEND flag is 1, and write the transmit data to the ICDRT register. If the IIC receives no acknowledge from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the IIC suspends transfer of the next data.
4. Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the IIC drives the SCLn line low on the 9th falling edge of the SCL clock.
5. When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read ICDRR to complete the processing. This releases the SCLn line.
6. On detecting the stop condition, the IIC automatically sets the ICSR1.HOA, GCA, and AASy (y = 0 to 2) flags, the ICSR2.TDRE and TEND flags, and the ICCR2.TRS bit to 0, and enters slave receive mode.
7. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

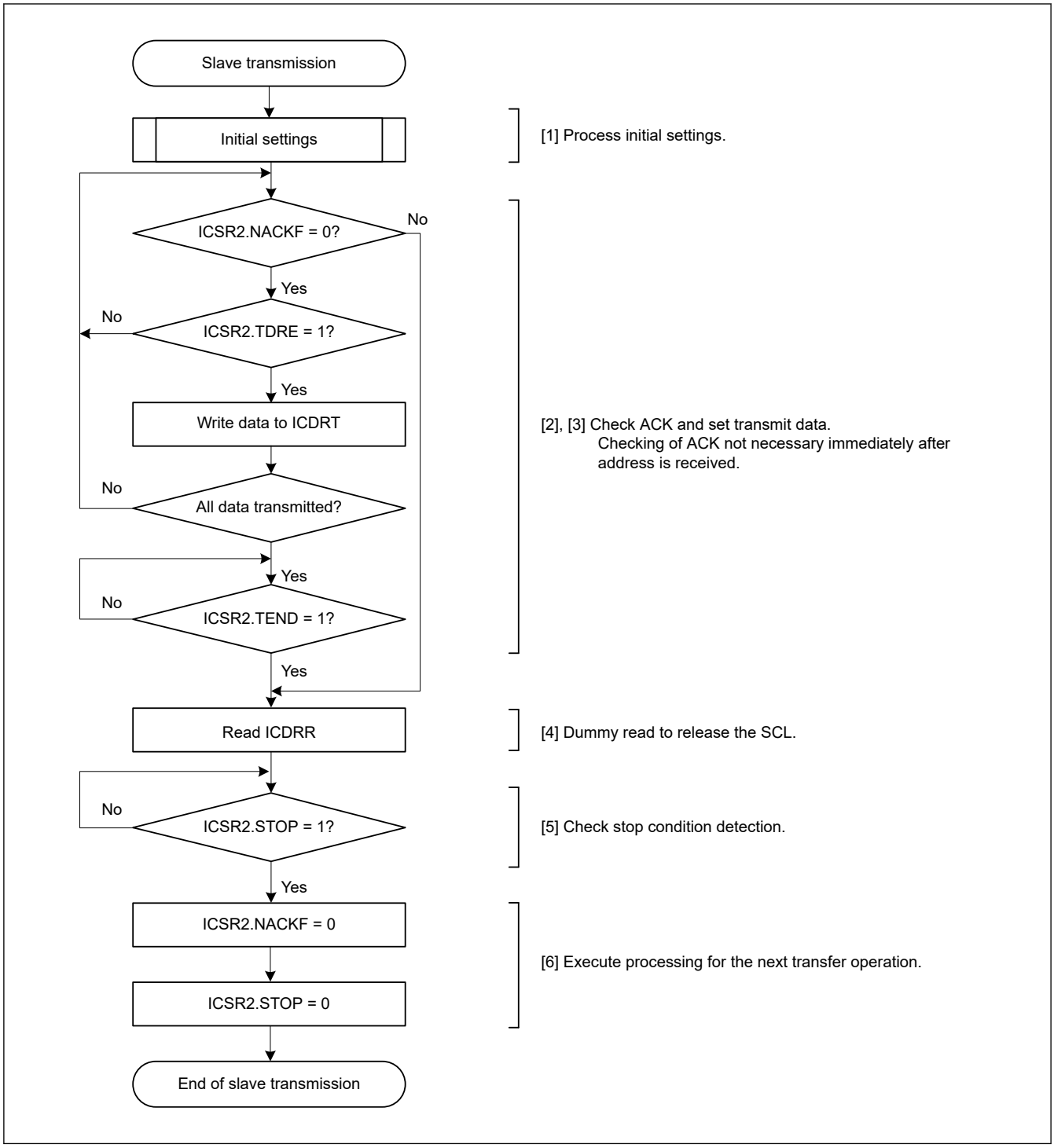


Figure 34.15 Example slave transmission flow

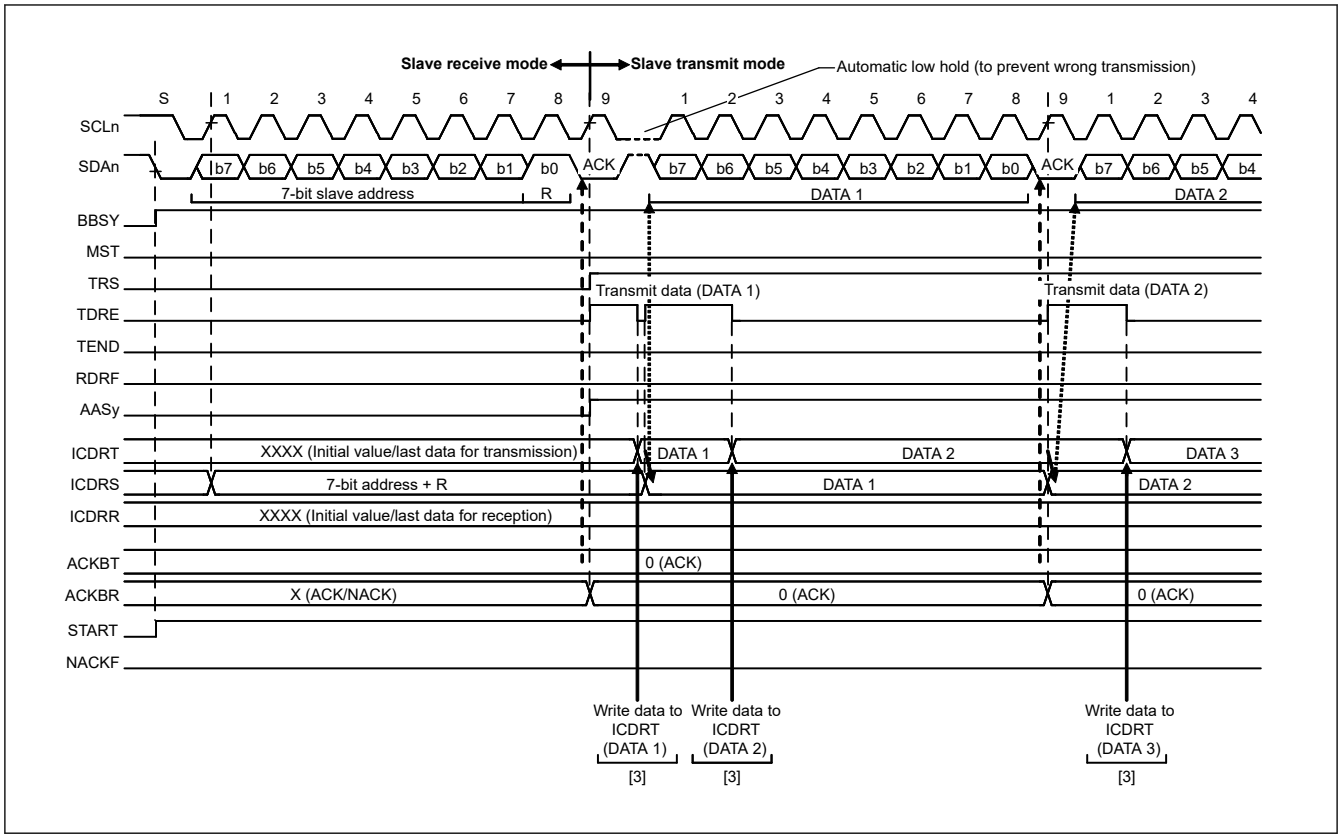


Figure 34.16 Slave transmit operation timing (1) with 7-bit address format

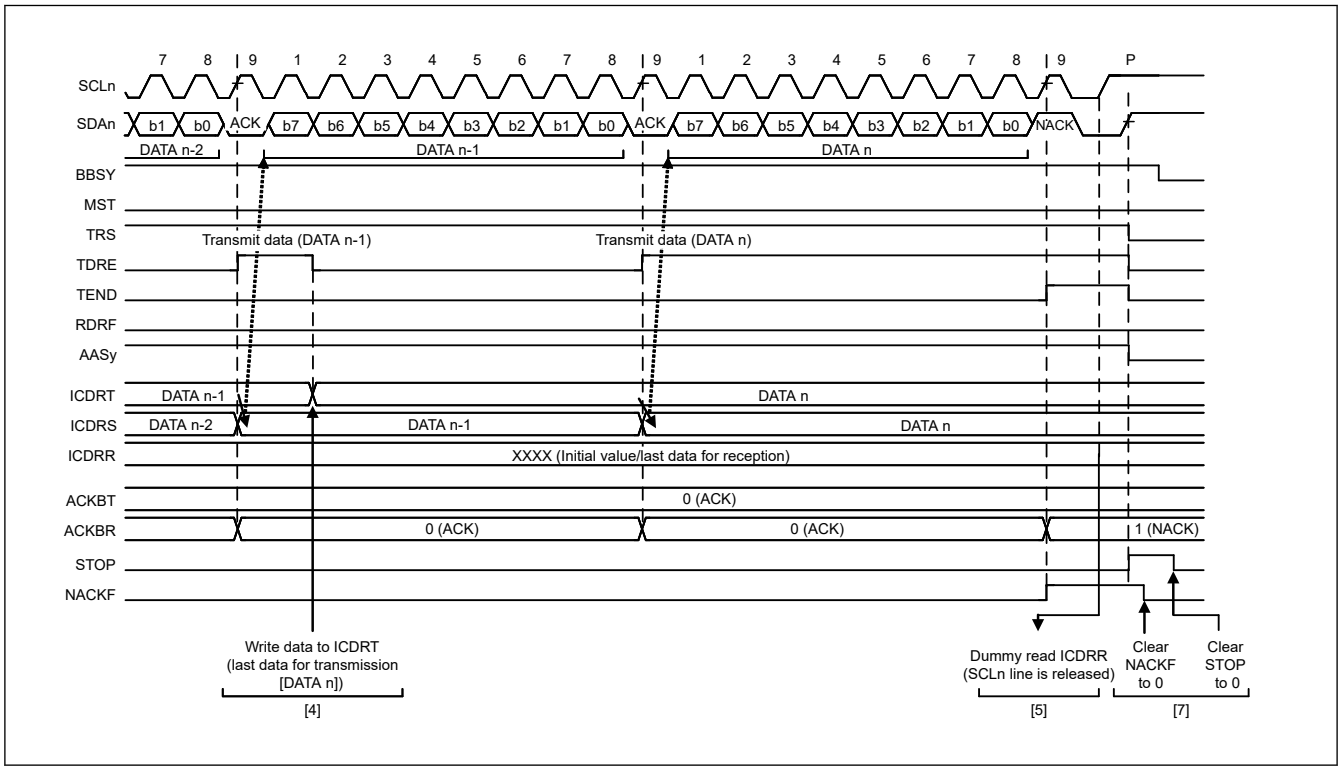


Figure 34.17 Slave transmit operation timing (2)

34.4.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the IIC returns acknowledgments as a slave device.

Figure 34.18 shows an example of slave reception, and Figure 34.19 and Figure 34.20 show the operation timing in slave reception.

To set up and perform slave reception:

1. To initialize the IIC, follow the procedure in section 34.4.2. Initial Settings.
   After the initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy (y = 0 to 2) flags to 1 on the rising edge of the 9th cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the 9th cycle of the SCL clock. If the value of the received R/W# bit is 0, the IIC continues to place itself in slave receive mode and sets the RDRF flag in ICSR2 to 1.
3. Check that the ICSR2.STOP flag is 0 and the ICSR2.RDRF flag is 1, and then dummy read the ICDRR register.
   The dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected.
4. When ICDRR is read, the IIC automatically sets the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and the next byte is received while the RDRF flag is still set to 1, the IIC holds the SCLn line low until 1 SCL cycle before the point where RDRF must be set. In this case, reading ICDRR releases the SCLn line from being held at the low level. When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.
5. On detecting the stop condition, the IIC automatically clears the ICSR1.HOA, GCA, and AASy (y = 0 to 2) flags to 0.
6. Check that the ICSR2.STOP flag is 1, then set the ICSR2.STOP flag to 0 for the next transfer operation.

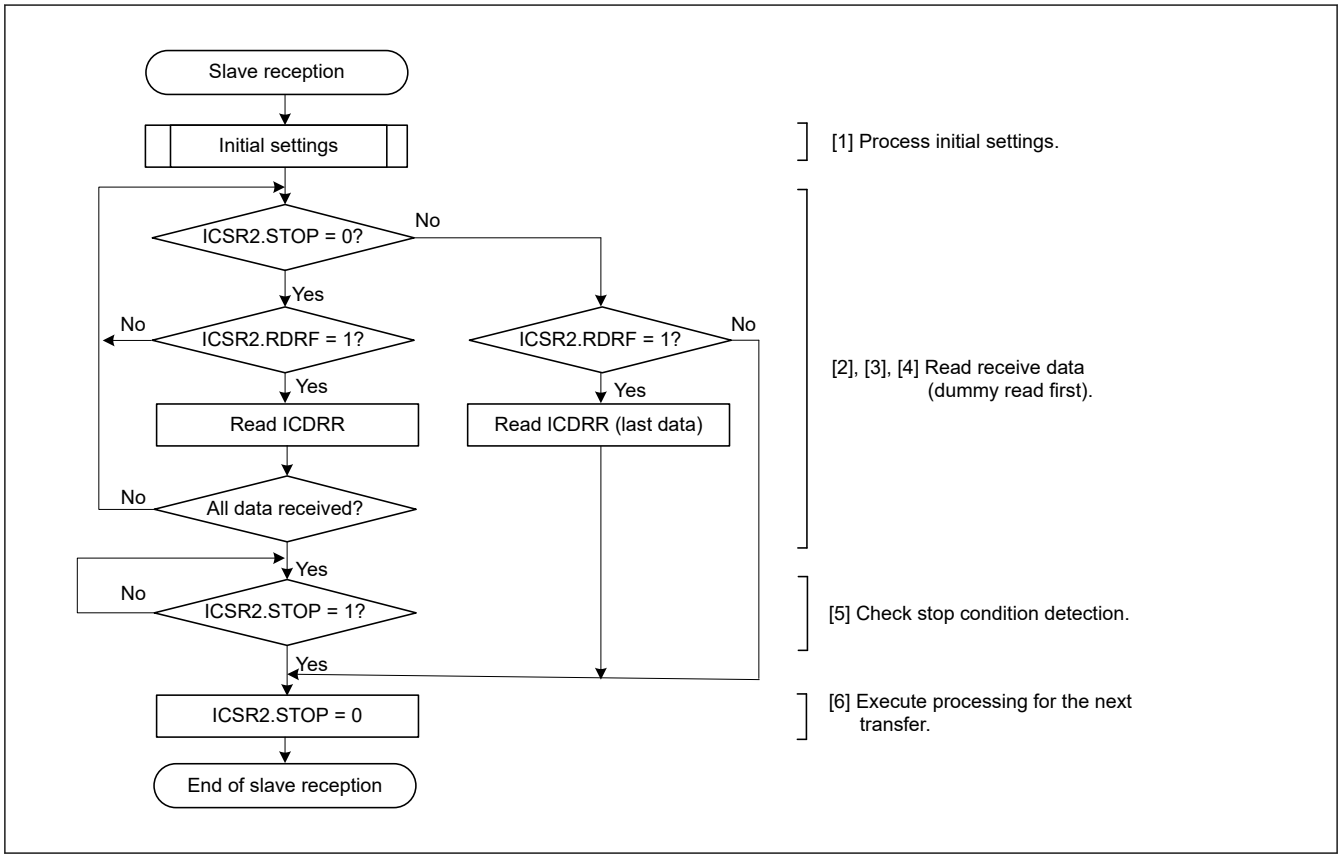


Figure 34.18 Example slave reception flow

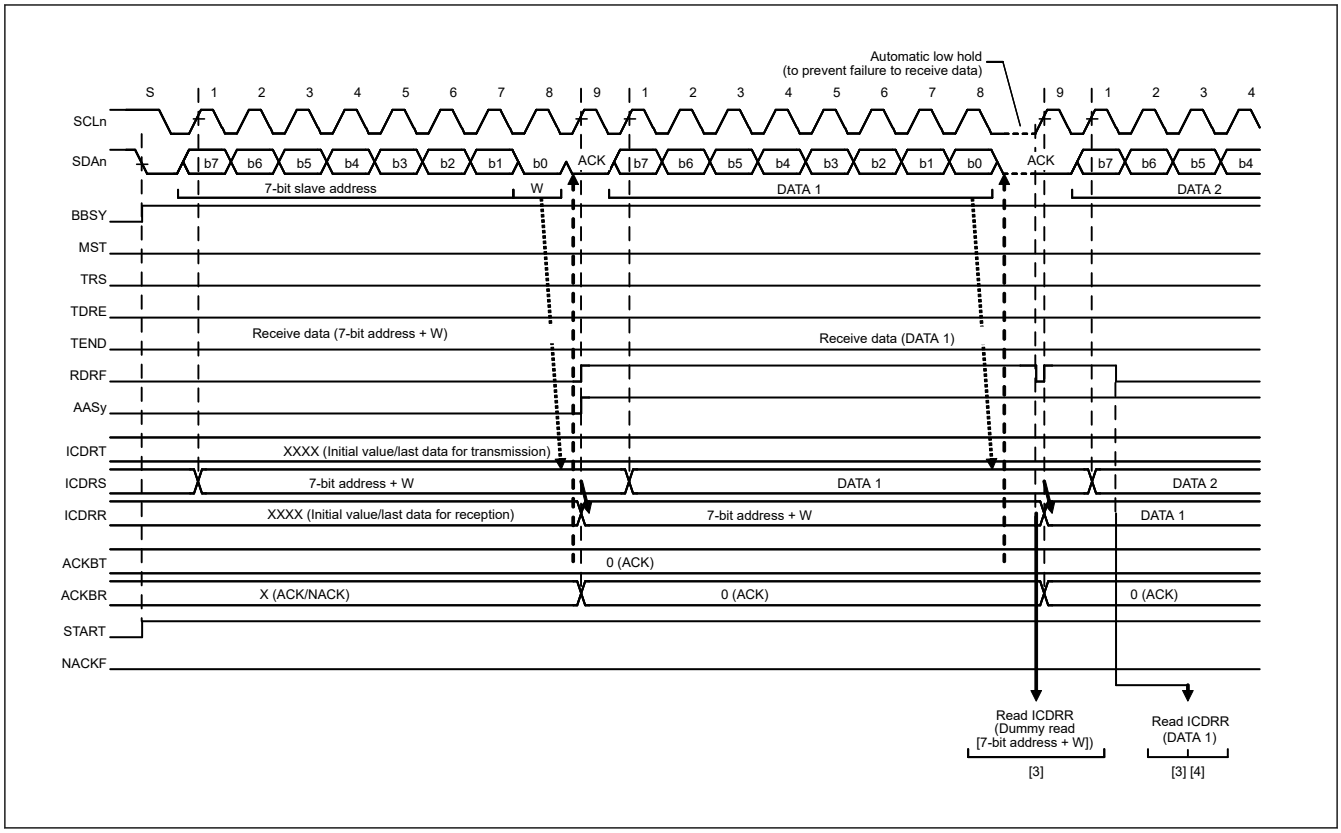


Figure 34.19 Slave receive operation timing (1) with 7-bit address format, when RDRFS = 0

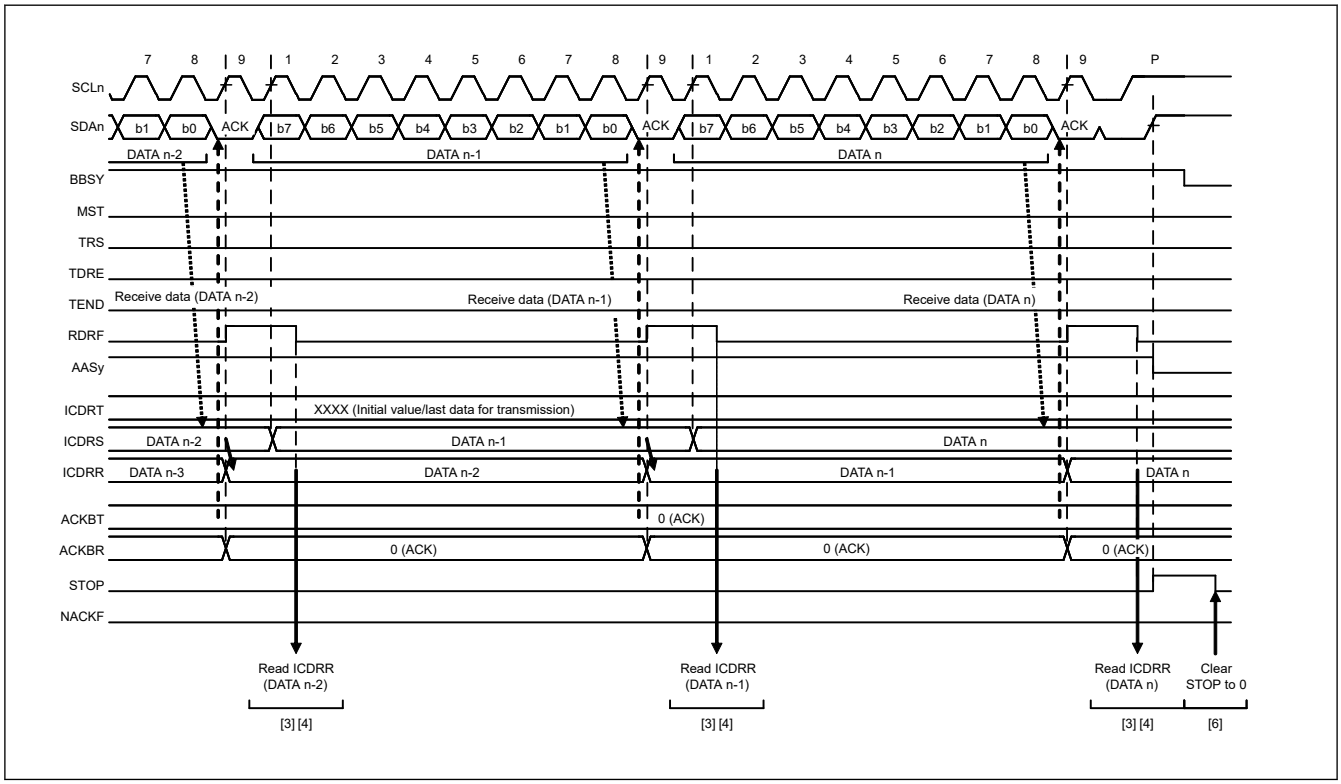


Figure 34.20 Slave receive operation timing (2) when RDRFS = 0

34.5 SCL Synchronization Circuit

For generation of the SCL clock, IIC starts counting the value for the high-level period specified in ICBRH when it detects a rising edge on the SCLn line, and drives the SCLn line low when it completes counting. When IIC detects the falling edge

of the SCLn line, it starts counting the value for the low-level period specified in ICBRL, and then stops driving the SCLn line, releasing the line when it completes counting. The IIC repeats this process to generate the SCL clock.

If multiple master devices are connected to the I^2C bus, a collision of SCL signals might arise because of contention with another master device. In such cases, the master devices must synchronize their SCL signals. Because this synchronization of SCL signals must be bit by bit, IIC is equipped with an SCL synchronization circuit to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When IIC detects a rising edge on the SCLn line and starts counting the high-level period specified in ICBRH, and the level on the SCLn line falls because an SCL signal is being generated by another master device, IIC performs the following:

1. Stops counting when it detects the falling edge.
2. Drives the level on the SCLn line low.
3. Starts counting the low-level period specified in ICBRL.

When the IIC finishes counting the low-level period, it stops driving the SCLn line low to release the line. If the lowlevel period of the SCL clock signal from the other master device is longer than the low-level period set in the IIC, the low-level period of the SCL signal is extended. When the low-level period for the other master device ends, the SCL signal rises because the SCLn line is released. When the IIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, when SCL signals from more than one master are contending, the high-level period of the SCL signal is synchronized with that of the clock with the narrower period, and the low-level period of the SCL signal is synchronized with that of the clock with the broader period. However, such synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.

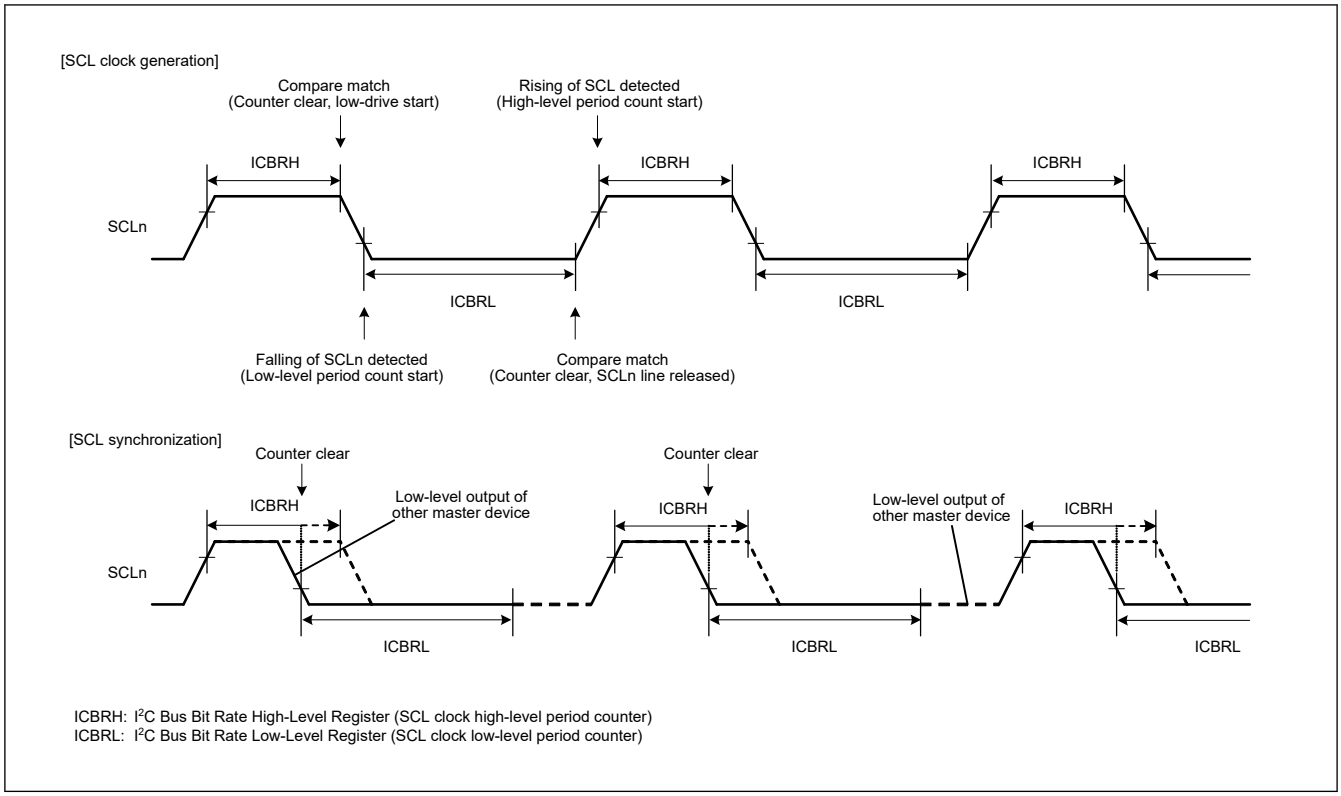


Figure 34.21 Generation and synchronization of SCL signal from IIC

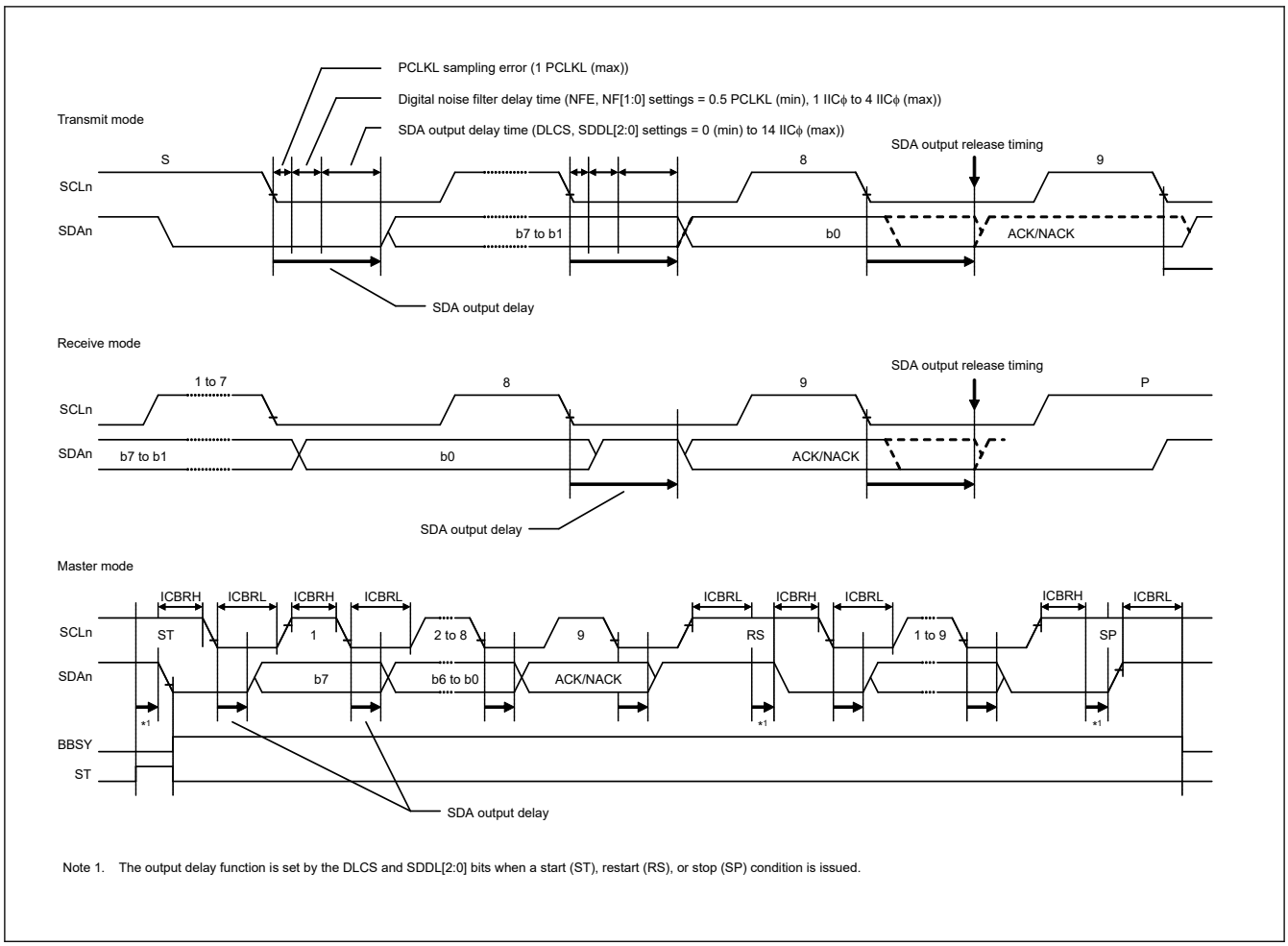
34.6 SDA Output Delay Function

The IIC module provides a function for delaying the output on the SDA line. The delay can be applied to all outputs on the SDA line, including the issuing of start, restart, and stop conditions, data, and the ACK and NACK signals.

With this function, SDA output is delayed from the detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval during which the SCL clock is low. This approach helps prevent erroneous operation of communications devices, with the aim of satisfying the 300 ns minimum data-hold time requirement of the SMBus

specification. The output delay function is enabled by setting the SDDL[2:0] bits in ICMR2 to any value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled, for example, the DLCS bit in ICMR2 selects the clock source for the SDA output delay counter, either as the internal base clock (IICΦ) for the IIC module or as the internal base clock divided by two (IICΦ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in ICMR2. When the delay count is reached, the IIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.



Note 1. The output delay function is set by the DLCS and SDDL[2:0] bits when a start (ST), restart (RS), or stop (SP) condition is issued.

Figure 34.22 SDA output delay function

34.7 Digital Noise Filter Circuits

The internal circuitry sees the states of the SCLn and SDAn pins through analog and digital noise-filter circuits. Figure 34.23 shows a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the IIC consists of four flip-flop circuit stages connected in series, and a match-detection circuit.

The number of valid stages in the digital noise filter is selected in the NF[1:0] bits in ICMR3. The selected number of valid stages determines the noise-filtering capability as a period from 1 to 4 IICΦ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on falling edges of the IICΦ signal. When the input signal level matches the output level of the number of valid flip-flop circuit stages as selected in the NF[1:0] bits in ICMR3, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is saved.

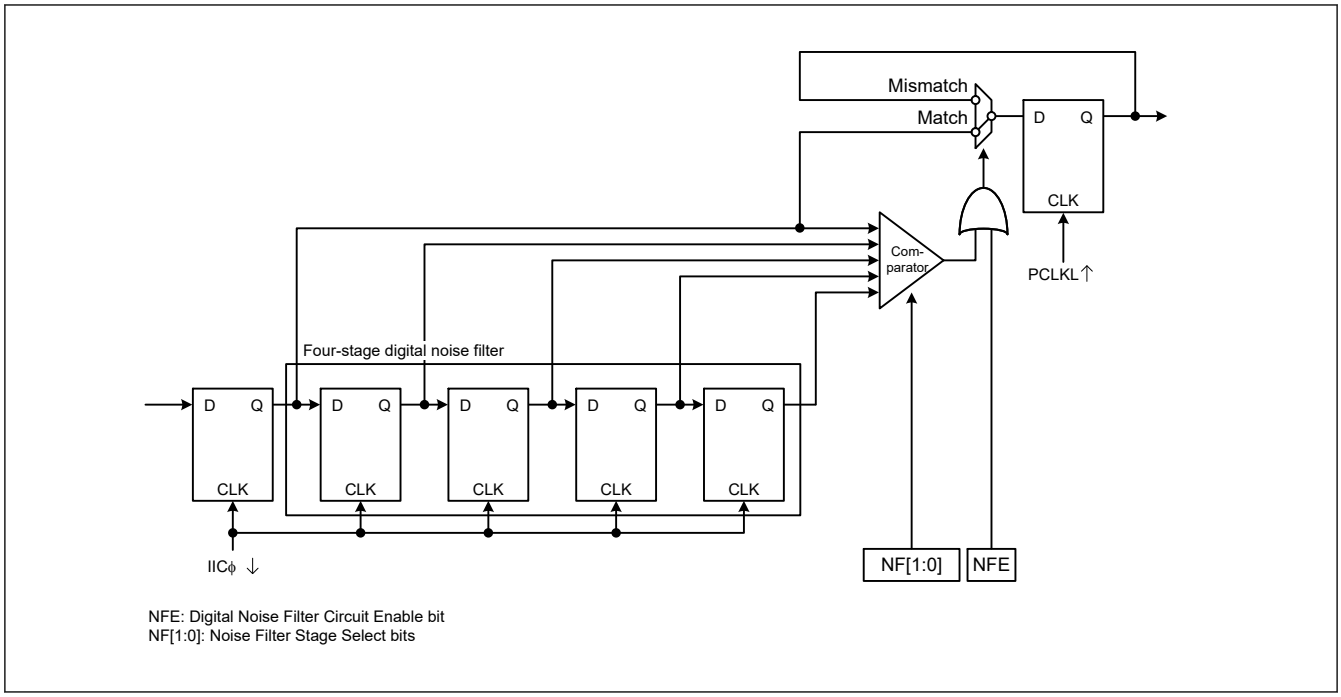


Figure 34.23 Digital noise filter circuit block diagram

34.8 Address Match Detection

IIC can set three unique slave addresses in addition to the general call address and host address. The slave addresses can be 7-bit or 10-bit slave addresses.

34.8.1 Slave-Address Match Detection

IIC can set three unique slave addresses and has a slave address detection function for each unique slave address. When the SARyE bit (y = 0 to 2) in ICSER is set to 1, the slave addresses set in SARUy and SARLy (y = 0 to 2) can be detected.

When the IIC detects a match of the set slave address, the associated AASy (y = 0 to 2) flag in ICSR1 is set to 1 on the rising edge of the 9th SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the subsequent R/W# bit. This causes a receive data full interrupt (IICn_RXI) or transmit data empty interrupt (IICn_TXI) to be generated. The AASy flag identifies which slave address is specified.

Figure 34.24 to Figure 34.26 show the AASy flag set timing in three cases.

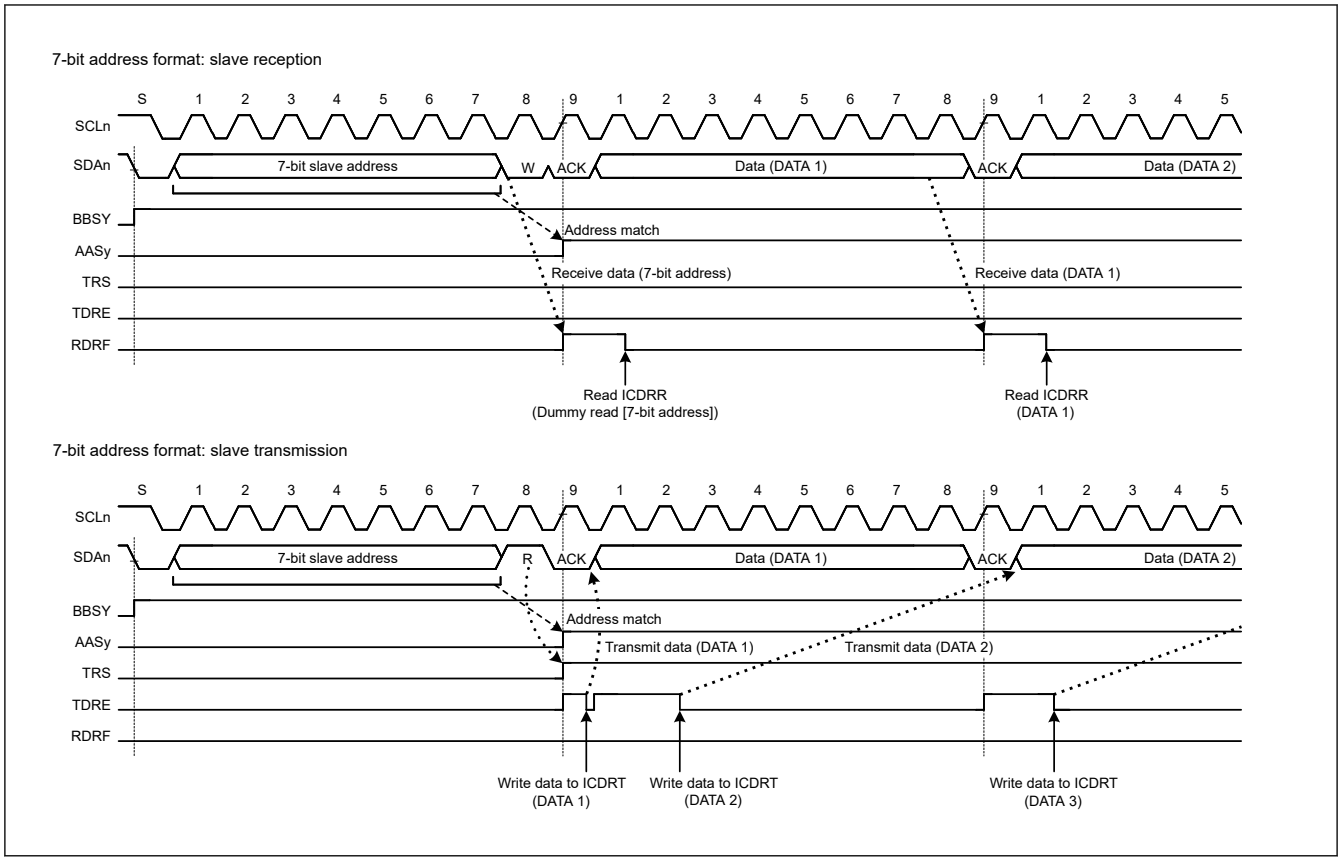


Figure 34.24 AASy flag set timing with 7-bit address format

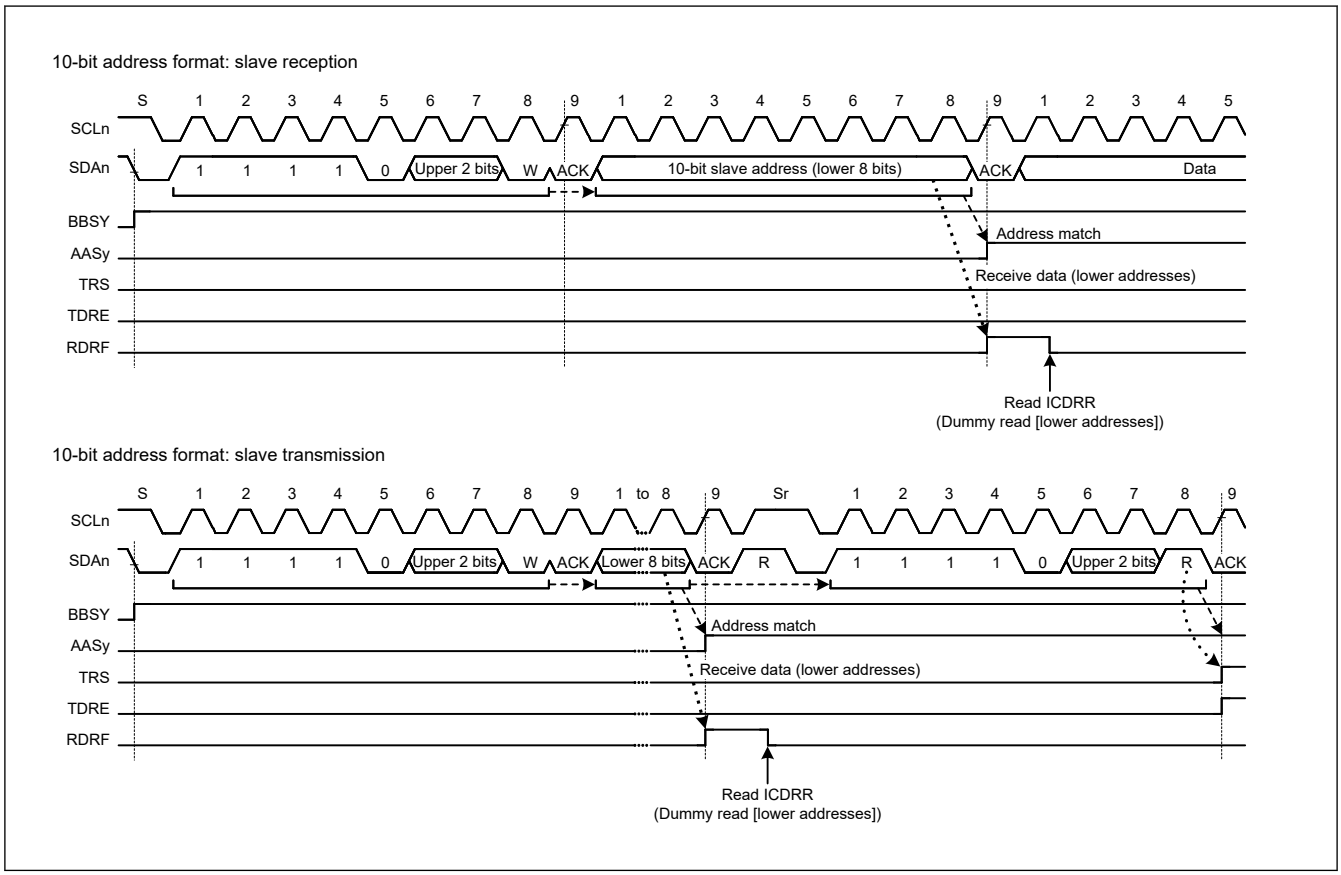


Figure 34.25 AASy flag set timing with 10-bit address format

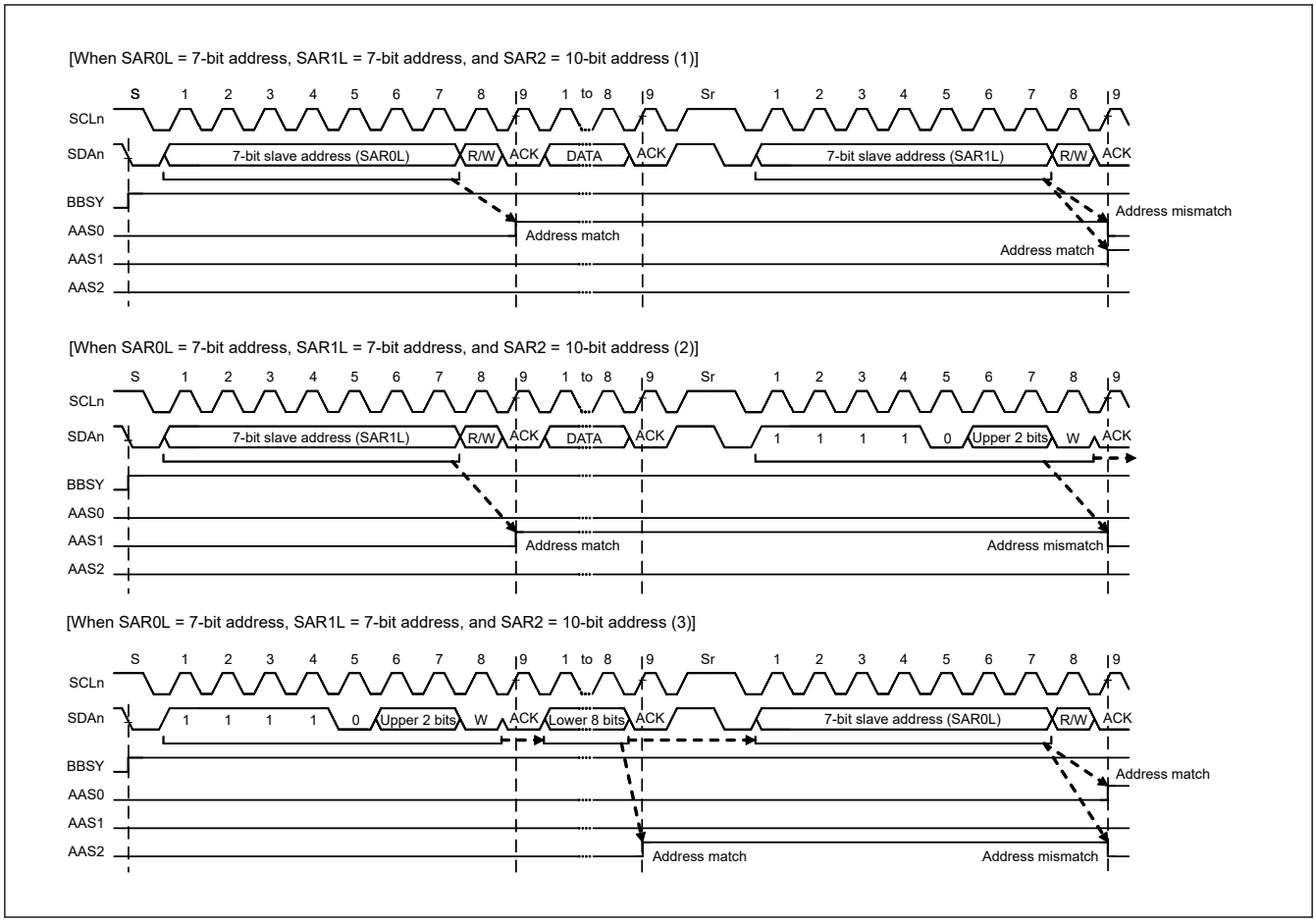


Figure 34.26 AASy flag set and clear timing with 7- and 10-bit address formats mixed

34.8.2 Detection of General Call Address

The IIC provides detection of the general call address (0000 000b + 0 [W]). General call address detection is enabled by setting the GCAE bit in ICSER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1 [R] (start byte), the IIC recognizes this as the address of a slave device with an all-zero address, but not as the general call address.

When the IIC detects the general call address, both the GCA flag in ICSR1 and the RDRF flag in ICSR2 set to 1 on the rising edge of the 9th cycle of the SCL clock. This leads to the generation of a receive data full interrupt (IICn_RXI). The value of the GCA flag can be checked to confirm that the general call address is transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

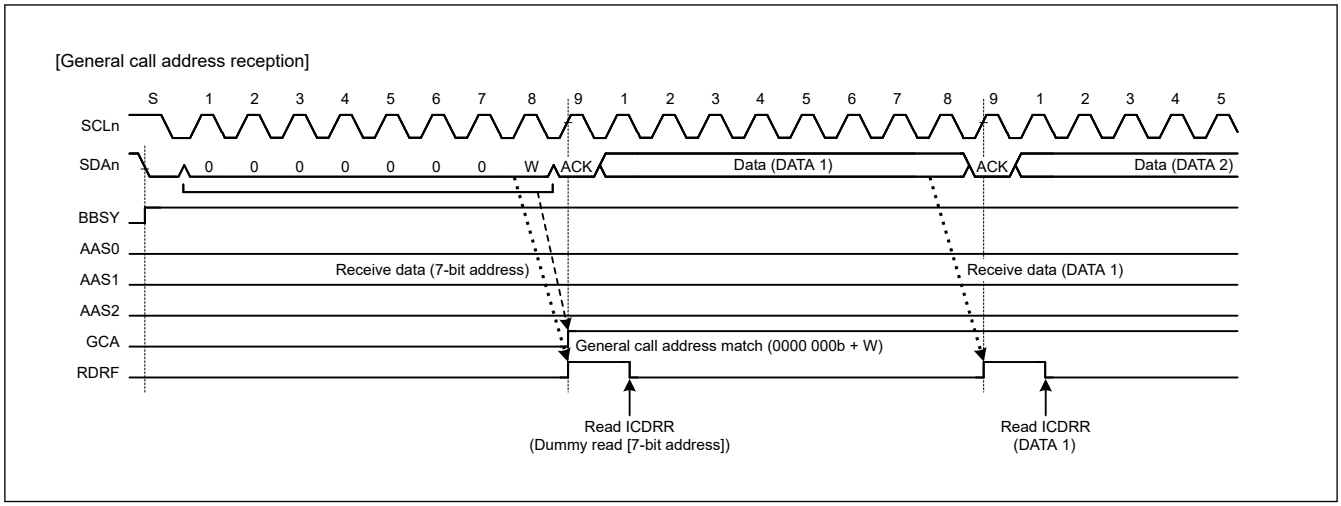


Figure 34.27 Timing of GCA flag setting during reception of general call address

34.8.3 Device ID Address Detection

The IIC module provides detection of the device ID address in compliance with the I^2C bus specification, revision 03.

When IIC receives 1111 100b as the first byte after a start or restart condition is issued with the DIDE bit in ICSER set to 1, it recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the 8th SCL clock cycle when the subsequent R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the IIC sets the associated AASy (y = 0 to 2) flag in ICSR1 to 1.

When the first byte received after the issue of a start or restart condition matches the device ID address (1111 100b) again and the subsequent R/W# bit is 1, IIC does not compare the second and subsequent bytes, and sets the ICSR2.TDRE flag to 1.

In the device ID address detection function, IIC sets the DID flag to 0 if a match with the IIC slave address is not obtained, or a match with the device ID address is not obtained after a match with the IIC slave address, and a restart condition is detected. If the first byte after the detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, IIC sets the DID flag to 1 and compares the second and subsequent bytes with the slave address of IIC. If the R/W# bit is 1, the DID flag holds the previous value and IIC does not compare the second and subsequent bytes. In this way, reception of a device ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Additionally, prepare the device ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device ID field as normal transmit data. For details on the information that must be included in device ID fields, contact NXP Semiconductors.

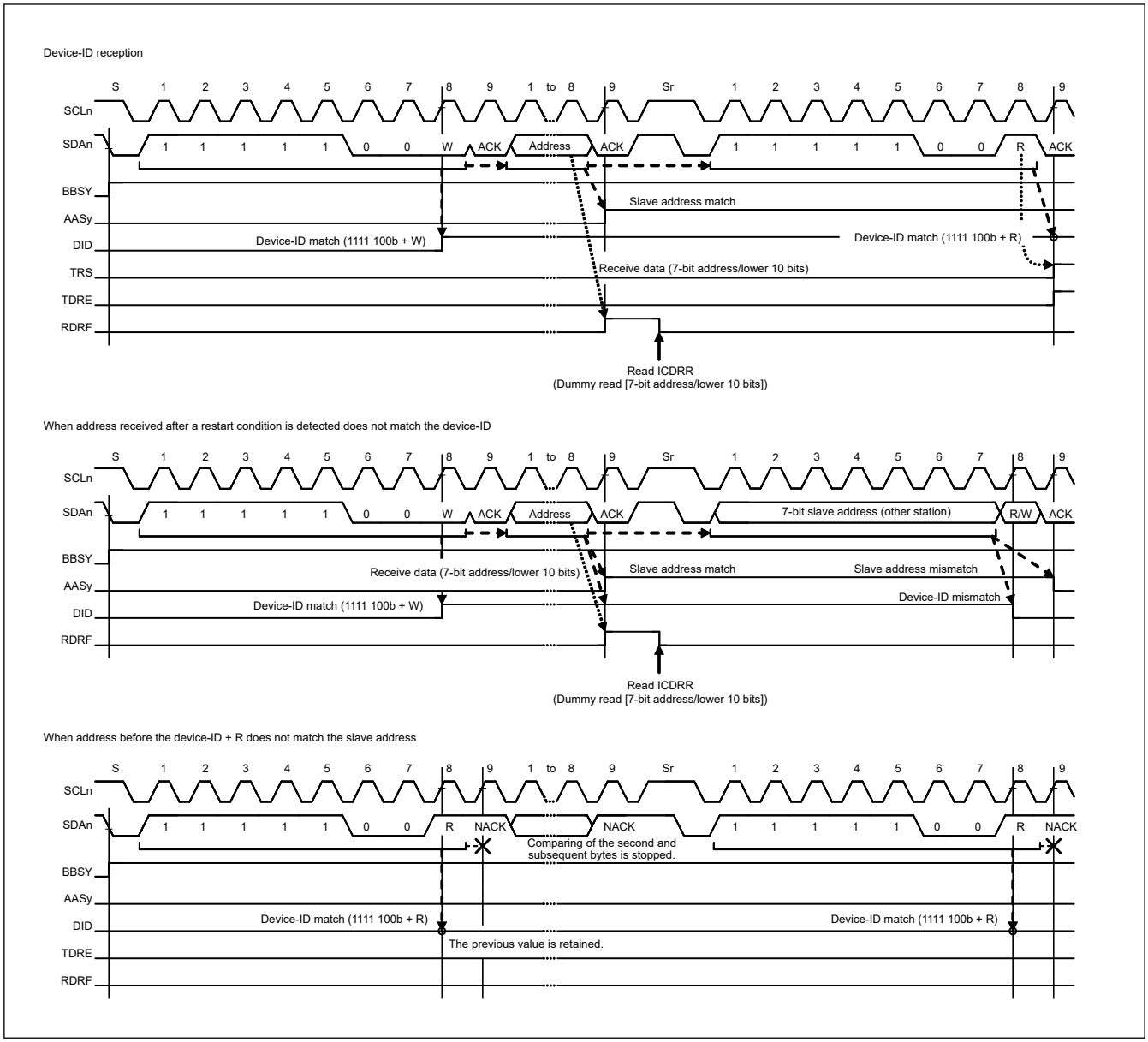


Figure 34.28 AASy and DID flag set and clear timing during reception of device ID

34.8.4 Host Address Detection

The IIC provides host address detection while the SMBus is operating. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the IIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the IIC detects the host address, the HOA flag in ICSR1 is set to 1 on the rising edge of the 9th SCL clock cycle. At the same time, the RDRF flag in ICSR2 is set to 1 when the R/W# bit is 0. This causes a receive data full interrupt (IICn_RXI) to be generated. The HOA flag indicates that the host address was sent from another device.

If the bit following the host address (0001 000b) is a read bit (R/W# bit = 1), the IIC can also detect the host address. After the host address is detected, the IIC operates in the same manner as in normal slave operation.

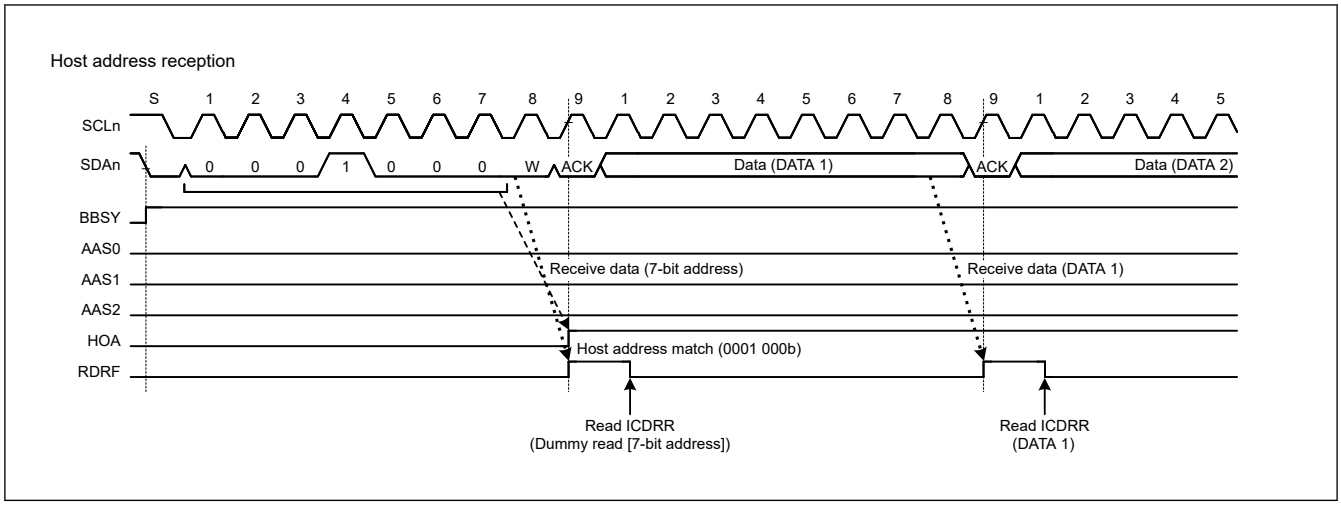


Figure 34.29 HOA flag set timing during reception of host address

34.9 Automatic Low-Hold Function for SCL

34.9.1 Function to Prevent Wrong Transmission of Transmit Data

If the I^{2}C Bus Shift Register (ICDRS) is empty when data has not been written to the I^{2}C Bus Transmit Data Register (ICDRT) with the IIC in transmission mode (TRS bit = 1 in ICCR2), the SCLn line is automatically held at the low level over subsequent intervals. This low-hold period is extended until the transmit data is written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low-level interval after a start or restart condition is issued
- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

Slave transmit mode

- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

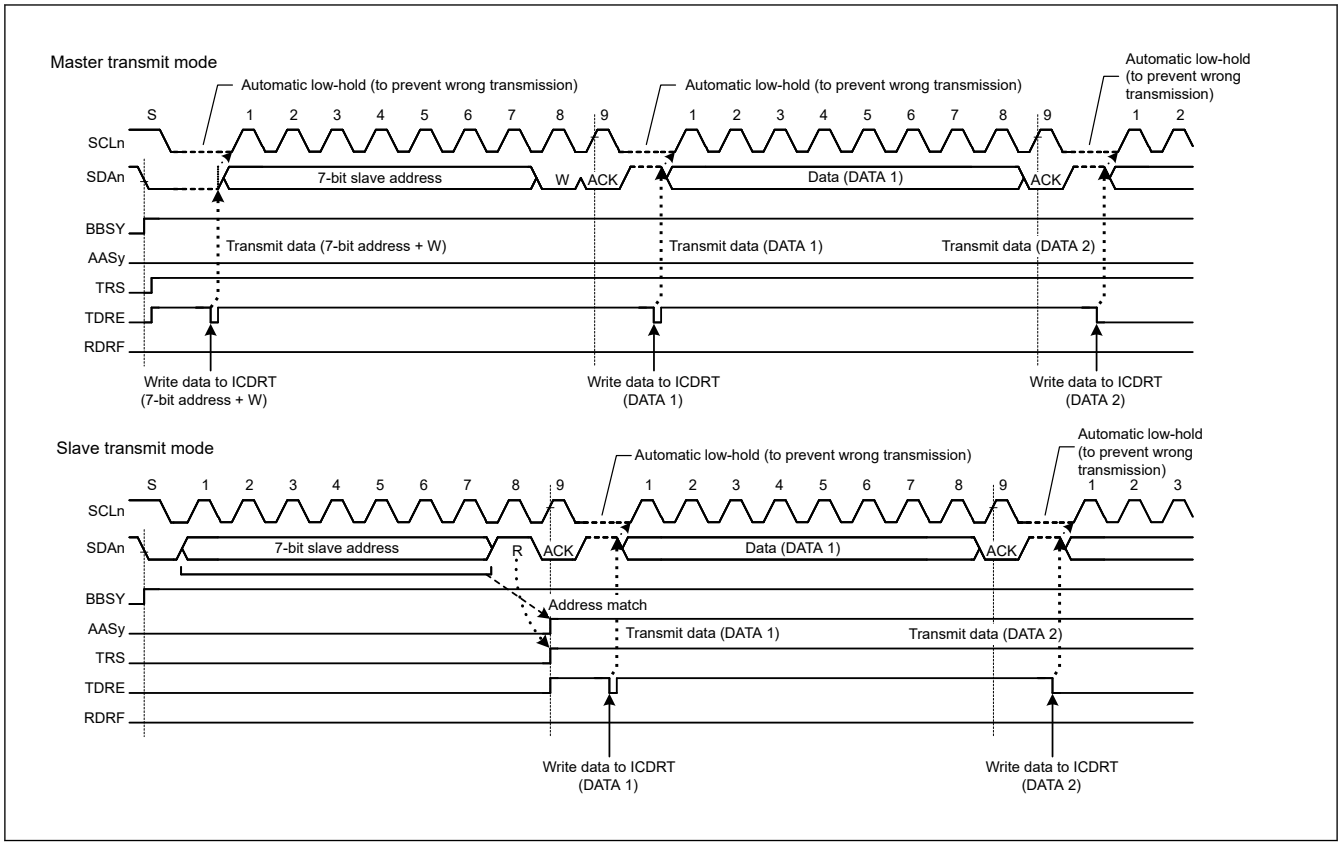


Figure 34.30 Automatic low-hold operation in transmit mode

34.9.2 NACK Reception Transfer Suspension Function

This function suspends transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). This function is enabled when the NACKE bit in ICFER is set to 1. If the next transmit data is already written (TDRE flag = 0 in ICSR2) when NACK is received, the next data transmission on the falling edge of the 9th SCL clock cycle is automatically suspended. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit and receive operations are discontinued. To restore transmit or receive operations, set the NACKF flag to 0. In master transmit mode, after a restart or stop condition is issued, set the NACKF flag to 0, and then issue a start condition again.

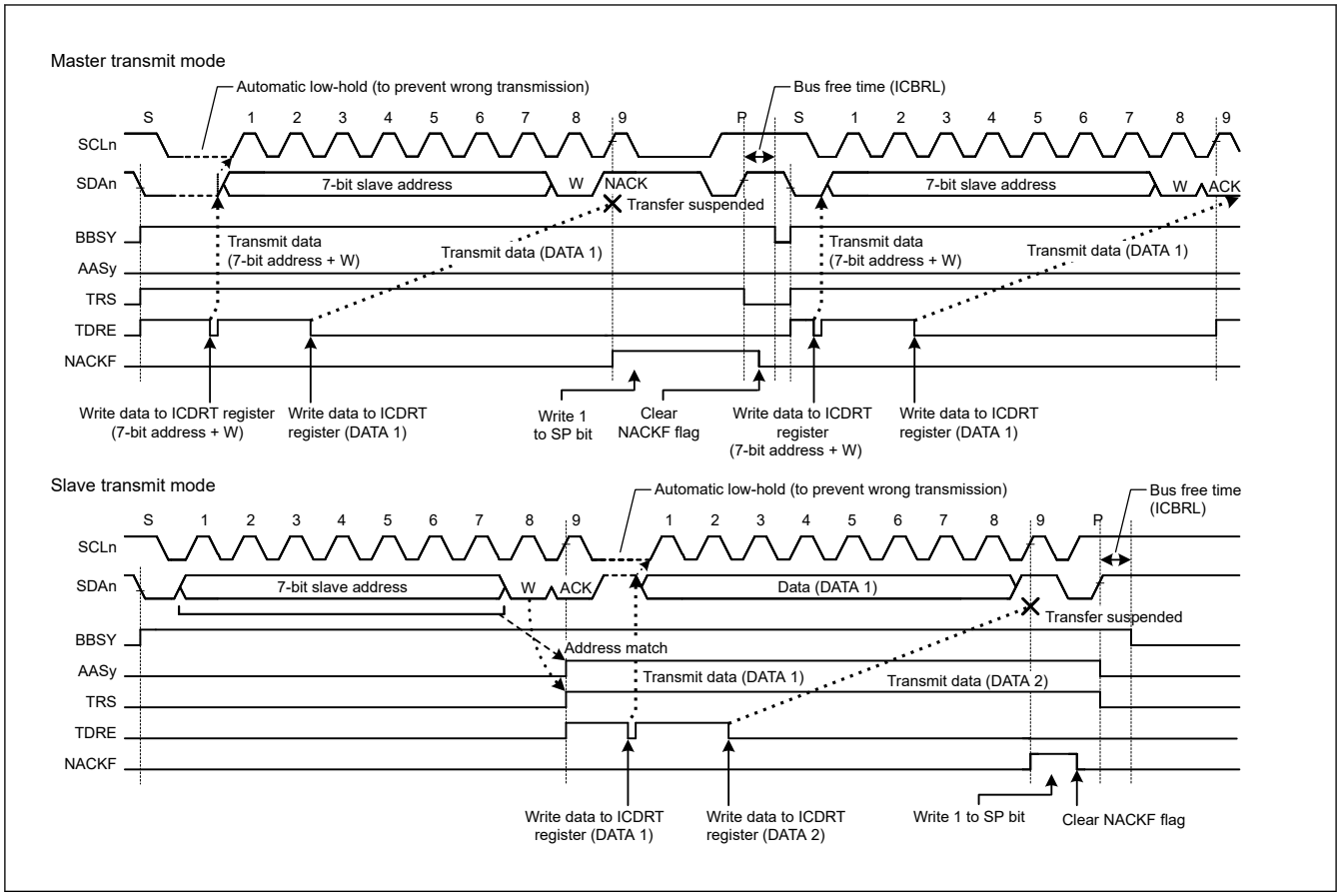


Figure 34.31 Suspension of data transfer when NACK is received, when NACKE = 1

34.9.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the IIC holds the SCLn line low automatically immediately before the next data is received to prevent failure to receive data.

This function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the IIC slave address is designated after a stop condition is issued. This function does not interfere with other communication because the IIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Periods in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) 1-byte receive operation and automatic low-hold function using the WAIT bit

When the WAIT bit in ICMR3 is set to 1, IIC performs a 1-byte receive operation using the WAIT bit function. Additionally, when the ICMR3.RDRFS bit is 0, IIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the 8th SCL clock cycle to the falling edge of the 9th SCL clock cycle, and automatically holds the SCLn line low on the falling edge of the 9th SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables byte-wise receive operation.

The WAIT bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

(2) 1-byte receive operation (ACK/NACK transmission control) and automatic low-hold function using the RDRFS bit

When the RDRFS bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the RDRFS bit function.

When the RDRFS bit is set to 1, the RDRF (receive data full) flag in ICSR2 is set to 1 on the rising edge of the 8th SCL clock cycle, and the SCLn line is automatically held low on the falling edge of the 8th SCL clock cycle. This low-hold

is released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation through the ACK or NACK transmission control based on the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

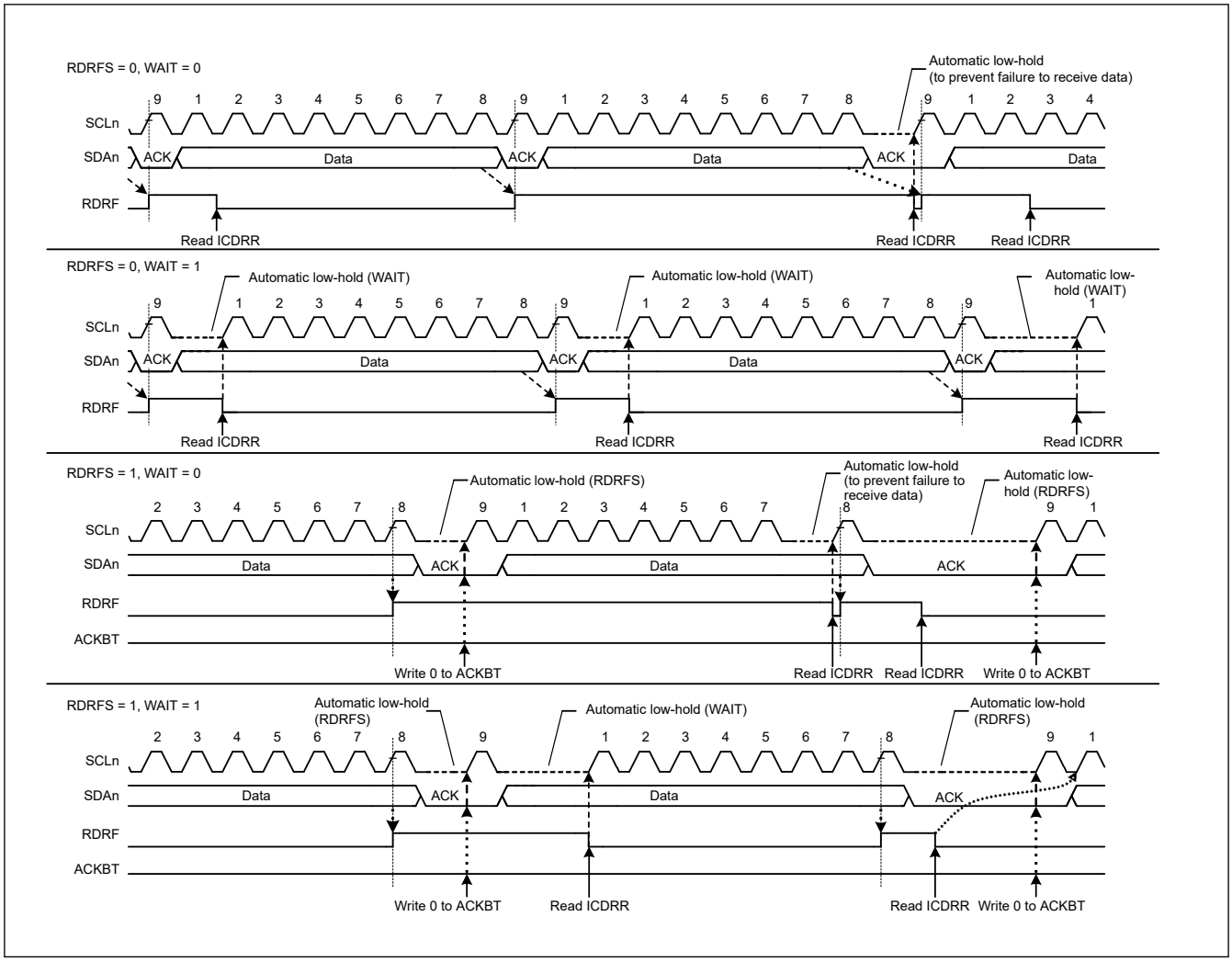


Figure 34.32 Automatic low-hold operation in receive mode using the RDRFS and WAIT bits

34.10 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I^2C bus standard, IIC provides functions to prevent double-issue of a start condition, detect arbitration-lost during the transmission of NACK, and detect arbitration lost in slave transmit mode.

34.10.1 Master Arbitration-Lost Detection (MALE Bit)

IIC drives the SDAn line low to issue a start condition. However, if the SDAn line was already driven low by another master device issuing a start condition, IIC regards its own start condition as an error and considers this a loss in arbitration.

Priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), IIC regards this as a double-issuing-of-start-condition error and considers itself to have lost the arbitration. This prevents a failure of transfer resulting from a start condition issued while transfer is in progress.

When a start condition is issued successfully, if the transmit data including the address bits (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state) and a low level is detected on the SDAn line, IIC loses the arbitration.

After a loss in arbitration of mastership, the IIC immediately enters slave receive mode. If a slave address, including the general call address, matches its own address at this time, the IIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the MALE bit in ICFER is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Mismatching of the internal level for output on SDA and the level on the SDAn line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 is set to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDAn line in master transmit mode (MST and TRS bits = 11b in ICCR2).

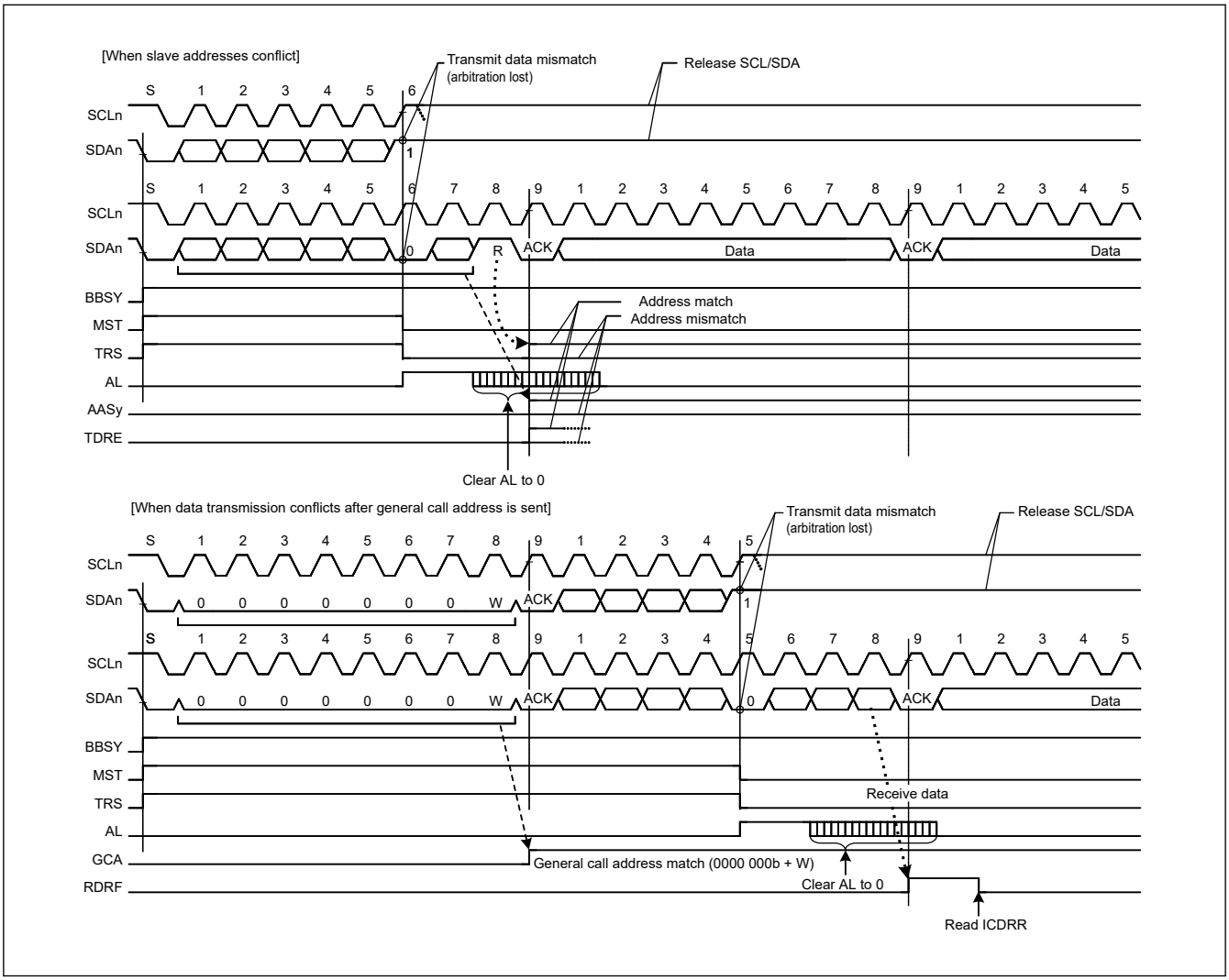


Figure 34.33 Examples of master arbitration-lost detection when MALE = 1

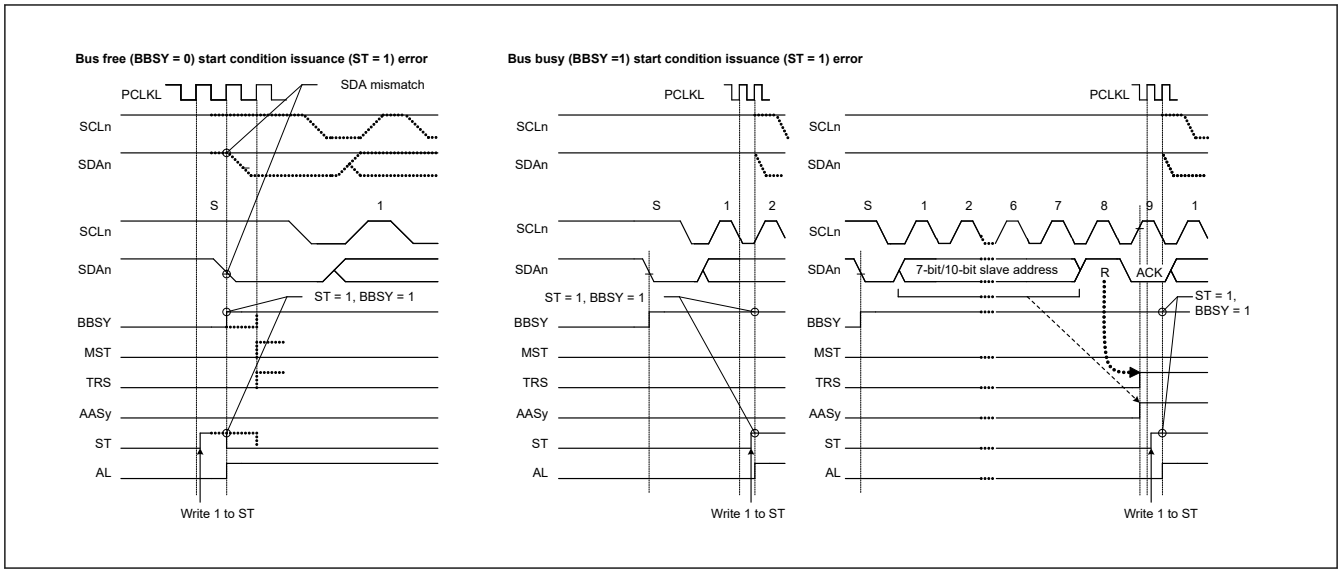


Figure 34.34 Arbitration-lost when start condition is issued when MALE = 1

34.10.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

This function causes arbitration to be lost if the internal SDA output level does not match the level on the SDAn line during transmission of NACK in receive mode. Arbitration is lost because of a conflict between NACK and ACK transmissions when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send or receive the same information through a single slave device. Figure 34.35 shows an example of arbitration-lost detection during the transmission of NACK.

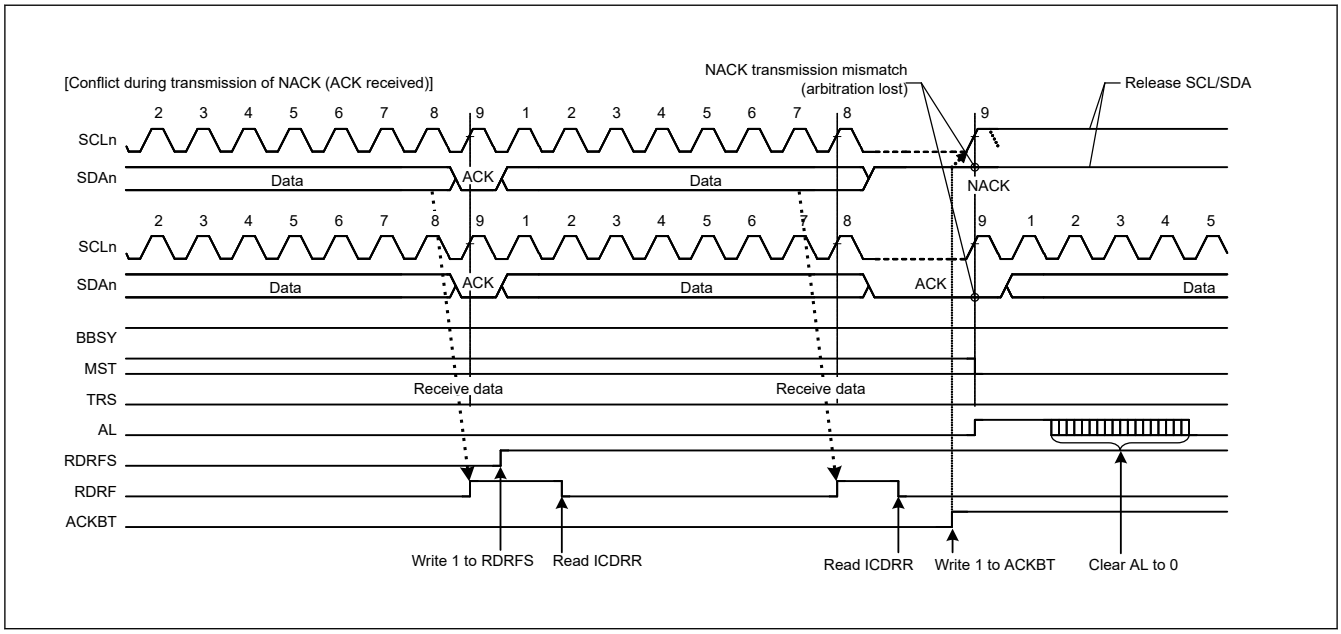


Figure 34.35 Example of arbitration-lost detection during transmission of NACK when NALE = 1

The following description explains arbitration-lost detection using an example in which two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in either master A or B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Master A sends NACK when it has received the 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received the required 4 bytes of data.

The NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect the ACK transmitted by master B and issues a stop condition. The stop condition issue conflicts with the SCL clock output of master B, which disrupts communication.

When IIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost. If arbitration is lost during transmission of NACK, IIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing, such as 0xFF transmission processing, which is required if the UDID (Unique Device Identifier) of the assigned address does not match in the Get UDID general processing after the Assign Address command.

The IIC detects arbitration-lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDAn line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3).

34.10.3 Slave Arbitration-Lost Detection (SALE Bit)

This function causes arbitration to be lost if the transmit data (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state), and the low level is detected on the SDAn line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When IIC loses slave arbitration, IIC is immediately released from the slave-matched state and enters slave receive mode.

This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates subsequent redundant processing for the transmission of 0xFF.

IIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDAn line in slave transmit mode (MST and TRS bits = 01b in ICCR2).

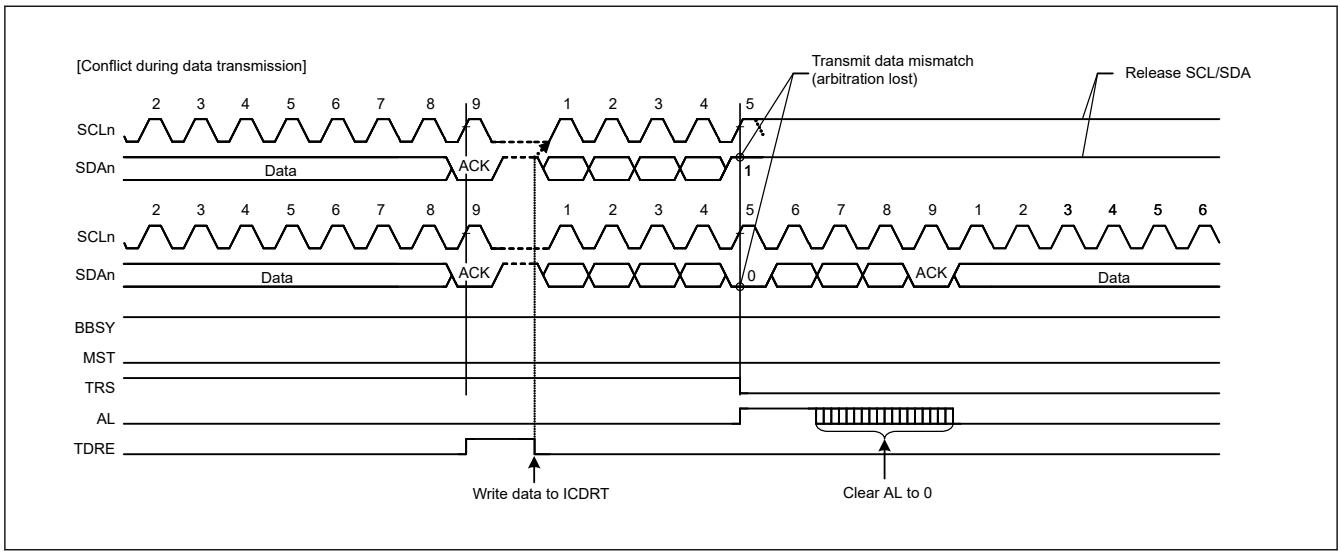


Figure 34.36 Example of slave arbitration-lost detection when SALE = 1

34.11 Start, Restart, and Stop Condition Issuing Function

34.11.1 Issuing a Start Condition

IIC issues a start condition when the ST bit in ICCR2 is set to 1. When the ST bit is set to 1, a start condition request is made, and IIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, IIC automatically shifts to the master transmit mode.

To issue a start condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the time set in ICBRH and the start condition hold time elapse.
3. Drive the SCLn line low (high level to low level).
4. Detect low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

34.11.2 Issuing a Restart Condition

IIC issues a restart condition when the RS bit in ICCR2 is set to 1.

When the RS bit is set to 1, a restart condition request is made, and IIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a restart condition:

1. Release the SDAn line.
2. Ensure that the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRL and the restart condition setup time elapse.
5. Drive the SDAn line low (high level to low level).
6. Ensure the time set in ICBRH and the restart condition hold time elapse.
7. Drive the SCLn line low (high level to low level).
8. Detect a low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

Note: When issuing restart condition requests, write the slave address to ICDRT after confirming that ICCR2.RS = 0. Data written while ICCR2.RS = 1 is not forwarded because of the retransmission condition before the occurrence.

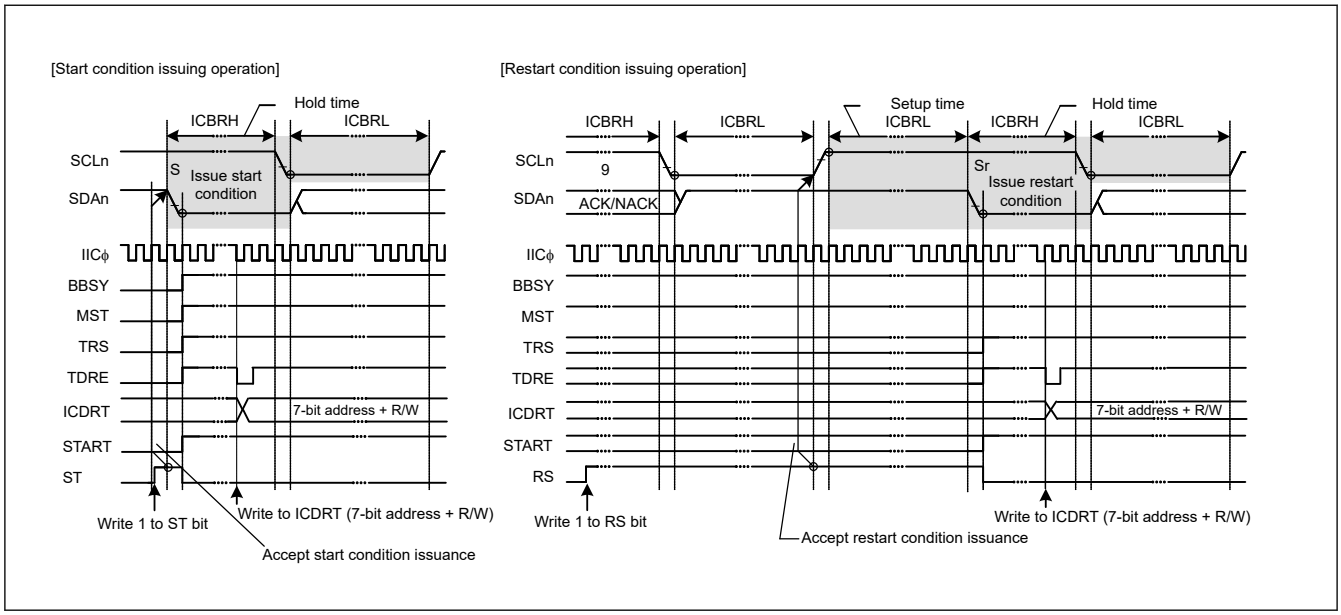


Figure 34.37 Start and restart condition issue timing using the ST and RS bits

Figure 34.38 shows the operation timing when a restart condition is issued after the master transmission.

To issue a restart condition after the master transmission:

1. Initialize the IIC using the details provided in section 34.4.2. Initial Settings.
2. Read the BBSY flag in IICR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition issuance request). On receiving the request, IIC issues a start condition. At the same time, the BBSY flag and the START flag in ICSR2 are automatically set to 1 and the ST bit automatically sets to 0. If the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, IIC recognizes that a start condition is successfully issued as requested by the ST bit. The MST and TRS bits in ICCR2 automatically set to 1, placing IIC in master transmit mode. The TDRE flag in ICSR2 also automatically sets to 1 when the TRS bit is set to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. After the transmit data is written to ICDRT, the TDRE flag automatically sets to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again sets to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode according to the value of the transmitted R/W# bit. If the value of the R/W# bit is 0, IIC continues in master transmit mode. If the ICSR2.NACKF flag is 1 at this time, indicating that no slave device recognized the address or that there was an error in communications, write 1 to ICCR2.SP bit to issue a stop condition. To transmit data with an address in the 10-bit format, start by writing 11110b, the 2 upper bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the 8 lower bits of the slave address to ICDRT.
4. After confirming that the TDRE flag in ICSR2 is 1, write the data for transmission to the ICDRT register. IIC automatically holds the SCLn line low until the data for transmission is ready, and a restart or stop condition is issued.
5. After all bytes of data for transmission are written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1. Then, after checking that the START flag in ICSR2 is 1, set the START flag in ICSR2 to 0.
6. Set the RS bit in ICCR2 to 1 (restart condition issuance request). Upon receiving the request, IIC issues a restart condition.
7. After checking that the START flag in ICSR2 is 1, write the value for transmission (the slave address and the R/W# bit) to ICDRT.

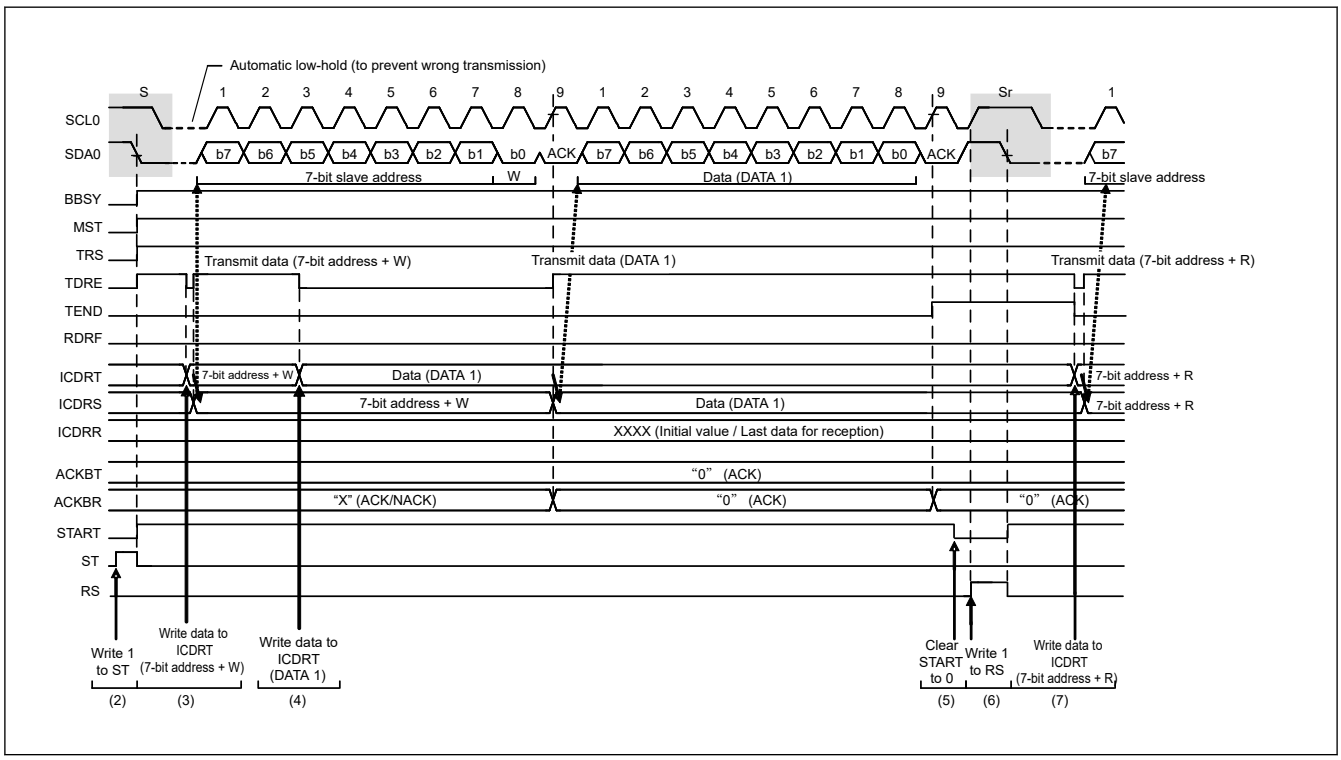


Figure 34.38 Restart condition issue timing after master transmission

34.11.3 Issuing a Stop Condition

IIC issues a stop condition when the SP bit in ICCR2 is set to 1. When the SP bit is set to 1, a stop condition request is made, and the IIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a stop condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure that the time set in ICBRH and the stop condition setup time elapse.
5. Release the SDAn line (low level to high level).
6. Ensure the time set in ICBRL and the bus free time elapse.
7. Clear the BBSY flag to 0 to release the bus mastership.

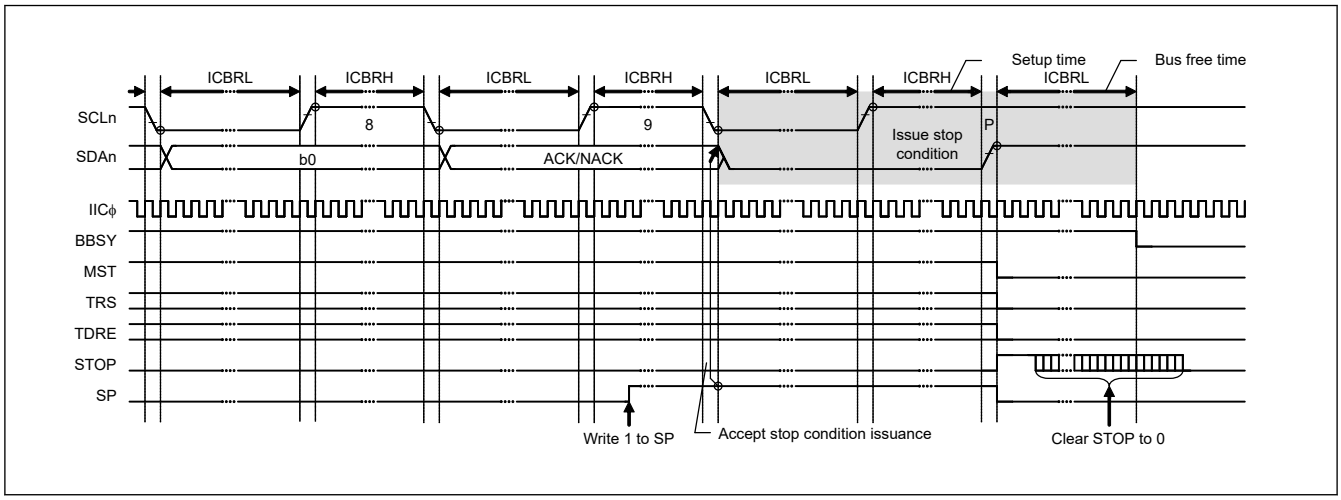


Figure 34.39 Stop condition issue timing using the SP bit

34.12 Bus Hanging

If the clock signals from the master and slave devices are out of synchronization because of noise or other factors, the I^2C bus might hang with a fixed level on the SCLn line or SDAn line.

To manage bus hanging, the IIC has the following:

- A timeout function to detect hanging by monitoring the SCLn line
- A function for the output of an extra SCL clock cycle to release the bus from a hung state because of clock signals being out of synchronization
- An IIC reset function
- An internal reset function

By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the IIC or its communicating partner is placing the low level on the SCLn or SDAn lines.

34.12.1 Timeout Function

The timeout function can detect when the SCLn line is stuck longer than the predetermined time. IIC can detect an abnormal bus state by monitoring that the SCLn line is stuck low or high for a predetermined time.

The timeout function monitors the SCLn line state and counts the low-level or high-level period using the internal counter.

The timeout function resets the internal counter each time the SCLn line changes (rises or falls), but continues to count unless the SCLn line changes. If the internal counter overflows because no SCLn line changes, IIC can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state when the SCLn line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1)
- IIC slave address is detected (ICSR1 register is not 0x00) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0)
- The bus is free (ICCR2.BBSY flag is 0) while a start condition is requested (ICCR2.ST bit is 1)

The internal counter of the timeout function uses the internal reference clock (IICΦ) set in the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCLn line level (low, high, or both levels) during which this counter is activated can be selected in the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are set to 0, the internal counter is disabled.

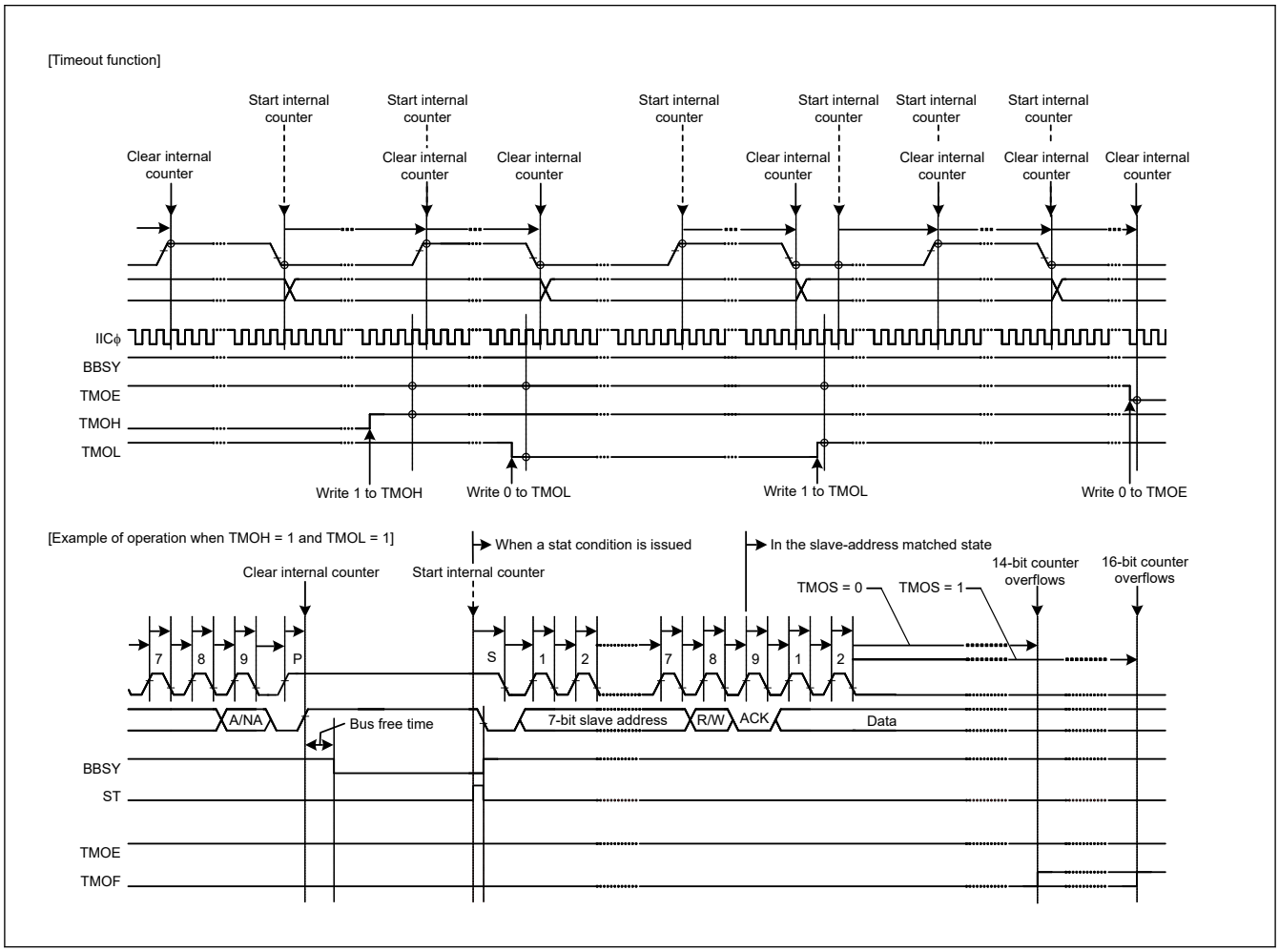


Figure 34.40 Timeout function using the TMOE, TMOS, TMOH, and TMOL bits

34.12.2 Extra SCL Clock Cycle Output Function

In master mode, this function outputs extra SCL clock cycles to release the SDAn line of the slave device from being held at the low level because the master is out of synchronization with the slave device.

This function is mainly used in master mode to release the SDAn line of the slave device from being fixed low by including extra cycles of SCL output from IIC. It uses single cycles of the SCL clock for a bus error where IIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this function in normal situations.

Using it when communications are proceeding correctly leads to malfunctioning.

When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the transfer rate specified in the CKS[2:0] bits in ICMR1, and in the ICBRH and ICBRL registers, is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically set to 0. More extra clock cycles can be output consecutively by writing 1 to the CLO bit through software after having read CLO = 0.

When the IIC module is in master mode and the slave device is holding the SDAn line at the low level because synchronization with the slave device is lost because of effects like noise, the output of a stop condition is not possible.

This function can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held at the low level, and recover the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming the release of the SDAn line by the slave device, complete the communication by reissuing the stop condition.

Use this function with the MALE bit in ICFER set to 0 (master arbitration-lost detection disabled). If the MALE bit is set to 1 (enabled), arbitration is lost when the value of the SDAO bit in ICCR1 does not match the state of the SDAn line.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is free (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)
- When the communication device does not hold the SCLn line low.

Figure 34.41 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

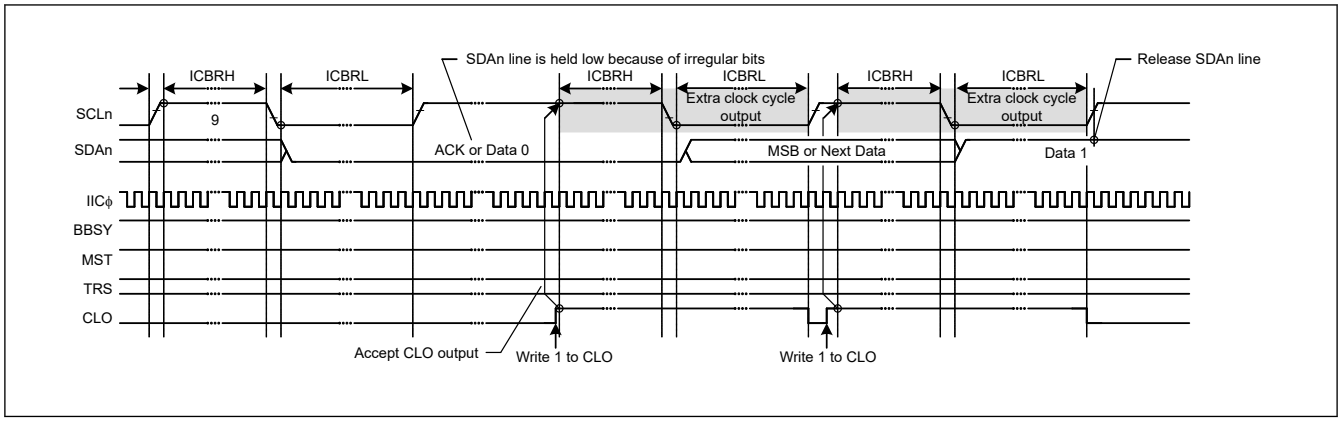


Figure 34.41 Extra SCL clock cycle output function using the CLO bit

34.12.3 IIC Reset and Internal Reset

The IIC module has two types of resets:

- IIC reset, which initializes all registers, including the BBSY flag in ICCR2
- Internal reset, which releases the IIC from the slave-address matched state and initializes the internal counter while saving other settings.

After issuing a reset, always set the IICRST bit in ICCR1 to 0.

Both types of resets are valid for release from bus-hung states, because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing a reset during slave operation might lead to a loss of synchronization between the master device clock and the slave device clock, therefore, avoid this when possible. In addition, monitoring of the bus state, such as for the presence of a start condition, is not possible during an IIC reset (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the IIC and internal resets, see section 34.15. Resets, Registers, and Function States When Issuing Each Condition.

34.13 SMBus Operation

IIC supports data communication conforming to the SMBus Specification, version 2.0. To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, ICBRH, and ICBRL. In addition, specify the values in the DLCS bit in ICMR2 and the

SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. When the IIC is used only as a slave device, the transfer rate setting is not required, but ICBRL must be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (SARL0, SARL1, and SARL2), and set the associated FS bit (7- or 10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

34.13.1 SMBus Timeout Measurement

(1) Measuring slave device timeout

The following period (timeout interval: $T_{LOW:\ SEXT}$) must be measured for slave devices in SMBus communication:

- From start condition to stop condition

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the internal timer using the IIC start condition detection interrupt (STIn) and stop condition detection interrupt (SPIn). The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:\ SEXT}$: 25 ms (maximum) of the SMBus standard.

If the time measured with the internal timer exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the IIC. When an internal reset is issued, the IIC stops driving the bus for the SCLn and SDAn pins and makes the SCLn/SDAn pin output high-impedance, which releases the bus.

(2) Measuring master device timeout

The following periods (timeout interval: $T_{LOW:\ MEXT}$) must be measured for master devices in SMBus communication:

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition

To measure timeout for master devices, measure these periods with the internal timer using the IIC start condition detection interrupt (STIn), stop condition detection interrupt (SPIn), transmit end interrupt (IICn_TEI), or receive data full interrupt (IICn_RXI). The measured timeout period must be within the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (maximum) of the SMBus standard, and the total of all $T_{LOW:\ MEXT}$ values from the start condition to stop condition must be within $T_{LOW:\ SEXT}$: 25 ms (maximum).

For the ACK receive timing (rising edge of the 9th SCL clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). Perform byte-wise transmit operations in master transmit mode, and hold the RDRFS bit in ICMR3 at 0 until the byte immediately before the reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

If the period measured with internal timer exceeds the total clock low-level extended period (master device), $T_{LOW:\ MEXT}$: 10 ms (maximum) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout, $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (stop writing data to ICDRT).

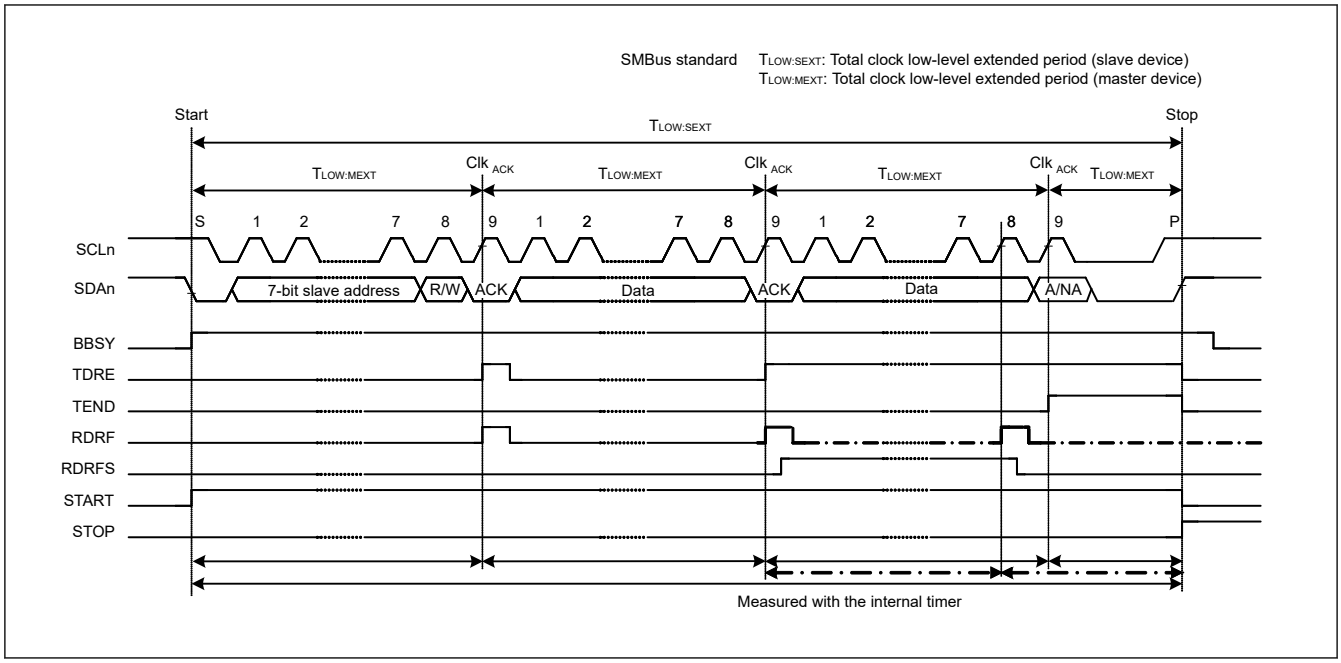


Figure 34.42 SMBus timeout measurement

34.13.2 Packet Error Code (PEC)

The LSI provides a CRC calculator, which enables transmission of a packet error code (PEC) or allows checking the received data in SMBus data communication for the IIC. For the CRC-generating polynomials of the CRC calculator, see section 38, CRC Operation Unit (CRC).

In master transmit mode, the PEC data can be generated by writing all transmit data to the CRC Data Input register (CRCDIR) in the CRC calculator.

In master receive mode, the PEC data can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC Data Output Register (CRCDOR) with the received PEC data.

To send ACK or NACK based on the match or mismatch result when the final byte is received as a result of the PEC code check, set the RDRFS bit in ICMR3 to 1 before the rising edge of the 8th SCL clock cycle during reception of the final byte, and hold the SCLn line low on the falling edge of the 8th clock cycle.

34.13.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For a product using the LSI to operate as an SMBus host or ARP master, the host address (0001 000b) sent from the slave device must be detected as a slave address, and so the IIC has a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in ICSER to 1. Operation after the host address is detected is the same as normal slave operation.

34.14 Interrupt Sources

The IIC issues four types of interrupt requests:

- Transfer error or event occurrence (detection of arbitration-lost, NACK, timeout, start or restart condition, or stop condition)
- Receive data full
- Transmit data empty
- Transmit end

Table 34.11 lists details about the interrupt requests. The receive data full and transmit data empty interrupts can activate data transfer by the DMAC.

Table 34.11 Interrupt sources

Symbol	Interrupt source	Interrupt flag	DMAC activation	Interrupt condition
IICn_EEI*4	Transfer error or event generation	AL	Not possible	AL = 1, ALIE = 1
		NACKF		NACKF = 1, NAKIE = 1
		TMOF		TMOF = 1, TMOIE = 1
		START		START = 1, STIE = 1
		STOP		STOP = 1, SPIE = 1
IICn_RXI*2 *4	Receive data full	RDRF	Possible	RDRF = 1, RIE = 1
IICn_TXI*1 *4	Transmit data empty	TDRE	Possible	TDRE = 1, TIE = 1
IICn_TEI*3 *4	Transmit end	TEND	Not possible	TEND = 1, TEIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. When an interrupt flag is cleared or masked, read the relevant flag again to check whether clearing or masking is complete, and then return from interrupt handling. Not doing so creates the possibility of repeated processing of the same interrupt.
Note 1. Because IICn_TXI is an edge-detected interrupt, it does not require clearing. Additionally, the TDRE flag in ICSR2 (condition for IICn_TXI) is automatically set to 0 when transmit data is written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).
Note 2. Because IICn_RXI is an edge-detected interrupt, it does not require clearing. Additionally, the RDRF flag in ICSR2 (condition for IICn_RXI) automatically is set to 0 when data is read from ICDRR.
Note 3. When using the IICn_TEI interrupt, clear the TEND flag in ICSR2 in the IICn_TEI interrupt handling. The TEND flag in ICSR2 automatically is set to 0 when transmit data is written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).
Note 4. Channel number (n = 0 to 2).

Clear or mask each flag during interrupt handling.

34.15 Resets, Registers, and Function States When Issuing Each Condition

IIC provides chip reset, IIC reset, and internal reset functions. Table 34.12 lists the IIC resets, registers, and function states when issuing each condition.

Table 34.12 Resets, registers, and function states when issuing each condition (1 of 2)

Registers		Chip reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICCR1	ICE, IICRST	Reset	Saved	Saved	Saved	Saved
	SCLO, SDAO		Reset	Reset		
	Others			Saved		
ICCR2	BBSY	Reset	Reset	Saved	Set	Saved
	ST			Reset	Saved	Saved
	TRS, MST				Set or saved	Reset
	Others				Reset	Reset or Saved
ICMR1	BC[2:0]	Reset	Reset	Reset	Reset	Saved
	Others			Saved	Saved	
ICMR2		Reset	Reset	Saved	Saved	Saved
ICMR3		Reset	Reset	Saved	Saved	Saved
ICFER		Reset	Reset	Saved	Saved	Saved
ICSER		Reset	Reset	Saved	Saved	Saved
ICIER		Reset	Reset	Saved	Saved	Saved
ICSR1		Reset	Reset	Reset	Saved	Reset

Table 34.12 Resets, registers, and function states when issuing each condition (2 of 2)

Registers		Chip reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICSR2	TDRE, TEND	Reset	Reset	Reset	Saved	Reset
	START				Set	
	STOP				Saved	Set
	Others				Saved	Saved
SARL0, SARL1, SARL2, SARU0, SARU1, SARU2		Reset	Reset	Saved	Saved	Saved
ICBRH, ICBRL		Reset	Reset	Saved	Saved	Saved
ICDRT		Reset	Reset	Saved	Saved	Saved
ICDRR		Reset	Reset	Saved	Saved	Saved
ICDRS		Reset	Reset	Reset	Saved	Saved
Timeout function		Reset	Reset	Operating	Operating	Operating
Bus free time measurement		Reset	Reset	Operating	Operating	Operating

34.16 Usage Notes

34.16.1 Settings for the Module-Stop Function

IIC operation can be disabled or enabled using Module Stop Control Register G (MSTPCRG01). IIC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 9, Low-Power Consumption Function.

35. CAN-FD Interface (CANFD)

35.1 Overview

The LSI has a 2-channel CAN-FD module (CANFD) that complies with ISO 11898-1 (2015) Standards. CANFD transmits and receives both formats of messages, namely the standard identifier (11 bits) (identifier is hereafter referred to as ID) and extended ID (29 bits). Table 35.1 lists the specifications of the CANFD, Figure 35.1 shows a block diagram of the CANFD, and Table 35.2 lists the I/O pins.

Table 35.1 CANFD specifications (1 of 2)

Parameter	Description
Number of channels	Two channels
Protocol	CAN-FD ISO 11898-1 (2015)
Communication speed	• Classical CAN mode: 1 Mbps • CAN FD mode Nominal bit rate: max. 1 Mbps Data bit rate: max. 8 Mbps
Buffer	Total 192 buffers (when frame size is 76 bytes) • Individual buffers: 64 buffers (32 buffers × 2 channels) Transmit buffer: 32 buffers per channel Transmit queue: 4 queues per channel (shared with transmit buffer, up to 16 buffers allocatable) • Shared buffers: 128 buffers for all channels Receive buffer: 0 to 32 buffers Receive FIFO buffer: 8 FIFO buffers (up to 128 buffers allocatable to each) Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 128 buffers allocatable to each)
Reception function	• Receives data frames and remote frames • Selects ID format (standard ID, extended ID, or both IDs) to be received • Sets interrupt enable/disable for each FIFO • Mirror function (reception of messages transmitted from the CAN node itself) • Timestamp function (to record message reception time as a 16-bit timer value)
Reception filter function	• Selects receive message according to 256 receive rules • Sets the number of receive rules (0 to 255) for each channel • Acceptance filter processing: Sets ID and mask for each receive rule • DLC filter processing: Enables DLC filter check for each acceptance rule
Receive message transfer function	• Routing function Transfers receive messages to arbitrary destinations (can be transferred to up to 8 buffers) Transfer destination: Receive buffer, receive FIFO buffer, and/or transmit/receive FIFO buffer • Label addition function Stores label information together with a message in a receive buffer and FIFO buffer
Transmission function	• Transmits data frames and remote frames • Selects ID format (standard ID, extended ID, or both IDs) to be transmitted • Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer • Selects ID priority transmission or transmit buffer number priority transmission • Transmit request can be aborted (possible to confirm with a flag) • One-shot transmission function
Interval transmission function	Transmit message at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)
Transmit ques function	Transmits all stored messages according to the ID priority
Transmit history function	Stores the history information of transmission-completed messages Adds timestamp (recording message transmission time as a 16-bit timer value) to the history information
Gateway function	Transmits a received message automatically

Table 35.1 CANFD specifications (2 of 2)

Parameter	Description
Bus off recovery mode selection	Selects the method for returning from bus-off state. ● ISO 11898-1 compliant ● Automatic entry to channel halt mode at bus-off entry ● Automatic entry to channel halt mode at bus-off end ● Transition to channel halt mode by program request ● Transition to the error-active state by program request (forcible return from the bus-off state)
Error status monitoring	● Monitors CAN protocol errors (stuff error, from error, ACK error, CRC error, bit error, ACK delimiter error, and bus dominant lock) ● Defects error status transitions (error warning, error passive, bus-off entry, and bus-off recovery) ● Reads the error counter ● Monitors DLC errors
Interrupt source	8 interrupt sources ● Global interrupts (2 sources) RX FIFO interrupt Global error interrupt ● Channel interrupts (3 sources/channel) Channel transmit interrupt Channel error interrupt Common RX FIFO interrupt 10 DMA requests ● 8 for RX FIFO ● 2 for the first common FIFO (1 source/channel)
CAN clock source	Selects the PCLKM or the PCLKCAN (40 MHz or 80 MHz)
Test function	Test function for user evaluation ● Listen-only mode ● Self-test mode 0 (external loopback) ● Self-test mode 1 (internal loopback) ● Restricted operation mode ● RAM test (read/write test) ● Inter-channel communication test (CRC error test enabled)
Low-power consumption function	Module-stop state can be set to reduce power consumption

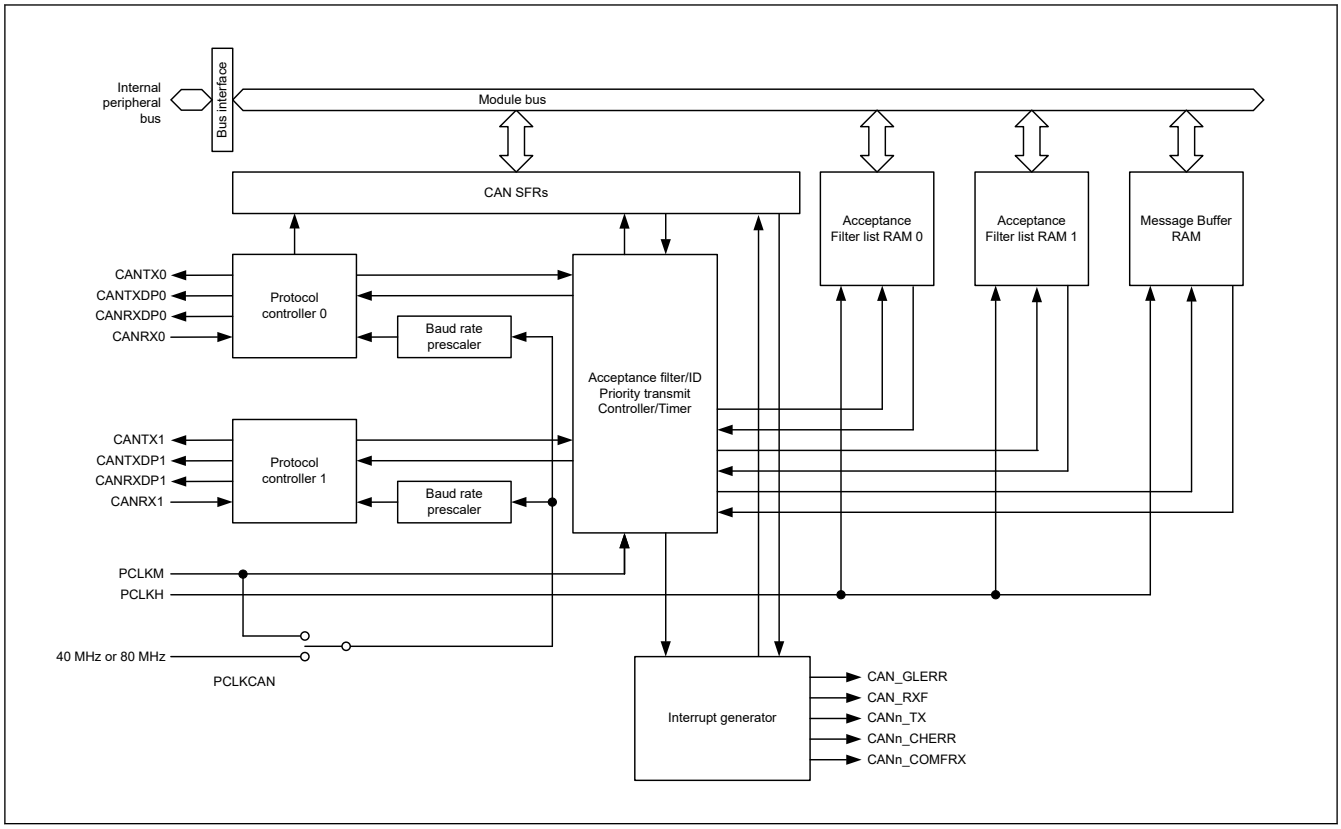


Figure 35.1 Block diagram of CANFD

Table 35.2 lists the input/output pins of the CANFD module.

Table 35.2 Pin configuration of CANFD

Channel	Pin name	I/O	Function
CAN0	CANRX0	Input	CAN0 receive data input
	CANTX0	Output	CAN0 transmit data output
	CANRXDP0	Output	CAN0 receive data phase output
	CANTXDP0	Output	CAN0 transmit data phase output
CAN1	CANRX1	Input	CAN1 receive data input
	CANTX1	Output	CAN1 transmit data output
	CANRXDP1	Output	CAN1 receive data phase output
	CANTXDP1	Output	CAN1 transmit data phase output

Table 35.3 CANFD interrupt sources

Name	Interrupt sources
CAN_RXF	RX FIFO interrupt
CAN_GLERR	Global error interrupt
CAN_RF_DMAREQm	RX FIFO m DMA request
CANn_TX	Channel n TX interrupt
CANn_CHERR	Channel n CAN error interrupt
CANn_COMFRX	Channel n Common RX FIFO or TXQ interrupt
CANn_CF_DMAREQ	Channel n First common FIFO DMA request

Note: m = 0 to 7, n = 0, 1

35.2 Register Map

Table 35.4 CANFD register map (1 of 3)

Address	Register symbol	Register name	Write protection
0x8004_0000 + 0x10 × n	CFDCnNCFG	Channel n Nominal Bit Rate Configuration Register (n = 0, 1)	—
0x8004_0004 + 0x10 × n	CFDCnCTR	Channel n Control Register (n = 0, 1)	—
0x8004_0008 + 0x10 × n	CFDCnSTS	Channel n Status Register (n = 0, 1)	—
0x8004_000C + 0x10 × n	CFDCnERFL	Channel n Error Flag Register (n = 0, 1)	—
0x8004_1400 + 0x20 × n	CFDCnDCFG	Channel n Data Bit Rate Configuration Register (n = 0, 1)	—
0x8004_1404 + 0x20 × n	CFDCnFDCFG	Channel n CAN-FD Configuration Register (n = 0, 1)	—
0x8004_1408 + 0x20 × n	CFDCnFDCTR	Channel n CAN-FD Control Register (n = 0, 1)	—
0x8004_140C + 0x20 × n	CFDCnFDSTS	Channel n CAN-FD Status Register (n = 0, 1)	—
0x8004_1410 + 0x20 × n	CFDCnFDCRC	Channel n CAN-FD CRC Register (n = 0, 1)	—
0x8004_0080	CFDGIPV	Global IP Version Register	—
0x8004_0084	CFDGCFG	Global Configuration Register	—
0x8004_0088	CFDGCTR	Global Control Register	—
0x8004_008C	CFDGSTS	Global Status Register	—
0x8004_0090	CFDGERFL	Global Error Flag Register	—
0x8004_1300	CFDGTINTSTS0	Global TX Interrupt Status Register 0	—
0x8004_0094	CFDGTSC	Global Timestamp Counter Register	—
0x8004_0098	CFDGAFLECTR	Global Acceptance Filter List Entry Control Register	—
0x8004_009C	CFDGAFLCFG0	Global Acceptance Filter List Configuration Register 0	—
0x8004_1800 + 0x10 × (n - 1)	CFDGAFLIDn	Global Acceptance Filter List ID Register n (n = 1 to 16)	—
0x8004_1804 + 0x10 × (n - 1)	CFDGAFLMn	Global Acceptance Filter List Mask Register n (n = 1 to 16)	—
0x8004_1808 + 0x10 × (n - 1)	CFDGAFLP0n	Global Acceptance Filter List Pointer 0 Register n (n = 1 to 16)	—
0x8004_180C + 0x10 × (n - 1)	CFDGAFLP1n	Global Acceptance Filter List Pointer 1 Register n (n = 1 to 16)	—
0x8004_00AC	CFDRMNB	RX Message Buffer Number Register	—
0x8004_00B0	CFDRMND0	RX Message Buffer New Data Register 0	—
0x8004_00C0 + 0x04 × n	CFDRFCCn	RX FIFO Configuration/Control Register n (n = 0 to 7)	—
0x8004_00E0 + 0x04 × n	CFDRFSTSn	RX FIFO Status Register n (n = 0 to 7)	—
0x8004_0100 + 0x04 × n	CFDRFPCTRn	RX FIFO Pointer Control Register n (n = 0 to 7)	—
0x8004_0120 + 0x04 × n	CFDCFCCn	Common FIFO Configuration/Control Register n (n = 0 to 5)	—
0x8004_0180 + 0x04 × n	CFDCFCCEn	Common FIFO Configuration/Control Enhancement Register n (n = 0 to 5)	—
0x8004_01E0 + 0x04 × n	CFDCFSTSn	Common FIFO Status Register n (n = 0 to 5)	—
0x8004_0240 + 0x04 × n	CFDCFPCTRn	Common FIFO Pointer Control Register n (n = 0 to 5)	—
0x8004_02A0	CFDFESTS	FIFO Empty Status Register	—
0x8004_02A4	CFDFFSTS	FIFO Full Status Register	—
0x8004_02A8	CFDFMSTS	FIFO Message Lost Status Register	—
0x8004_02AC	CFDRFISTS	RX FIFO Interrupt Flag Status Register	—
0x8004_02B0	CFDCFRISTS	Common FIFO RX Interrupt Flag Status Register	—
0x8004_02B4	CFDCFTISTS	Common FIFO TX Interrupt Flag Status Register	—
0x8004_02C4	CFDFFFSTS	FIFO FDC Full Status Register	—
0x8004_02C0	CFDCFMOWSTS	Common FIFO Message Overwrite Status Register	—

Table 35.4 CANFD register map (2 of 3)

Address	Register symbol	Register name	Write protection
0x8004_02B8	CFDCFOFRISTS	Common FIFO One Frame RX Interrupt Flag Status Register	—
0x8004_02BC	CFDCFOFTISTS	Common FIFO One Frame TX Interrupt Flag Status Register	—
0x8004_1330	CFDCDTCT	DMA Transfer Control Register	—
0x8004_1334	CFDCDTSTS	DMA Transfer Status Register	—
0x8004_1340	CFDCDTTCT	DMA TX Transfer Control Register	—
0x8004_1344	CFDCDTTSTS	DMA TX Transfer Status Register	—
0x8004_1350 + 0x04 × n	CFDGRINTSTSn	Global RX Interrupt Status Register n (n = 0, 1)	—
0x8004_02D0 + 0x01 × n	CFDTMCn	TX Message Buffer Control Register n (n = 0 to 127)	—
0x8004_07D0 + 0x01 × n	CFDTMSTSn	TX Message Buffer Status Register n (n = 0 to 127)	—
0x8004_0CD0 + 0x04 × f	CFDTMTRSTSf	TX Message Buffer Transmission Request Status Register f (f = 0 to 3)	—
0x8004_0D70 + 0x04 × f	CFDTMTARSTSf	TX Message Buffer Transmission Abort Request Status Register f (f = 0 to 3)	—
0x8004_0E10 + 0x04 × f	CFDTMTCSTSf	TX Message Buffer Transmission Completion Status Register f (f = 0 to 3)	—
0x8004_0EB0 + 0x04 × f	CFDTMTASTSf	TX Message Buffer Transmission Abort Status Register f (f = 0 to 3)	—
0x8004_0F50 + 0x04 × f	CFDTMIECf	TX Message Buffer Transmission Interrupt Enable Register f (f = 0 to 3)	—
0x8004_1000 + 0x04 × n	CFDTXQCC0n	TX Queue Configuration/Control Register 0n (n = 0, 1)	—
0x8004_1060 + 0x04 × n	CFDTXQCC1n	TX Queue Configuration/Control Register 1n (n = 0, 1)	—
0x8004_10C0 + 0x04 × n	CFDTXQCC2n	TX Queue Configuration/Control Register 2n (n = 0, 1)	—
0x8004_1120 + 0x04 × n	CFDTXQCC3n	TX Queue Configuration/Control Register 3n (n = 0, 1)	—
0x8004_1020 + 0x04 × n	CFDTXQSTS0n	TX Queue Status Register 0n (n = 0, 1)	—
0x8004_1080 + 0x04 × n	CFDTXQSTS1n	TX Queue Status Register 1n (n = 0, 1)	—
0x8004_10E0 + 0x04 × n	CFDTXQSTS2n	TX Queue Status Register 2n (n = 0, 1)	—
0x8004_1140 + 0x04 × n	CFDTXQSTS3n	TX Queue Status Register 3n (n = 0, 1)	—
0x8004_1040 + 0x04 × n	CFDTXQPCTR0n	TX Queue Pointer Control Register 0n (n = 0, 1)	—
0x8004_10A0 + 0x04 × n	CFDTXQPCTR1n	TX Queue Pointer Control Register 1n (n = 0, 1)	—
0x8004_1100 + 0x04 × n	CFDTXQPCTR2n	TX Queue Pointer Control Register 2n (n = 0, 1)	—
0x8004_1160 + 0x04 × n	CFDTXQPCTR3n	TX Queue Pointer Control Register 3n (n = 0, 1)	—
0x8004_1180	CFDTXQESTS	TX Queue Empty Status Register	—
0x8004_1184	CFDTXQFISTS	TX Queue Full Interrupt Status Register	—
0x8004_1188	CFDTXQMSTS	TX Queue Message Lost Status Register	—
0x8004_1190	CFDTXQISTS	TX Queue Interrupt Status Register	—
0x8004_1194	CFDTXQOFTISTS	TX Queue One Frame TX Interrupt Status Register	—
0x8004_1198	CFDTXQOFRISTS	TX Queue One Frame RX Interrupt Status Register	—
0x8004_119C	CFDTXQFSTS	TX Queue Full Status Register	—
0x8004_1200 + 0x04 × n	CFDTHLCCn	TX History List Configuration/Control Register n (n = 0, 1)	—
0x8004_1220 + 0x04 × n	CFDTHLSTSn	TX History List Status Register n (n = 0, 1)	—
0x8004_8000 + 0x08 × n	CFDTHLACC0n	Channel n TX History List Access Register 0 (n = 0, 1)	—
0x8004_8004 + 0x08 × n	CFDTHLACC1n	Channel n TX History List Access Register 1 (n = 0, 1)	—
0x8004_1240 + 0x04 × n	CFDTHLPCTRn	TX History List Pointer Control Register n (n = 0, 1)	—

Table 35.4 CANFD register map (3 of 3)

Address	Register symbol	Register name	Write protection
0x8004_1380	CFDGRSTC	Global Reset Control Register	—
0x8004_1384	CFDGFCMC	Global Flexible CAN Mode Configuration Register	—
0x8004_138C	CFDGFTBAC	Global Flexible Transmission Buffer Assignment Configuration Register	—
0x8004_1308	CFDGTSTCFG	Global Test Configuration Register	—
0x8004_130C	CFDGTSTCTR	Global Test Control Register	—
0x8004_1314	CFDGFDCFG	Global FD Configuration Register	—
0x8004_131C	CFDGLOCKK	Global Lock Key Register	—
0x8004_8400 + 0x04 × n	CFDRPGACCn	RAM Test Page Access Register n (n = 0 to 63)	—
0x8004_1418 + 0x20 × n	CFDCnBLCT	Channel n Bus Load Control Register (n = 0, 1)	—
0x8004_141C + 0x20 × n	CFDCnBLSTS	Channel n Bus Load Status Register (n = 0, 1)	—

Table 35.5 CANFD related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0 (NONSAFETY)	—	MSTPCRD.MSTPCRD10	SLVACCCTL1.CANFD_SL

35.3 Register Descriptions

35.3.1 CFDCnNCFG : Channel n Nominal Bit Rate Configuration Register (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x0000 + 0x10 × n

Bit position:	31						25	24						17	16						10	9						0
Bit field:	NTSEG2[6:0]							NTSEG1[7:0]								NSJW[6:0]							NBRP[9:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	NBRP[9:0]	Nominal Bit Rate Prescaler Nominal bit rate prescaler division ratio setting	R/W
16:10	NSJW[6:0]	Nominal Bit Rate Resynchronization Jump Width Control 0x00: Setting prohibited 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W
24:17	NTSEG1[7:0]	Nominal Bit Rate Time Segment 1 Control 0x00: Setting prohibited 0x01: 2 Tq 0x02: 3 Tq ⋮ 0xFE: 255 Tq 0xFF: 256 Tq	R/W
31:25	NTSEG2[6:0]	Nominal Bit Rate Time Segment 2 Control 0x00: Setting prohibited 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W

The CFDCnNCFG register is used to configure the transmission/reception nominal bit rate parameters of the channels. Do not write this register in CH_OPERATION or CH_SLEEP mode. Set this register in CH_RESET or CH_HALT mode.

NBRP[9:0] bits (Nominal Bit Rate Prescaler)

The NBRP[9:0] bits are used to define the peripheral bus clock periods contained in a time quantum.

NSJW[6:0] bits (Nominal Bit Rate Resynchronization Jump Width Control)

The NSJW[6:0] bits set the synchronization jump width. A value from 1 to 128 time quanta can be set.

NTSEG1[7:0] bits (Nominal Bit Rate Time Segment 1 Control)

The NTSEG1[7:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. These bits contain the propagation segment. Configure a Tq value only between 2 and 256, inclusive. See section 35.5.1.2. CAN Bit Timing for more details.

NTSEG2[6:0] bits (Nominal Bit Rate Time Segment 2 Control)

The NTSEG2[6:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. Configure a Tq value only between 2 and 128, inclusive.

35.3.2 CFDCnCTR : Channel n Control Register (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x0004 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ROM	CRCT	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	TDCVFIE	SOCOIE	EOCOIE	TAIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
1:0	CHMDC[1:0]	Mode Select 0 0: Channel Communication mode 0 1: Channel Reset mode 1 0: Channel Halt mode 1 1: Setting prohibited	R/W
2	CSLPR	Channel Stop Mode 0: Other than Channel Stop mode 1: Channel Stop mode	R/W
3	RTBO	Forcible Return from Bus-Off When this bit is set to 1, forcible return from the bus-off state is made. This bit is always read as 0.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt disabled 1: Bus error interrupt enabled	R/W
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt disabled 1: Error warning interrupt enabled	R/W
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt disabled 1: Error passive interrupt enabled	R/W
11	BOEIE	Bus-Off Entry Interrupt Enable 0: Bus-off entry interrupt disabled 1: Bus-off entry interrupt enabled	R/W

Bit	Symbol	Function	R/W
12	BORIE	Bus-Off Recovery Interrupt Enable 0: Bus-off recovery interrupt disabled 1: Bus-off recovery interrupt enabled	R/W
13	OLIE	Overload Interrupt Enable 0: Overload interrupt disabled 1: Overload interrupt enabled	R/W
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt disabled 1: Bus lock interrupt enabled	R/W
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt disabled 1: Arbitration lost interrupt enabled	R/W
16	TAIE	Transmission Abort Interrupt Enable 0: Transmission abort interrupt disabled 1: Transmission abort interrupt enabled	R/W
17	EOCOIE	Error Occurrence Counter Overflow Interrupt Enable 0: Error occurrence counter overflow interrupt disabled 1: Error occurrence counter overflow interrupt enabled	R/W
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt Enable 0: Successful occurrence counter overflow interrupt disabled 1: Successful occurrence counter overflow interrupt enabled	R/W
19	TDCVFIE*1	Transceiver Delay Compensation Violation Interrupt Enable 0: Transceiver delay compensation violation interrupt disabled 1: Transceiver delay compensation violation interrupt enabled	R/W
20	—	This bit is read as 0. The write value should be 0.	R/W
22:21	BOM[1:0]	Bus-Off Recovery Mode Select 0 0: ISO11898-1 compliant 0 1: Entry to Channel Halt mode automatically at bus-off entry 1 0: Entry to Channel Halt mode automatically at bus-off end 1 1: Entry to Channel Halt mode (in bus-off state) by program request	R/W
23	ERRD	Error Display Mode Select 0: Error flags are displayed only for the first error information after bits [14:8] in the CFDCnERFL register are all cleared 1: Error flag for all error information are displayed	R/W
24	CTME	Communication Test Mode Enable 0: Communication test mode is disabled 1: Communication test mode is enabled	R/W
26:25	CTMS[1:0]	Communication Test Mode Select 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (External loopback mode) 1 1: Self-test mode 1 (Internal loopback mode)	R/W
29:27	—	These bits are read as 0. The write value should be 0.	R/W
30	CRCT	CRC Error Test Enable 0: The first bit of reception ID field is not inverted 1: The first bit of reception ID field is inverted	R/W
31	ROM*1	Restricted Operation Mode Enable 0: Restricted operation mode disabled 1: Restricted operation mode enabled	R/W

Note 1. These bits are not available in the classical CAN function.

The CFDCnCTR register controls the modes of the related channel. It is used to enable generation of interrupts if errors are detected on the CAN bus connected to this channel. It is also used to configure the channel in test mode.

CHMDC[1:0] bits (Mode Select)

The CHMDC[1:0] bits can be used to configure modes of the CAN channel.

CAN mode transitions are described in more details in section 35.4.3. Channel Modes.

Setting CHMDC[1:0] bits to 11b has no effect. When the CAN-FD module is in GL_HALT mode, these bits can only be set to 10b or 01b. These bits cannot be set in CH_SLEEP mode.

These bits can change automatically when transitioning to Halt mode by the CFDCnCTR.BOM settings.

If CPU write access to CFDCnCTR.CHMDC occurs at the same time when the CAN channel enters Halt mode (at the start of bus-off when CFDCnCTR.BOM[1:0] = 01b, or at the end of bus-off when CFDCnCTR.BOM[1:0] = 10b), then the CPU write access has the highest priority.

The CAN channel changes the value of CFDCnCTR.CHMDC within the Channel Control Registers for the specified cases only if the CFDCnCTR.CHMDC[1:0] value is 00b (Operation mode).

CSLPR bit (Channel Stop Mode)

When the CSLPR bit is 1, a Sleep mode request is generated for the corresponding CAN channel

When this bit is 0, a request to exit Sleep mode is generated for the related CAN-FD channel.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

RTBO bit (Forcible Return from Bus-Off)

When the protocol controller of the CAN channel enters bus-off state, you can force a recovery from bus-off state by setting the RTBO bit in the Channel Control Register to 1.

The error state changes from bus-off state to integrating with a maximum delay of 1 CAN bit time.

When the RTBO bit is set to 1, the REC and TEC registers are initialized and the Bus-Off Status bit (Channel Bus-off Status, CFDCnSTS.BOSTS) is set to 0.

Registers other than the REC and TEC registers are not initialized by this command. Even if CFDCnCTR.BORIE is set, a bus-off recovery interrupt is not generated by this recovery from the bus-off state.

The RTBO bit cannot be set in CH_SLEEP mode. Setting this bit in any state other than bus-off state has no effect and the bit is cleared immediately. The read value is always 0.

Return from the Bus-Off command should be used only when CFDCnCTR.BOM[1:0] is set to 00b.

Only write to this bit when the related CAN-FD channel is in CH_OPERATION mode. This bit is automatically cleared when set by software.

BEIE bit (Bus Error Interrupt Enable)

When the BEIE and the CFDCnERFL.BEF bits are both 1, an error interrupt request is generated.

This bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

EWIE bit (Error Warning Interrupt Enable)

When the EWIE and the CFDCnERFL.EWF bits are both 1, an error interrupt request is generated.

The EWIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

EPIE bit (Error Passive Interrupt Enable)

An error interrupt request is generated when the EPIE bit and the CFDCnERFL.EPF are both 1.

The EPIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

BOEIE bit (Bus-Off Entry Interrupt Enable)

When the BOEIE and the CFDCnERFL.BOEF bits are both 1, an error interrupt request is generated.

The BOEIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

BORIE bit (Bus-Off Recovery Interrupt Enable)

When the BORIE and the CFDCnERFL.BORF bits are both 1, an error interrupt request is generated.

The BORIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

OLIE bit (Overload Interrupt Enable)

When the OLIE and the CFDCnERFL.OVLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

BLIE bit (Bus Lock Interrupt Enable)

When the BLIE and the CFDCnERFL.BLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

ALIE bit (Arbitration Lost Interrupt Enable)

When the ALIE and the CFDCnERFL.ALF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

TAIE bit (Transmission Abort Interrupt Enable)

When the TAIE bit is 1 and a transmission is successfully aborted from the transmit buffer, an interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

EOCOIE bit (Error Occurrence Counter Overflow Interrupt Enable)

When the EOCOIE bit is 1 and the CFDCnFDSTS.EOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The EOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

SOCOIE bit (Successful Occurrence Counter Overflow Interrupt Enable)

When the SOCOIE bit is 1 and the CFDCnFDSTS.SOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The SOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

TDCVFIE bit (Transceiver Delay Compensation Violation Interrupt Enable)

When the TDCVFIE bit is 1 and the CFDCnFDSTS.TDCVF bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The TDCVFIE bit cannot be set in CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET mode. Do not set this bit when in Classical-only mode.

Note: This bit is not available in the classical CAN function.

BOM[1:0] bits (Bus-Off Recovery Mode Select)

The BOM[1:0] bits control the timing of the recovery from Bus-Off mode of the CAN-FD Channel.

Do not write to these bits in CH_SLEEP mode. Only write to these bits when the related CAN-FD channel is in CH_RESET mode.

ERRD bit (Error Display Mode Select)

The ERRD bit controls the display mode of the error flag bits [14:8] in the Channel Error Flag Register (CFDCnERFL).

If the ERRD bit is 0 and more than one errors occur at the same time, the error flag bits are set for all the errors that occurred at the same time. No further errors are flagged until the error flag bits [14:8] are cleared.

Do not write to the ERRD bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

CTME bit (Communication Test Mode Enable)

The CTME bit enables the channel test modes.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_HALT mode.

CTMS[1:0] bits (Communication Test Mode Select)

The CTMS[1:0] bits are used to select the required test mode.

Do not write to these bits in CH_SLEEP or CH_RESET mode. Only write to these bits when the related CAN-FD channel is in CH_HALT mode.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

CRCT bit (CRC Error Test Enable)

The CRCT bit checks the internal CRC generator logic of the protocol controller.

This bit inverts the first bit (ID bit) of the CAN message data stream that is being received, so that the internal generated CRC result does not match the received CRC value of the frame. Refer to the bit stuffing rule when using this feature for the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The internal generated CRC value is always observed in the following registers:

- CFDCnERFL.CRCREG (classical CAN frames)
- CFDCnFDCRC.CRCREG (CAN-FD frames)

Some limitations exist when using this bit:

- It is not possible to use this feature with CAN nodes connected to the device externally, only with nodes connected to the internal CAN bus communication can be used
- One CAN node can send a reference message and the receiver node can invert one bit of the incoming bit stream

Note: The transmitter and receiver modes share the same CRC generator, therefore it is not necessary to consider the modes separately when testing this limitation.

The CRC Error Test mode is enabled if the CRCT (new control signal that inverts the first bit of the bit stream) and CTME bits are both 1.

Do not write to the CRCT bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_HALT mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

ROM bit (Restricted Operation Mode Enable)

When the ROM and CTME bits are both 1, the restricted operation mode is enabled. This mode should only be used in basic test mode (CFDCnCTR.CTMS[1:0] = 00b).

The ROM bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_HALT mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. Do not set this bit when in Classical-only mode.

Note: This bit is not available in the classical CAN function.

35.3.3 CFDCnSTS : Channel n Status Register (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x0008 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TEC[7:0]								REC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ESIF	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
0	CRSTSTS	Channel Reset Status Flag 0: Not in Channel Reset mode 1: In Channel Reset mode	R
1	CHLTSTS	Channel Halt Status Flag 0: Not in Channel Halt mode 1: In Channel Halt mode	R
2	CSLPSTS	Channel Stop Status Flag 0: Not in Channel Sleep mode 1: In Channel Sleep mode	R
3	EPSTS	Error Passive Status Flag 0: Not in error passive state 1: In error passive state	R
4	BOSTS	Bus-Off Status Flag 0: Not in bus-off state 1: In bus-off state	R
5	TRMSTS	Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state	R
6	RECSTS	Receive Status Flag 0: Bus idle, in transmission or bus off state 1: In reception	R
7	COMSTS	Communication Status Flag 0: Communication is not ready 1: Communication is ready	R
8	ESIF*1	Error State Indication Flag 0: No CANFD message whose ESI bit is recessive has been received 1: At least one CANFD message whose ESI bit is recessive has been received	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
23:16	REC[7:0]	Reception Error Count The receive error counter (REC) can be read.	R
31:24	TEC[7:0]	Transmission Error Count The transmit error counter (TEC) can be read.	R

Note 1. This bit is not available in the classical CAN function.

The CFDCnSTS register shows the mode, error, and transmission/reception status of the related channel together with its reception and transmission error count values.

CRSTSTS bit (Channel Reset Status Flag)

The CRSTSTS bit indicates whether the related CAN channel is in Reset mode.

This bit is set automatically when the related CAN channel enters Channel Reset mode. When the mode is changed from Reset mode to Sleep mode, the CRSTSTS bit remains 1.

This bit is cleared automatically when the related CAN channel exits the Channel Reset mode, except when changing to Sleep mode.

CHLTSTS bit (Channel Halt Status Flag)

The CHLTSTS bit indicates whether the related CAN channel is in Halt mode.

This bit is set automatically when the related CAN module enters Halt mode, and is cleared automatically when the related CAN module exits Halt mode.

CSLPSTS bit (Channel Stop Status Flag)

The CSLPSTS bit indicates whether the related CAN channel is in Sleep mode.

This bit is set automatically when the related CAN-FD channel enters Sleep mode, and is cleared automatically when the related CAN-FD channel exits Sleep mode.

EPSTS bit (Error Passive Status Flag)

The EPSTS bit indicates whether the related CAN-FD channel has entered the error passive state.

This bit is set automatically when the value of the CAN Transmission or Reception Counter Register exceeds the value of 0x7F.

This bit is cleared automatically when the related CAN-FD channel exits the error passive state or enters Reset mode.

BOSTS bit (Bus-Off Status Flag)

The BOSTS bit indicates whether the related CAN-FD channel has entered the error bus-off state.

This bit is set automatically when the value of the related CAN Transmission Error Count Register exceeds 0xFF and the related CAN-FD channel is in the bus-off state (CAN Transmission Error Count Register > 0xFF).

This bit is cleared automatically when the related CAN-FD channel exits bus-off state.

TRMSTS bit (Transmit Status Flag)

The TRMSTS bit indicates whether the related CAN-FD channel is transmitting a message.

This bit is set automatically when the related CAN-FD channel is operating as a transmitter node or is in the bus-off state.

This bit is cleared automatically when the related CAN-FD channel is in the bus-idle state or starts operating as a receiver node.

RECSTS bit (Receive Status Flag)

The RECSTS bit indicates whether the related CAN-FD channel is receiving a message.

This bit is set automatically when the related CAN-FD channel is operating as a receiver node.

This bit is cleared automatically when the related CAN-FD channel is in the bus-idle state or starts operating as a transmitter node.

COMSTS bit (Communication Status Flag)

The COMSTS bit indicates whether the related CAN-FD channel is ready for communication.

This bit is set automatically when the related CAN-FD channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting the Reset or Halt mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

Note: This bit is 1 during bus-off state.

ESIF bit (Error State Indication Flag)

The ESIF bit is set when the ESI bit is sampled recessive for a reception CAN message without any error. When in Loopback or Mirror mode, the self-transmitted messages are considered reception messages.

If a set from the CAN-FD channel occurs simultaneously with a clear by a write access, then the bit is set.

This bit is cleared by writing 0 to it. This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

REC[7:0] bits (Reception Error Count)

The REC[7:0] bits increment or decrement the counter value according to error status of the CAN-FD channel during reception, and display the value of the REC error counter.

The value in bus-off state is indeterminate.

These bits are cleared automatically when the CAN-FD module enters GL_RESET or the CAN-FD channel is in CH_RESET mode.

TEC[7:0] bits (Transmission Error Count)

The TEC[7:0] bits increment or decrement the counter value according to error status of the CAN-FD channel during transmission, and display the value of the TEC error counter.

Only write to these bits when in test mode and CAN-FD channel is in CH_HALT mode.

Do not read data from this register while CFDCnCTR.TRWE is set.

These bits are cleared automatically when CAN-FD module is in GL_RESET or CAN-FD channel is in CH_RESET mode.

35.3.4 CFDCnERFL : Channel n Error Flag Register (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x000C + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CRCREG[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BEF	Bus Error Flag 0: Channel bus error not detected 1: Channel bus error detected	R/W*1
1	EWF	Error Warning Flag 0: Channel error warning not detected 1: Channel error warning detected	R/W*1
2	EPF	Error Passive Flag 0: Channel error passive not detected 1: Channel error passive detected	R/W*1
3	BOEF	Bus-Off Entry Flag 0: Channel bus-off entry not detected 1: Channel bus-off entry detected	R/W*1
4	BORF	Bus-Off Recovery Flag 0: Channel bus-off recovery not detected 1: Channel bus-off recovery detected	R/W*1
5	OVLF	Overload Flag 0: Channel overload not detected 1: Channel overload detected	R/W*1

Bit	Symbol	Function	R/W
6	BLF	Bus Lock Flag 0: Channel bus lock not detected 1: Channel bus lock detected	R/W*1
7	ALF	Arbitration Lost Flag 0: Channel arbitration-lost not detected 1: Channel arbitration-lost detected	R/W*1
8	SERR	Stuff Error Flag 0: Channel stuff error not detected 1: Channel stuff error detected	R/W*1
9	FERR	Form Error Flag 0: Channel form error not detected 1: Channel form error detected	R/W*1
10	AERR	Acknowledge Error Flag 0: Channel acknowledge error not detected 1: Channel acknowledge error detected	R/W*1
11	CERR	CRC Error Flag 0: Channel CRC error not detected 1: Channel CRC error detected	R/W*1
12	B1ERR	Recessive Bit Error Flag 0: Channel recessive bit error not detected 1: Channel recessive bit error detected	R/W*1
13	B0ERR	Dominant Bit Error Flag 0: Channel dominant bit error not detected 1: Channel dominant bit error detected	R/W*1
14	ADERR	Acknowledge Delimiter Error Flag 0: Channel acknowledge delimiter error not detected 1: Channel acknowledge delimiter error detected	R/W*1
15	—	This bit is read as 0. The write value should be 0.	R/W
30:16	CRCREG[14:0]	CRC Calculation Data (CRC length: 15 bits) These bits show the CRC value calculated based on the transmit message or receive message.	R
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. To clear each flag of this register, software must write 0 to the corresponding bit. These flags cannot be set to 1 by software.

The CFDCnERFL register shows the status of various error conditions detectable regardless of the setting of the related CAN Channel Error Interrupt Enable Register. It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) for a description of error occurrence conditions.

Only write to this register when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

BEF bit (Bus Error Flag)

The BEF bit indicates a detection of a CAN channel bus error state, flagged by bits [14:8] in this register.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when a bus error is detected, and is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

EWF bit (Error Warning Flag)

The EWF bit indicates whether an error warning condition has been detected for the CAN channel.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when either TEC or REC exceeds 0x5F.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x5F. Therefore, if the TEC or REC remains > 0x5F and the EWF bit is cleared by software, it is not set again until both the TEC and REC go below 0x60 and either TEC or REC crosses over again from a value 0x5F to a value > 0x5F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

EPF bit (Error Passive Flag)

The EPF bit indicates a detection of a CAN channel error passive state.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state becomes error passive state.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x7F. Therefore, if the TEC or REC remains > 0x7F and the bit is cleared by software, it is not set again until both the TEC and REC go below 0x80 and either TEC or REC crosses over again from a value ≤ 0x7F to a value > 0x7F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

BOEF bit (Bus-Off Entry Flag)

The BOEF bit indicates a detection of a CAN channel bus-off entry state.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state enters the bus-off state.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, then the bit is set.

BORF bit (Bus-Off Recovery Flag)

The BORF bit indicates a detection of a CAN channel bus-off recovery state.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically if CAN channel recovers from bus-off state in the following conditions:

- When CFDCnCTR.BOM[1:0] is 00b and normal recovery (11 consecutive recessive bits × 128 times detected) occurs
- When CFDCnCTR.BOM[1:0] is 10b and normal recovery (11 consecutive recessive bits × 128 times detected) occurs
- When CFDCnCTR.BOM[1:0] is 11b and normal recovery (11 consecutive recessive bits × 128 times detected) occurs.

The bit is not set if CAN channel recovers from bus-off state in the following conditions:

- When CAN Reset mode is requested
- When CFDCnCTR.RTBO is set to 1 (the CAN channel returns to error active)
- When CFDCnCTR.BOM[1:0] is 01b
- When CFDCnCTR.BOM[1:0] is 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set.

OVLF bit (Overload Flag)

The OVLF flag indicates a detection of a CAN channel overload state.

The OVLF bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

BLF bit (Bus Lock Flag)

The BLF bit indicates a detection of a CAN channel bus lock condition.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

ALF bit (Arbitration Lost Flag)

The ALF bit indicates a detection of a CAN channel bus arbitration-lost condition.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

The bit is set automatically when an arbitration-lost condition is detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

SERR bit (Stuff Error Flag)

The SERR bit indicates a detection of a CAN stuff error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a stuff error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

FERR bit (Form Error Flag)

The FERR bit indicates a detection of a CAN form error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

AERR bit (Acknowledge Error Flag)

The AERR bit indicates a detection of a CAN acknowledge error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when an acknowledge error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

CERR bit (CRC Error Flag)

The CERR bit indicates a detection of a CAN CRC error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a CRC error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

B1ERR bit (Recessive Bit Error Flag)

The B1ERR bit indicates a detection of a recessive bit error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000001b.

B0ERR bit (Dominant Bit Error Flag)

The B0ERR bit indicates a detection of a dominant bit error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

ADERR bit (Acknowledge Delimiter Error Flag)

The ADERR bit indicates a detection of an acknowledge delimiter bit error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected during the acknowledge delimiter state of frame transmission. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

CRCREG[14:0] bits (CRC Calculation Data (CRC length: 15 bits))

The CRCREG[14:0] bits read the calculated CRC value when CFDCnCTR.CTME bit is 1 for the channel.

If CFDCnCTR.CTME bit is 0, then these bits are always read as 0.

These bits show the CAN2.0 CRC value calculated by the CAN-FD channel logic when the CTME bit is enabled.

The CFDCnERFL.CRCREG value is updated in the first bit of the CRC field of the CAN frame (reception and transmission).

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

35.3.5 CFDCnDCFG : Channel n Data Bit Rate Configuration Register (n = 0, 1)

This register is not available in the classical CAN function.

Base address: CANFD = 0x8004_0000

Offset address: 0x1400 + 0x20 × n

Bit position:	31				27				24				19				16				12				8	7							0
Bit field:	—	—	—	—	DSJW[3:0]				—	—	—	—	DTSEG2[3:0]				—	—	—	DTSEG1[4:0]					DBRP[7:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DBRP[7:0]	Data Bit Rate Prescaler Division Ratio Setting When the set value = P (0 to 255), the data bit rate prescaler divides fCAN by (P + 1).	R/W
12:8	DTSEG1[4:0]	Data Bit Rate Time Segment 1 Control 0x00: Setting prohibited 0x01: 2 Tq 0x02: 3 Tq 0x03: 4 Tq ⋮ 0x1E: 31 Tq 0x1F: 32 Tq	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
19:16	DTSEG2[3:0]	Data Bit Rate Time Segment 2 Control 0x0: Setting prohibited 0x1: 2 Tq 0x2: 3 Tq 0x3: 4 Tq ⋮ 0xE: 15 Tq 0xF: 16 Tq	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	DSJW[3:0]	Data Bit Rate Resynchronization Jump Width Control 0x0: 1 Tq 0x1: 2 Tq ⋮ 0xE: 15 Tq 0xF: 16 Tq	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCnDCFG register configures the transmission/reception data bit rate parameters of the channels.

The channel of Classical-only mode does not perform configuration of this register.

Do not write to this register in CH_OPERATION or CH_SLEEP mode. Only write to this register in CH_RESET or CH_HALT mode.

DBRP[7:0] bits (Data Bit Rate Prescaler Division Ratio Setting)

The DBRP[7:0] bits define the peripheral bus clock periods contained in a time quantum.

DTSEG1[4:0] bits (Data Bit Rate Time Segment 1 Control)

The DTSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 time quanta can be set.

The DTSEG1[4:0] bits are also used to set the propagation segment.

Do not write any other value to these bits. See section 35.5.1.2. CAN Bit Timing for more details.

DTSEG2[3:0] bits (Data Bit Rate Time Segment 2 Control)

The DTSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A value from 2 to 16 time quanta can be set.

Do not write any other value to these bits.

DSJW[3:0] bits (Data Bit Rate Resynchronization Jump Width Control)

The DSJW[3:0] bits set the synchronization jump width. A value from 1 to 16 time quanta can be set.

35.3.6 CFDCnFDCFG : Channel n CAN-FD Configuration Register (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1404 + 0x20 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFDTE	CLOE	REFE	FDOE	—	GWBRS	GWFDF	GWEN	TDCO[7:0]							
Value after reset:	0	0*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ESIC	TDCE	TDCOC	—	—	—	—	—	EOCCFG[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	EOCCFG[2:0]	Error Occurrence Counter Configuration 0 0 0: All transmit messages and receive messages 0 0 1: All transmit messages 0 1 0: All receive messages 0 1 1: Setting prohibited 1 0 0: Only data phase of transmitted or received CANFD message 1 0 1: Only data phase of transmitted CANFD message 1 1 0: Only data phase of received CANFD message 1 1 1: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	TDCOC*2	Transmitter Delay Compensation Offset Configuration 0: Measurement and offset 1: Offset-only	R/W
9	TDCE*2	Transceiver Delay Compensation Enable 0: Transceiver delay compensation disabled 1: Transceiver delay compensation enabled	R/W
10	ESIC*2	Error State Indication Configuration 0: The ESI bit in the frame represents the error state of the node itself 1: The ESI bit in the frame represents the error state of the message buffer if the node itself is not in error passive. If the node is in error passive, then the ESI bit is driven by the node itself.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TDCO[7:0]*2	Transceiver Delay Compensation Offset These bits are set to the transmitter delay compensation offset value.	R/W
24	GWEN*2	CAN2.0, CAN-FD Multi Gateway Enable 0: Multi gateway disabled 1: Multi gateway enabled	R/W
25	GWFDF*2	Gateway FDF Configuration Bit 0: Gateway frame is transmitted as Classical CAN frame 1: Gateway frame is transmitted as CAN-FD frame	R/W
26	GWBRS*2	Gateway BRS Configuration Bit 0: Gateway frame is transmitted with BRS = 0 1: Gateway frame is transmitted with BRS = 1	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	FDOE*2	FD-Only Enable 0: FD-only mode disabled 1: FD-only mode enabled	R/W
29	REFE	RX Edge Filter Enable 0: RX edge filter disabled 1: RX edge filter enabled	R/W
30	CLOE*2	Classical CAN-Only Enable 0: Classical-only mode disabled 1: Classical-only mode enabled	R/W
31	CFDTE	CAN-FD Frame Distinction Enable 0: CAN-FD frame distinction disabled 1: CAN-FD frame distinction enabled	R/W

Note 1. This bit is set to 1 for RZ/T2M device with CANFD disabled.
Note 2. These bits are not available in the classical CAN function.

The CFDCnFDCFG register configures which communication direction (transmitter/receiver) errors are counted.

EOCCFG[2:0] bits (Error Occurrence Counter Configuration)

The EOCCFG[2:0] bits select which type of CAN frame configuration and direction, including protocol errors are counted.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

TDCOC bit (Transmitter Delay Compensation Offset Configuration)

The TDCOC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CAN-FD channel. If the bit is set to 0, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1, the position of the SSP is defined only by the offset.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

TDCE bit (Transceiver Delay Compensation Enable)

The TDCE bit enables the transceiver delay compensation for the CAN-FD channel.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

ESIC bit (Error State Indication Configuration)

Bus controllers that are used as CAN-to-CAN gateway support that in every forwarded CAN-FD message. The ESI flag does not change to reflect the status of the gateway, bridge, or router but instead the flag is sent as it was in the original message.

The ESIC bit controls the transmission of either the ESI flag information or the message of ESI flag information (CFDCFFDCSTSn.CFESI or CFDTMFDCTRn.TMESI).

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

TDCO[7:0] bits (Transceiver Delay Compensation Offset)

The TDCO[7:0] bits set the secondary sample point offset. How this value is used, depends on the CFDCnFDCFG.TDCOC setting.

If CFDCnFDCFG.TDCOC = 0, the transceiver delay compensation result is equal to the Trv_Delay (measured delay) + the value in CFDCnFDCFG.TDCO, rounded down to the nearest integer number of time quanta. Otherwise, the result is equal to the value in CFDCnFDCFG.TDCO. See section 35.5.1.5. Transmitter Delay Compensation for details on how CFDCnFDCFG.TDCO is used.

The actual offset value is interpreted as TDCO + 1. For example, if 4 is set in TDCO, the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.

Do not write to the TDCO[7:0] bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set to this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

GWEN bit (CAN2.0, CAN-FD Multi Gateway Enable)

When the GWEN bit is enabled, a multi gateway is enabled. Message received on one node can be routed to another node using the COM FIFO when they are configured as gateway FIFO (CFDCFCCn.CFM[1:0] = 10b). Furthermore, when TX Queue is set as Gateway mode, the message received on one node can be stored in TX Queue and can be sent to another node.

The FDF and BRS bits of the routed message can be changed by the configuration value of the CFDCnFDCFG.GWFDF and CFDCnFDCFG.GWBRS bits. By this, the transmitted value of these bits can be replaced.

Example:

CFDCnFDCFG.GWEN = 1 on channel y

CFDCnFDCFG.GWFDF = 1

If a Classical CAN frame is received on channel x and routed to a gateway FIFO or TX Queue of channel y, then this CAN frame is sent on channel y as a CAN-FD frame because of the CFDCnFDCFG.GWFDF bit.

Table 35.6 shows how the message information is changed depending on the received and configured data.

Table 35.6 Modified message information by received and configured data

Routed CAN frame	Routed received DLC	CAN BRS bit	Configured CFDCnFDCFG.GWFDF bit	Gateway message DLC	Gateway message BRS bit	Gateway message frame type
CAN2.0	≤ 8	N/A	1	≤ 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN2.0	> 8	N/A	1	= 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN-FD	≤ 8	None	1	≤ 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN-FD	> 8	None	1	> 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN2.0	≤ 8	N/A	0	≤ 8	N/A	CAN2.0
CAN2.0	> 8	N/A	0	> 8	N/A	CAN2.0
CAN-FD	≤ 8	None	0	≤ 8	N/A	CAN2.0
CAN-FD	> 8	None	0	= 8	N/A	CAN2.0

Note:
- This gateway is limited to an 8-byte data payload for different frame type. If routing and target frame type is the same, the data length code (DLC) value remains the same. If the source frame is a CAN-FD with more than 8 data bytes, then on classical destination node, the data payload is reduced to 8 bytes. Only the first 8 bytes of data perform gateway transmission.
- Transmission buffers other than the gateway FIFO are not affected by this feature.

Do not route remote frames through the gateway when CFDCnFDCFG.GWEN is set. When a destination node is CFDCnFDCFG.FDOE = 1, set CFDCnFDCFG.GWEN and CFDCnFDCFG.GWFDF to 1. When a destination node is CFDCnFDCFG.CLOE = 1, set CFDCnFDCFG.GWEN = 1 and CFDCnFDCFG.GWFDF = 0.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

This bit is not available in the classical CAN function.

GWFDF bit (Gateway FDF Configuration Bit)

When the GWEN bit is set to 1, the FDF bit of the transmitting gateway frame is replaced by the value of the GWFDF bit.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CAN-FD module is in CH_RESET mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

GWBRS bit (Gateway BRS Configuration Bit)

When the GWEN bit is set to 1, the BRS bit of the transmitting gateway frame is replaced by the value of CFDCnFDCFG.GWBRS.

In classical CAN frames, the GWBRS bit is invalid.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

FDOE bit (FD-Only Enable)

The FDOE bit enables the reception and transmission of CAN-FD-only frames. If enabled, communication in Classical CAN frame format is disabled. Transmission of Classical CAN frames is not possible because the FDF bit of the message buffer is a don't care (CFDCFFDCSTSn.CFFDF/CFDTMFDCTRn.TMFDF).

If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and response with error frames. When a Classical CAN frame is configured for transmitting, the FDF bit is sent as recessive, therefore an FD frame is sent. If the data length code (DLC) is configured of greater than 8 bytes, the remaining data bytes are padded with 0xCC.

The FDOE bit cannot be written in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Do not set CFDCnFDCFG.FDOE and CFDCnFDCFG.CLOE simultaneously.

This bit is not available in the classical CAN function.

REFE bit (RX Edge Filter Enable)

The REFE bit enables the RX edge filter during the IDLE detection (bus integration). When the bit is enabled, two consecutive dominant time quanta are required to detect a synchronization edge.

The REFE bit cannot be written in CH_OPERATION, CH_HALT and CH_SLEEP mode. Do not set this bit when in Classical-only mode and when CFDCnFDCFG.CFDTE = 0 (disabled CAN-FD frame distinction).

CLOE bit (Classical CAN-Only Enable)

The CLOE bit enables the Classical CAN-only mode. If this bit is 1, the protocol controller can only send classical frames and response with a form or CRC error on FD frames.

Do not set CFDCnFDCFG.CLOE and CFDCnFDCFG.FDOE simultaneously.

CFDCnFDCFG.CLOE	CFDCnFDCFG.FDOE	Channel mode
0	0	CAN-FD mode (Dual mode)
0	1	FD-only mode
1	0	Classical CAN-only mode
1	1	Reserved

Do not write to this bit in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Only write to these bits when the CAN-FD channel is in CH_RESET mode.

This bit is not available in the classical CAN function.

CFDTE bit (CAN-FD Frame Distinction Enable)

The CFDTE bit enables the CAN-FD frame distinction function. The CFDTE bit is required for Classical CAN-only mode (CFDCnFDCFG.CLOE = 1).

If this bit is 0, the protocol controller can only send Classical frames and response with a Form or CRC error on FD frames.

If this bit is 1, the protocol controller behaves according to the ISO 11898-1 (DIS 2015) specification. If the FDF bit is detected recessive, then the protocol controller enters the protocol exception state, and attempts to integrate back to the CAN communication.

Do not write this bit in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET mode.

35.3.7 CFDCnFDCTR : Channel n CAN-FD Control Register (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1408 + 0x20 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOCCLR	EOCCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EOCCLR	Error Occurrence Counter Clear 0: No error occurrence counter clear 1: Clear error occurrence counter	R/W
1	SOCCLR	Successful Occurrence Counter Clear 0: No successful occurrence counter clear 1: Clear successful occurrence counter	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCnFDCTR register (n = 0, 1) controls the error and successful occurrence counters.

EOCCLR bit (Error Occurrence Counter Clear)

The EOCCLR bit is used to clear the error occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CAN-FD logic and when the related CAN-FD channel is in CH_RESET mode.

SOCCLR bit (Successful Occurrence Counter Clear)

The SOCCLR bit is used to clear the successful occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CAN-FD logic and when the related CAN-FD channel is in CH_RESET mode.

35.3.8 CFDCnFDSTS : Channel n CAN-FD Status Register (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x140C + 0x20 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SOC[7:0]								EOC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TDCVF	—	—	—	—	—	SOCO	EOCO	TDCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TDCR[7:0]	Transceiver Delay Compensation Result	R

Bit	Symbol	Function	R/W
8	EOCO	Error Occurrence Counter Overflow Flag 0: Error occurrence counter has not overflowed 1: Error occurrence counter has overflowed	R/W
9	SOCO	Successful Occurrence Counter Overflow Flag 0: Successful occurrence counter has not overflowed 1: Successful occurrence counter has overflowed	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	TDCVF	Transceiver Delay Compensation Violation Flag 0: Transceiver delay compensation violation has not occurred 1: Transceiver delay compensation violation has occurred	R/W
23:16	EOC[7:0]	Error Occurrence Counter These bits show the error occurrence counter value.	R
31:24	SOC[7:0]	Successful Occurrence Counter These bits show the successful occurrence counter value.	R

The CFDCnFDSTS register (n = 0, 1) indicates the transceiver compensation delay result and its related FIFO message lost status.

TDCR[7:0] bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.

The measured delay is a multiple of the CAN channel DLL clock. The result depends on the CFDCnFDCFG.TDCOC configuration and the offset value in CFDCnFDCFG.TDCO. See section 35.5.1.5. Transmitter Delay Compensation for details on how this value is derived.

The TDCR[7:0] bits are updated at the falling edge between the FDF bit and the RES bit when CFDCnFDCFG.TDCOC = 0 and the transceiver delay compensation is enabled (CFDCnFDCFG.TDCE = 1).

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

EOCO bit (Error Occurrence Counter Overflow Flag)

The EOCO bit indicates whether the related CAN channel error occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDCnFDSTS.EOC is 0xFF and a CAN bus error is detected based on the configuration defined in CFDCnFDCFG.EOCCFG.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit.

SOCO bit (Successful Occurrence Counter Overflow Flag)

The SOCO bit indicates whether the related CAN channel successful occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDCnFDSTS.SOC is 0xFF and a successful message reception or successful message transmission occurs.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit.

TDCVF bit (Transceiver Delay Compensation Violation Flag)

The CAN-FD module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.

The transceiver delay has some variations depending on the physical parameters such as temperature. The result bit CFDCnFDSTS.TDCR is updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCVF bit captures this violation.

This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times - 2 clk_dlc) and the internal bit is overrun.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit.

EOC[7:0] bits (Error Occurrence Counter)

The EOC[7:0] bits are used together with the SOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

This higher error rate can be detected depending on the configuration of the CFDCnFDCFG.EOCCFG bits.

The EOC[7:0] bits are set only by CAN-FD module logic. These bits are cleared by writing 1 to CFDCnFDCTR.EOCCLR. Writing any other value has no effect.

These bits are updated when an error occurs, according to the configuration of the CFDCnFDCFG.EOCCFG bits. When the counter reaches the value of 0xFF, the update stops.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

SOC[7:0] bits (Successful Occurrence Counter)

The SOC[7:0] bits are used together with the EOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

The SOC[7:0] bits are set only by CAN-FD module logic. Writing any other value has no effect.

These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of 0xFF, the update stops. In Loopback mode, the counter is incremented twice.

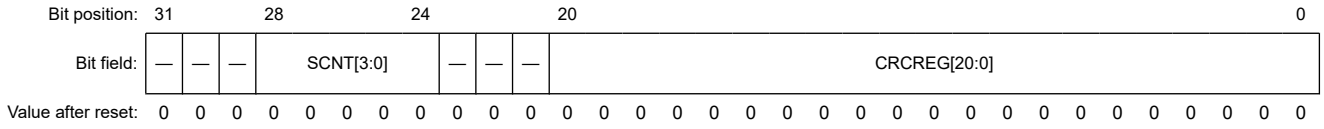
These bits are cleared by writing 1 to CFDCnFDCTR.SOCCLR.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

35.3.9 CFDCnFDCRC : Channel n CAN-FD CRC Register (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1410 + 0x20 × n



Bit	Symbol	Function	R/W
20:0	CRCREG[20:0]	CRC Register Value These bits show the CRC value calculated for the CAN-FD frame.	R
24:21	—	These bits are read as 0.	R
28:25	SCNT[3:0]	Stuff Bit Count These bits shows the stuff bit count (mod 8) for the CAN-FD frame.	R
31:29	—	These bits are read as 0.	R

The CFDCnFDCRC register (n = 0, 1) holds the CRC value calculated for the CAN-FD frame.

CRCREG[20:0] bits (CRC Register Value)

The CRCREG[20:0] bits contain the CRC value calculated by the CAN-FD channel logic when the CFDCnCTR.CTME bit is enabled.

The CFDCnFDCRC.CRCREG value is updated in the first bit of the CRC field of the CAN-FD frame (reception and transmission).

When the CFDCnCTR.CTME bit is 0, the CRCREG[20:0] bits are always read as 0.

When bit 17th of the CRC field is used, CRCREG[20:17] are always read as 0.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

SCNT[3:0] bits (Stuff Bit Count)

The SCNT[3:0] bits contain the stuff count value of the CAN-FD frame. These bits indicate the number of inserted stuff bits (modulo 8, Gray-coded) for a CAN-FD frame when the CFDCnCTR.CTME bit is enabled in CFDCnFDCRC.SCNT[3:1]. SCNT[0] is the parity bit.

When the CFDCnCTR.CTME bit is 0, the SCNT[3:0] bits are always read as 0.

The SCNT value is updated in the first bit of CRC field of the CAN-FD frame (reception and transmission).

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

35.3.10 CFDGIPV : Global IP Version Register

Base address: CANFD = 0x8004_0000

Offset address: 0x0080

Bit position:	31	29														16								9	8	7							0
Bit field:	—	—	PSI[13:0]														—	—	—	—	—	—	IPT[1:0]		IPV[7:0]								
Value after reset:	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	*1	0	1	0	0	0	0	1	0

Bit	Symbol	Function	R/W
7:0	IPV[7:0]	IP Version These bits show the main release number.	R
9:8	IPT[1:0]	IP Type These bits show the IP type used in the product. 0 0: CAN IP 0 1: CAN-FD IP 1 x: Reserved	R
15:10	—	These bits are read as 0.	R
29:16	PSI[13:0]	Parameter Status Information These bits show the IP configuration.	R
31:30	—	These bits are read as 0.	R

Bits	Description
[29:27]	Number of channel (110b: 2 channels)
[26:24]	Number of TXMB (001b: 32/channel)
[23:21]	Number of AFL entry (001b: 128/channel)
[20:17]	Number of pool buffer (0010b: 64/channel)
[16]	Same ID overwrite function of TXQ (0b: Support)

Note 1. The value depends on RZ/T2 package type.

The CFDGIPV register shows the release version of the CAN-FD module.

35.3.11 CFDGCFG : Global Configuration Register

Base address: CANFD = 0x8004_0000

Offset address: 0x0084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ITRCP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSBTCS[2:0]			TSSS	TSP[3:0]				—	—	CMPOC	DCS	MME	DRE	DCE	TPRI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TPRI	Transmission Priority 0: ID priority 1: Message buffer number priority	R/W
1	DCE	DLC Check Enable 0: DLC check disabled 1: DLC check enabled	R/W
2	DRE	DLC Replacement Enable 0: DLC replacement disabled 1: DLC replacement enabled	R/W
3	MME	Mirror Mode Enable 0: Mirror mode disabled 1: Mirror mode enabled	R/W
4	DCS	Data Link Controller Clock Select 0: PCLKM 1: PCLKCAN	R/W
5	CMPOC	CAN-FD Message Payload Overflow Configuration 0: Message is rejected 1: Message payload is cut to fit to configured message size	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
11:8	TSP[3:0]	Timestamp Prescaler 0x0: Timestamp prescaler = 1 0x1: Timestamp prescaler = 2 0x2: Timestamp prescaler = 4 0x3: Timestamp prescaler = 8 ⋮ 0xD: Timestamp prescaler = 8192 0xE: Timestamp prescaler = 16384 0xF: Timestamp prescaler = 32768	R/W
12	TSSS	Timestamp Source Select 0: Source clock for timestamp counter is peripheral clock 1: Source clock for timestamp counter is bit time clock	R/W
15:13	TSBTCS[2:0]	Timestamp Bit Time Channel Select 0 0 0: Select clock from channel 0 0 0 1: Select clock from channel 1 Others: Setting prohibited	R/W
31:16	ITRCP[15:0]	Interval Timer Reference Clock Prescaler FIFO interval timer prescaler value	R/W

The CFDGCFG register is used to select the transmission priority to be used for all the TX message buffers and the clock source for the CAN protocol engine of all CAN channels. The CFDGCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

TPRI bit (Transmission Priority)

The TPRI bit selects the transmission priority for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

Message buffer number priority should not be used together with TX queue transmission.

DCE bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

DRE bit (DLC Replacement Enable)

When the DRE bit is 1 and the DCE is 1, the CAN-FD stores the configured value (CFDGAFLP0n.GAFLDLC) of the DLC in the destination RX message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination RX message buffer or FIFO buffer is unchanged.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

MME bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

DCS bit (Data Link Controller Clock Select)

The DCS bit selects the clock source for CAN communication.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

CMPOC bit (CAN-FD Message Payload Overflow Configuration)

The CMPOC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size CFDRMNB.RMPLS, CFDRFCCn.RFPLS, and CFDCFCCn.CFPLS. The received message payload is always compared with the available message payload size in the message buffer.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

When this bit is set and payload overflow occurs, the DLC value is stored in the RX message buffer or FIFO buffer unchanged.

TSP[3:0] bits (Timestamp Prescaler)

The value configured in the TSP[3:0] bits defines the period of the clock source used for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

TSSS bit (Timestamp Source Select)

The TSSS bit allows the selection of the clock source for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode. Additionally, do not set this bit to 1 when CAN-FD communication is used.

Note: The bit time clock varies depending on the nominal and data rate bit configuration.

TSBTCS[2:0] bits (Timestamp Bit Time Channel Select)

The TSBTCS[2:0] bits allow the selection of the bit time clock of a particular channel for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

ITRCP[15:0] bits (Interval Timer Reference Clock Prescaler)

The ITRCP[15:0] bits allow the definition of a reference clock for the FIFO interval timer source clock.

When these bits are 0x0000, the timer is disabled.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

35.3.12 CFDGCTR : Global Control Register

Base address: CANFD = 0x8004_0000

Offset address: 0x0088

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MOWEIE	QMEIE	—	QOWEIE	CMPOFIE	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
1:0	GMDC[1:0]	Global Mode Control 0 0: Global operation mode request 0 1: Global reset mode request 1 0: Global halt mode request 1 1: Keep current value	R/W
2	GSLPR	Global Sleep Request 0: Global sleep request disabled 1: Global sleep request enabled	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	DEIE	DLC Check Interrupt Enable 0: DLC check interrupt disabled 1: DLC check interrupt enabled	R/W
9	MEIE	Message Lost Error Interrupt Enable 0: Message lost error interrupt disabled 1: Message lost error interrupt enabled	R/W
10	THLEIE	TX History List Entry Lost Interrupt Enable 0: TX history list entry lost interrupt disabled 1: TX history list entry lost interrupt enabled	R/W
11	CMPOFIE	CAN-FD Message Payload Overflow Flag Interrupt Enable 0: CAN-FD message payload overflow flag interrupt disabled 1: CAN-FD message payload overflow flag interrupt enabled	R/W
12	QOWEIE	TXQ Message Overwrite Error Interrupt Enable 0: TXQ message overwrite error interrupt disabled 1: TXQ message overwrite error interrupt enabled	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	QMEIE	TXQ Message Lost Error Interrupt Enable 0: TXQ message lost error interrupt disabled 1: TXQ message lost error interrupt enabled	R/W
15	MOWEIE	Message Lost Error Interrupt Enable 0: GW FIFO message lost error interrupt disabled 1: GW FIFO message lost error interrupt enabled	R/W
16	TSRST	Timestamp Reset 0: Timestamp not reset 1: Timestamp reset	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGCTR register controls the global mode of the CAN-FD module and the timestamp function. The register also enables and disables the global error interrupts.

GMDC[1:0] bits (Global Mode Control)

The GMDC[1:0] bits can be used to configure the modes for the CAN-FD module. Additionally, if CFDGCTR.GSLPR bit is 1 when the CAN-FD module is in Reset mode, the CAN-FD module transits to Global Sleep mode. Additionally, if CFDGCTR.GSLPR is 1 when the CAN-FD module is in Reset Mode, then the CAN-FD module transits to Global Sleep Mode.

Setting the GMDC[1:0] bits to 11b has no effect. Mode transition is described in detail in section 35.4.2. Global Modes.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

GSLPR bit (Global Sleep Request)

The GSLPR bit globally selects the sleep request for CAN-FD module including all CAN channels. Channel sleep request is set automatically for all channels.

Only write to this bit when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

DEIE bit (DLC Check Interrupt Enable)

When the DEIE bit is 1, an interrupt is generated if a DLC error is detected in the received frames.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

MEIE bit (Message Lost Error Interrupt Enable)

When the MEIE bit is 1, an interrupt is generated if a message lost condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

THLEIE bit (TX History List Entry Lost Interrupt Enable)

When the THLEIE bit is 1, an interrupt is generated if a TX history list entry lost condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CMPOFIE bit (CAN-FD Message Payload Overflow Flag Interrupt Enable)

When the CMPOFIE bit is 1, an interrupt is generated when a CAN-FD message payload overflow condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

QOWEIE bit (TXQ Message Overwrite Error Interrupt Enable)

When the QOWEIE bit is 1, an interrupt is generated when a TXQ message overwrite condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

QMEIE bit (TXQ Message Lost Error Interrupt Enable)

When the QMEIE bit is 1, an interrupt is generated when a TXQ message lost condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

MOWEIE bit (Message Lost Error Interrupt Enable)

When the MOWEIE bit is 1, an interrupt is generated in GW mode and a GW FIFO message over write condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

TSRST bit (Timestamp Reset)

When the TSRST bit is 1, the Global Timestamp Register is reset to 0x0000.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode.

Read value is always 0.

This bit is cleared automatically by the CAN-FD logic.

35.3.13 CFDGSTS : Global Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x008C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	GRAM INIT	GSLP STS	GHLT STS	GRST STS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Bit	Symbol	Function	R/W
0	GRSTSTS	Global Reset Status 0: Not in Reset mode 1: In Reset mode	R
1	GHLTSTS	Global Halt Status 0: Not in Halt mode 1: In Halt mode	R
2	GSLPSTS	Global Sleep Status 0: Not in Sleep mode 1: In Sleep mode	R
3	GRAMINIT	Global RAM Initialization 0: RAM initialization is complete 1: RAM initialization is ongoing	R
31:4	—	These bits are read as 0.	R

The CFDGSTS register indicates the global status of the CAN-FD module.

GRSTSTS bit (Global Reset Status)

The GRSTSTS bit indicates the status of Global CAN-FD module Reset mode.

This bit is set automatically when the CAN-FD module enters GL_RESET mode. When the mode changes from GL_RESET mode to GL_SLEEP mode, this bit remains set.

This bit is cleared automatically when the CAN-FD module exits the GL_RESET mode.

GHLTSTS bit (Global Halt Status)

The GHLTSTS bit indicates the status of Global CAN-FD module Halt mode.

This bit is set automatically when the CAN-FD module enters GL_HALT mode.

This bit is cleared automatically when the CAN-FD module exits the GL_HALT mode.

GSLPSTS bit (Global Sleep Status)

The GSLPSTS bit indicates the status of Global CAN-FD module Sleep mode.

This bit is set automatically when the CAN-FD module enters GL_SLEEP mode.

This bit is cleared automatically when the CAN-FD module exits the GL_SLEEP mode.

GRAMINIT bit (Global RAM Initialization)

The GRAMINIT bit indicates the status of Global CAN-FD module RAM initialization.

This bit is set automatically when the CAN-FD module enters GL_SLEEP mode after a hardware reset.

This bit is cleared automatically when the CAN-FD module completed RAM initialization.

35.3.14 CFDGERFL : Global Error Flag Register

Base address: CANFD = 0x8004_0000

Offset address: 0x0090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	MOWES	QMES	—	QOWES	CMPOF	THLES	MES	DEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DEF	DLC Error Flag 0: DLC error not detected 1: DLC error detected	R/W
1	MES	Message Lost Error Status 0: Message lost error not detected 1: Message lost error detected	R
2	THLES	TX History List Entry Lost Error Status 0: TX history list entry lost error not detected 1: TX history list entry lost error detected	R
3	CMPOF	CAN-FD Message Payload Overflow Flag 0: CAN-FD message payload overflow not detected 1: CAN-FD message payload overflow detected	R/W
4	QOWES	TXQ Message Overwrite Error Status 0: TXQ message overwrite error not detected 1: TXQ message overwrite error detected	R
5	—	This bit is read as 0. The write value should be 0.	R/W
6	QMES	TXQ Message Lost Error Status 0: TXQ message lost error not detected 1: TXQ message lost error detected	R
7	MOWES	Message Overwrite Error Status 0: Message overwrite error not detected 1: Message overwrite error detected	R
15:8	—	These bits are read as 0. The write value should be 0.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGERFL egister indicates the detection of global errors.

DEF bit (DLC Error Flag)

The DEF bit indicates the error status of the DLC.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when a DLC error is detected in a received frame.

The bit is cleared by writing 0 to it.

This bit is cleared automatically in GL_RESET mode.

MES bit (Message Lost Error Status)

The MES bit indicates status of the message lost error.

This bit is set automatically when a FIFO message lost error is detected.

This bit is cleared automatically when:

- All FIFO message lost flags are cleared
- The CAN-FD module is in GL_RESET mode.

THLES bit (TX History List Entry Lost Error Status)

The THLES bit indicates status of the TX history list entry lost error.

This bit is set automatically when a TX history list entry lost error is detected.

This bit is cleared automatically when:

- All TX history list entry lost flags are cleared
- The CAN-FD module is in GL_RESET mode.

CMPOF bit (CAN-FD Message Payload Overflow Flag)

The CMPOF bit is set automatically when a CAN-FD message payload overflow is detected on at least one channel.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode.

This bit is cleared by writing 0 to it. Writing 1 to this bit has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared automatically in GL_RESET mode.

QOWES bit (TXQ Message Overwrite Error Status)

The QOWES bit is set automatically when a TXQ message overwrite error is detected.

This bit is cleared automatically when all TXQ message overwrite flags are cleared.

This bit is cleared automatically in GL_RESET mode.

QMES bit (TXQ Message Lost Error Status)

The QMES bit is set automatically when a TXQ message lost error is detected.

This bit is cleared automatically when all TXQ message lost flags are cleared.

This bit is cleared automatically in GL_RESET mode.

MOWES bit (Message Overwrite Error Status)

The MOWES bit is set automatically when GW mode and COMFIFO message overwrite error is detected.

This bit is cleared automatically when all COMFIFO Message Overwrite flags are cleared.

This bit is cleared automatically in GL_RESET mode.

35.3.15 CFDGTINTSTS0 : Global TX Interrupt Status Register 0

Base address: CANFD = 0x8004_0000

Offset address: 0x1300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CFOTIF1	TQOFIF1	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	CFOTIF0	TQOFIF0	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSIF0	TX Successful Interrupt Flag Channel 0 0: Channel 0 TX Successful Interrupt flag not set 1: Channel 0 TX Successful Interrupt flag set	R
1	TAIF0	TX Abort Interrupt Flag Channel 0 0: Channel 0 TX Abort Interrupt flag not set 1: Channel 0 TX Abort Interrupt flag set	R
2	TQIF0	TX Queue Interrupt Flag Channel 0 0: Channel 0 TX Queue Interrupt flag not set 1: Channel 0 TX Queue Interrupt flag set	R
3	CFTIF0	COM FIFO TX/GW Mode Interrupt Flag Channel 0 0: Channel 0 COM FIFO TX/GW Mode Interrupt flag not set 1: Channel 0 COM FIFO TX/GW Mode Interrupt flag set	R
4	THIF0	TX History List Interrupt Channel 0 0: Channel 0 TX History List Interrupt flag not set 1: Channel 0 TX History List Interrupt flag set	R
5	TQOFIF0	TX Queue One Frame Transmission Interrupt Flag Channel 0 0: Channel 0 TX Queue One Frame Transmission Interrupt flag not set 1: Channel 0 TX Queue One Frame Transmission Interrupt flag set	R
6	CFOTIF0	COM FIFO One Frame Transmission Interrupt Flag Channel 0 0: Channel 0 COM FIFO One Frame Transmission Interrupt flag not set 1: Channel 0 COM FIFO One Frame Transmission Interrupt flag set	R
7	—	This bit is read as 0.	R
8	TSIF1	TX Successful Interrupt Flag Channel 1 0: Channel 1 TX Successful Interrupt flag not set 1: Channel 1 TX Successful Interrupt flag set	R
9	TAIF1	TX Abort Interrupt Flag Channel 1 0: Channel 1 TX Abort Interrupt flag not set 1: Channel 1 TX Abort Interrupt flag set	R
10	TQIF1	TX Queue Interrupt Flag Channel 1 0: Channel 1 TX Queue Interrupt flag not set 1: Channel 1 TX Queue Interrupt flag set	R
11	CFTIF1	COM FIFO TX/GW Mode Interrupt Flag Channel 1 0: Channel 1 COM FIFO TX/GW Mode Interrupt flag not set 1: Channel 1 COM FIFO TX/GW Mode Interrupt flag set	R
12	THIF1	TX History List Interrupt Channel 1 0: Channel 1 TX History List Interrupt flag not set 1: Channel 1 TX History List Interrupt flag set	R
13	TQOFIF1	TX Queue One Frame Transmission Interrupt Flag Channel 1 0: Channel 1 TX Queue One Frame Transmission Interrupt flag not set 1: Channel 1 TX Queue One Frame Transmission Interrupt flag set	R
14	CFOTIF1	COM FIFO One Frame Transmission Interrupt Flag Channel 1 0: Channel 1 COM FIFO One Frame Transmission Interrupt flag not set 1: Channel 1 COM FIFO One Frame Transmission Interrupt flag set	R
31:15	—	These bits are read as 0.	R

The CFDGTINTSTS0 register indicates the detection of transmit specific interrupts.

TSIFn bit (TX Successful Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.TSIFn bit is set to 1 when the TX Successful Interrupt flag of the related channel is set (when the interrupt is enabled).
This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TAIFn bit (TX Abort Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.TAIFn bit is set to 1 when the TX Abort Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TQIFn bit (TX Queue Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.TQIFn bit is set to 1 when the TX Queue Interrupt flag of the related channel is set (when the interrupt is enabled).
This bit is cleared automatically:

- When the related TX Queue Interrupt flag is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

CFTIFn bit (COM FIFO in TX/GW Mode Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.CFTIFn bit is set to 1 when the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

THIFn bit (TX History List Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.THIFn bit is set to 1 when the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TQOFIFn bit (TX Queue One Frame Transmission Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.TQOFIFn bit is set to 1 when the TX Queue One Frame Transmission Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX Queue One Frame Transmission Interrupt flag is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

CFOTIFn bit (COM FIFO One Frame Transmission Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.CFOTIFn bit is set to 1 when the related COM FIFO One Frame Transmission Interrupt flag (CFDCFSTSn.CFOFTXIF) is set (when the Interrupt is enabled).

This bit is cleared automatically:

- When the related COM FIFO One Frame Transmission Interrupt flag (CFDCFSTSn.CFOFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

35.3.16 CFDGTSC : Global Timestamp Counter Register

Base address: CANFD = 0x8004_0000

Offset address: 0x0094

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TS[15:0]	Timestamp value	R
31:16	—	These bits are read as 0.	R

The CFDGTSC register stores the timestamp based on the selected configuration.

TS[15:0] bits (Timestamp value)

The Timestamp value is stored in the Global Timestamp Counter register based on the configuration of TSSS, TSBTCS, and TSP. The proper incrementing of the timestamp counter cannot be guaranteed when transitioning to halt state.

Do not write to bits TS[15:0] when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

The TS[15:0] bits are cleared automatically in GL_RESET mode.

35.3.17 CFDGAFLECTR : Global Acceptance Filter List Entry Control Register

Base address: CANFD = 0x8004_0000

Offset address: 0x0098

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	AFLDAE	—	—	—	—	AFLPN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	AFLPN[3:0]	Acceptance Filter List Page Number Select an Acceptance Filter List page	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	AFLDAE	Acceptance Filter List Data Access Enable 0: Acceptance Filter List data access disabled 1: Acceptance Filter List data access enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGAFLECTR register is used to select the Global Acceptance Filter List page for reading or writing entries into the Global Acceptance Filter List.

AFLPN[3:0] bits (Acceptance Filter List Page Number)

The AFLPN[3:0] bits select the page number to access the desired RAM area of the Acceptance Filter List. One Acceptance Filter List page consists of 16 Acceptance Filter List entries.

Read/write accesses to the Acceptance Filter List can only be performed through a fixed window.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode. Enter only the values between 0x0 and 0xF, inclusive.

AFLDAE bit (Acceptance Filter List Data Access Enable)

The AFLDAE bit prevents write access to the Acceptance Filter List when cleared after configuration of the Acceptance Filter List.

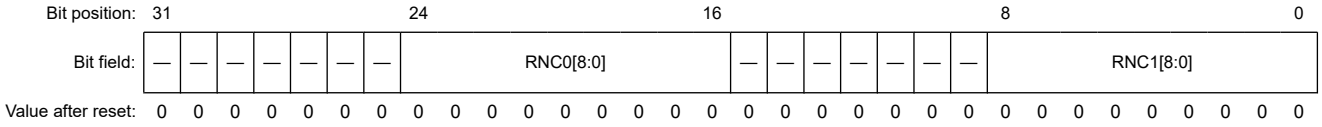
Data can be read from the Acceptance Filter List independent of the status of this bit.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. Set this bit to enable write access for the Acceptance Filter List.

35.3.18 CFDGAFLCFG0 : Global Acceptance Filter List Configuration Register 0

Base address: CANFD = 0x8004_0000

Offset address: 0x009C



Bit	Symbol	Function	R/W
8:0	RNC1[8:0]	Rule Number for Channel 1 Number of rules dedicated to channel 1	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
24:16	RNC0[8:0]	Rule Number for Channel 0 Number of rules dedicated to channel 0	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGAFLCFG0 register is used to define the number of rules for entries in the Acceptance Filter List, applicable for channels 0 to 1.

The total number of available entries in the Acceptance Filter List is 256 for 2 CAN channels. However, the filters can be allocated flexibly to the different channels depending on requirements as long as both of the following conditions are satisfied:

- The maximum number of acceptance filter per channel is 256
- The total number of rules defined for all channels is not exceeding the number of available entries in the Acceptance Filter List.

RNCn[8:0] bits (Rule Number for Channel n (n = 0, 1))

The RNCn[8:0] bits define the number of rules in the Acceptance Filter List for channel n.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

35.3.19 CFDGAFLIDn : Global Acceptance Filter List ID Register n (n = 1 to 16)

Base address: CANFD = 0x8004_0000

Offset address: 0x1800 + 0x10 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	GAFLID[28:0]	Global Acceptance Filter List Entry ID Field ID part of the Global Acceptance Filter List entry	R/W
29	GAFLLB	Global Acceptance Filter List Entry Loopback Configuration 0: Global Acceptance Filter List entry ID for acceptance filtering with attribute RX 1: Global Acceptance Filter List entry ID for acceptance filtering with attribute TX	R/W
30	GAFLRTR	Global Acceptance Filter List Entry RTR Field 0: Data frame 1: Remote frame	R/W
31	GAFLIDE	Global Acceptance Filter List Entry IDE Field 0: Standard identifier of rule entry ID is valid for acceptance filtering 1: Extended identifier of rule entry ID is valid for acceptance filtering	R/W

The CFDGAFLIDn register (n = 1 to 16) is used to configure the ID field for the rules of entries in the Global Acceptance Filter List.

GAFLID[28:0] bits (Global Acceptance Filter List Entry ID Field)

The GAFLID[28:0] bits represent the CAN identifier (ID) field of each entry in the Global Acceptance Filter List.

The acceptance filter process compares this field against the ID of a received CAN message. For alignment of these bits in standard and extended frame formats, see section 35.3.88. Identifier Bits Alignment.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLLB bit (Global Acceptance Filter List Entry Loopback Configuration)

The GAFLLB bit selects whether entry in the Global Acceptance Filter List gets the attribute RX or TX.

This attribute determines the validity of the entry in Mirror mode, Loopback test mode, and during standard (non-loopback) reception. See section 35.6.5. Loopback Modes for detailed description of the validity of the Global Acceptance Filter List entry depending on transmitter/receiver case, the type of loopback mode, and RX/TX attribute.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLRTR bit (Global Acceptance Filter List Entry RTR Field)

The GAFLRTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLIDE bit (Global Acceptance Filter List Entry IDE Field)

The GAFLIDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

35.3.20 CFDGAFLMn : Global Acceptance Filter List Mask Register n (n = 1 to 16)

Base address: CANFD = 0x8004_0000

Offset address: 0x1804 + 0x10 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLIDEM	GAFLRTRM	GAFLIFL1	GAFLIDM[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLIDM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	GAFLIDM[28:0]	Global Acceptance Filter List ID Mask Field Global Acceptance Filter List Mask field bits for ID field	R/W
29	GAFLIFL1	Global Acceptance Filter List Information Label 1 Global Acceptance Filter List information label bit 1	R/W
30	GAFLRTRM	Global Acceptance Filter List Entry RTR Mask 0: RTR bit is not used for ID matching 1: RTR bit is used for ID matching	R/W
31	GAFLIDEM	Global Acceptance Filter List IDE Mask 0: IDE bit is not used for ID matching 1: IDE bit is used for ID matching	R/W

The CFDGAFLMn register (n = 1 to 16) is used to configure the Mask field of each rule for entries in the Global Acceptance Filter List.

GAFLIDM[28:0] bits (Global Acceptance Filter List ID Mask Field)

GAFLIDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Global Acceptance Filter List entry.

0	Corresponding STD-ID/EXT-ID bit is not used for ID matching
1	Corresponding STD-ID/EXT-ID bit is used for ID matching

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLIFL1 bit (Global Acceptance Filter List Information Label 1)

The GAFLIFL1 bit allows the configuration of a 2-bit information label to be attached to a received message accepted by the associated entry in the Global Acceptance Filter List. This bit is a MSB bit of an information label.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

This bit is stored in the Information Label Field [1] (CFDRMFDSTS.RMIFL[1], CFDRFFDSTS.RFIFL[1], CFDCFFDCSTSn.CFIFL[1]) of the storage location of an incoming message.

Note: This bit is stored in CFDTHLACC1n.TIFL[1] when CFDTHLCCn.THLDGE = 1 is set up using Gateway function.

GAFLRTRM bit (Global Acceptance Filter List Entry RTR Mask)

The GAFLRTRM bit allows the configuration of the RTR mask bit for each entry in the Global Acceptance Filter List.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLIDEM bit (Global Acceptance Filter List IDE Mask)

The GAFLIDEM bit allows the configuration of the IDE mask bit for each entry in the Global Acceptance Filter List.

When the IDE mask bit is 0, the ID comparison depends on the received IDE bit.

If the received IDE bit is 0, the STD-ID comparison takes place.

If the received IDE bit is 1, the EXT-ID comparison takes place.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

35.3.21 CFDGAFLP0n : Global Acceptance Filter List Pointer 0 Register n (n = 1 to 16)

Base address: CANFD = 0x8004_0000

Offset address: 0x1808 + 0x10 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFL RMV	—	—	GAFLRMDP[4:0]					GAFLI FL0	GAFL SRD2	GAFL SRD1	GAFL SRD0	GAFLDLC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	GAFLDLC[3:0]	Global Acceptance Filter List DLC Field Minimum number of data bytes in a data frame required for acceptance	R/W
4	GAFLSRD0	Global Acceptance Filter List Select Routing Destination 0 0: Routing target is CFIFO0 1: Routing target is TX Queue 0 instead of CFIFO0	R/W
5	GAFLSRD1	Global Acceptance Filter List Select Routing Destination 1 0: Routing target is CFIFO1 1: Routing target is TX Queue 1 instead of CFIFO1	R/W
6	GAFLSRD2	Global Acceptance Filter List Select Routing Destination 2 0: Routing target is CFIFO2 1: Routing target is TX Queue 2 instead of CFIFO2	R/W
7	GAFLIFL0	Global Acceptance Filter List Information Label 0	R/W
12:8	GAFLRMDP[4:0]	Global Acceptance Filter List RX Message Buffer Direction Pointer RX message buffer number for storage of received messages	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	GAFLRMV	Global Acceptance Filter List RX Message Buffer Valid 0: Single message buffer direction pointer is invalid 1: Single message buffer direction pointer is valid	R/W
31:16	GAFLPTR[15:0]	Global Acceptance Filter List Pointer	R/W

The CFDGAFLP0n register (n = 1 to 16) is used to configure the data length code (DLC), software pointer, single message buffer select, and message buffer direction pointer for each rule entry in the Global Acceptance Filter List.

GAFLDLC[3:0] bits (Global Acceptance Filter List DLC Field)

The GAFLDLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Global Acceptance Filter List (automatic DLC filter function).

DLC filter process is only passed if the DLC value of the message accepted by an entry in the Global Acceptance Filter List is equal to or higher than the DLC value configured for this associated Global Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0.

Table 35.7 shows DLC value that can be configured.

Table 35.7 Configuration of DLC value

Format	DLC[3:0]	Description
CAN and CAN-FD	0000b	DLC of received message = 0 or more (DLC filter check is disabled)
CAN and CAN-FD	0001b	DLC of received message = 1 or more
CAN and CAN-FD	0010b	DLC of received message = 2 or more
CAN and CAN-FD	0011b	DLC of received message = 3 or more
CAN and CAN-FD	0100b	DLC of received message = 4 or more
CAN and CAN-FD	0101b	DLC of received message = 5 or more
CAN and CAN-FD	0110b	DLC of received message = 6 or more
CAN and CAN-FD	0111b	DLC of received message = 7 or more
CAN	1xxxb	DLC of received message = 8 or more
CAN-FD	1000b	DLC of received message = 8 or more*1
CAN-FD	1001b	DLC of received message = 12 or more*1
CAN-FD	1010b	DLC of received message = 16 or more*1
CAN-FD	1011b	DLC of received message = 20 or more*1
CAN-FD	1100b	DLC of received message = 24 or more*1
CAN-FD	1101b	DLC of received message = 32 or more*1
CAN-FD	1110b	DLC of received message = 48 or more*1
CAN-FD	1111b	DLC of received message = 64*1

Note 1. This setting is not available in the classical CAN function.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLSRD0 bit (Global Acceptance Filter List Select Routing Destination 0)

The GAFLSRD0 bit changes a copy destination to CFIFO0 or TXQ0 by routing.

If this bit is set as 1, the preset value of CFDGAFLP1n.GAFLFDP selects TXQ0.

If this bit is set to 0, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO0.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLSRD1 bit (Global Acceptance Filter List Select Routing Destination 1)

The GAFLSRD1 bit changes a copy destination to CFIFO1 or TXQ1 by routing.

If this bit is set to 1, the preset value of CFDGAFLP1n.GAFLFDP selects TXQ1.

If this bit is set to 0, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO1.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLSRD2 bit (Global Acceptance Filter List Select Routing Destination 2)

The GAFLSRD2 bit changes a copy destination to CFIFO2 or TXQ2 by routing.

If this bit is set to 1, the preset value of CFDGAFLP1n.GAFLFDP selects TXQ2.

If this bit is set to 0, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO2.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the CAN-FD module is in CH_RESET or CH_HALT mode.

GAFLIFL0 bit (Global Acceptance Filter List Information Label 0)

The GAFLIFL0 bit allows the configuration of a 2-bit information label that can be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a LSB bit of an information label.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

This bit is stored in Information Label Field[0] (CFDRMFDSTS.RMIFL[0], CFDRFFDSTS.RFIFL[0], CFDCFFDCSTSn.CFIFL[0]) of the storage location of an incoming message.

This bit is stored in CFDTHLACC1n.TIFL[0] when CFDTHLCCn.THLDGE = 1 is set up using the gateway function.

GAFLRMDP[4:0] bits (Global Acceptance Filter List RX Message Buffer Direction Pointer)

The GAFLRMDP[4:0] bits allow the configuration of a single reception message buffer as the destination target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. The value entered is the single destination message buffer number.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

CFDRMNB.NRXMB[7:0] is the value entered in the RX Message Buffer Number Register to configure the number of RX message buffers. The value to be entered in CFDGAFLP0n.GAFLRMDP[4:0] bits should only be between 0x00 and CFDRMNB.NRXMB[7:0] to 1 less.

If CFDRMNB.NRXMB[7:0] = 0x00, the GAFLRMV bit should be configured as 0.

GAFLRMV bit (Global Acceptance Filter List RX Message Buffer Valid)

The GAFLRMV bit allows the enabling or disabling of a single reception message buffer as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLPTR[15:0] bits (Global Acceptance Filter List Pointer)

The GAFLPTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the related Global Acceptance Filter List entry. The pointer is added during message storage in the Message Buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

35.3.22 CFDGAFLP1n : Global Acceptance Filter List Pointer 1 Register n (n = 1 to 16)

Base address: CANFD = 0x8004_0000

Offset address: 0x180C + 0x10 × (n - 1)

Bit position:	31														13																0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAFLFDP[13:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	GAFLFDP[13:0]	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGAFLP1n register (n = 1 to 16) is used to configure the FIFO direction pointer fields in each Rule Entry of the Global Acceptance Filter List.

GAFLFDP[13:0] bits (Global Acceptance Filter List FIFO Direction Pointer)

The GAFLFDP[13:0] bits allow the configuration of FIFO buffers as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. Each bit of the CFDGAFLP1n.GAFLFDP[13:0] is configured as dedicated FIFO.

Bit	Value (binary)	Function
0	0	Disable RX FIFO 0 as target for reception
	1	Enable RX FIFO 0 as target for reception
1	0	Disable RX FIFO 1 as target for reception
	1	Enable RX FIFO 1 as target for reception
2	0	Disable RX FIFO 2 as target for reception
	1	Enable RX FIFO 2 as target for reception
3	0	Disable RX FIFO 3 as target for reception
	1	Enable RX FIFO 3 as target for reception
4	0	Disable RX FIFO 4 as target for reception
	1	Enable RX FIFO 4 as target for reception
5	0	Disable RX FIFO 5 as target for reception
	1	Enable RX FIFO 5 as target for reception
6	0	Disable RX FIFO 6 as target for reception
	1	Enable RX FIFO 6 as target for reception
7	0	Disable RX FIFO 7 as target for reception
	1	Enable RX FIFO 7 as target for reception
8	0	Disable Common FIFO 0 and Channel 0 TX Queue 0 as target for reception
	1	GAFLSRD0 = 0: Enable Common FIFO 0 as target for reception GAFLSRD0 = 1: Enable Channel 0 TX Queue 0 as target for reception
9	0	Disable Common FIFO 1 and Channel 0 TX Queue 1 as target for reception
	1	GAFLSRD1 = 0: Enable Common FIFO 1 as target for reception GAFLSRD1 = 1: Enable Channel 0 TX Queue 1 as target for reception
10	0	Disable Common FIFO 2 and Channel 0 TX Queue 2 as target for reception
	1	GAFLSRD2 = 0: Enable Common FIFO 2 as target for reception GAFLSRD2 = 1: Enable Channel 0 TX Queue 2 as target for reception
11	0	Disable Common FIFO 3 and Channel 1 TX Queue 0 as target for reception
	1	GAFLSRD0 = 0: Enable Common FIFO 3 as target for reception GAFLSRD0 = 1: Enable Channel 1 TX Queue 0 as target for reception
12	0	Disable Common FIFO 4 and Channel 1 TX Queue 1 as target for reception
	1	GAFLSRD1 = 0: Enable Common FIFO 4 as target for reception GAFLSRD1 = 1: Enable Channel 1 TX Queue 1 as target for reception
13	0	Disable Common FIFO 5 and Channel 1 TX Queue 2 as target for reception
	1	GAFLSRD2 = 0: Enable Common FIFO 5 as target for reception GAFLSRD2 = 1: Enable Channel 1 TX Queue 2 as target for reception

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0. Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

For storage in TX queue, TX queue buffers of a target that is in GW mode is possible.

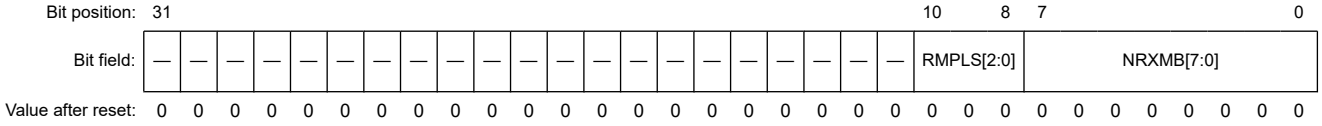
Only one of the following configurations is valid:

- Up to 8 destination FIFO buffers
- 7 destination FIFO buffers plus one RX message buffer
- 8 destination TX queue buffers
- 7 destination TX queue buffers plus one RX message buffer
- A maximum of 8 destinations in all at FIFO buffer and TX queue buffer.

35.3.23 CFDRMNB : RX Message Buffer Number Register

Base address: CANFD = 0x8004_0000

Offset address: 0x00AC



Bit	Symbol	Function	R/W
7:0	NRXMB[7:0]	Number of RX Message Buffers	R/W
10:8	RMPLS[2:0]*1	Reception Message Buffer Payload Data Size 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits are not available in the classical CAN function.

The CFDRMNB register is used to configure the total number of RX message buffers allocated to all channels.

NRXMB[7:0] bits (Number of RX Message Buffers)

The NRXMB[7:0] bits are used to configure the number of RX message buffers.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Enter only values between 0 and 32 inclusive, with 0x00 indicating that no RX message buffer is allocated.

RMPLS[2:0] bits (Reception Message Buffer Payload Data Size)

The RMPLS[2:0] bits are used to configure the message buffer payload data size.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

35.3.24 CFDRMND0 : RX Message Buffer New Data Register 0

Base address: CANFD = 0x8004_0000

Offset address: 0x00B0

Bit position:	31																												0
Bit field:	RMNS[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RMNS[31:0]	RX Message Buffer New Data Status 0: New data not stored in corresponding RX message buffer 1: New data stored in corresponding RX message buffer	R/W

The CFDRMND0 specifies the new data storage status of the RX message buffers.

RMNS[31:0] bits (RX Message Buffer New Data Status)

The RMNSu[31:0] bits indicate the status of new data for the corresponding RX message buffer. RMNS bit [0] corresponds to RX message buffer [0] and so on.

no_of_channels = 2

no_of_CFDRMBCPs_per_channel = (number of RX message buffer components per channel) = 16

no_of_CFDRMBCPs = no_of_channels × no_of_CFDRMBCPs_per_channel (2 * 16 = 32)

no_of_bits_per_register = 32

no_of_CFDRMNDs (number of CFDRMND Registers) = no_of_CFDRMBCPs / no_of_bits_per_register (32 / 32 = 1)

t = [0 ... no_of_CFDRMNDs - 1]

u = [t * 32 ... (no_of_CFDRMBCPs − ((no_of_CFDRMNDs − 1 − t) * 32) − 1)]

t can be calculated from the target of the New Data Status flag (u) using the formula t = floor (u / 32)

Bit position can be calculated using the formula (u − (t * 32))

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode. Writing 1 has no effect.

These bits cannot be cleared when message storage in the corresponding RX message buffer is in progress.
Do not use the bit clear instruction to clear these bits. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

These bits are set automatically when storage of new messages are in the corresponding RX message buffer. These bits are cleared by writing 0. These bits are cleared automatically when the CAN-FD module is in GL_RESET mode.

When CFDRMNB.RMPLS = 000b (maximum 8 bytes payload), the duration of message storage is 6 PCLKM cycles.

When CFDRMNB.RMPLS > 000b, the duration of message storage is 6 PCLKM cycles + 1 for each 4 bytes (maximum of 20 PCLKM cycles for 64 bytes).

Note: This feature is not available in the classical CAN function.

35.3.25 CFDRFCCn : RX FIFO Configuration/Control Register n (n = 0 to 7)

Base address: CANFD = 0x8004_0000

Offset address: 0x00C0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	RFPLS[2:0]			—	—	RFIE	RFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFE	RX FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W
1	RFIE	RX FIFO Interrupt Enable 0: FIFO interrupt generation disabled 1: FIFO interrupt generation enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	RFPLS[2:0]*1	Rx FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	RFDC[2:0]	RX FIFO Depth Configuration 0 0 0: FIFO Depth = 0 messages 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 messages 1 0 1: FIFO Depth = 48 messages 1 1 0: FIFO Depth = 64 messages 1 1 1: FIFO Depth = 128 messages	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	RFIM	RX FIFO Interrupt Mode 0: Interrupt generated when RX FIFO counter reaches RFIGCV value from values smaller than RFIGCV 1: Interrupt generated at the end of every received message storage	R/W
15:13	RFIGCV[2:0]	RX FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
16	RFFIE	RX FIFO Full Interrupt Enable 0: FIFO interrupt generation disabled 1: FIFO interrupt generation enabled	R/W

Bit	Symbol	Function	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits are not available in the classical CAN function.

The CFDRFCCn registers (n = 0 to 7) are used to configure and control the eight RX FIFOs.

RFE bit (RX FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0, the RX FIFO is cleared to empty.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

This bit can only be set if the configured FIFO depth is greater than 000b (CFDRFCCn.RFDC[2:0] > 000b).

Set the RFE bit with a separate write access to the CFDRFCCn register, after all the other bits in the CFDRFCCn register are set.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

RFIE bit (RX FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

RFPLS[2:0] bits (Rx FIFO Payload Data Size Configuration)

The RFPLS[2:0] bits define the message data payload allocation in the RAM.

This is the maximum number of bytes which can be received by this FIFO.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

RFDC[2:0] bits (RX FIFO Depth Configuration)

The RFDC[2:0] bits select the depth of the FIFO in terms of the number of messages. If the FIFO depth is configured to 0 messages, the FIFO cannot be used.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

RFIM bit (RX FIFO Interrupt Mode)

The RFIM bit selects the interrupt generation condition for the FIFO.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

RFIGCV[2:0] bits (RX FIFO Interrupt Generation Counter Value)

The RFIGCV[2:0] bits select the counter value of the FIFO for generation of FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

The setting of the RFIGCV[2:0] bits should be synchronized with the RFDC[2:0] bits.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

RFFIE bit (RX FIFO Full Interrupt Enable)

The RFFIE bit enables generation of the RX FIFO full interrupt. Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

The following content shows examples of interruptions:

- Interruption output in number of arbitrary stages (CFDRFCCn.RFIGCV)
- Interruption output in FIFO full state.

Note: Management of the receiving data of FIFO can be performed by these notices of interruption.

35.3.26 CFDRFSTSn : RX FIFO Status Register n (n = 0 to 7)

Base address: CANFD = 0x8004_0000

Offset address: 0x00E0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFMC[7:0]								—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	RFEMP	RX FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	RFFLL	RX FIFO Full 0: FIFO not full 1: FIFO full	R
2	RFMLT	RX FIFO Message Lost 0: No message lost in FIFO 1: FIFO message lost	R/W
3	RFIF	RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied 1: FIFO interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
15:8	RFMC[7:0]	RX FIFO Message Count Number of messages stored in FIFO	R
16	RFFIF	RX FIFO Full Interrupt Flag 0: FIFO full interrupt condition not satisfied 1: FIFO full interrupt condition satisfied	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The CFDRFSTSn register (n = 0 to 7) shows the status of messages stored in the corresponding FIFO buffers.

RFEMP bit (RX FIFO Empty)

The RFEMP bit is set automatically when:

- The RFMC bit is 0
- RX FIFO is disabled by setting the CFDRFCCn.RFE bit to 0
- The CAN-FD module is in GL_RESET mode.

The RFEMP bit is cleared automatically when the first message is stored in the RX FIFO buffer.

RFFLL bit (RX FIFO Full)

The RFFLL bit is set automatically when the number of CAN messages stored in the FIFO buffer matches the configured FIFO depth.

The RFFLL is cleared automatically when:

- The number of CAN messages stored in the FIFO buffer is less than the configured FIFO depth
- RX FIFO is disabled by setting the CFDRFCCn.RFE bit to 0
- The CAN-FD module is in GL_RESET mode.

RFMLT bit (RX FIFO Message Lost)

Only write to the RFMLT bit when CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically whenever a message is lost due to attempted storage when the FIFO buffer is already full. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode.

RFIF bit (RX FIFO Interrupt Flag)

The RFIF bit is set automatically when the configured interrupt condition is satisfied. This bit is not automatically cleared when the RX FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

The bit is cleared by writing 0 to it. The bit is also cleared when CAN-FD module is in GL_RESET mode.

RFMC[7:0] bits (RX FIFO Message Count)

The RFMC[7:0] bits indicate the number of CAN messages stored in the RX FIFO buffer that can be read by the CPU.

These bits are cleared automatically when the FIFO is disabled and when the CAN-FD module is in GL_RESET mode.

RFFIF bit (RX FIFO Full Interrupt Flag)

The RFFIF bit is not cleared automatically when the RX FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the FIFO full interrupt condition is satisfied. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared by writing 0 to it.

The bit is also cleared when the CAN-FD module is in GL_RESET mode.

35.3.27 CFDRFPCTRn : RX FIFO Pointer Control Register n (n = 0 to 7)

Base address: CANFD = 0x8004_0000

Offset address: 0x0100 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFPC[7:0]	RX FIFO Pointer Control Increments read pointer of the corresponding RX FIFO buffers	R/W
31:8	—	The write value should be 0.	R/W

The CFDRFPCTRn register (n = 0 to 7) can be used to increment the read pointer of the corresponding RX FIFO buffers.

RFPC[7:0] bits (RX FIFO Pointer Control)

When the value 0xFF is written to the RFPC bits, the pointer of the corresponding RX FIFO buffer is moved to the next FIFO entry. Only write 0xFF to these registers when the corresponding RX FIFO buffer is enabled and not empty.

The read value from these bits is always 0x00.

Only write to these bits when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

Do not write to the RX FIFO Pointer Control registers when DMA is enabled.

35.3.28 CFDCFCCn : Common FIFO Configuration/Control Register n (n = 0 to 5)

Base address: CANFD = 0x8004_0000

Offset address: 0x0120 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFITT[7:0]								CFDC[2:0]			CFTML[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CFIGCV[2:0]			CFIM	CFITR	CFITSS	CFM[1:0]		—	CFPLS[2:0]			—	CFTXIE	CFRXIE	CFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFE	Common FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W
1	CFRXIE	Common FIFO RX Interrupt Enable 0: FIFO interrupt generation disabled for Frame RX 1: FIFO interrupt generation enabled for Frame RX	R/W
2	CFTXIE	Common FIFO TX Interrupt Enable 0: FIFO interrupt generation disabled for Frame TX 1: FIFO interrupt generation enabled for Frame TX	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CFPLS[2:0]*1	Common FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
9:8	CFM[1:0]	Common FIFO Mode 0 0: RX FIFO mode 0 1: TX FIFO mode 1 0: CAN – CAN GW FIFO mode 1 1: Reserved	R/W
10	CFITSS	Common FIFO Interval Timer Source Select 0: Reference clock (× 1 / × 10 period) 1: Bit time clock of related channel (FIFO is linked to fixed channel)	R/W
11	CFITR	Common FIFO Interval Timer Resolution 0: Reference clock period × 1 1: Reference clock period × 10	R/W

Bit	Symbol	Function	R/W
12	CFIM	Common FIFO Interrupt Mode 0: RX FIFO mode: RX interrupt generated when Common FIFO counter reaches CFIGCV value from a lower value TX FIFO mode: TX interrupt generated when Common FIFO transmits the last message successfully GW FIFO mode: For RX interrupt flag: Interrupt generated when FIFO counter increments and reaches the value configured in CFIGCV For TX interrupt flag: Interrupt generated when FIFO transmits the last message successfully 1: RX FIFO mode: RX interrupt generated at the end of every received message storage TX FIFO mode: TX interrupt generated for every successfully transmitted message GW FIFO mode: For RX interrupt flag: Interrupt generated when a message is stored in the FIFO For TX interrupt flag: Interrupt generated when a message is successfully transmitted from the FIFO	R/W
15:13	CFIGCV[2:0]	Common FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
20:16	CFTML[4:0]	Common FIFO TX Message Buffer Link Transmission scan link position of the corresponding channel	R/W
23:21	CFDC[2:0]	Common FIFO Depth Configuration 0 0 0: FIFO Depth = 0 messages 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 messages 1 0 1: FIFO Depth = 48 messages 1 1 0: FIFO Depth = 64 messages 1 1 1: FIFO Depth = 128 messages	R/W
31:24	CFITT[7:0]	Common FIFO Interval Transmission Time Delay the start of transmission from the FIFO if configured in TX or GW mode, delay is a multiple of basic Interval Timer Clock Source unit	R/W

Note 1. These bits are not available in the classical CAN function.

The CFDCFCCn register (n = 0 to 5) is used to configure the Common FIFOs.

CFE bit (Common FIFO Enable)

The CFE bit enables the FIFO when set. FIFO is disabled when this bit is cleared.

This bit can also be used, by clearing it, to abort transmission from Common FIFO when configured in TX mode or GW mode, or to stop reception into the Common FIFO in RX mode.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.

This bit can only be set if the configured FIFO depth is greater than 0 (CFDC bit > 0).

Set the CFE bit with a separate write access to the CFDCFCCn register, after all the other bits in this register are set.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

This bit is also cleared automatically when the related channel is in CH_RESET mode if the FIFO is configured in TX or GW mode.

CFRXIE bit (Common FIFO RX Interrupt Enable)

The CFRXIE bit enables generation of FIFO interrupts when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CFTXIE bit (Common FIFO TX Interrupt Enable)

The CFTXIE bit enables generation of common FIFO interrupts when the interrupt flag is set after transmission of a frame from the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CFPLS[2:0] bits (Common FIFO Payload Data Size Configuration)

The CFPLS[2:0] bits define the message data payload allocation in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.

For details, see section 35.7. FIFO Buffers and Normal Message Buffer Configuration.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

CFM[1:0] bits (Common FIFO Mode)

The CFM[1:0] bits select the mode of the FIFO. When a hardware reset is applied, all the Common FIFO buffers are configured in RX FIFO mode. Do not configure these bits to 11b.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

CFITSS bit (Common FIFO Interval Timer Source Select)

The CFITSS bit selects the basic clock source for the Interval Transmission Timer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. In addition, do not write to this bit when the CFE bit is set to 1.

Do not write 1 to this bit when CAN-FD communication is used.*1

Note: The bit time clock can vary depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

CFITR bit (Common FIFO Interval Timer Resolution)

The CFITR bit selects the resolution of the reference clock for the Interval Transmission Timer (peripheral clock is the source for the reference clock).

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1.

CFIM bit (Common FIFO Interrupt Mode)

The CFIM bit selects the interrupt generation condition for the FIFO buffer.

Do not write to this bit in GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

CFIGCV[2:0] bits (Common FIFO Interrupt Generation Counter Value)

The CFIGCV[2:0] bits select the message counter value for the generation of FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

The setting of these bits should be synchronized with the CFDC[2:0] bits.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

CFTML[4:0] bits (Common FIFO TX Message Buffer Link)

The CFTML[4:0] bits select the normal transmit message buffer position where the TX or GW FIFO is linked to, for transmission scanning.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

CFDC[2:0] bits (Common FIFO Depth Configuration)

The CFDC[2:0] bits select the depth of the common FIFO in terms of the number of messages. If the FIFO depth is configured to 0 message, the FIFO cannot be used.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

CFITT[7:0] bits (Common FIFO Interval Transmission Time)

The CFITT[7:0] bits select the delay in the start of transmission for all messages transmitted from this FIFO buffer when configured in TX or GW mode. The delay is a multiple of the basic interval timer clock source period (reference clock × 1, reference clock × 10, or bit time clock of the related CAN channel).

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

Do not write to these bits when the CFE bit is set to 1.

When CFDGCFG.ITRCP[15:0] = 0x0000, set the CFITT[7:0] bits to 0x0000.

35.3.29 CFDCFCCEn : Common FIFO Configuration/Control Enhancement Register n (n = 0 to 5)

Base address: CANFD = 0x8004_0000

Offset address: 0x0180 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFBME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFMOWM	—	—	—	—	—	CFOFTXIE	CFOFRXIE	CFFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFFIE	Common FIFO Full Interrupt Enable 0: FIFO Interrupt generation disabled 1: FIFO Interrupt generation enabled	R/W
1	CFOFRXIE	Common FIFO One Frame Reception Interrupt Enable 0: One Frame RX interrupt generation disabled 1: One Frame RX interrupt generation enabled	R/W
2	CFOFTXIE	Common FIFO One Frame Transmission Interrupt Enable 0: One Frame TX interrupt generation disabled 1: One Frame TX interrupt generation enabled	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	CFMOWM	Common FIFO Message Overwrite Mode 0: Message discarded mode 1: Message overwrite mode	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	CFBME	Common FIFO Buffering Mode Enable 0: Transmission from Common FIFO 1: Transmission halt from Common FIFO	R/W

Bit	Symbol	Function	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCFCCEn register (n = 0 to 5) is used to configure the Common FIFOs.

CFFIE bit (Common FIFO Full Interrupt Enable)

The CFFIE bit enables generation of the FIFO full interrupt when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

The following content shows examples of interruptions:

1. Interruption output in number of arbitrary stages (CFDCFCCn.CFIGCV).
2. Interruption output in FIFO full state.

Management of the receiving data of FIFO can be performed by these notices of interruption.

CFOFRXIE bit (Common FIFO One Frame Reception Interrupt Enable)

The CFOFRXIE bit enables generation of the one frame reception interrupt when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CFOFTXIE bit (Common FIFO One Frame Transmission Interrupt Enable)

The CFOFTXIE bit enables generation of the one frame transmission interrupt when the interrupt flag is set after transmission of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CFMOWM bit (Common FIFO Message Overwrite Mode)

When the CFMOWM bit is 0, a receiving message is discarded and FIFO is full. When the CFMOWM bit is 1, a receiving message is overwritten and FIFO is full.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

Only write 1 to this bit when the common FIFO is in GW mode.

Do not write 1 to this bit when the CFE bit is 1.

CFBME bit (Common FIFO Buffering Mode Enable)

When the CFBME bit is 0, messages are transmitted from FIFO. When the CFBME bit is 1, messages are not transmitted from FIFO.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. Additionally, do not write 1 to this bit when the CFE bit is 1.

35.3.30 CFDCFSTSn : Common FIFO Status Register n (n = 0 to 5)

Base address: CANFD = 0x8004_0000

Offset address: 0x01E0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	CFMOW	—	—	—	—	—	CFOFTXIF	CFOFRXIF	CFFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CFMC[7:0]								—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	CFEMP	Common FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	CFFLL	Common FIFO Full 0: FIFO not full 1: FIFO full	R
2	CFMLT	Common FIFO Message Lost 0: FIFO message not lost 1: FIFO message lost	R/W
3	CFRXIF	Common RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame reception 1: FIFO interrupt condition satisfied after frame reception	R/W
4	CFTXIF	Common TX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame transmission 1: FIFO Interrupt condition satisfied after frame transmission	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	CFMC[7:0]	Common FIFO Message Count Number of messages stored in FIFO	R
16	CFFIF	Common FIFO Full Interrupt Flag 0: Interrupt condition not satisfied for FIFO full interrupt 1: Interrupt condition satisfied for FIFO full interrupt	R/W
17	CFOFRXIF	Common FIFO One Frame Reception Interrupt Flag For each FIFO that receives a frame, a corresponding interrupt is set.	R/W
18	CFOFTXIF	Common FIFO One Frame Transmission Interrupt Flag For each FIFO that transmits a frame, a corresponding interrupt is set.	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
24	CFMOW	Common FIFO Message Overwrite 0: No message overwrite occurred in FIFO 1: Message overwrite occurred in FIFO	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCFSTSn register (n = 0 to 5) shows the status of messages stored in the corresponding FIFO buffers.

CFEMP bit (Common FIFO Empty)

The CFEMP bit is set automatically when:

- The CPU has read all messages from the FIFO configured in RX mode
- All messages have been transmitted from the FIFO configured in TX or GW mode
- The FIFO is disabled by setting the CFE bit to 0
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET when FIFO configured in TX or GW mode.

The CFEMP bit is cleared automatically when:

- The first reception message is stored in the FIFO buffer when configured in RX mode
- The first message to be transmitted is stored in the FIFO buffer when configured in TX or GW mode.

CFFLL bit (Common FIFO Full)

The CFFLL bit is set automatically when the number of CAN messages stored in the FIFO matches the configured FIFO depth.

The CFFLL bit is cleared automatically when:

- The number of CAN messages stored in the FIFO is less than the configured FIFO depth
- The FIFO is disabled by setting the CFE bit to 0

- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode when FIFO buffer is configured in TX or GW mode.

CFMLT bit (Common FIFO Message Lost)

The CFMLT bit is set automatically whenever a message is lost due to attempted storage of a new message when FIFO is already full in RX or GW mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The CFMLT bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFRXIF bit (Common RX FIFO Interrupt Flag)

The CFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers when configured in GW mode or RX mode.

The CFRXIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in GW mode.

CFTXIF bit (Common TX FIFO Interrupt Flag)

The CFTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO buffer configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers configured in GW or TX mode.

The CFTXIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFMC[7:0] bits (Common FIFO Message Count)

The CFMC[7:0] bits indicate the following:

- Number of CAN messages stored by the CPU in the FIFO buffer configured in TX mode pending for transmission
- Number of CAN messages stored in the FIFO buffer configured in RX mode by CAN-FD to be read by the CPU
- Number of CAN messages stored by the CAN-FD in the GW FIFO pending for transmission.

The CFMC[7:0] bits are cleared automatically when:

- The FIFO is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFFIF bit (Common FIFO Full Interrupt Flag)

The CFFIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the FIFO Full Interrupt condition is satisfied for Common FIFO buffers when configured in GW or RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The CFFIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFOFRXIF bit (Common FIFO One Frame Reception Interrupt Flag)

The CFOFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for the FIFO buffer configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the One Frame Reception Interrupt condition is satisfied for the Common FIFO buffers when configured in GW or RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The CFOFRXIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFOFTXIF bit (Common FIFO One Frame Transmission Interrupt Flag)

The CFOFTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the One Frame Transmission Interrupt condition is satisfied for Common FIFO buffers configured in GW mode or TX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO is configured in TX or GW mode.

CFMOW bit (Common FIFO Message Overwrite)

The CFMOW bit is set automatically whenever a message is an overwrite storage of a new message when CFDCFCCEn.CFMOWM = 1 and FIFO is already full in GW mode.

If the set from the CAN channel occurs simultaneously with the clear by a write access, the bit is set.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO buffer configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The CFMOW bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

35.3.31 CFDCFPCTRn : Common FIFO Pointer Control Register n (n = 0 to 5)

Base address: CANFD = 0x8004_0000

Offset address: 0x0240 + 0x04 × n

Bit position:	31																								7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CFPC[7:0]	Common FIFO Pointer Control Increments read or write pointer of the corresponding Common FIFO buffers depending on the mode configuration.	W
31:8	—	The write value should be 0.	W

The CFDCFPCTRn register (n = 0 to 5) can be used to increment the read or write pointer of the corresponding Common FIFO buffer.

CFPC[7:0] bits (Common FIFO Pointer Control)

When the value 0xFF is written into the CFPC[7:0] bits, the read pointer of the corresponding Common FIFO buffer (when configured in RX mode), or the write pointer of the corresponding Common FIFO buffer (when configured in TX mode) moves to the next FIFO entry.

The read value from these bits is always 0x00.

Only write to these bits when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

Only write 0xFF to this register when:

- The Common FIFO buffer is enabled and is not empty if configured in RX mode
- The Common FIFO buffer is enabled and is not full if configured in TX mode
- The Common FIFO buffer is enabled and is not configured in GW mode.

Do not write to the Common FIFO Pointer Control registers when DMA is enabled.

35.3.32 CFDFESTS : FIFO Empty Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x02A0

Bit position:	31																	13					8	7						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXEMP[5:0]						RFXEMP[7:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	RFXEMP[7:0]	RX FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
13:8	CFXEMP[5:0]	Common FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
31:14	—	These bits are read as 0.	R

The CFDFESTS register shows status of the empty bits of the FIFO buffers.

RFXEMP[7:0] bits (RX FIFO Empty Status)

Bit [7] (RFXEMP[7]) is associated with FIFO index 7 and bit [0] (RFXEMP[0]) is associated with FIFO index 0.

The RFXEMP[7:0] bits are set when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFXEMP[5:0] bits (Common FIFO Empty Status)

Bit [13] (CFXEMP[5]) is associated with common FIFO index 5 and bit [8] (CFXEMP[0]) is associated with common FIFO index 0.

The CFXEMP[5:0] bits are set when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

35.3.33 CFDFFSTS : FIFO Full Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x02A4

Bit position:	31																	13					8	7						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXFLL[5:0]						RFXFLL[7:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFXFLL[7:0]	RX FIFO Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
13:8	CFXFLL[5:0]	Common FIFO Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
31:14	—	These bits are read as 0.	R

The CFDFFSTS register shows status of the full bits of the FIFO buffers.

RFXFLL[7:0] bits (RX FIFO Full Status)

The RFXFLL[7:0] bits are cleared when CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFXFLL[5:0] bits (Common FIFO Full Status)

The CFXFLL[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

35.3.34 CFDFMSTS : FIFO Message Lost Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x02A8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXMLT[5:0]						RFXMLT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFXMLT[7:0]	RX FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
13:8	CFXMLT[5:0]	Common FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
31:14	—	These bits are read as 0.	R

The CFDFMSTS register shows status of the message lost bits of the FIFO buffers.

RFXMLT[7:0] bits (RX FIFO Message Lost Status)

The RFXMLT[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFXMLT[5:0] bits (Common FIFO Message Lost Status)

The CFXMLT[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

35.3.35 CFDRFISTS : RX FIFO Interrupt Flag Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x02AC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	RFXFFLL[7:0]								—	—	—	—	—	—	—	—	RFXIF[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFXIF[7:0]	RX FIFO[x] Interrupt Flag Status 0: Corresponding RX FIFO Interrupt flag not set 1: Corresponding RX FIFO Interrupt flag set	R
15:8	—	These bits are read as 0.	R
23:16	RFXFFLL[7:0]	RX FIFO[x] Interrupt Full Flag Status 0: Corresponding RX FIFO Interrupt Full flag not set 1: Corresponding RX FIFO Interrupt Full flag set	R
31:24	—	These bits are read as 0.	R

The CFDRFISTS register shows status of the interrupt flag bits of the RX FIFO buffers.

RFXIF[7:0] bits (RX FIFO[x] Interrupt Flag Status)

Each bit is set automatically when the corresponding interrupt flag bit is set in the RX FIFO Status Registers.

The RFXIF[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding interrupt flag bit is cleared in the RX FIFO Status Registers.

RFXFFLL[7:0] bits (RX FIFO[x] Interrupt Full Flag Status)

Each bit is set automatically when the corresponding interrupt full flag bit is set in the RX FIFO Status Registers.

The RFXFFLL[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding interrupt full flag bit is cleared in the RX FIFO Status Registers.

35.3.36 CFDCFRISTS : Common FIFO RX Interrupt Flag Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x02B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXRXIF[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CFXRXIF[5:0]	Common FIFO RX Interrupt Flag Status 0: Corresponding Common FIFO RX Interrupt flag not set 1: Corresponding Common FIFO RX Interrupt flag set	R
31:6	—	These bits are read as 0.	R

The CFDCFRISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

CFXRXIF[5:0] bits (Common FIFO RX Interrupt Flag Status)

Each bit is set automatically when the corresponding RX interrupt flag bit is set in the Common FIFO Status Registers.

The CFXRXIF[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding RX interrupt flag bit is cleared in the Common FIFO Status Registers.

35.3.37 CFDCFTISTS : Common FIFO TX Interrupt Flag Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x02B4

Bit position:	31																								5						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXTXIF[5:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CFXTXIF[5:0]	Common FIFO TX Interrupt Flag Status 0: Corresponding Common FIFO TX Interrupt flag not set 1: Corresponding Common FIFO TX Interrupt flag set	R
31:6	—	These bits are read as 0.	R

The CFDCFTISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

CFXTXIF[5:0] bits (Common FIFO TX Interrupt Flag Status)

Each bit is set automatically when the corresponding TX interrupt flag bit is set in the Common FIFO Status Registers.

The CFXTXIF[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding TX interrupt flag bit is cleared in the Common FIFO Status Registers.

35.3.38 CFDFFFSTS : FIFO FDC Full Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x02C4

Bit position:	31																		13					8	7						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXFFLL[5:0]						RFXFFLL[7:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFXFFLL[7:0]	RX FIFO FDC Level Full Status 0: Corresponding FIFO full interrupt not set 1: Corresponding FIFO full interrupt set	R
13:8	CFXFFLL[5:0]	COMMON FIFO FDC Level Full Status 0: Corresponding FIFO full interrupt not set 1: Corresponding FIFO full interrupt set	R
31:14	—	These bits are read as 0.	R

The CFDFFFSTS register shows status of the full interrupt flag bits of the FIFO buffers.

RFXFFLL[7:0] bits (RX FIFO FDC Level Full Status)

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

The RFXFFLL[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

CFXFFLL[5:0] bits (COMMON FIFO FDC Level Full Status)

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

The CFXFFLL[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

35.3.39 CFDCFMOWSTS : Common FIFO Message Overwrite Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x02C0

Bit position:	31																								5	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXMOW[5:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CFXMOW[5:0]	Common FIFO Massage Overwrite Status 0: Corresponding FIFO Overwrite flag not set 1: Corresponding FIFO Overwrite flag set	R
31:6	—	These bits are read as 0.	R

The CFDCFMOWSTS register shows the status of the Interrupt Flag bits of the Common FIFO Buffers.

CFXMOW[5:0] bits (Common FIFO Massage Overwrite Status)

The CFXMOW[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode. This register is only valid in GW mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

35.3.40 CFDCFOFRISTS : Common FIFO One Frame RX Interrupt Flag Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x02B8

Bit position:	31																								5	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXOFRXIF[5:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CFXOFRXIF[5:0]	Common FIFO One Frame RX Interrupt Flag Status 0: Corresponding Common FIFO One Frame RX Interrupt flag not set 1: Corresponding Common FIFO One Frame RX Interrupt flag set	R
31:6	—	These bits are read as 0.	R

The CFDCFOFRISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

CFXOFRXIF[5:0] bits (Common FIFO One Frame RX Interrupt Flag Status)

Each bit is set automatically when the corresponding One Frame RX Interrupt flag bit is set in the Common FIFO Status Registers.

The CFXOFRXIF[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding One Frame RX Interrupt flag bit is cleared in the Common FIFO Status Registers.

35.3.41 CFDCFOFTISTS : Common FIFO One Frame TX Interrupt Flag Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x02BC

Bit position:	31																								5	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXOFTXIF[5:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CFXOFTXIF[5:0]	Common FIFO One Frame TX Interrupt Flag Status 0: Corresponding Common FIFO One Frame TX Interrupt flag not set 1: Corresponding Common FIFO One Frame TX Interrupt flag set	R
31:6	—	These bits are read as 0.	R

The CFDCFOFTISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

CFXOFTXIF[5:0] bits (Common FIFO One Frame TX Interrupt Flag Status)

Each bit is set automatically when the corresponding One Frame TX Interrupt flag bit is set in the Common FIFO Status Registers.

The CFXOFTXIF[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding One Frame TX Interrupt flag bit is cleared in the Common FIFO Status Registers.

35.3.42 CFDCDTCT : DMA Transfer Control Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1330

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFDMAE1	CFDMAE0	RFDMAE7	RFDMAE6	RFDMAE5	RFDMAE4	RFDMAE3	RFDMAE2	RFDMAE1	RFDMAE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMAE0	DMA Transfer Enable for RX FIFO 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
1	RFDMAE1	DMA Transfer Enable for RX FIFO 1 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
2	RFDMAE2	DMA Transfer Enable for RX FIFO 2 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
3	RFDMAE3	DMA Transfer Enable for RX FIFO 3 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W

Bit	Symbol	Function	R/W
4	RFDMAE4	DMA Transfer Enable for RX FIFO 4 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
5	RFDMAE5	DMA Transfer Enable for RX FIFO 5 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
6	RFDMAE6	DMA Transfer Enable for RX FIFO 6 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
7	RFDMAE7	DMA Transfer Enable for RX FIFO 7 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
8	CFDMAE0	DMA Transfer Enable for Common FIFO 0 of Channel 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
9	CFDMAE1	DMA Transfer Enable for Common FIFO 0 of Channel 1 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCDTCT register controls the start and stop of DMA transfer operation.

RFDMAEn (n = 0 to 7) bit (DMA Transfer Enable for RX FIFOn)

Number of RX FIFOs = 8

The RFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

CFDMAEn (n = 0, 1) bit (DMA Transfer Enable for Common FIFO 0 of Channel 0, 1)

The CFDMAEn bit enables or disables DMA transfer request for common FIFO 0 of channel 0 or 1. Only Common FIFO 0 can be linked to a DMA channel with this bit. Common FIFO 1 cannot be linked to a DMA channel.

To link Common FIFO 2, see bit CFDCDTTCT.CFDMAEn in section 35.3.44. CFDCDTTCT : DMA TX Transfer Control Register.

The CFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

Do not enable a DMA transfer for a Common FIFO that is configured as TX or GW FIFO.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

35.3.43 CFDCDTSTS : DMA Transfer Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1334

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFDMASTS1	CFDMASTS0	RFDMASTS7	RFDMASTS6	RFDMASTS5	RFDMASTS4	RFDMASTS3	RFDMASTS2	RFDMASTS1	RFDMASTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMASTS0	DMA Transfer Status for RX FIFO 0 0: DMA transfer stopped 1: DMA transfer on going	R
1	RFDMASTS1	DMA Transfer Status for RX FIFO 1 0: DMA transfer stopped 1: DMA transfer on going	R
2	RFDMASTS2	DMA Transfer Status for RX FIFO 2 0: DMA transfer stopped 1: DMA transfer on going	R
3	RFDMASTS3	DMA Transfer Status for RX FIFO 3 0: DMA transfer stopped 1: DMA transfer on going	R
4	RFDMASTS4	DMA Transfer Status for RX FIFO 4 0: DMA transfer stopped 1: DMA transfer on going	R
5	RFDMASTS5	DMA Transfer Status for RX FIFO 5 0: DMA transfer stopped 1: DMA transfer on going	R
6	RFDMASTS6	DMA Transfer Status for RX FIFO 6 0: DMA transfer stopped 1: DMA transfer on going	R
7	RFDMASTS7	DMA Transfer Status for RX FIFO 7 0: DMA transfer stopped 1: DMA transfer on going	R
8	CFDMASTS0	DMA Transfer Status only for Common FIFO 0 of Channel 0 0: DMA transfer stopped 1: DMA transfer on going	R
9	CFDMASTS1	DMA Transfer Status only for Common FIFO 0 of Channel 1 0: DMA transfer stopped 1: DMA transfer on going	R
31:10	—	These bits are read as 0.	R

The CFDCDTSTS register shows the status of the DMA transfer.

RFDMASTSn (n = 0 to 7) bit (DMA Transfer Status for RX FIFO n)

Number of RX FIFOs = 8

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.RFDMAEn (see CFDCDTCT.RFDMAEn bit in section 35.3.42. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the RFDMASTSn bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

CFDMASTSn bit (DMA Transfer Status only for Common FIFO 0 of Channel 0, 1)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.CFDMAEn (see CFDCDTCT.CFDMAEn bit in section 35.3.42. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the CFDMASTSn bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

35.3.44 CFDCDTTCT : DMA TX Transfer Control Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1340

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDMAE1	CFDMAE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TQ3DMAE1	TQ3DMAE0	—	—	—	—	—	—	TQ0DMAE1	TQ0DMAE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TQ0DMAE0	DMA TX Transfer Enable for TXQ 0 of Channel 0 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
1	TQ0DMAE1	DMA TX Transfer Enable for TXQ 0 of Channel 1 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	TQ3DMAE0	DMA TX Transfer Enable for TXQ 3 of Channel 0 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
9	TQ3DMAE1	DMA TX Transfer Enable for TXQ 3 of Channel 1 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	CFDMAE0	DMA TX Transfer Enable for Common FIFO 2 of Channel 0 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
17	CFDMAE1	DMA TX Transfer Enable for Common FIFO 2 of Channel 1 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCDTTCT register controls the start and stop of DMA transfer operation.

TQ0DMAEn (n = 0, 1) bit (DMA TX Transfer Enable for TXQ 0 of Channel n)

The TQ0DMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

TQ3DMAEn (n = 0, 1) bit (DMA TX Transfer Enable for TXQ 3 of Channel n)

The TQ3DMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

CFDMAEn (n = 0, 1) bit (DMA TX Transfer Enable for Common FIFO 2 of Channel n)

The CFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

Only common FIFO 2 can be linked to a DMA channel with this bit. Common FIFO 1 cannot be linked to a DMA channel.

To link Common FIFO 0, see CFDCDTCT.CFDMAEn bit in section 35.3.42. CFDCDTCT : DMA Transfer Control Register.

Do not enable a DMA transfer for a Common FIFO that is configured as RX or GW FIFO.

The CFDMAEn bit is cleared when the CAN-FD module is in GL_RESET mode.

35.3.45 CFDCDTTSTS : DMA TX Transfer Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1344

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDMASTS1	CFDMASTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TQ3DMASTS1	TQ3DMASTS0	—	—	—	—	—	—	TQ0DMASTS1	TQ0DMASTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TQ0DMASTS0	DMA TX Transfer Status for TXQ0 of Channel 0 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
1	TQ0DMASTS1	DMA TX Transfer Status for TXQ0 of Channel 1 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
7:2	—	These bits are read as 0.	R
8	TQ3DMASTS0	DMA TX Transfer Status for TXQ3 of Channel 0 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
9	TQ3DMASTS1	DMA TX Transfer Status for TXQ3 of Channel 1 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
15:10	—	These bits are read as 0.	R
16	CFDMASTS0	DMA TX Transfer Status for Common FIFO 2 of Channel 0 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
17	CFDMASTS1	DMA TX Transfer Status for Common FIFO 2 of Channel 1 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
31:18	—	These bits are read as 0.	R

The CFDCDTTSTS register shows the status of the DMA TX transfer.

TQ0DMASTSn (n = 0, 1) bit (DMA TX Transfer Status for TXQ 0 of Channel n)

The TQ0DMASTSn bit is set when the CFDCDTTCT.TQ0DMAEn bit in the corresponding CFDCDTTCT register is set (see section 35.3.44. CFDCDTTCT : DMA TX Transfer Control Register).

This bit is cleared when:

- The CFDCDTTCT.TQ0DMAEn bit in the corresponding CFDCDTTCT register is cleared
- The CAN-FD module is in GL_RESET mode.

TQ3DMASTSn (n = 0, 1) bit (DMA TX Transfer Status for TXQ 3 of Channel n)

The TQ3DMASTSn bit is set when the CFDCDTTCT.TQ3DMAEn bit in the corresponding CFDCDTTCT register is set (see section 35.3.44. CFDCDTTCT : DMA TX Transfer Control Register).

This bit is cleared when:

- The CFDCDTTCT.TQ3DMAEn bit in the corresponding CFDCDTTCT register is cleared

- The CAN-FD module is in GL_RESET mode.

CFDMASTSn (n = 0, 1) bit (DMA TX Transfer Status only for Common FIFO 2 of Channel n)

The CFDMASTSn bit is set when the CFDCDTTCT.CFDMAEn bit in the corresponding CFDCDTTCT register is set (see section 35.3.44. CFDCDTTCT : DMA TX Transfer Control Register).

This bit is cleared when:

- The CFDCDTTCT.CFDMAEn bit in the corresponding CFDCDTTCT register is cleared
- The CAN-FD module is in GL_RESET mode.

35.3.46 CFDGRINTSTSn : Global RX Interrupt Status Register n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1350 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CFOFRIF[2:0]			—	CFRFIF[2:0]			—	—	—	—	—	CFRIF[2:0]			—	—	BQOFRIF[1:0]		—	QOFRIF[2:0]			—	BQFIF[1:0]			—	—	QFIF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	QFIF[2:0]	TXQ Full Interrupt Flag Channel n (n = 0, 1) 0: Corresponding TXQ Full Interrupt flag is not set 1: Corresponding TXQ Full Interrupt flag is set	R
3	—	This bit is read as 0.	R
5:4	BQFIF[1:0]	Borrowed TXQ Full Interrupt Flag Channel n (n = 0, 1) 0: Corresponding TXQ Full Interrupt flag is not set 1: Corresponding TXQ Full Interrupt flag is set	R
7:6	—	These bits are read as 0.	R
10:8	QOFRIF[2:0]	TXQ One Frame RX Interrupt Flag Channel n (n = 0, 1) 0: Corresponding TXQ One Frame RX Interrupt flag is not set 1: Corresponding TXQ One Frame RX Interrupt flag is set	R
11	—	This bit is read as 0.	R
13:12	BQOFRIF[1:0]	Borrowed TXQ One Frame RX Interrupt Flag Channel n (n = 0, 1) 0: Corresponding TXQ One Frame RX Interrupt flag is not set 1: Corresponding TXQ One Frame RX Interrupt flag is set	R
15:14	—	These bits are read as 0.	R
18:16	CFRIF[2:0]	Common FIFO RX Interrupt Flag Channel n (n = 0, 1) 0: Corresponding Common FIFO RX Interrupt flag is not set 1: Corresponding Common FIFO RX Interrupt flag is set	R
23:19	—	These bits are read as 0.	R
26:24	CFRFIF[2:0]	Common FIFO FDC Level Full Interrupt Flag Channel n (n = 0, 1) 0: Corresponding Common FIFO Full Interrupt flag is not set 1: Corresponding Common FIFO Full Interrupt flag is set	R
27	—	This bit is read as 0.	R
30:28	CFOFRIF[2:0]	Common FIFO One Frame RX Interrupt Flag Channel n (n = 0, 1) 0: Corresponding Common FIFO One Frame RX Interrupt flag is not set 1: Corresponding Common FIFO One Frame RX Interrupt flag is set	R
31	—	This bit is read as 0.	R

QFIF[2:0] bits (TXQ Full Interrupt Flag Channel n (n = 0, 1))

The QFIF[2:0] bits are set automatically when the TXQ Full Interrupt flag of the related channel is set when the interrupt is enabled.

The QFIF[2:0] bits are cleared automatically when:

- The related TXQ result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

BQFIF[1:0] bits (Borrowed TXQ Full Interrupt Flag Channel n (n = 0, 1))

The BQFIF[1:0] bits are set when a flexible transmission buffer assignment function is used and the borrowed TXQ is in full status. Operation is the same as CFDGRINTSTSn.QFIF.

The bit of the channel that lends TXMB is a reserved bit.

QOFRIF[2:0] bits (TXQ One Frame RX Interrupt Flag Channel n (n = 0, 1))

The QOFRIF[2:0] bits are set automatically when the TXQ One Frame RX Interrupt flag of the related channel is set when the interrupt is enabled.

The QOFRIF[2:0] bits are cleared automatically when:

- The related TXQ result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET or CH_RESET mode.

BQOFRIF[1:0] bits (Borrowed TXQ One Frame RX Interrupt Flag Channel n (n = 0, 1))

The BQOFRIF[1:0] bits are set when a flexible transmission buffer assignment function is used and the borrowed TXQ receives one frame. Operation is the same as CFDGRINTSTSn.QOFRIF.

The bit of the channel that lends TXMB is a reserved bit.

CFRIF[2:0] bits (Common FIFO RX Interrupt Flag Channel n (n = 0, 1))

The CFRIF[2:0] bits are set automatically when the Common FIFO RX Interrupt flag of the related channel is set when the interrupt is enabled.

The CFRIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET or CH_RESET mode.

CFRFIF[2:0] bits (Common FIFO FDC Level Full Interrupt Flag Channel n (n = 0, 1))

The CFRFIF[2:0] bits are set automatically when the Common FIFO Full Interrupt flag of the related channel is set when the interrupt is enabled.

The CFRFIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

CFOFRIF[2:0] bits (Common FIFO One Frame RX Interrupt Flag Channel n (n = 0, 1))

The CFOFRIF[2:0] bits are set automatically when the Common FIFO One Frame RX Interrupt flag of the related channel is set when the interrupt is enabled.

The CFOFRIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

35.3.47 CFDTMCn : TX Message Buffer Control Register n (n = 0 to 127)

Base address: CANFD = 0x8004_0000

Offset address: 0x02D0 + 0x01 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMTR	TX Message Buffer Transmission Request 0: TX Message buffer transmission not requested 1: TX message buffer transmission requested	R/W
1	TMTAR	TX Message Buffer Transmission Abort Request 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R/W
2	TMOM	TX Message Buffer One-shot Mode 0: TX message buffer not configured in one-shot mode 1: TX message buffer configured in one-shot mode	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTMCn register (n = 0 to 127) configures the TX message buffer functions.

TMTR bit (TX Message Buffer Transmission Request)

When the TMTR bit is set, the CAN-FD module logic tries to transmit the message stored in the corresponding message buffer.

Only write to this bit when the related CAN-FD module is in CH_HALT or CH_OPERATION mode.

Do not set this bit if the corresponding TX message buffer is linked to a COM FIFO in TX or GW mode or is a part of TX Queue.

This bit cannot be directly cleared by a CPU write access.

This bit can only be set when the Transmission Result flag bits (CFDTMSTSn.TMTRF[1:0]) in the CFDTMSTSn register corresponding to the message buffer are cleared to 00b.

The TMTR bit is automatically cleared by the:

- CAN-FD module logic at the end of a successful transmission
- CAN-FD module logic at the end of a transmission abort, requested by the corresponding CFDTMCn.TMTAR bit
- CAN-FD module logic when there is a detection of a CAN bus error or arbitration loss if CFDTMCn.TMOM bit is set for the message buffer
- CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

TMTAR bit (TX Message Buffer Transmission Abort Request)

When the TMTAR bit is set, the CAN-FD module logic tries to abort the transmission of the frame stored in the corresponding message buffer.

In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame may be transmitted successfully from the message buffer. The message buffer selection is released by entering CH_HALT mode.

However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (RX pin) before it starts transmission from the selected message buffer.

Only write to the TMTAR bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. This bit can only be set when the related transmit request TMTR bit is set.

The TMTAR bit cannot be cleared by a CPU write access. Clearing of this bit by CAN-FD has priority over setting by a CPU write access.

The TMTAR bit is automatically cleared by:

- The CAN-FD module logic at the end of a successful transmission
- The CAN-FD module logic at the end of a transmission abort
- The CAN-FD module logic when there is detection of a CAN bus error or arbitration loss
- The CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel enters CH_RESET mode.

TMOM bit (TX Message Buffer One-shot Mode)

When the TMOM bit is set, the CAN-FD module logic tries to transmit the message only once.

If the transmission is successful, the CFDTMSTSn.TMTRF[1:0] bits are set to 10b or 11b. Otherwise, the transmission is automatically aborted and CFDTMSTSn.TMTRF[1:0] bits are set to 01b due to a bus error or a bus arbitration lost.

The TMOM bit remains set if the transmission has completed successfully or aborted due to an error or a loss of arbitration.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Set this bit at the same time as the TMTR bit. Clear this bit with a write access.

If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.

The TMOM bit is automatically cleared by the CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

35.3.48 CFDTMSTSn : TX Message Buffer Status Register n (n = 0 to 127)

Base address: CANFD = 0x8004_0000

Offset address: 0x07D0 + 0x01 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMTSTS	TX Message Buffer Transmission Status 0: No on-going transmission 1: On-going transmission	R
2:1	TMTRF[1:0]	TX Message Buffer Transmission Result Flag 0 0: No result 0 1: Transmission aborted from the TX message buffer 1 0: Transmission successful from the TX message buffer and transmission abort was not requested 1 1: Transmission successful from the TX message buffer and transmission abort was requested	R/W
3	TMTRM	TX Message Buffer Transmission Request Mirrored 0: TX message buffer transmission not requested 1: TX message buffer transmission requested	R
4	TMTARM	TX Message Buffer Transmission Abort Request Mirrored 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTMSTSn register (n = 0 to 127) shows status of the transmission and transmission abort for the corresponding message buffers.

TMTSTS bit (TX Message Buffer Transmission Status)

The TMTSTS bit is set automatically at the start of the transmission from the corresponding TX message buffer.

This bit is cleared automatically when:

- Transmission stops
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

TMTRF[1:0] bits (TX Message Buffer Transmission Result Flag)

The TMTRF[1:0] bits show the result for the corresponding TX message buffer. The status is as follows:

- 00b: Transmission in progress or has not been requested
- 01b: Transmission has been aborted from the corresponding TX message buffer
- 10b: Transmission was successful from the corresponding TX message buffer and the CFDTMCn.TMTAR bit was not set for this TX message buffer
- 11b: Transmission was successful from the corresponding TX message buffer, but the CFDTMCn.TMTAR bit was set for this TX message buffer.

Only write to these bits when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

The TMTRF[1:0] bits are cleared automatically when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

TMTRM bit (TX Message Buffer Transmission Request Mirrored)

The TMTRM bit is set when the CFDTMCn.TMTR bit in the corresponding CFDTMCn register is set.

This bit is cleared when the CFDTMCn.TMTR bit in the corresponding CFDTMCn register is cleared.

TMTARM bit (TX Message Buffer Transmission Abort Request Mirrored)

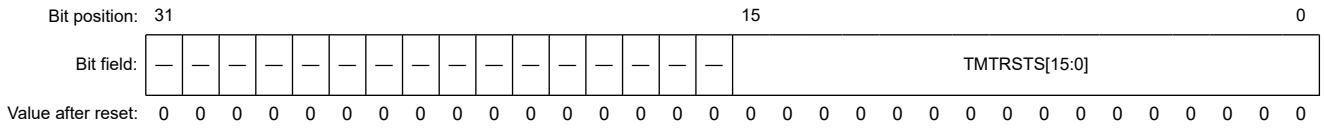
The TMTARM bit is set when the CFDTMCn.TMTAR bit in the corresponding CFDTMCn register is set.

This bit is cleared when the CFDTMCn.TMTAR bit in the corresponding CFDTMCn register is cleared.

35.3.49 CFDTMTRSTSf : TX Message Buffer Transmission Request Status Register f (f = 0 to 3)

Base address: CANFD = 0x8004_0000

Offset address: 0x0CD0 + 0x04 × f



Bit	Symbol	Function	R/W
15:0	TMTRSTS[15:0]	TX Message Buffer Transmission Request Status 0: Transmission not requested for corresponding TX message buffer 1: Transmission requested for corresponding TX message buffer	R
31:16	—	These bits are read as 0.	R

TMTRSTS[15:0] bits (TX Message Buffer Transmission Request Status)

The TMTRSTS[15:0] bits show status of the CFDTMCn.TMTR bits of the TX Message Buffer Control Registers.

Alignment of the TMTRSTS[15:0] bits is shown in Table 35.8.

Table 35.8 Alignment of TMTRSTS[15:0] mirror bits

Bit position	TX message buffer number
n × 64 - fmin × 32	n × 64 + 0
n × 64 + 1 - fmin × 32	n × 64 + 1
⋮	⋮
n × 64 + 31 - fmin × 32	n × 64 + 31
n × 64 + 32 - fmax × 32	n × 64 + 32
n × 64 + 33 - fmax × 32	n × 64 + 33
⋮	⋮
n ×64 + 62 - fmax × 32	n × 64 + 62
n × 64 + 63 - fmax × 32	n × 64 + 63

Note: When n = 0, fmin = 0, fmax = 1
When n = 1, fmin = 2, fmax = 3

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers (CFDTMCn), and only when the message buffer does not belong to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

35.3.50 CFDTMTARSTSf : TX Message Buffer Transmission Abort Request Status Register f (f = 0 to 3)

Base address: CANFD = 0x8004_0000

Offset address: 0x0D70 + 0x04 × f

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMTARSTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TMTARSTS[15:0]	TX Message Buffer Transmission Abort Request Status 0: Transmission abort not requested for corresponding TX message buffer 1: Transmission abort requested for corresponding TX message buffer	R
31:16	—	These bits are read as 0.	R

TMTARSTS[15:0] bits (TX Message Buffer Transmission Abort Request Status)

The TMTARSTS[15:0] bits show status of the CFDTMCn.TMTAR bits of the TX Message Buffer Control Registers.

Alignment of the TMTARSTS[15:0] bits is shown in Table 35.9.

Table 35.9 Alignment of TMTARSTS[15:0] mirror bits (1 of 2)

Bit position	TX message buffer number
n × 64 - fmin × 32	n × 64 + 0
n × 64 + 1 - fmin × 32	n × 64 + 1
⋮	⋮
n × 64 + 31 - fmin × 32	n × 64 + 31
n × 64 + 32 - fmax × 32	n × 64 + 32
n × 64 + 33 - fmax × 32	n × 64 + 33

Table 35.9 Alignment of TMTARSTS[15:0] mirror bits (2 of 2)

Bit position	TX message buffer number
⋮	⋮
n × 64 + 62 - fmax × 32	n × 64 + 62
n × 64 + 63 - fmax × 32	n × 64 + 63

Note: When n = 0, fmin = 0, fmax = 1
When n = 1, fmin = 2, fmax = 3

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers, and when the message buffer belongs to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

35.3.51 CFDTMTCSTSf : TX Message Buffer Transmission Completion Status Register f (f = 0 to 3)

Base address: CANFD = 0x8004_0000

Offset address: 0x0E10 + 0x04 × f

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMTCSTS[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TMTCSTS[15:0]	TX Message Buffer Transmission Completion Status 0: Transmission not completed for corresponding TX message buffer 1: Transmission completed for corresponding TX message buffer	R
31:16	—	These bits are read as 0.	R

TMTCSTS[15:0] bits (TX Message Buffer Transmission Completion Status)

The TMTCSTS[15:0] bits show status of successful completion of the TX Message Buffer Status Registers.

Alignment of the TMTCSTS[15:0] bits is shown in Table 35.10.

Table 35.10 Alignment of TMTCSTS[15:0] mirror bits

Bit position	TX message buffer number
n × 64 - fmin × 32	n × 64 + 0
n × 64 + 1 - fmin × 32	n × 64 + 1
⋮	⋮
n × 64 + 31 - fmin × 32	n × 64 + 31
n × 64 + 32 - fmax × 32	n × 64 + 32
n × 64 + 33 - fmax × 32	n × 64 + 33
⋮	⋮
n × 64 + 62 - fmax × 32	n × 64 + 62
n × 64 + 63 - fmax × 32	n × 64 + 63

Note: When n = 0, fmin = 0, fmax = 1
When n = 1,fmin = 2, fmax = 3

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Status Registers.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Status Registers
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

If a CAN channel enters CH_RESET mode, then the bits related to that channel are cleared.

35.3.52 CFDTMTASTSf : TX Message Buffer Transmission Abort Status Register f (f = 0 to 3)

Base address: CANFD = 0x8004_0000

Offset address: 0x0EB0 + 0x04 × f

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMTASTS[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TMTASTS[15:0]	TX Message Buffer Transmission Abort Status 0: Transmission not aborted for corresponding TX message buffer 1: Transmission aborted for corresponding TX message buffer	R
31:16	—	These bits are read as 0.	R

TMTASTS[15:0] bits (TX Message Buffer Transmission Abort Status)

The TMTASTS[15:0] bits show status of the successful transmission abort of the corresponding TX message buffer.

Alignment of the TMTASTS[15:0] bits is shown in Table 35.11.

Table 35.11 Alignment of TMTASTS[15:0] mirror bits

Bit position	TX message buffer number
n × 64 - fmin × 32	n × 64 + 0
n × 64 + 1 - fmin × 32	n × 64 + 1
⋮	⋮
n × 64 + 31 - fmin × 32	n × 64 + 31
n × 64 + 32 - fmax × 32	n × 64 + 32
n × 64 + 33 - fmax × 32	n × 64 + 33
⋮	⋮
n × 64 + 62 - fmax × 32	n × 64 + 62
n × 64 + 63 - fmax × 32	n × 64 + 63

Note: When n = 0, fmin = 0, fmax = 1
When n = 1, fmin = 2, fmax = 3

Each bit is set automatically when the CFDTMSTSn.TMTRF[1:0] bits are set to 01b in the corresponding TX Message Buffer Status Register.

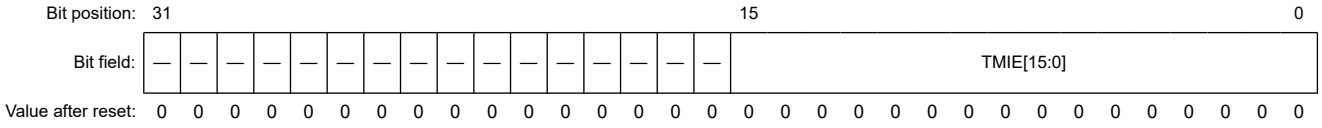
Each bit is cleared automatically when:

- The CFDTMSTSn.TMTRF[1:0] bits are cleared in the corresponding TX Message Buffer Status Register
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

35.3.53 CFDTMIECf : TX Message Buffer Transmission Interrupt Enable Register f (f = 0 to 3)

Base address: CANFD = 0x8004_0000

Offset address: 0x0F50 + 0x04 × f



Bit	Symbol	Function	R/W
15:0	TMIE[15:0]	TX Message Buffer Interrupt Enable 0: TX message buffer interrupt disabled for corresponding TX message buffer 1: TX message buffer interrupt enabled for corresponding TX message buffer	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

TMIE[15:0] bits (TX Message Buffer Interrupt Enable)

If the TMIE[15:0] bits are set, an interrupt is generated at the end of a successful transmission from the corresponding message buffer.

See section 35.8. Interrupt and DMA for TX Message Buffer Interrupt specification.

Alignment of the TMIE[15:0] bits is shown in Table 35.12.

Table 35.12 Alignment of TMIE[15:0] bits

Bit position	TX message buffer number
n × 64 - fmin × 32	n × 64 + 0
n × 64 + 1 - fmin × 32	n × 64 + 1
⋮	⋮
n × 64 + 31 - fmin × 32	n × 64 + 31
n × 64 + 32 - fmax × 32	n × 64 + 32
n × 64 + 33 - fmax × 32	n × 64 + 33
⋮	⋮
n × 64 + 62 - fmax × 32	n × 64 + 62
n × 64 + 63 - fmax × 32	n × 64 + 63

Note: When n = 0, fmin = 0, fmax = 1
When n = 1, fmin = 2, fmax = 3

Do not write to the TMIE[15:0] bits when:

- The CAN-FD module is in GL_SLEEP mode
- The related CAN-FD channel is in CH_SLEEP mode
- The corresponding TX message buffer is part of a TX Queue
- The corresponding TX message buffer is linked to a Common FIFO with the CFDCFCCn.CFTML[4:0] bits.

35.3.54 CFDTXQCC0n : TX Queue Configuration/Control Register 0n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1000 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	TXQOFRXIE	TXQFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	TXQGWE	TXQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
1	TXQGWE	TX Queue Gateway Mode Enable 0: TX Queue GW mode disabled 1: TX Queue GW mode enabled	R/W
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
12:8	TXQDC[4:0]	TX Queue Depth Configuration 0x00: 0 messages 0x01: Reserved 0x02: 3 messages 0x03: 4 messages ⋮ 0x0F: 16 messages Others: Setting prohibited	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIE	TXQ Full Interrupt Enable 0: TX Queue full interrupt generation disabled 1: TX Queue full interrupt generation enabled	R/W
17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable 0: One Frame RX interrupt generation disabled 1: One Frame RX interrupt generation enabled	R/W
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX interrupt generation disabled 1: One Frame TX interrupt generation enabled	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQCC0n register (n = 0, 1) is used to configure the TX Queue transmission.

TXQ0 is composed of TXMB0 to TXMB31 (at the maximum) when TXQE is enabled.

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC0n.TXQDC[4:0] = 0x00).

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

The TXQE bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue GW mode.

When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC0n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC0n.TXQE bit is 1.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[0] up to MB[31] depending on the configured depth.

When using TXQ1 and TXQ0 simultaneously, the total depth of TXQ1 and TXQ0 should be 32 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQFIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit in Gateway mode (CFDTXQCC0n.TXQGWE = 1).

TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQOFRXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit in GW mode (CFDTXQCC0n.TXQGWE = 1).

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

35.3.55 CFDTXQCC1n : TX Queue Configuration/Control Register 1n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1060 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	TXQOFRXIE	TXQFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	TXQGWE	TXQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
1	TXQGWE	TX Queue Gateway Mode Enable 0: TX Queue GW mode disabled 1: TX Queue GW mode enabled	R/W
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
12:8	TXQDC[4:0]	TX Queue Depth Configuration 0x00: 0 messages 0x01: Reserved 0x02: 3 messages 0x03: 4 messages ⋮ 0x0F: 16 messages Others: Setting prohibited	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIE	TXQ Full Interrupt Enable 0: TX Queue full interrupt generation disabled 1: TX Queue full interrupt generation enabled	R/W
17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable 0: One Frame RX interrupt generation disabled 1: One Frame RX interrupt generation enabled	R/W
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX interrupt generation disabled 1: One Frame TX interrupt generation enabled	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQCC1n register (n = 0, 1) is used to configure the TX Queue transmission.

TXQ1 is composed of TXMB31 to TXMB0 (at the maximum) when TXQE is enabled.

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC1n.TXQDC[4:0] = 0x00).

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue Gateway mode.

When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC1n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC1n.TXQE bit is 1.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[31] down to MB[0] depending on the configured depth.

When using TXQ1 and TXQ0 simultaneously, the total depth of TXQ1 and TXQ0 should be 32 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQFIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit when in Gateway mode (CFDTXQCC1n.TXQGWE = 1).

TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQOFRXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit when in Gateway mode (CFDTXQCC1n.TXQGWE = 1).

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

35.3.56 CFDTXQCC2n : TX Queue Configuration/Control Register 2n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x10C0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	TXQOFRXIE	TXQFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	TXQGWE	TXQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
1	TXQGWE	TX Queue Gateway Mode Enable 0: TX Queue GW mode disabled 1: TX Queue GW mode enabled	R/W
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
12:8	TXQDC[4:0]	TX Queue Depth Configuration 0x00: 0 messages 0x01: Reserved 0x02: 3 messages 0x03: 4 messages ⋮ 0x0F: 16 messages Others: Setting prohibited	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIE	TXQ Full Interrupt Enable 0: TX Queue full interrupt generation disabled 1: TX Queue full interrupt generation enabled	R/W
17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable 0: One Frame RX interrupt generation disabled 1: One Frame RX interrupt generation enabled	R/W
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX interrupt generation disabled 1: One Frame TX interrupt generation enabled	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQCC2n register (n = 0, 1) is used to configure the TX Queue transmission.

TXQ2 is composed of TXMB32 to TXMB63 (at the maximum) when TXQE is enabled.

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC2n.TXQDC[4:0] = 0x00).

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue Gateway mode. When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC2n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC2n.TXQE bit is 1.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[32] up to MB[63] depending on the configured depth.

When using TXQ3 and TXQ2 simultaneously, the total depth of TXQ3 and TXQ2 should be 32 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION

TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQFIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit when in GW mode (CFDTXQCC2n.TXQGWE = 1).

TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQOFRXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit when in GW mode (CFDTXQCC2n.TXQGWE = 1).

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

35.3.57 CFDTXQCC3n : TX Queue Configuration/Control Register 3n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1120 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	—	TXQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
12:8	TXQDC[4:0]	TX Queue Depth Configuration 0x00: 0 messages 0x01: Reserved 0x02: 3 messages 0x03: 4 messages ⋮ 0x0F: 16 messages Others: Setting prohibited	R/W
17:13	—	These bits are read as 0. The write value should be 0.	R/W
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX interrupt generation disabled 1: One Frame TX interrupt generation enabled	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQCC3n register (n = 0, 1) is used to configure the TX Queue transmission.

TXQ3 is composed of TXMB63 to TXMB32 (at the maximum) when TXQE is enabled.

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC3n.TXQDC[4:0] = 0x00).

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC3n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC3n.TXQE bit is 1.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[63] down to MB[32] depending on the configured depth.

When using TXQ3 and TXQ2 simultaneously, the total depth of TXQ3 and TXQ2 should be 32 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS3n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

35.3.58 CFDTXQSTS0n : TX Queue Status Register 0n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1020 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	TXQMLT	TXQOFTXIF	TXQOFRXIF	TXQFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TXQMC[5:0]	TX Queue Message Count Number of messages in the TX Queue.	R
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIF	TXQ Full Interrupt Flag When TXQ is in full status, an interrupt is set.	R/W

Bit	Symbol	Function	R/W
17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag When TXQ receives one frame, an interrupt is set.	R/W
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag When one frame transmits from TXQ, an interrupt is set.	R/W
19	TXQMLT	TXQ Message Lost 0: No message lost in TXQ 1: TXQ message lost	R/W
20	TXQMOW	TXQ Message Overwrite 0: No message overwrite in TXQ 1: Message overwrite in TXQ	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQSTS0n register (n = 0, 1) shows the status of the TX Queue of corresponding CAN channel.

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Only in Gateway mode (CFDTXQCC0n.TXQGWE = 1) that this bit is set automatically when the TX Queue transits to a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in Gateway mode of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in Gateway mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC0n.TXQOWE = 1 and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

35.3.59 CFDTXQSTS1n : TX Queue Status Register 1n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1080 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	TXQMLT	TXQOFTXIF	TXQOFRXIF	TXQFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TXQMC[5:0]	TX Queue Message Count Number of messages in the TX Queue	R
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIF	TXQ Full Interrupt Flag When TXQ is in full status, an interrupt is set.	R/W
17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag When TXQ receives one frame, an interrupt is set.	R/W
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag When one frame transmits from TXQ, an interrupt is set.	R/W
19	TXQMLT	TXQ Message Lost 0: No message lost in TXQ 1: TXQ message lost	R/W
20	TXQMOW	TXQ Message Overwrite 0: No message overwrite in TXQ 1: Message overwrite in TXQ	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQSTS1n register (n = 0, 1) shows the status of the TX Queue of corresponding CAN channel.

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Only in Gateway mode (CFDTXQCC1n.TXQGWE = 1) that this bit is set automatically when TX Queue transits to a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in Gateway mode of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in Gateway mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC1n.TXQOWE = 1 and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

35.3.60 CFDTXQSTS2n : TX Queue Status Register 2n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x10E0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	TXQMLT	TXQOFTXIF	TXQOFRXIF	TXQFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TXQMC[5:0]	TX Queue Message Count Number of messages in the TX Queue.	R
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIF	TXQ Full Interrupt Flag When TXQ is in full status, an interrupt is set.	R/W
17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag When TXQ receives one frame, an interrupt is set.	R/W
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag When one frame transmits from TXQ, an interrupt is set.	R/W
19	TXQMLT	TXQ Message Lost 0: No message lost in TXQ 1: TXQ message lost	R/W
20	TXQMOW	TXQ Message Overwrite 0: No message overwrite in TXQ 1: Message overwrite in TXQ	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQSTS2n register (n = 0, 1) shows the status of the TX Queue of corresponding CAN Channel.

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Only in Gateway mode (CFDTXQCC2n.TXQGWE = 1) that this bit is set automatically when the TX Queue transits to a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in Gateway mode of the TX queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in Gateway mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC2n.TXQOWE = 1 and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

35.3.61 CFDTXQSTS3n : TX Queue Status Register 3n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1140 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	—	TXQOFTXIF	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TXQMC[5:0]	TX Queue Message Count Number of messages in the TX Queue	R
17:14	—	These bits are read as 0. The write value should be 0.	R/W
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag When one frame transmits from TXQ, an interrupt is set.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	TXQMOW	TXQ Message Overwrite 0: No message overwrite in TXQ 1: Message overwrite in TXQ	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQSTS3n register (n = 0, 1) shows the status of the TX Queue of corresponding CAN Channel.

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

This bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC3n.TXQOWE = 1 and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

35.3.62 CFDTXQPCTR0n : TX Queue Pointer Control Register 0n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1040 + 0x04 × n

Bit position:	31																							7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

The CFDTXQPCTR0n register (n = 0, 1) is used to confirm storage of a full message in the corresponding TX Queue buffers.

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

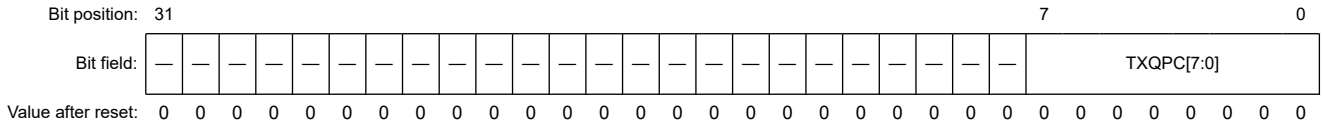
Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

35.3.63 CFDTXQPCTR1n : TX Queue Pointer Control Register 1n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x10A0 + 0x04 × n



Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

The CFDTXQPCTR1n register (n = 0, 1) is used to confirm storage of a full message in the corresponding TX Queue buffers.

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. You cannot write to the FIFO control registers when DMA is enabled.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

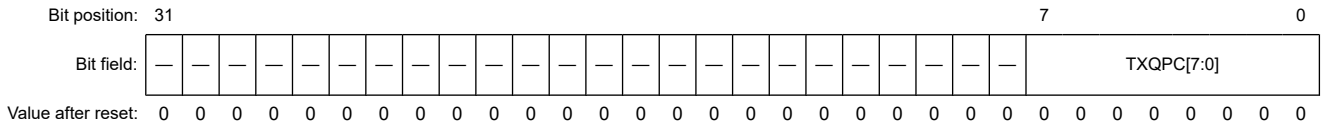
Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

35.3.64 CFDTXQPCTR2n : TX Queue Pointer Control Register 2n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1100 + 0x04 × n



Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

The CFDTXQPCTR2n register (n = 0, 1) is used to confirm storage of a full message in the corresponding TX Queue buffers.

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. You cannot write to the FIFO control registers when DMA is enabled.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

35.3.65 CFDTXQPCTR3n : TX Queue Pointer Control Register 3n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1160 + 0x04 × n

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

The CFDTXQPCTR3n register (n = 0, 1) is used to confirm storage of a full message in the corresponding TX Queue buffers.

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in Gateway mode.

35.3.66 CFDTXQESTS : TX Queue Empty Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1180

Bit position:	31																					8	7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQxEMP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	TXQxEMP[7:0]	TXQ Empty Status 0: TXQ not empty 1: TXQ empty	R
31:8	—	These bits are read as 0.	R

The CFDTXQESTS register shows the status of the empty bits of the TXQ buffers.

TXQxEMP[7:0] bits (TXQ Empty Status)

Each bit is set automatically when the corresponding bit is set in the TX Queue Empty Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Empty Status Register.

This bit is set when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3

35.3.67 CFDTXQFISTS : TX Queue Full Interrupt Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1184

Bit position:	31																								6	4	2	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1FULL[2:0]	—	TXQ0FULL[2:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TXQ0FULL[2:0]	TXQ Full Interrupt Status Flag for Channel 0 0: TXQ full interrupt not set 1: TXQ full interrupt set	R
3	—	This bit is read as 0.	R
6:4	TXQ1FULL[2:0]	TXQ Full Interrupt Status Flag for Channel 1 0: TXQ full interrupt not set 1: TXQ full interrupt set	R
31:7	—	These bits are read as 0.	R

The CFDTXQFISTS register shows the status of the full interrupt bits of the TXQ buffers.

TXQnFULL[2:0] (n = 0, 1) bits (TXQ Full Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Full Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Full Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Reserved
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Reserved

35.3.68 CFDTXQMSTS : TX Queue Message Lost Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1188

Bit position:	31																								6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1ML[2:0]			—	TXQ0ML[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TXQ0ML[2:0]	TXQ Message Lost Status Flag for Channel 0 0: TXQ message lost flag not set 1: TXQ message lost flag set	R
3	—	This bit is read as 0.	R
6:4	TXQ1ML[2:0]	TXQ Message Lost Status Flag for Channel 1 0: TXQ message lost flag not set 1: TXQ message lost flag set	R
31:7	—	These bits are read as 0.	R

The CFDTXQMSTS register shows the status of the message lost bits of the TXQ buffers.

TXQnML[2:0] (n = 0, 1) bits (TXQ Message Lost Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Message Lost Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Message Lost Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Reserved
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Reserved

35.3.69 CFDTXQISTS : TX Queue Interrupt Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1190

Bit position:	31																								15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1ISF[3:0]				TXQ0ISF[3:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	TXQ0ISF[3:0]	TXQ Interrupt Status Flag for Channel 0 0: TXQ Interrupt flag not set 1: TXQ Interrupt flag set	R

Bit	Symbol	Function	R/W
7:4	TXQ1ISF[3:0]	TXQ Interrupt Status Flag for Channel 1 0: TXQ Interrupt flag not set 1: TXQ Interrupt flag set	R
31:8	—	These bits are read as 0.	R

The CFDTXQISTS register shows the status of the interrupt flag of the TXQ buffers.

TXQnISF[3:0] (n = 0, 1) bits (TXQ Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3

35.3.70 CFDTXQOFTISTS : TX Queue One Frame TX Interrupt Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1194

Bit position:	31																								7				3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1OFTISF[3:0]				TXQ0OFTISF[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	TXQ0OFTISF[3:0]	TXQ One Frame TX Interrupt Status Flag for Channel 0 0: TXQ One Frame TX Interrupt flag not set 1: TXQ One Frame TX Interrupt flag set	R
7:4	TXQ1OFTISF[3:0]	TXQ One Frame TX Interrupt Status Flag for Channel 1 0: TXQ One Frame TX Interrupt flag not set 1: TXQ One Frame TX Interrupt flag set	R
31:8	—	These bits are read as 0.	R

The CFDTXQOFTISTS register shows the status of the One Frame TX Interrupt flag of the TXQ buffers.

TXQnOFTISF[3:0] (n = 0, 1) bits (TXQ One Frame TX Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue One Frame TX Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue One Frame TX Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3

35.3.71 CFDTXQOFRISTS : TX Queue One Frame RX Interrupt Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1198

Bit position:	31																								6	4	2	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1OFRISF[2:0]		—	TXQ0OFRISF[2:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TXQ0OFRISF[2:0]	TXQ One Frame RX Interrupt Status Flag for Channel 0 0: TXQ One Frame RX Interrupt flag not set 1: TXQ One Frame RX Interrupt flag set	R
3	—	This bit is read as 0.	R
6:4	TXQ1OFRISF[2:0]	TXQ One Frame RX Interrupt Status Flag for Channel 1 0: TXQ One Frame RX Interrupt flag not set 1: TXQ One Frame RX Interrupt flag set	R
31:7	—	These bits are read as 0.	R

The CFDTXQOFRISTS register shows the status of the One Frame RX Interrupt flag of the TXQ buffers.

TXQnOFRISF[2:0] (n = 0, 1) bits (TXQ One Frame RX Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue One Frame RX Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue One Frame RX Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Reserved
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Reserved

35.3.72 CFDTXQFSTS : TX Queue Full Status Register

Base address: CANFD = 0x8004_0000

Offset address: 0x119C

Bit position:	31																						7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1FSF[3:0]				TXQ0FSF[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	TXQ0FSF[3:0]	TXQ Full Status Flag for Channel 0 0: TXQ Full flag not set 1: TXQ Full flag set	R
7:4	TXQ1FSF[3:0]	TXQ Full Status Flag for Channel 1 0: TXQ Full flag not set 1: TXQ Full flag set	R
31:8	—	These bits are read as 0.	R

The CFDTXQFSTS register shows the status of the Full Status flag bits of the TXQ buffers.

TXQnFSF[3:0] (n = 0, 1) bits (TXQ Full Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Full Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Full Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3

35.3.73 CFDTHLCCn : TX History List Configuration/Control Register n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1200 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	THLDGE	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	THLE	TX History List Enable 0: TX History List disabled 1: TX History List enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	THLIE	TX History List Interrupt Enable 0: TX History List Interrupt disabled 1: TX History List Interrupt enabled	R/W
9	THLIM	TX History List Interrupt Mode 0: Interrupt generated if TX History List level reaches ¾ of the TX History List depth 1: Interrupt generated for every successfully stored entry	R/W
10	THLDTE	TX History List Dedicated TX Enable 0: TX FIFO + TX Queue 1: Flat TX MB + TX FIFO + TX Queue	R/W
11	THLDGE	TX History List Dedicated Gateway Enable 0: Not dedicated Gateway FIFO + Gateway TX Queue 1: Dedicated Gateway FIFO + Gateway TX Queue	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTHLCCn register (n = 0, 1) configures the TX History List functions.

THLE bit (TX History List Enable)

The THLE bit enables the TX History List buffer when it is set.

You cannot write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

THLIE bit (TX History List Interrupt Enable)

The THLIE bit enables the generation of the TX History List interrupt when it is set.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

THLIM bit (TX History List Interrupt Mode)

The THLIM bit selects the interrupt generation condition for the FIFO.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

THLDTE bit (TX History List Dedicated TX Enable)

The THLDTE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

THLDGE bit (TX History List Dedicated Gateway Enable)

The THLDGE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

35.3.74 CFDTHLSTSn : TX History List Status Register n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1220 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	THLMC[5:0]						—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	THLEMP	TX History List Empty 0: TX History List not empty 1: TX History List empty	R
1	THLFLL	TX History List Full 0: TX History List not full 1: TX History List full	R
2	THLELT	TX History List Entry Lost 0: No entry lost in TX History List 1: TX History List entry Lost	R/W
3	THLIF	TX History List Interrupt Flag 0: TX History List interrupt condition not satisfied 1: TX History List interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
13:8	THLMC[5:0]	TX History List Message Count Number of messages stored in TX History List	R
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTHLSTSn register (n = 0, 1) shows the status of data stored in the TX History List buffer.

THLEMP bit (TX History List Empty)

The THLEMP bit is set automatically when the CPU has read all the entries from the TX History List buffer.

This bit is cleared automatically when the first entry is stored to the TX History List.

This bit is set automatically when:

- TX History List is disabled
- The related CAN-FD channel is in CH_RESET mode.

THLFLL bit (TX History List Full)

The THLFLL bit is set automatically when the number of entries in the TX History List buffer matches the TX History List depth.

Each TX History List can store up to 32 entries (each channel has a dedicated TX History List).

This bit is cleared automatically when:

- The number of entries in the TX History List buffer is less than the TX History List depth
- The TX History List is disabled
- The related CAN-FD channel is in CH_RESET mode.

THLELT bit (TX History List Entry Lost)

The THLELT bit is set when a new entry cannot be stored because the related TX History List buffer is already full.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

THLIF bit (TX History List Interrupt Flag)

The THLIF bit is set when the configured interrupt condition is satisfied.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

This bit is automatically cleared in CH_RESET mode.

THLMC[5:0] bits (TX History List Message Count)

The THLMC[5:0] bits show the number of transmitted messages stored in the TX History List.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

35.3.75 CFDTHLACC0n : Channel n TX History List Access Register 0 (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x8000 + 0x08 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TMTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TGW	—	—	—	—	—	BN[6:0]							BT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	BT[2:0]	Buffer Type 0 0 1: Flat TX message buffer 0 1 0: TX FIFO message buffer number and gateway FIFO message number 1 0 0: TX Queue message buffer number Others: Setting prohibited	R
9:3	BN[6:0]	Buffer Number Number of the message buffer	R
14:10	—	These bits are read as 0.	R
15	TGW	Transmit Gateway Buffer Indication 0: No transmission from gateway 1: Transmission from gateway	R
31:16	TMTS[15:0]	Transmit Timestamp Transmit timestamp value for software drivers	R

The CFDTHLACC0n register (n = 0, 1) provides access to the entry in the TX History List based on the read timestamp value.

BT[2:0] bits (Buffer Type)

The BT[2:0] bits indicate whether data has been stored following a transmission from a FIFO buffer, a TX Queue or a TX message buffer.

BN[6:0] bits (Buffer Number)

The BN[6:0] bits show the message buffer from which transmission was successfully completed. If a message from a Common FIFO is transmitted, then these bits show the message buffer that is linked to the Common FIFO for transmission.

TGW bit (Transmit Gateway Buffer Indication)

The TGW bit is automatically set to 1 when transmission is completed in GW mode.

TMTS[15:0] bits (Transmit Timestamp)

The TMTS[15:0] bits indicate the timestamp for use by software drivers.

35.3.76 CFDTHLACC1n : Channel n TX History List Access Register 1 (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x8004 + 0x08 × n

Bit position:	31														17	16	15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TIFL[1:0]			TID[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TID[15:0]	Transmit ID These bits indicate that message buffer reference ID, TX FIFO reference ID, or AFL pointer field is stored for software drivers.	R
17:16	TIFL[1:0]	Transmit Information Label These bits indicate that message buffer information label, TX FIFO information label, or AFL information label is stored for software drivers.	R
31:18	—	These bits are read as 0.	R

The CFDTHLACC1n register (n = 0, 1) provides access to entry in the TX History List based on the read pointer value.

TID[15:0] bits (Transmit ID)

The TID[15:0] bits indicate whether the message buffer reference ID (CFDTMFDCTRn.TMPTR) or the TX FIFO reference ID (CFDCFFDCSTSn.CFPTR) is for use by software drivers.

When transmission in Gateway mode, these bits indicate the AFL pointer field (CFDGAFLP0n.GAFLPTR) instead of the message buffer reference ID (CFDTMFDCTRn.TMPTR).

TIFL[1:0] bits (Transmit Information Label)

The TIFL[1:0] bits indicate whether the message buffer information label (CFDTMFDCTRn.TMIFL) or the TX FIFO information label (CFDCFFDCSTSn.CFIFL) is for use by software drivers.

When transmission in Gateway mode, these bits indicate the AFL pointer field (CFDGAFLMn.GAFLIFL1 and CFDGAFLP0n.GAFLIFL0) instead of the MB information label (CFDTMFDCTRn.TMIFL).

35.3.77 CFDTHLPCTRn : TX History List Pointer Control Register n (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1240 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	THLPC[7:0]	TX History List Pointer Control Increments the write pointer to the TX History List in the corresponding channel	W
31:8	—	The write value should be 0.	W

The CFDTHLPCTRn register (n = 0, 1) is used to increment the read pointer of the TX History List.

THLPC[7:0] bits (TX History List Pointer Control)

When 0xFF is written to the THLPC[7:0] bits, the read pointer of the TX History List is moved to the next TX History List entry address.

The read value from these bits is always 0x00. Only write to these bits when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Only write 0xFF to these registers when the corresponding TX History List is enabled and not empty.

35.3.78 CFDGRSTC : Global Reset Control Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1380

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	KEY[7:0]								—	—	—	—	—	—	—	SRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRST	Software Reset 0: Normal state 1: Software reset state	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits control the validity of rewriting of the SRST bit.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

SRST bit (Software Reset)

When the SRST bit is set, the CAN-FD module is in the same state as hardware reset. When a reset is required, write 1 then write 0 to this bit.

This bit is cleared when the CAN-FD module is in GL_SLEEP mode.

When this bit is cleared, the RAM initialization sequence does not operate. The configuration of RAM is performed by software.

The RAM is not initialized when software reset is performed during the initialization of RAM. Software must perform the initialization of RAM.

KEY[7:0] bits (Key Code)

When 0xC4 is written in the KEY[7:0] bits, a write to the SRST bit is valid.

The read value from these bits is always 0x00.

35.3.79 CFDGFCMC : Global Flexible CAN Mode Configuration Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1384

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FLXC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLXC0	Flexible CAN Mode between Channel 0 and Channel 1 0: Normal mode 1: Flexible CAN mode	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Flexible CAN mode configured in the CFDGFCMC register and Flexible transmission buffer assignment configured in the CFDGFTBAC register should not be used simultaneously.

FLXC0 bit (Flexible CAN Mode between Channel 0 and Channel 1)

When the FLXC0 bit is set, Channel 0 and Channel 1 of a CAN-FD module are in Flexible CAN mode.

Channel 1 uses TX/RX terminal of Channel 0. The TX/RX terminal of Channel 1 cannot be used.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

35.3.80 CFDGFTBAC : Global Flexible Transmission Buffer Assignment Configuration Register

Base address: CANFD = 0x8004_0000

Offset address: 0x138C

Bit position:	31																												3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FLXMB0[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FLXMB0[3:0]	Flexible Transmission Buffer Assignment between Channel 0 and Channel 1 By setting these bits, the even channel can use the configured number of TX mailboxes of the odd channel 0x0: 0 0x1: 4 0x2: 8 0x3: 12 0x4: 16 0x5: 20 0x6: 24 0x7: 28 0x8: 32 Others: Setting prohibited	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Flexible transmission buffer assignment configured in the CFDGFTBAC register and Flexible CAN mode configured in the CFDGFCMC register should not be used simultaneously.

FLXMB0[3:0] bits (Flexible Transmission Buffer Assignment between Channel 0 and Channel 1)

Channel 0 can use the number TXMB of channel 1 from 0 to 32 by the configuration of these bits.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

35.3.81 CFDGTSTCFG : Global Test Configuration Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1308

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	RTMPS[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	C1ICBCE	C0ICBCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	C0ICBCE	Channel 0 Internal CAN Bus Communication Test Mode Enable 0: Channel 0 internal CAN bus communication disabled 1: Channel 0 internal CAN bus communication enabled	R/W
1	C1ICBCE	Channel 1 Internal CAN Bus Communication Test Mode Enable 0: Channel 1 internal CAN bus communication disabled 1: Channel 1 internal CAN bus communication enabled	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
25:16	RTMPS[9:0]	RAM Test Mode Page Select Select a RAM test mode page	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGTSTCFG register is used to configure the CAN channels joining the internal CAN bus communication test mode and the RAM test mode page.

CnICBCE (n = 0, 1) bits (Channel n Internal CAN Bus Communication Test Mode Enable)

When the C0ICBCE and C1ICBCE bits are set and CAN-FD module is configured in the internal CAN bus communication test mode, then CAN channel n joins the internal CAN bus communication test mode operation.

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

Only write to these bits when the CAN-FD module is in GL_HALT mode.

These bits are cleared automatically when the CAN-FD module is in GL_RESET mode.

RTMPS[9:0] bits (RAM Test Mode Page Select)

The RTMPS[9:0] bits select the RAM page mode for CPU read/write access when the CAN-FD module is configured in RAM test mode.

See section 35.10.2.1. RAM Test Mode for the RAM test mode specification.

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

Only enter values from 0 to 15 (0x00F) for the AFL RAM and 16 to 76 (0x04C) for the message buffer RAM.

The setting range of these bits depends on the combination of parameters.

Only write to these bits when the CAN-FD module is in GL_HALT mode.

These bits are cleared automatically when the related CAN-FD channel is in GL_RESET mode.

35.3.82 CFDGTSTCTR : Global Test Control Register

Base address: CANFD = 0x8004_0000

Offset address: 0x130C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBCTME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ICBCTME	Internal CAN Bus Communication Test Mode Enable 0: Internal CAN Bus Communication test mode disabled 1: Internal CAN Bus Communication test mode enabled	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	RTME	RAM Test Mode Enable 0: RAM test mode disabled 1: RAM test mode enabled	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGTSTCTR register is used to control the global test modes of the CAN-FD module.

ICBCTME bit (Internal CAN Bus Communication Test Mode Enable)

When the ICBCTME bit is set, internal CAN bus communication is enabled for the CAN channels that are configured for internal CAN bus communication participation. See section 35.10.2.2. Internal CAN Bus Communication Mode for the specification of internal CAN bus communication test mode.

Only write to this bit when the CAN-FD module is in GL_HALT mode.

Clear this bit when the CAN-FD module is in GL_HALT mode.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

RTME bit (RAM Test Mode Enable)

When the RTME bit is set, the CAN-FD module is configured in RAM test mode. See section 35.10.2.1. RAM Test Mode for RAM test mode specification.

Only write to this bit when the CAN-FD module is in GL_HALT mode.

Clear this bit when the CAN-FD module is in GL_HALT mode.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

35.3.83 CFDGFDCFG : Global FD Configuration Register

Base address: CANFD = 0x8004_0000

Offset address: 0x1314

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TSCCFG[1:0]		—	—	—	—	—	—	—	RPED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RPED	RES Bit Protocol Exception Disable 0: Protocol exception event detection enabled 1: Protocol exception event detection disabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
9:8	TSCCFG[1:0]	Timestamp Capture Configuration 0 0: Timestamp capture at the sample point of SOF (start of frame) 0 1: Timestamp capture at frame valid indication 1 0: Timestamp capture at the sample point of RES bit 1 1: Reserved	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

RPED bit (RES Bit Protocol Exception Disable)

The RPED bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is enabled, the protocol exception event detection is disabled, and the protocol controller transmits an error frame when the protocol exception event is detected (RES bit is sampled recessive).

Only write to this bit when the CAN-FD module is in GL_RESET mode.

TSCCFG[1:0] bits (Timestamp Capture Configuration)

The TSCCFG[1:0] bits configure the different capture points of the timestamp for transmission and reception.

When CFDGFDCFG.TSCCFG[1:0] = 10b, the timestamp capture is performed for CAN-FD frames at RES bit and for Classical frames at the start of frame.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

35.3.84 CFDGLOCKK : Global Lock Key Register

Base address: CANFD = 0x8004_0000

Offset address: 0x131C

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LOCK[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	LOCK[15:0]	Lock Key Key bits for unlocking the protection of test modes	W
31:16	—	The write value should be 0.	W

The CFDGLOCKK register is a write-only register that is used to unlock the protection for special test bits.

See section 35.10.2. Global Test Modes for Lock key specification.

LOCK[15:0] bits (Lock Key)

The unlock key sequence must be written in the LOCK[15:0] bits to configure the CAN-FD module in RAM test modes.

The read value from these bits is always 0x0000.

You cannot write to these bits when the CAN-FD module is in GL_SLEEP or GL_RESET mode.

Do not write to these bits when the CAN-FD module is in GL_OPERATION mode.

35.3.85 CFDRPGACCn : RAM Test Page Access Register n (n = 0 to 63)

Base address: CANFD = 0x8004_0000

Offset address: 0x8400 + 0x04 × n

Bit position:	31																													0
Bit field:	RDTA[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RDTA[31:0]	RAM Data Test Access RAM data bytes	R/W

RDTA[31:0] bits (RAM Data Test Access)

Data can be read from or written into the RDTA[31:0] bits when the CAN-FD module is configured in RAM test mode.

Only write to this bit when the CAN-FD module is in GL_HALT mode and RAM test mode is enabled.

If data is read when RAM test mode is not enabled, then it is always read as 0x00000000.

Software data should be read/written in the RAM Test Page Access registers during RAM test mode.

35.3.86 CFDCnBLCT : Channel n Bus Load Control Register (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x1418 + 0x20 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BLCLD	—	—	—	—	—	—	—	BLCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BLCE	Bus Load Counter Enable 0: Bus load counter disabled 1: Bus load counter enabled	R/W

Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	BLCLD	Bus Load Counter Load When CFDCnBLCT.BLCLD is set, it is reset after a bus load counter value is loaded.	W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

BLCE bit (Bus Load Counter Enable)

The BLCE bit indicates the enabling signal of a bus load counter. Write to this bit after setting up the CFDCnNCFG register.

When this bit is 0, the bus load counter stops but the counter is not clear.

The bit is cleared when the related CAN-FD channel is in CH_RESET mode.

Do not write to this bit in CH_SLEEP mode.

BLCLD bit (Bus Load Counter Load)

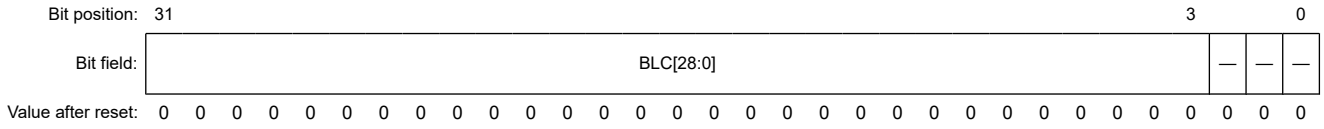
When the BLCLD bit is set, the bus load counter value is loaded into CFDCnBLSTS.BLC and the bus load counter is reset.

The read value is always 0.

35.3.87 CFDCnBLSTS : Channel n Bus Load Status Register (n = 0, 1)

Base address: CANFD = 0x8004_0000

Offset address: 0x141C + 0x20 × n



Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0.	R
31:3	BLC[28:0]	Bus Load Counter Status These bits indicate the bus load counter value.	R

BLC[28:0] bits (Bus Load Counter Status)

When the CFDCnBLCT.BLCLD bit is set, the bus load counter value is loaded to BLC[28:0] bits and the bus load counter is reset.

These bits are set only by the CAN-FD module. Writing any value has no effect.

35.3.88 Identifier Bits Alignment

Standard Identifier (11 bits) format: ID28 – ID18 is aligned to b10 to b0, b11 to b28 should be 0.

Extended Identifier (29 bits) format: ID28 – ID0 is aligned to b28 – b0

Table 35.13 Standard Identifier (11-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 0	RTR	—	0	0	0	0	0	0	0	0	0	0	0	0	0

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18

Table 35.14 Extended Identifier (29-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 1	RTR	—	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

35.3.89 Message Buffer Component Structure

35.3.89.1 Start Addresses

The start address for each of the Message Buffer component is calculated using the number of related Message Buffer components and the number of channels.

number_of_channels	2
number_of_RMBCPs_per_channel	16
number_of_RFMBCPs	8
number_of_CFMBCPs_per_channel	3
number_of_TMBCPs_per_channel	64

The start addresses for each register in the Message Buffer component are depicted in Table 35.15.

Table 35.15 Message Buffer Component Register Start Addresses

b = Message buffer component index	MBCP	Register	p	Start address n = 0, 1
0 to number_of_RMBCPs_per_channel - 1	RMBCPb[i]	RMID	x	0x2000 + b × 0x0080 + n × 0x800
		RMPTR	x	0x2004 + b × 0x0080 + n × 0x800
		RMFDSTS	x	0x2008 + b × 0x0080 + n × 0x800
		RMDFp	0 to 15	0x200C + p × 0x0004 + b × 0x0080 + n × 0x800
0 to number_of_RFMBCPs - 1	RFMBCPb[i]	RFID	x	0x6000 + b × 0x0080
		RFPTR	x	0x6004 + b × 0x0080
		RFFDSTS	x	0x6008 + b × 0x0080
		RFDFp	0 to 15	0x600C + p × 0x0004 + b × 0x0080
0 to number_of_CFMBCPs_per_channel - 1	CFMBCPb[i]	CFID	x	0x6400 + b × 0x0080 + n × 0x180
		CFPTR	x	0x6404 + b × 0x0080 + n × 0x180
		CFFDCSTS	x	0x6408 + b × 0x0080 + n × 0x180
		CFDFp	0 to 15	0x640C + p × 0x0004 + b × 0x0080 + n × 0x180
0 to number_of_TMBCPs_per_channel - 1	TMBCPb[i]	TMID	x	0x10000 + b × 0x0080 + n × 0x2000
		TMPTR	x	0x10004 + b × 0x0080 + n × 0x2000
		TMFDCTR	x	0x10008 + b × 0x0080 + n × 0x2000
		TMDFp	0 to 15	0x1000C + p × 0x0004 + b × 0x0080 + n × 0x2000

The message buffer configuration consists of four types of Message Buffer components:

- RX Message Buffer Component (CFDRMBCPb[i])
- RX FIFO Access Message Buffer Component (CFDRFMBCPb[i])
- Common FIFO Access Message Buffer Component (CFDCFMBCPb[i])
- TX Message Buffer Component (CFDTMBCPb[i]).

Where b = the Message Buffer component index having a range that varies based on the type of Message Buffer component and i = channel index that has a range from 0 to n.

For a summary of this configuration, see Figure 35.28. For a detailed description of the number of and the different types of message buffers, see section 35.7. FIFO Buffers and Normal Message Buffer Configuration

As described in section 35.3. Register Descriptions, each Message Buffer component consists of the following registers:

- Identifier (ID)
- Pointer (PTR)
- Data Field (DFp).

p is the Data Field register index having a range that varies based on the type of Message Buffer component.

Rc is the Message Buffer Component register where c = Message Buffer Component register index having a range that varies based on the type of Message Buffer component.

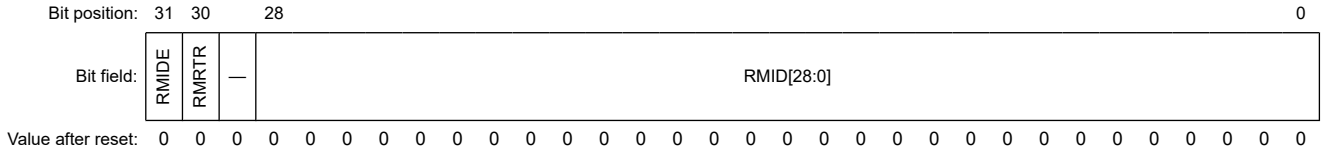
A description of the registers, their associated bits and their accessibility are shown below the summary and detailed figures of each component.

In each of the figures, a cell that contains '-' means reserved and has the same behavior as reserved bits for registers in section 35.3.89. Message Buffer Component Structure.

35.3.89.2 CFDRMIDn : RX Message Buffer ID Register n (n = 0 to 31)

Base address: CANFD = 0x8004_0000

Offset address: 0x2000 + 0x080 × n



Bit	Symbol	Function	R/W
28:0	RMID[28:0]	RX Message Buffer ID Field STD-ID/EXT-ID fields	R
29	—	These bits are read as 0.	R
30	RMRTR	RX Message Buffer RTR Bit 0: Data frame 1: Remote frame	R
31	RMIDE	RX Message Buffer IDE Bit 0: STD-ID is stored 1: EXT-ID is stored	R

The CFDRMIDn register (n = 0 to 31) stores the ID field, IDE bit, and RTR bit of the received message.

RMID[28:0] bits (RX Message Buffer ID Field)

The RMID[28:0] are the bits of the STD-ID/EXT-ID fields of the message stored in the RX message buffer.

For alignment of these bits in standard and extended frame format, see section 35.3.88. Identifier Bits Alignment

See section 35.3.89.1. Start Addresses for details on how to interpret the structure of this buffer component.

RMRTR bit (RX Message Buffer RTR Bit)

The RMRTR bit shows whether a data frame or a remote frame was stored in the RX message buffer.

Note: There are no remote frames in CAN-FD format. When a CAN-FD frame is received, the register reflects the state of the received value (the RRS bit in FD frame format).

RMIDE bit (RX Message Buffer IDE Bit)

The RMIDE bit shows whether message with Standard Identifier or Extended Identifier was stored in the RX message buffer.

35.3.89.3 CFDRMPTRn : RX Message Buffer Pointer Register (n = 0 to 31)

Base address: CANFD = 0x8004_0000

Offset address: 0x2004 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RMDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	RMTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RMTS[15:0]	RX Message Buffer Timestamp Field Timestamp value stored for the message in the RX message buffer	R/W
27:16	—	These bits are read as 0. The write value should be 0.	R/W
31:28	RMDLC[3:0]	RX Message Buffer DLC Field Number of data bytes received in a CAN frame.	R/W

The CFDRMPTRn register (n = 0 to 31) stores the DLC and Timestamp fields for the received message.

RMTS[15:0] bits (RX Message Buffer Timestamp Field)

The RMTS[15:0] bits store the timestamp value taken at the capture point as configured by CFDGFDCFG.TSCCFG of the received message.

RMDLC[3:0] bits (RX Message Buffer DLC Field)

The RMDLC[3:0] bits store the number of data bytes that were received in the RX message buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

Note: The maximum capacity of the buffer belongs to CFDRMNB.RMPLS and this is not available in the classical CAN function.

35.3.89.4 CFDRMFDSTSn : RX Message Buffer CAN-FD Status Register (n = 0 to 31)

Base address: CANFD = 0x8004_0000

Offset address: 0x2008 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RMPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RMIFL[1:0]		—	—	—	—	—	RMFDF	RMBRS	RMESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMESI*1	Error State Indicator bit 0: CAN-FD frame received from error active node 1: CAN-FD frame received from error passive node	R/W
1	RMBRS*1	Bit Rate Switch bit 0: CAN-FD frame received with no bit rate switch 1: CAN-FD frame received with bit rate switch	R/W

Bit	Symbol	Function	R/W
2	RMFDF*1	CAN FD Format bit 0: Non CAN-FD frame received 1: CAN-FD frame received	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
9:8	RMIFL[1:0]	RX Message Buffer Information Label Field	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
31:16	RMPTR[15:0]	RX Message Buffer Pointer Field	R/W

Note 1. This bit is not available in the classical CAN function.

The CFDRMFDSTSn register (n = 0 to 31) shows the status of the FDF, BRS, and ESI bits, and pointer of the received CAN-FD frame.

RMESI bit (Error State Indicator bit)

The RMESI bit has the same value as the ESI bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMBRS bit (Bit Rate Switch bit)

The RMBRS bit has the same value as the BRS bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMFDF bit (CAN FD Format bit)

The RMFDF bit has the same value as the FDF bit of the received CAN-FD frame.

Note: This bit is not available in the classical CAN function.

RMIFL[1:0] bits (RX Message Buffer Information Label Field)

The RMIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

RMPTR[15:0] bits (RX Message Buffer Pointer Field)

The RMPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

35.3.89.5 CFDRMDFp_n : RX Message Buffer Data Field p Register n (p = 0 to 15, n = 0 to 31)

Base address: CANFD = 0x8004_0000

Offset address: 0x200C + 0x004 × p + 0x080 × n

Bit position:	31 … 24	23 … 16	15 … 8	7 … 0
Bit field:	RMDB_HH[7:0]	RMDB_HL[7:0]	RMDB_LH[7:0]	RMDB_LL[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	RMDB_LL[7:0]	RX Message Buffer Data Byte (4 × p)	R/W
15:8	RMDB_LH[7:0]	RX Message Buffer Data Byte (4 × p + 1)	R/W
23:16	RMDB_HL[7:0]	RX Message Buffer Data Byte (4 × p + 2)	R/W
31:24	RMDB_HH[7:0]*1	RX Message Buffer Data Byte (4 × p + 3)	R/W

Note 1. These bits are not available in the classical CAN function.

The CFDRMDFp_n register (p = 0 to 15, n = 0 to 31) stores the data bytes (4 × p) to data bytes (4 × p + 3) of the received message.

RMDB_LL[7:0] bits (RX Message Buffer Data Byte (4 × p))

The RMDB_LL[7:0] bits store data bytes (4 × p) of the message in the RX message buffer. Unused data bytes are filled with 0x00.

RMDB_LH[7:0] bits (RX Message Buffer Data Byte (4 × p + 1))

The RMDB_LH[7:0] bits store data bytes (4 × p + 1) of the message in the RX message buffer. Unused Data Bytes are filled with 0x00.

RMDB_HL[7:0] bits (RX Message Buffer Data Byte (4 × p + 2))

The RMDB_HL[7:0] bits store data bytes (4 × p + 2) of the message in the RX message buffer. Unused data bytes are filled with 0x00.

RMDB_HH[7:0] bits (RX Message Buffer Data Byte (4 × p + 3))

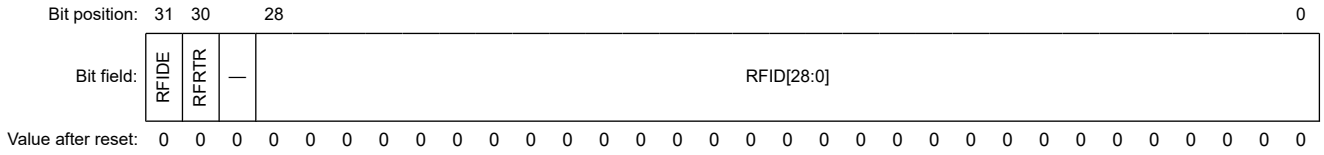
The RMDB_HH[7:0] bits store data bytes (4 × p + 3) of the message in the RX message buffer. Unused data bytes are filled with 0x00.

These bits are not available in the classical CAN function.

35.3.89.6 CFDRFIDn : RX FIFO Access ID Register n (n = 0 to 7)

Base address: CANFD = 0x8004_0000

Offset address: 0x6000 + 0x080 × n



Bit	Symbol	Function	R/W
28:0	RFID[28:0]	RX FIFO Buffer ID Field STD-ID/EXT-ID fields	R
29	—	This bit is read as 0.	R
30	RFRTR	RX FIFO Buffer RTR bit 0: Data frame 1: Remote frame	R
31	RFIDE	RX FIFO Buffer IDE bit 0: STD-ID has been received 1: EXT-ID has been received	R

The CFDRFIDn register (n = 0 to 7) stores the ID field, IDE bit, and RTR bit of the message.

RFID[28:0] bits (RX FIFO Buffer ID Field)

The RFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

For alignment of these bits in standard and extended frame format, see section 35.3.88. Identifier Bits Alignment.

RFRTR bit (RX FIFO Buffer RTR bit)

The RFRTR bit shows whether a data frame or a remote frame was stored in the FIFO buffer.

Note: There are no remote frames in CAN-FD format. When a CAN-FD frame was received, the register reflects the state of the received value (RRS bit in FD frame format).

RFIDE bit (RX FIFO Buffer IDE bit)

The RFIDE bit shows whether message with the Standard Identifier or Extended Identifier was received in the FIFO buffer.

35.3.89.7 CFDRFPTRn : RX FIFO Access Pointer Register n (n = 0 to 7)

Base address: CANFD = 0x8004_0000

Offset address: 0x6004 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	RFTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RFTS[15:0]	RX FIFO Timestamp Value Timestamp value of the received CAN frame	R
27:16	—	These bits are read as 0.	R
31:28	RFDLC[3:0]	RX FIFO Buffer DLC Field Number of data bytes received in a CAN frame	R

The CFDRFPTRn register (n = 0 to 7) stores the DLC and Timestamp fields for the received message.

RFTS[15:0] bits (RX FIFO Timestamp Value)

The RFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message.

RFDLC[3:0] bits (RX FIFO Buffer DLC Field)

The RFDLC[3:0] bits store the number of data bytes that were received in the RX FIFO buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

35.3.89.8 CFDRFFDSTSn : RX FIFO Access CAN-FD Status Register n (n = 0 to 7)

Base address: CANFD = 0x8004_0000

Offset address: 0x6008 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFDRFPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RFIFL[1:0]		—	—	—	—	—	RFFDF	RFBRS	RFESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFESI*1	Error State Indicator bit 0: CAN-FD frame received from error active node 1: CAN-FD frame received from error passive node	R
1	RFBRS*1	Bit Rate Switch bit 0: CAN-FD frame received with no bit rate switch 1: CAN-FD frame received with bit rate switch	R
2	RFFDF*1	CAN FD Format bit 0: Non CAN-FD frame received 1: CAN-FD frame received	R
7:3	—	These bits are read as 0.	R
9:8	RFIFL[1:0]	RX FIFO Buffer Information Label Field	R
15:10	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
31:16	CFDRFPTR[15:0]	RX FIFO Buffer Pointer Field	R

Note 1. This bit is not available in the classical CAN function.

The CFDRFFDSTSn register (n = 0 to 7) shows the status of the FDF, BRS, and ESI bits, including the pointer of the received CAN-FD frame.

RFESI bit (Error State Indicator bit)

The RFESI bit has the same value as the ESI bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFBRS bit (Bit Rate Switch bit)

The RFBRS bit has the same value as the BRS bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFFDF bit (CAN FD Format bit)

The RFFDF bit has the same value as the FDF bit of the received CAN-FD frame.

Note: This bit is not available in the classical CAN function.

RFIFL[1:0] bits (RX FIFO Buffer Information Label Field)

The RFIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

CFDRFPTR[15:0] bits (RX FIFO Buffer Pointer Field)

The CFDRFPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

35.3.89.9 CFDRFDFpn : RX FIFO Access Data Field p Register n (p = 0 to 15, n = 0 to 7)

Base address: CANFD = 0x8004_0000

Offset address: 0x600C + 0x004 × p + 0x080 × n

Bit position:	31							24	23							16	15							8	7							0
Bit field:	RFDB_HH[7:0]								RFDB_HL[7:0]								RFDB_LH[7:0]								RFDB_LL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFDB_LL[7:0]	RX FIFO Buffer Data Byte (4 × p)	R
15:8	RFDB_LH[7:0]	RX FIFO Buffer Data Byte (4 × p + 1)	R
23:16	RFDB_HL[7:0]	RX FIFO Buffer Data Byte (4 × p + 2)	R
31:24	RFDB_HH[7:0]	RX FIFO Buffer Data Byte (4 × p + 3)	R

The CFDRFDFpn register (p = 0 to 15, n = 0 to 7) stores data bytes (4 × p) to data bytes (4 × p + 3) of the received message.

RFDB_LL[7:0] bits (RX FIFO Buffer Data Byte (4 × p))

The RFDB_LL[7:0] bits store data bytes (4 × p) of the message present in the FIFO buffer. Unused data bytes are filled with 0x00 according to the configured data payload size CFDRFCCn.RFPLS.

RFDB_LH[7:0] bits (RX FIFO Buffer Data Byte (4 × p + 1))

The RFDB_LH[7:0] bits store data bytes (4 × p + 1) of the message present in the FIFO buffer. Unused data bytes are filled with 0x00.

RFDB_HL[7:0] bits (RX FIFO Buffer Data Byte (4 × p + 2))

The RFDB_HL[7:0] bits store data bytes (4 × p + 2) of the message present in the FIFO buffer. Unused data bytes are filled with 0x00.

RFDB_HH[7:0] bits (RX FIFO Buffer Data Byte (4 × p + 3))

The RFDB_HH[7:0] bits store data bytes (4 × p + 3) of the message present in the FIFO buffer. Unused data bytes are filled with 0x00.

35.3.89.10 CFDCFIDn : Common FIFO Access ID Register n (n = 0 to 5)

Base address: CANFD = 0x8004_0000

Offset address: 0x6400 + 0x080 × n

Bit position:	31	30	29	28																											0
Bit field:	CFIDE	CFRTR	THLEN	CFID[28:0]																											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	CFID[28:0]	Common FIFO Buffer ID Field STD-ID/EXT-ID fields	R/W
29	THLEN	THL Entry Enable TX FIFO mode: 0: Entry is not to be stored in THL after successful TX 1: Entry is to be stored in THL after successful TX RX FIFO mode: Reserved, this bit is read as 0.	R/W
30	CFRTR	Common FIFO Buffer RTR bit 0: Data frame 1: Remote frame	R/W
31	CFIDE	Common FIFO Buffer IDE bit 0: STD-ID is to be transmitted or has been received 1: EXT-ID is to be transmitted or has been received	R/W

The CFDCFIDn register (n = 0 to 5) stores the ID field, IDE bit, and RTR bit of the message.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFID[28:0] bits (Common FIFO Buffer ID Field)

The CFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

For alignment of these bits in standard and extended frame format, see section 35.3.88. Identifier Bits Alignment.

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

THLEN bit (THL Entry Enable)

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFRTR bit (Common FIFO Buffer RTR bit)

The CFRTR bit selects whether a data frame or a remote frame is to be transmitted from or was received in the FIFO buffer.

Note: There are no remote frames in CAN FD format. When a CAN-FD frame is received (RX mode), the register reflects the state of the received value (RRS bit in FD frame format). When CAN-FD transmission (TX or GW mode CFDCFID.CFFDF = 1), the bit is always transmitted dominant (data frame).

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFIDE bit (Common FIFO Buffer IDE bit)

The CFIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from or was received in the FIFO buffer.

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

35.3.89.11 CFDCFPTRn : Common FIFO Access Pointer Register n (n = 0 to 5)

Base address: CANFD = 0x8004_0000

Offset address: 0x6404 + 0x080 × n

Bit position:	31			28												15															0
Bit field:	CFDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	CFTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CFTS[15:0]	Common FIFO Timestamp Value Timestamp value of the received CAN frame (FIFO in RX mode).	R/W
27:16	—	These bits are read as 0. The write value should be 0.	R/W
31:28	CFDLC[3:0]	Common FIFO Buffer DLC Field Number of data bytes received in a CAN frame, or transmitted in a CAN frame.	R/W

The CFDCFPTRn register (n = 0 to 5) stores the DLC and Timestamp fields.

In TX mode, you can read data from the FIFO buffer, only for the current entry based on the write pointer value, and not for the other entries.

CFTS[15:0] bits (Common FIFO Timestamp Value)

The CFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message (if FIFO is configured in RX mode).

In TX mode, you can read and write data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFDLC[3:0] bits (Common FIFO Buffer DLC Field)

The CFDLC[3:0] bits store the number of data bytes that were received in the FIFO buffer or are to be transmitted.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes.

In TX mode, you can read and write data from the FIFO buffers. Do not read data for the other entries in the FIFO when configured in TX mode.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

35.3.89.12 CFDCFFDCSTSn : Common FIFO Access CAN-FD Control/Status Register n (n = 0 to 5)

Base address: CANFD = 0x8004_0000

Offset address: 0x6408 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFIFL[1:0]		—	—	—	—	—	CFFDF	CFBRS	CFESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFESI*1	Error State Indicator bit 0: CAN-FD frame received or to transmit by error active node 1: CAN-FD frame received or to transmit by error passive node	R/W
1	CFBRS*1	Bit Rate Switch bit 0: CAN-FD frame received or to transmit with no bit rate switch 1: CAN-FD frame received or to transmit with bit rate switch	R/W
2	CFFDF*1	CAN FD Format bit 0: Non CAN-FD frame received or to transmit 1: CAN-FD frame received or to transmit	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CFIFL[1:0]	COMMON FIFO Buffer Information Label Field	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
31:16	CFPTR[15:0]	Common FIFO Buffer Pointer Field	R/W

Note 1. This bit is not available in the classical CAN function.

The CFDCFFDCSTSn register (n = 0 to 5) shows the status of the FDF, BRS, and ESI bits, including the pointer of the received CAN-FD frame or the CAN-FD frame to transmit.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFESI bit (Error State Indicator bit)

In TX mode, you can read and write data from FIFO buffers. In this mode, when the CAN-FD module is not in error passive, the CFESI bit equals the write value. Otherwise, it is a don't care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFESI bit is updated with the ESI bit value of the CAN-FD frame when it has been received, indicating the error state of the transmitting node. In RX or GW mode, 0 is stored to this bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFBRS bit (Bit Rate Switch bit)

In TX mode, you can read and write data from FIFO buffers. In this mode, the CAN-FD module either transmits a 0 to indicate no bit rate switch in the frame to be transmitted or a 1 to indicate a bit rate switch in the frame to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFBRS bit is updated with the BRS bit value of the CAN-FD frame when it has been received, indicating whether there is a bit rate switch (1) or (0) on the CAN-FD frame.

In RX or GW mode, 0 is stored to the CFBRS bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFFDF bit (CAN FD Format bit)

In TX mode, you can read and write data from FIFO buffers. In this mode, the CAN-FD module either transmits a 0 to indicate a CAN 2.0 frame is to be transmitted or a 1 to indicate a CAN-FD frame is to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFFDF bit is updated with the FDF bit value of the CAN frame when it has been received, indicating whether it is a CAN 2.0 frame (0) or a CAN-FD frame (1).

Note: This bit is not available in the classical CAN function.

CFIFL[1:0] bits (COMMON FIFO Buffer Information Label Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTSn.CFIFL[1:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The information label value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX or GW mode).

In TX mode, you can read and write data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFPTR[15:0] bits (Common FIFO Buffer Pointer Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTSn.CFPTR[15:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The pointer value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX or GW mode).

In TX mode, you can read and write data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

35.3.89.13 CFDCFDFpn : Common FIFO Access Data Field p Register n (p = 0 to 15, n = 0 to 5)

Base address: CANFD = 0x8004_0000

Offset address: 0x640C + 0x004 × p + 0x080 × n

Bit position:	31 … 24	23 … 16	15 … 8	7 … 0
Bit field:	CFDB_HH[7:0]	CFDB_HL[7:0]	CFDB_LH[7:0]	CFDB_LL[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	CFDB_LL[7:0]	Common FIFO Buffer Data Byte (4 × p)	R/W
15:8	CFDB_LH[7:0]	Common FIFO Buffer Data Byte (4 × p + 1)	R/W
23:16	CFDB_HL[7:0]	Common FIFO Buffer Data Byte (4 × p + 2)	R/W

Bit	Symbol	Function	R/W
31:24	CFDB_HH[7:0]	Common FIFO Buffer Data Byte (4 × p + 3)	R/W

The CFDCFDFpn register (p = 0 to 15, n = 0 to 5) stores data bytes (4 × p) to data bytes (4 × p + 3) of the message.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFDB_LL[7:0] bits (Common FIFO Buffer Data Byte (4 × p))

The CFDB_LL[7:0] bits store data bytes (4 × p) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

CFDB_LH[7:0] bits (Common FIFO Buffer Data Byte (4 × p + 1))

The CFDB_LH[7:0] bits store data bytes (4 × p + 1) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

CFDB_HL[7:0] bits (Common FIFO Buffer Data Byte (4 × p + 2))

The CFDB_HL[7:0] bits store data bytes (4 × p + 2) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

CFDB_HH[7:0] bits (Common FIFO Buffer Data Byte (4 × p + 3))

The CFDB_HH[7:0] bits store data bytes (4 × p + 3) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

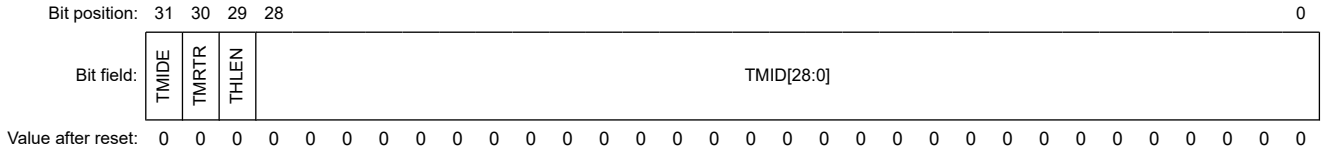
In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

Note 1. In RX or GW mode, unused data bytes are filled with 0x00 according to the configured data payload size CFDCFCCn.CFPLS, which is a CAN-FD feature not found in classical CAN.

35.3.89.14 CFDTMIDn : TX Message Buffer ID Register n (n = 0 to 127)

Base address: CANFD = 0x8004_0000

Offset address: 0x1_0000 + 0x080 × n



Bit	Symbol	Function	R/W
28:0	TMID[28:0]	TX Message Buffer ID Field STD-ID/EXT-ID fields	R/W
29	THLEN	Tx History List Entry 0: Entry not stored in THL after successful TX 1: Entry stored in THL after successful TX	R/W
30	TMRTR	TX Message Buffer RTR bit 0: Data frame 1: Remote frame	R/W
31	TMIDE	TX Message Buffer IDE bit 0: STD-ID is transmitted 1: EXT-ID is transmitted	R/W

The CFDTMIDn register (n = 0 to 127) stores the ID, IDE, RTR fields and history configuration of the message to be transmitted from the associated buffer.

TMID[28:0] bits (TX Message Buffer ID Field)

The TMID[28:0] bits are bits of the STD-ID/EXT-ID fields of the message stored in this TX message buffer.

For alignment of these bits in standard and extended frame format, see section 35.3.88. Identifier Bits Alignment.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

THLEN bit (Tx History List Entry)

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMRTR bit (TX Message Buffer RTR bit)

The TMRTR bit selects whether a data frame or remote frame is to be transmitted from this TX message buffer.

Note: There are no remote frames in CAN-FD format. For a CAN-FD transmission (CFDTMFDCTR.CFFDF = 1), this bit is always transmitted dominant (data frame).

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMIDE bit (TX Message Buffer IDE bit)

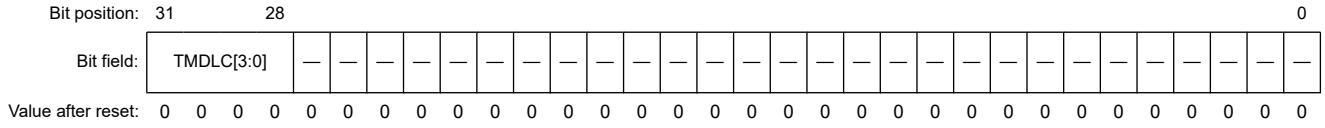
The TMIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from this TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

35.3.89.15 CFDTMPTRn : TX Message Buffer Pointer Register n (n = 0 to 127)

Base address: CANFD = 0x8004_0000

Offset address: 0x1_0004 + 0x080 × n



Bit	Symbol	Function	R/W
27:0	—	The read values are undefined. The write value should be 0.	R/W
31:28	TMDLC[3:0]	TX Message Buffer DLC Field Number of data bytes to be transmitted in a CAN frame.	R/W

The CFDTMPTRn register (n = 0 to 127) stores the DLC fields of the message to transmit from the associated buffer.

TMDLC[3:0] bits (TX Message Buffer DLC Field)

The TMDLC[3:0] bits select the number of data bytes to be transmitted from this TX message buffer when the corresponding TMRTR bit is configured as 0.

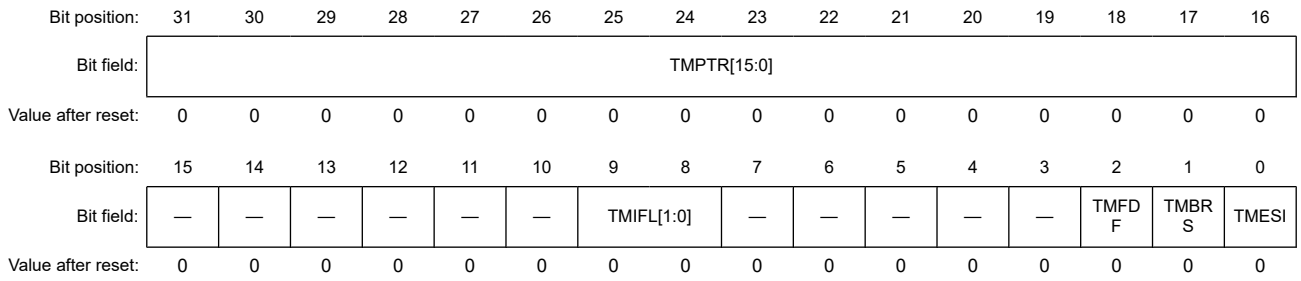
See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes to be transmitted.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

35.3.89.16 CFDTMFDCTRn : TX Message Buffer CAN-FD Control Register n (n = 0 to 127)

Base address: CANFD = 0x8004_0000

Offset address: 0x1_0008 + 0x080 × n



Bit	Symbol	Function	R/W
0	TMESI*1	Error State Indicator bit 0: CAN-FD frame to transmit by error active node 1: CAN-FD frame to transmit by error passive node	R/W
1	TMBRS*1	Bit Rate Switch bit 0: CAN-FD frame to transmit with no bit rate switch 1: CAN-FD frame to transmit with bit rate switch	R/W
2	TMFDF*1	CAN FD Format bit 0: Non CAN-FD frame to transmit 1: CAN-FD frame to transmit	R/W
7:3	—	The read values are undefined. The write value should be 0.	R/W
9:8	TMIFL[1:0]	TX Message Buffer Information Label Field	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W
31:16	TMPTR[15:0]	TX Message Buffer Pointer Field	R/W

Note 1. This bit is not available in the classical CAN function.

The CFDTMFDCTRn register (n = 0 to 127) shows the status of the FDF, BRS, and ESI bits, including the pointer fields of the CAN-FD frame to be transmitted.

TMESI bit (Error State Indicator bit)

If the channel is not in error passive, then the TMESI bit equals the write value, otherwise it is a don't care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

Do not write to the TMESI bit when the related CAN-FD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMBRS bit (Bit Rate Switch bit)

Do not write to the TMBRS bit when the related CAN-FD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMFDF bit (CAN FD Format bit)

Do not write to the TMFDF bit when the related CAN-FD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMIFL[1:0] bits (TX Message Buffer Information Label Field)

The TMIFL[1:0] bits store the information label value to be copied, together with additional message information, in the TX History List after successful transmission of the message.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMPTR[15:0] bits (TX Message Buffer Pointer Field)

The TMPTR[15:0] bits store the pointer value to be copied, together with additional message information in the TX History List after successful transmission of the message.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

35.3.89.17 CFDTMDFp_n : TX Message Buffer Data Field p Register n (p = 0 to 15, n = 0 to 127)

Base address: CANFD = 0x8004_0000

Offset address: 0x1_000C + 0x004 × p + 0x080 × n

Bit position:	31 … 24	23 … 16	15 … 8	7 … 0
Bit field:	TMDB_HH[7:0]	TMDB_HL[7:0]	TMDB_LH[7:0]	TMDB_LL[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	TMDB_LL[7:0]	TX Message Buffer Data Byte (4 × p)	R/W
15:8	TMDB_LH[7:0]	TX Message Buffer Data Byte (4 × p + 1)	R/W
23:16	TMDB_HL[7:0]	TX Message Buffer Data Byte (4 × p + 2)	R/W
31:24	TMDB_HH[7:0]	TX Message Buffer Data Byte (4 × p + 3)	R/W

The CFDTMDFp_n register (p = 0 to 15, n = 0 to 127) stores data bytes (4 × p) to data bytes (4 × p + 3) of the message to transmit from the associated buffer.

TMDB_LL[7:0] bits (TX Message Buffer Data Byte (4 × p))

Data bytes (4 × p)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMDB_LH[7:0] bits (TX Message Buffer Data Byte (4 × p + 1))

Data bytes (4 × p + 1)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMDB_HL[7:0] bits (TX Message Buffer Data Byte (4 × p + 2))

Data bytes (4 × p + 2)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMDB_HH[7:0] bits (TX Message Buffer Data Byte (4 × p + 3))

Data bytes (4 × p + 3)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

35.4 Operation

35.4.1 Overview

The modes of the CAN-FD module can be classified into 2 groups:

- Global modes
- Channel modes

35.4.2 Global Modes

Global modes are applicable for the complete CAN-FD module. The global modes of the CAN-FD module are:

- Global Sleep
- Global Reset
- Global Halt
- Global Operation

Figure 35.2 shows the possible transitions between the Global modes.

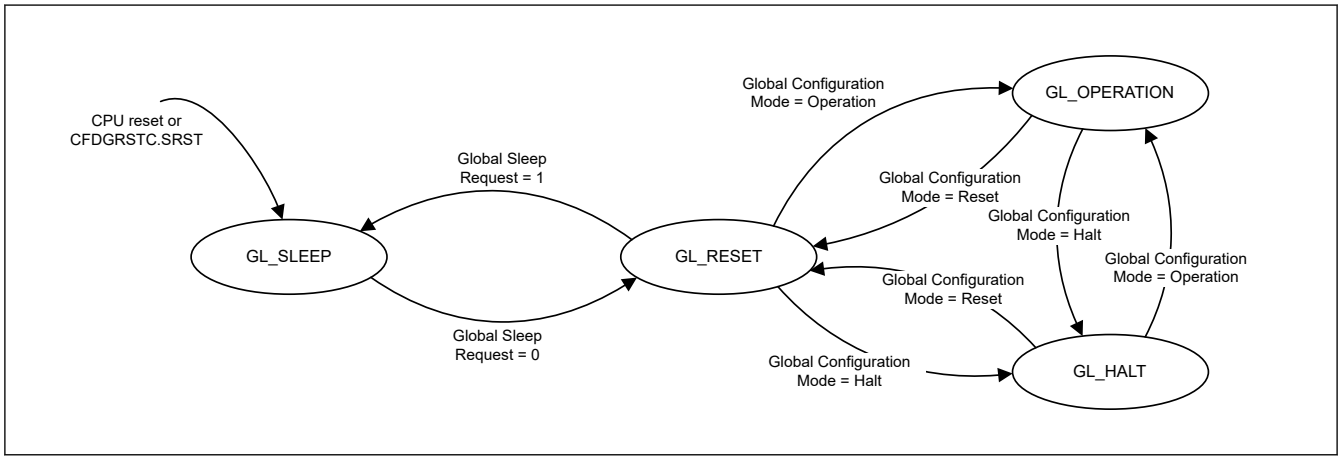


Figure 35.2 Transition between CAN-FD Global modes

Changes in the Global mode can affect the Channel mode. Table 35.16 shows the effect of a Global mode transition on a Channel mode.

Table 35.16 Possible CAN-FD Channel modes and Global modes

Current Global mode	Target Global mode			
	Sleep	**Reset**	**Halt**	**Operation**
Sleep	—	Ch-Sleep: Keep Ch-Reset: N/A Ch-Halt: N/A Ch-Oper: N/A	—	—
Reset	Ch-Sleep: Keep Ch-Reset: → Ch-Sleep Ch-Halt: N/A Ch-Oper: N/A	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A
Halt	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: N/A	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: N/A
Operation	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: → Ch-Reset	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: → Ch-Halt	—

35.4.2.1 Global Sleep Mode

After the release of a hardware reset or after setting and clearing the CFDGRSTC.SRST bit, the CAN-FD module automatically enters Global Sleep mode.

The CAN-FD module also enters the Global Sleep mode when the Global Sleep Request bit is set while it is in Global Reset mode. This control bit cannot be set in Global Halt mode or Global Operation mode.

Setting the Global Sleep Request bit sets all Channel Sleep Request bits and forces all channels into the Channel Sleep mode. Sleep mode is used for power saving purpose. When CAN-FD module is in Global Sleep mode, only the clock for CPU write access to the Global Sleep Mode Request bit is active. All other clocks are stopped and all other functions of the CAN-FD module are suspended.

Read access from all registers is still possible and all register values are preserved.

In Global Sleep mode, RAM access is prohibited because the logic which generates a RAM address does not operate.

After setting the Global Sleep Request bit, it is necessary to confirm that the Global Sleep status has been updated, indicating successful transition to Global Sleep mode before the Global Sleep Request bit can be cleared again.

Figure 35.3 shows the procedure for entering Global Sleep mode and Figure 35.4 shows the procedure for exiting Global Sleep mode.

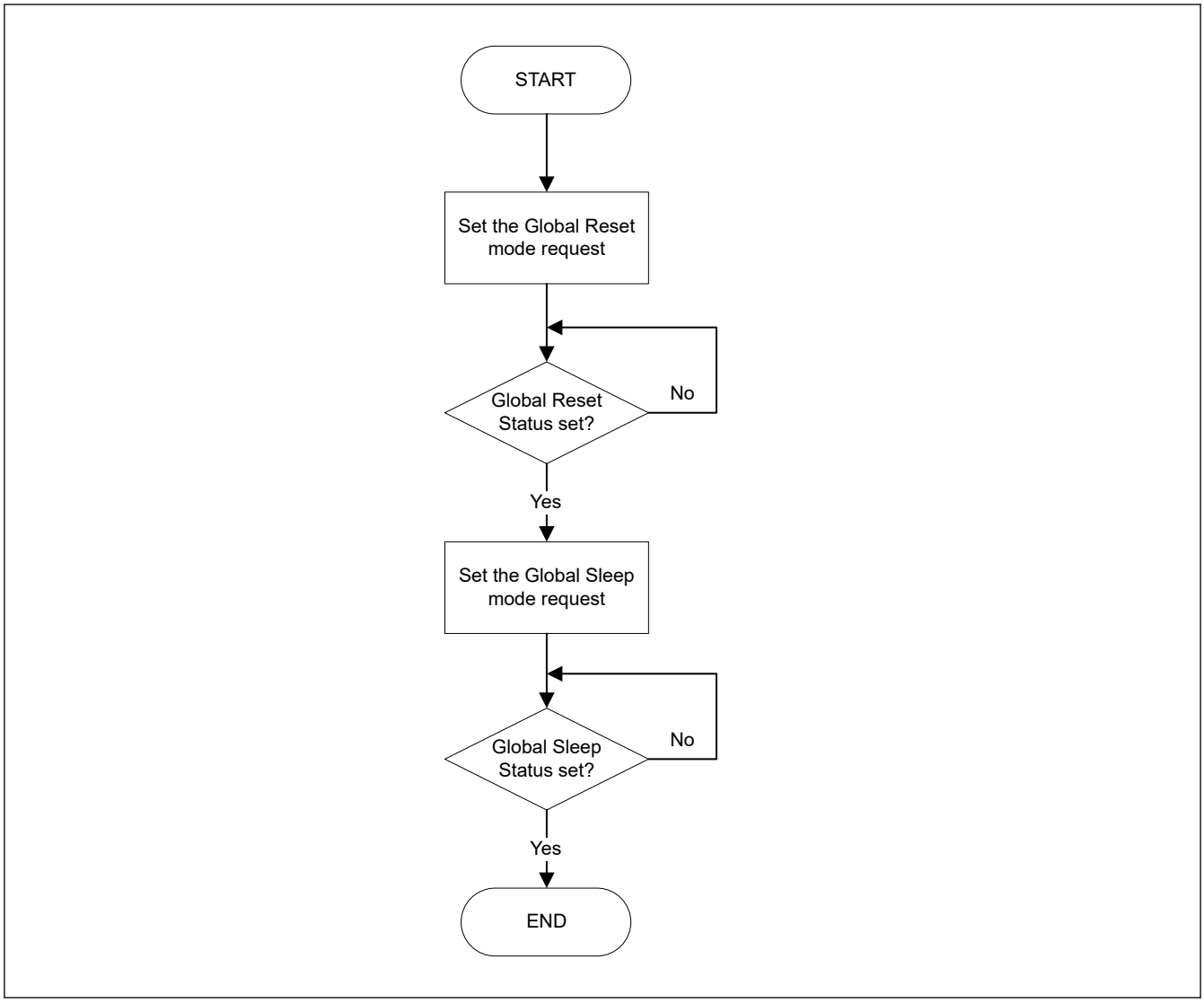


Figure 35.3 Procedure for entering Global Sleep mode

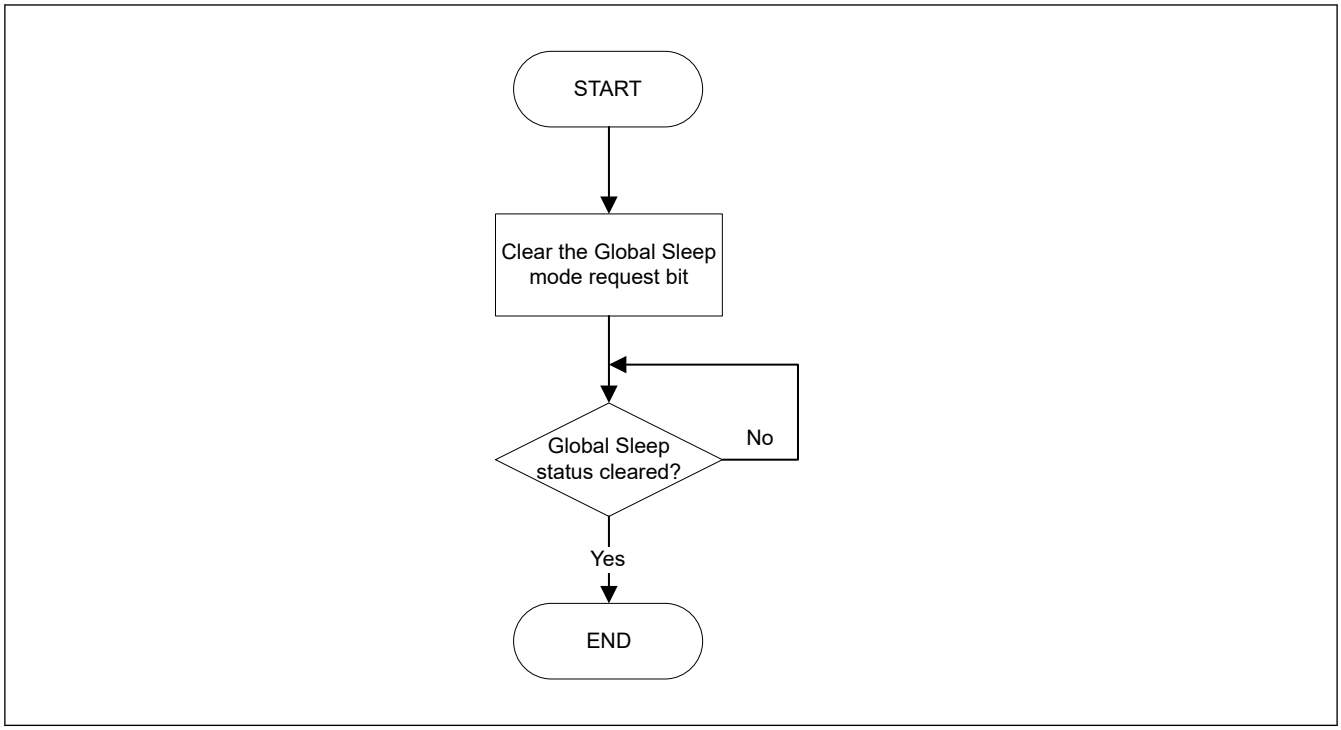


Figure 35.4 Procedure for exiting Global Sleep mode

35.4.2.2 Global Reset Mode

The CAN-FD module enters the Global Reset mode in the following ways:

- Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register is configured for Global Reset mode while the CAN-FD module is in Global Halt or Global Operation mode.
- Global Sleep Mode Request bit is cleared while CAN-FD module is in Global Sleep mode.

In Global Reset mode, all CAN-FD module functions are suspended and all status and flag registers are initialized.

Additionally all FIFOs and all channel TX Queues are disabled and transmission control bits are cleared.

Configuration registers (except the test mode registers) are not initialized in this mode to their device reset values and the CAN-FD module can be configured.

See section 35.4.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Reset mode.

Setting the Global mode to Reset by setting the Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register to 01b sets all Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers to 01b and forces all channels into the Channel Reset mode.

For channels that are already in Channel Reset mode or Channel Sleep mode, this automatic transition is not performed (CFDCnCTR.CHMDC[1:0] of related channel already set to 01b).

After setting Global Mode Control bits CFDGCTR.GMDC[1:0] to Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDGSTS.GRSTSTS in the Global Status Register has been updated, indicating successful transition to Global Reset mode before CFDGCTR.GMDC can be changed again.

Figure 35.5 shows the procedure for entering Global Reset mode and Figure 35.6 shows the procedure for exiting Global Reset mode.

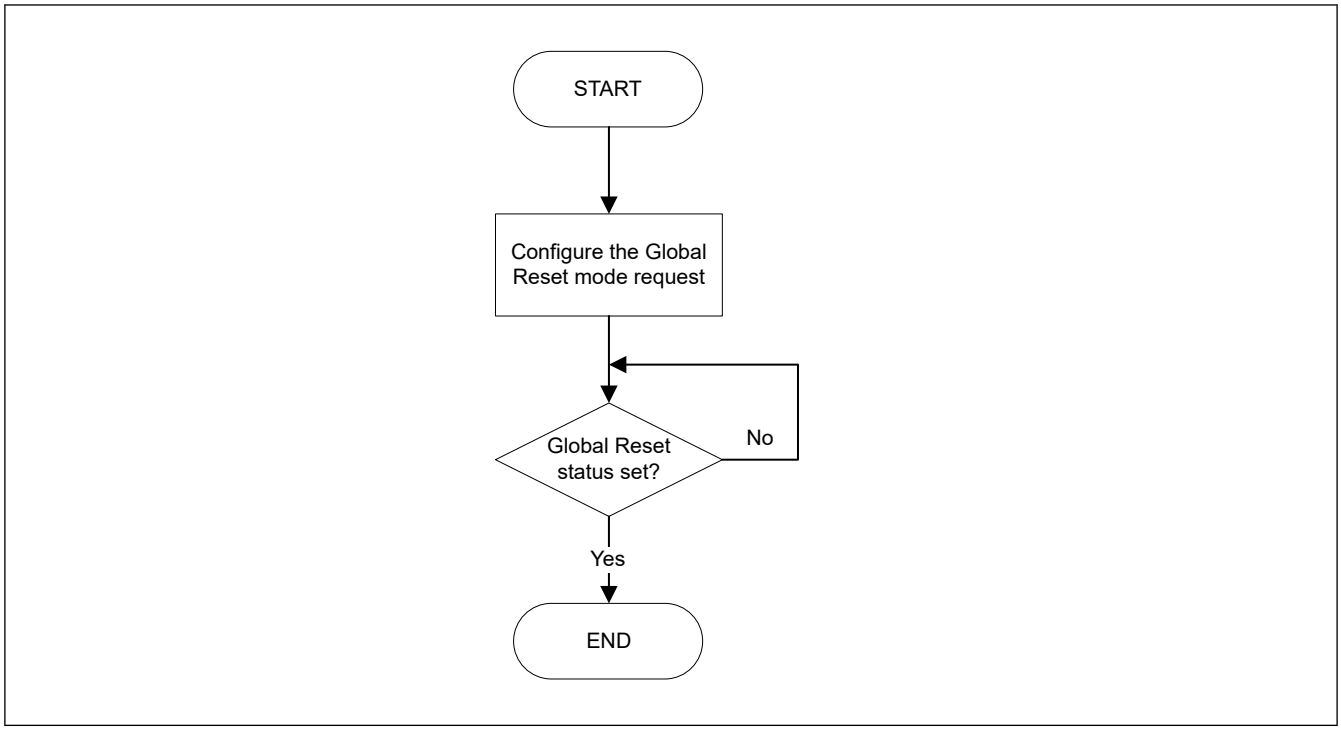


Figure 35.5 Procedure for entering Global Reset mode

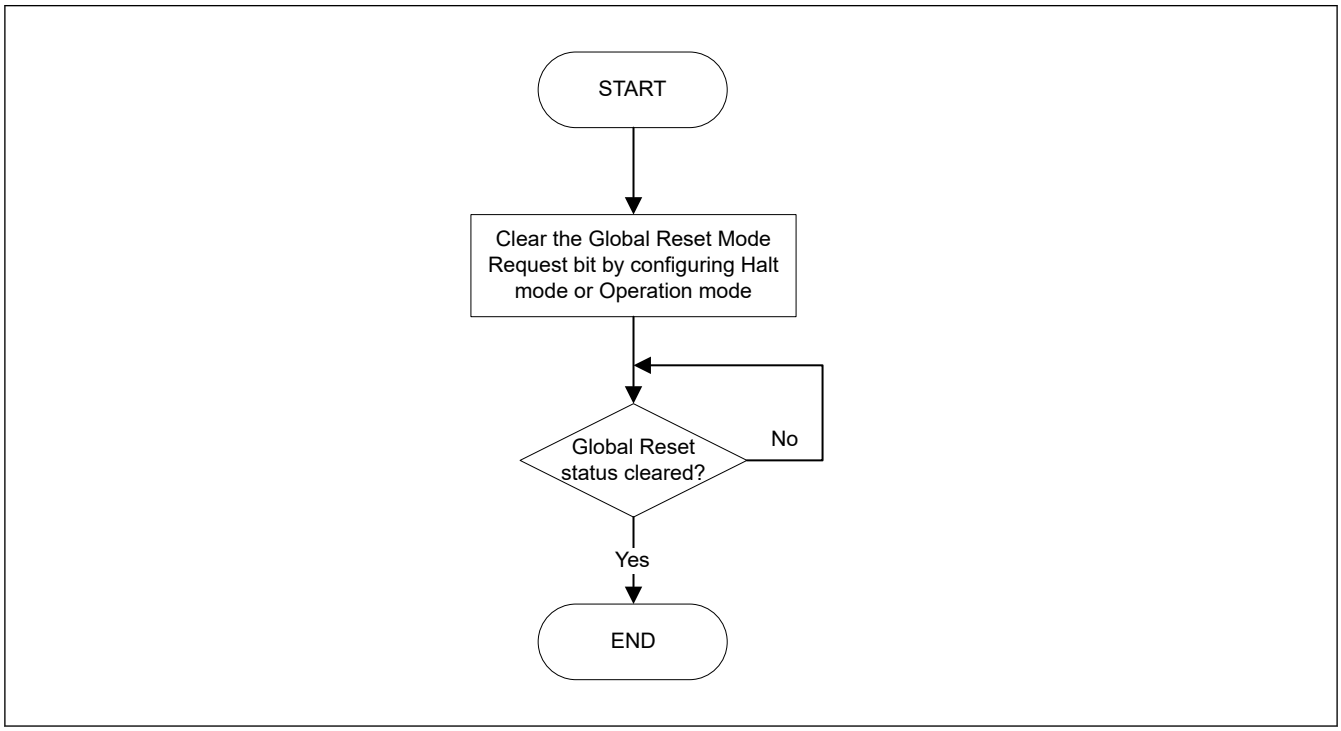


Figure 35.6 Procedure for exiting Global Reset mode

35.4.2.3 Global Halt Mode

The CAN-FD module enters the Global Halt mode in the following ways:

- Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register is configured for Global Halt mode while the CAN-FD module is in Global Reset mode:
  - The channel is in either Channel Reset or Channel Sleep mode and remains in this mode.

- Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register is configured for Global Halt mode while the CAN-FD module is in Global Operation mode:
  - All channels in Channel Reset, Channel Halt, or Channel Sleep mode remain in this mode
  - All channels in Channel Operation mode transit to Channel Halt mode
  - Global Halt Mode Status bit is set when all channels have left Channel Operation mode.

If a transmission or reception is ongoing for a channel, the transition to Channel Halt mode is delayed until completion of the communication.

Similarly, if a channel is in bus-off state, the full bus-off recovery sequence may be delayed depending on the channel configuration.

In Global Halt mode, all communications are suspended and CAN-FD logic does not cause any change to the Status and Flag registers (only when a channel is in the bus-off state that its REC and TEC values are cleared). Additionally, the test mode configuration and control registers are not initialized in this mode.

The Global Halt mode should be used to configure global module test modes.

See section 35.4.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Halt mode is performed.

Setting the Global mode to Halt mode by setting the Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register to 10b sets all Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers to 10b for the channels that are in Channel Operation mode and forces these channels into the Channel Halt mode.

For channels that are already in Channel Reset, Channel Halt, or Channel Sleep mode, this automatic transition is not performed.

Therefore, the Global Halt mode request can be used to shut down all CAN-FD channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channels).

After setting the Global Mode Control bits CFDGCTR.GMDC[1:0] to Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register has been updated to indicate a successful transition to Global Halt mode. Do not specify any other SFR setting until confirming CFDGSTS.GHLTSTS is set.

Figure 35.7 shows the procedure for entering Global Halt mode and Figure 35.8 shows the procedure for exiting Global Halt mode.

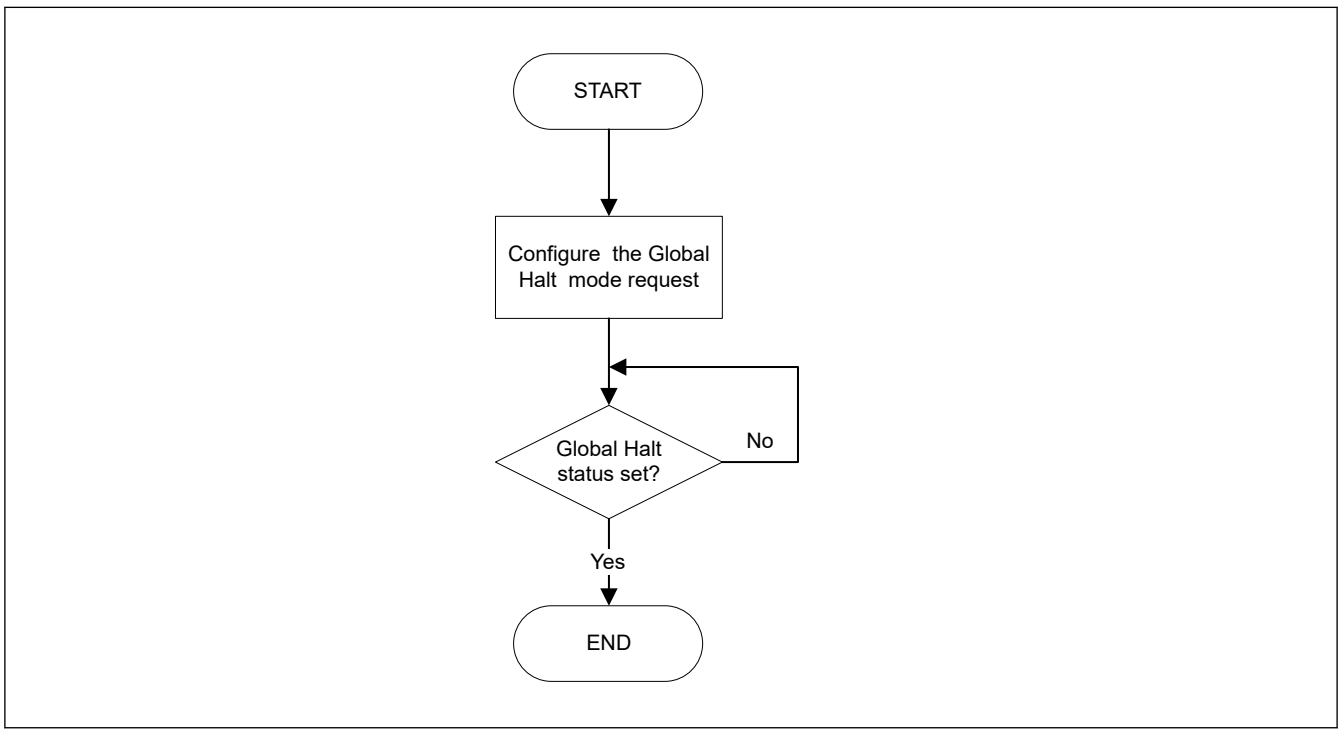


Figure 35.7 Procedure for entering Global Halt mode

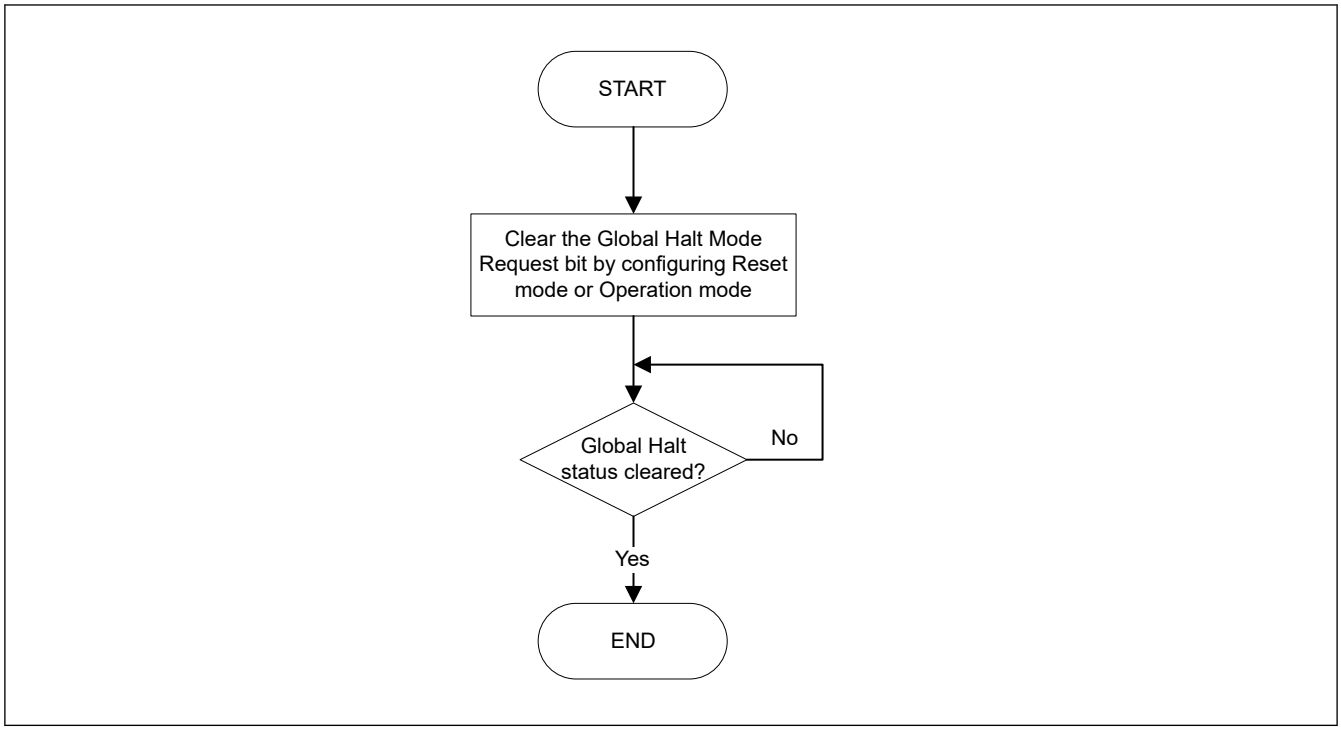


Figure 35.8 Procedure for exiting Global Halt mode

35.4.2.4 Global Operation Mode

The CAN-FD module enters the Global Operation mode when the Global Mode Configuration bits are set to Global Operation mode.

The CAN-FD channels can only be set to Channel Operation mode and start CAN communication when CAN-FD is in Global Operation mode.

After setting the Global Mode Control bits CFDGCTR.GMDC[1:0] to Global Operation mode, it is necessary to confirm that the Global Reset Mode Status bit CFDGSTS.GRSTSTS and the Global Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register have been cleared to indicate a successful transition to Global Operation mode before CFDGCTR.GMDC can be modified again.

Figure 35.9 shows the procedure for entering Global Operation mode and Figure 35.10 shows the procedure for exiting Global Operation mode.

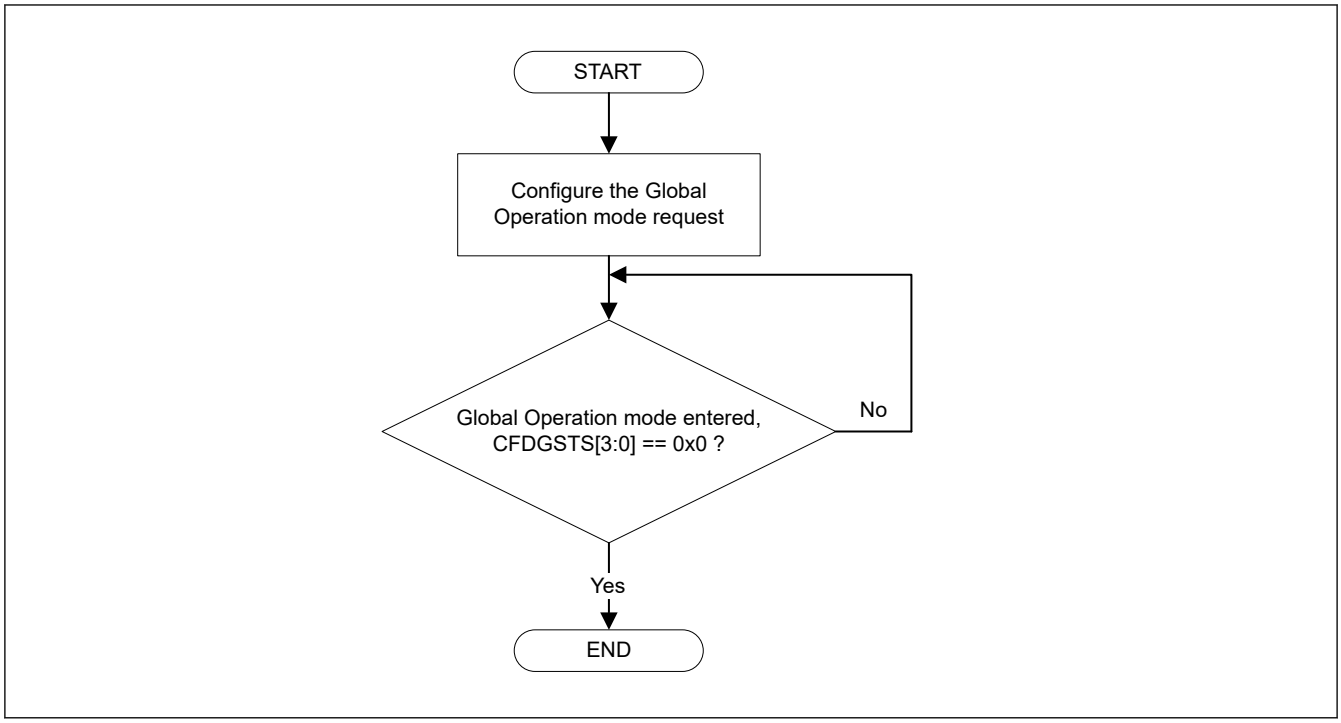


Figure 35.9 Procedure for entering Global Operation mode

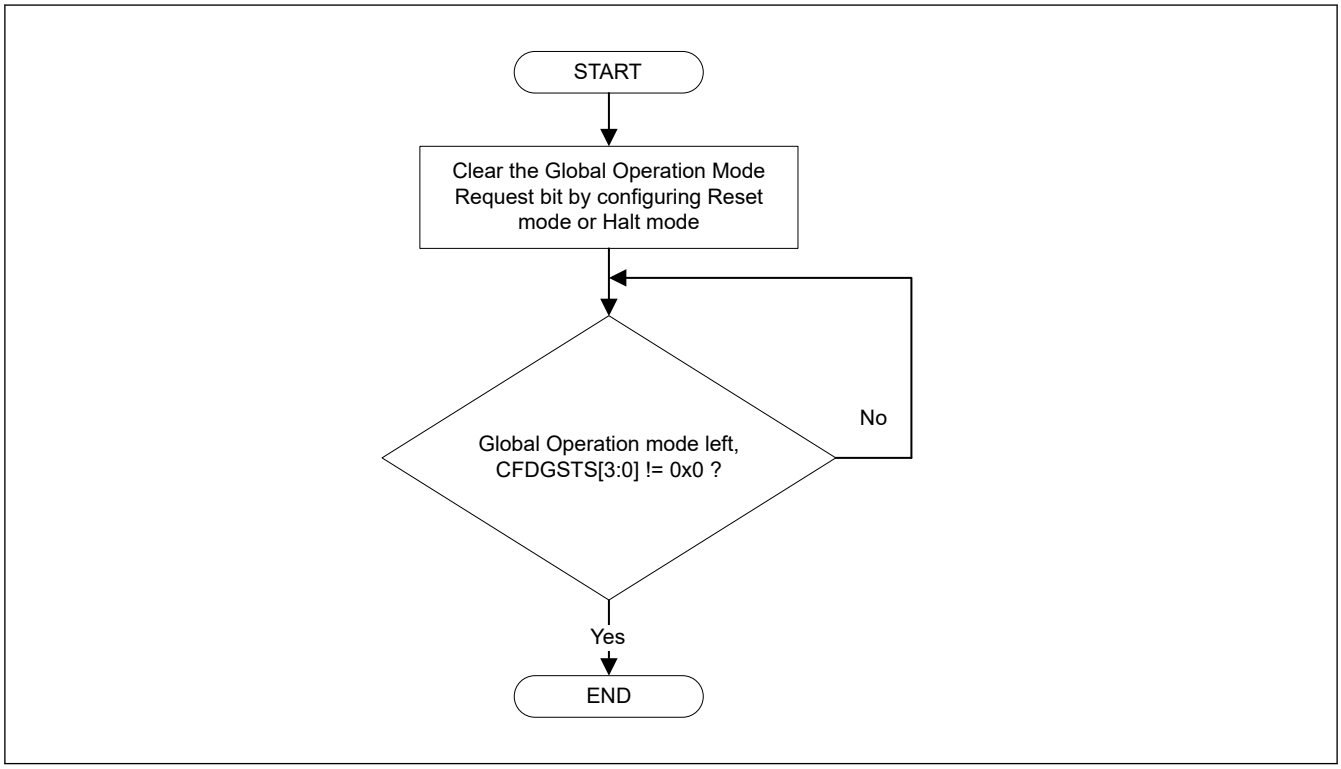


Figure 35.10 Procedure for exiting Global Operation mode

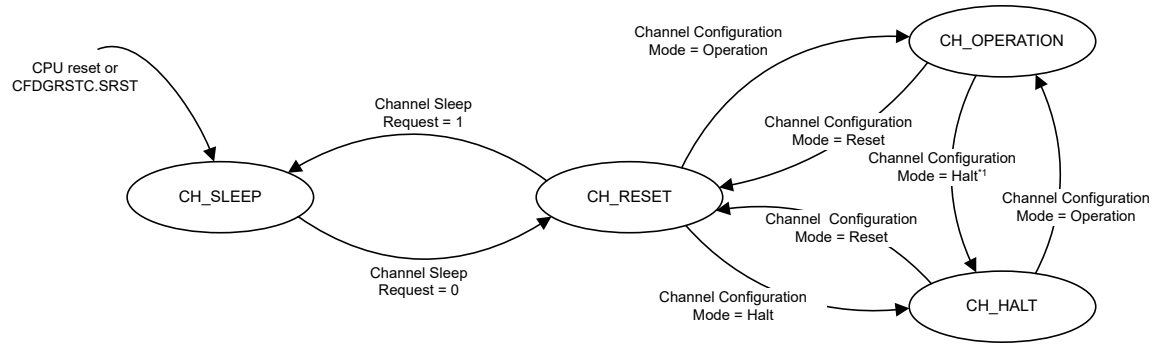
35.4.3 Channel Modes

Each CAN channel can be in one of the following four channel modes:

- Reset
- Halt
- Operation

- Sleep

Figure 35.11 shows possible transitions between the channel modes.



Note 1. The transition timing depends on BOM[1:0] setting in the Channel Control Register (CFDCnCTR):
When BOM[1:0] = 01b, the state transition timing is immediate after entering bus-off state
When BOM[1:0] = 10b, the state transition timing is at the end of bus-off state
When BOM[1:0] = 11b, the state transition timing is matching with the setting of CHMDC to HALT mode

Figure 35.11 Transition between CAN channel modes

35.4.3.1 CAN Channel Sleep Mode

After the release of a hardware reset or after setting and clearing the CFDGRSTC.SRST bit, each CAN channel of the CAN-FD module automatically enters Channel Sleep mode.

Each CAN channel also enters Channel Sleep mode when the related Channel Sleep Mode Request bit is set while the CAN channel is in Channel Reset mode. Do not set this control bit in Channel Halt mode or Channel Operation mode.

Entering the CAN Channel Sleep mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.

After setting the Channel Sleep Mode Request bit, it is necessary to confirm that the Channel Sleep mode status has been updated to indicate a successful transition to Channel Sleep mode before the Channel Sleep Mode Request bit can be cleared again.

During Channel Sleep mode, do not write to channel related registers. Read operation is still possible.

35.4.3.2 CAN Channel Reset Mode

A CAN-FD CAN channel enters the CAN Channel Reset mode in the following ways:

- Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers is configured for Channel Reset mode while the related CAN channel is in Channel Halt mode or Channel Operation mode
- Channel Sleep Mode Request bit is cleared while the related CAN channel is in Channel Sleep mode
- Global Mode Control bits CFDGCTR.GMDC[1:0] is set to Global Reset mode and CAN channel is not in Channel Sleep mode or Channel Reset mode.

In Channel Reset mode, all CAN channel status and flag registers are initialized.

Additionally all channel-related transmission control bits are cleared and the channel-related TX Queue is disabled.

Configuration registers (except the Channel Test Mode registers) are not initialized in this mode and the CAN channel can be configured for communication.

See section 35.4.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Reset mode is performed.

After setting the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] to Channel Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDCnSTS.CRSTSTS in the related Channel Status Registers has been updated to indicate a successful transition to Channel Reset mode before the related CFDCnCTR.CHMDC[1:0] bits can be modified again.

See Table 35.17 for the behavior of transitioning to Channel Reset mode while CAN communication is ongoing.

Table 35.17 Transition behavior in CAN Reset mode and Halt mode

Mode	State		
	Receiver	Transmitter	Bus-Off
CAN Channel Reset mode (CFDCnCTR.CHMDC[1:0] = 01b)	The CAN channel transits to Channel Reset mode without waiting for the completion of the ongoing reception.*1	The CAN channel transits to Channel Reset mode without waiting for the completion of the ongoing transmission.*1	The CAN channel transits to Channel Reset mode without waiting for the completion of the bus-off recovery.
CAN Channel Halt Mode (CFDCnCTR.CHMDC[1:0] = 10b)	The CAN channel transits to Channel Halt mode at the end of the ongoing reception or error.*2	The CAN channel transits to Channel Halt mode after completion of the ongoing transmission.	When CFDCnCTR.BOM[1:0] is set to 00b, a Channel Halt mode request is accepted only after the completion of the full bus-off recovery sequence. When CFDCnCTR.BOM[1:0] is set to 10b, the CAN channel transits automatically to Channel Halt mode after waiting for the completion of the bus-off recovery. When CFDCnCTR.BOM[1:0] is set to 01b, the CAN channel transits automatically to Channel Halt mode without waiting for the completion of the bus-off recovery. When CFDCnCTR.BOM[1:0] is set to 11b, the CAN channel transits to Channel Halt mode as soon as Channel Halt mode is requested (without waiting for the completion of the bus-off recovery).

Note 1. If the entry to Channel Reset is required only at the end of an ongoing communication, then Channel Halt mode can be requested first to prevent interruption of CAN communication by direct transition to Channel Reset mode. After the CAN channel enters Channel Halt mode, the Channel Reset mode can be requested.

Note 2. If CAN communication is locked at a dominant level after an error flag, software can detect this situation by monitoring the channel related Bus Lock flag and resolve the lock condition by setting the CAN channel to Channel Reset mode.

35.4.3.3 CAN Channel Halt Mode

A CAN-FD CAN channel enters the CAN Channel Halt mode in the following ways:

- Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers are configured for Channel Halt mode while the related CAN channel is in Channel Reset mode or Channel Operation mode
- Global Mode Control bits CFDGCTR.GMDC[1:0] are set to Global Halt mode and CAN channel is in Channel Operation mode.

In Channel Halt mode, all channel CAN communication is suspended but all status and flag registers remain unchanged during Channel Halt mode entry (except for the bus-off case where REC and TEC values are cleared for the channel).

In addition, the Channel Test Mode Configuration and Control registers are not initialized in this mode.

The Channel Halt mode should be used to configure Channel Test modes.

See section 35.4.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Halt mode is performed.

After setting the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] to Channel Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDCnSTS.CHLTSTS in the related Channel Status Register is updated to indicate a successful transition to Channel Halt mode before the related CFDCnCTR.CHMDC[1:0] bits can be modified again.

See Table 35.17 for the behavior of transitioning to Channel Halt mode while CAN communication is ongoing.

35.4.3.4 CAN Channel Operation Mode

The Channel Operation mode is activated by setting the CFDCnCTR.CHMDC[1:0] bits to 00b. If 11 consecutive recessive bits are detected after entering CAN Channel Operation mode, the CFDCnSTS.COMSTS bit is set and the CAN channel:

- Enables the functions of the channel communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters

At this point, the CAN channel can start transmission and reception of CAN messages.

Within the CAN Channel Operation mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (see Figure 35.12):

- Channel idle: The CAN channel is neither receiving nor transmitting
- Channel receives: The channel is receiving a CAN message sent by another CAN node
- Channel transmits: The channel is transmitting a CAN message

  Note: The channel may receive its own message simultaneously when Self Test mode is enabled.
- Channel is in bus-off state: The CAN channel is cut-off from CAN bus communication.

After setting the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] to Channel Operation mode, it is necessary to confirm that the Channel Reset Mode Status bit CFDCnSTS.CRSTSTS and the Channel Halt Mode Status bit CFDCnSTS.CHLTSTS in the Channel Status Register have been updated to indicate a successful transition to Channel Operation mode before the related CFDCnCTR.CHMDC[1:0] bits can be changed again.

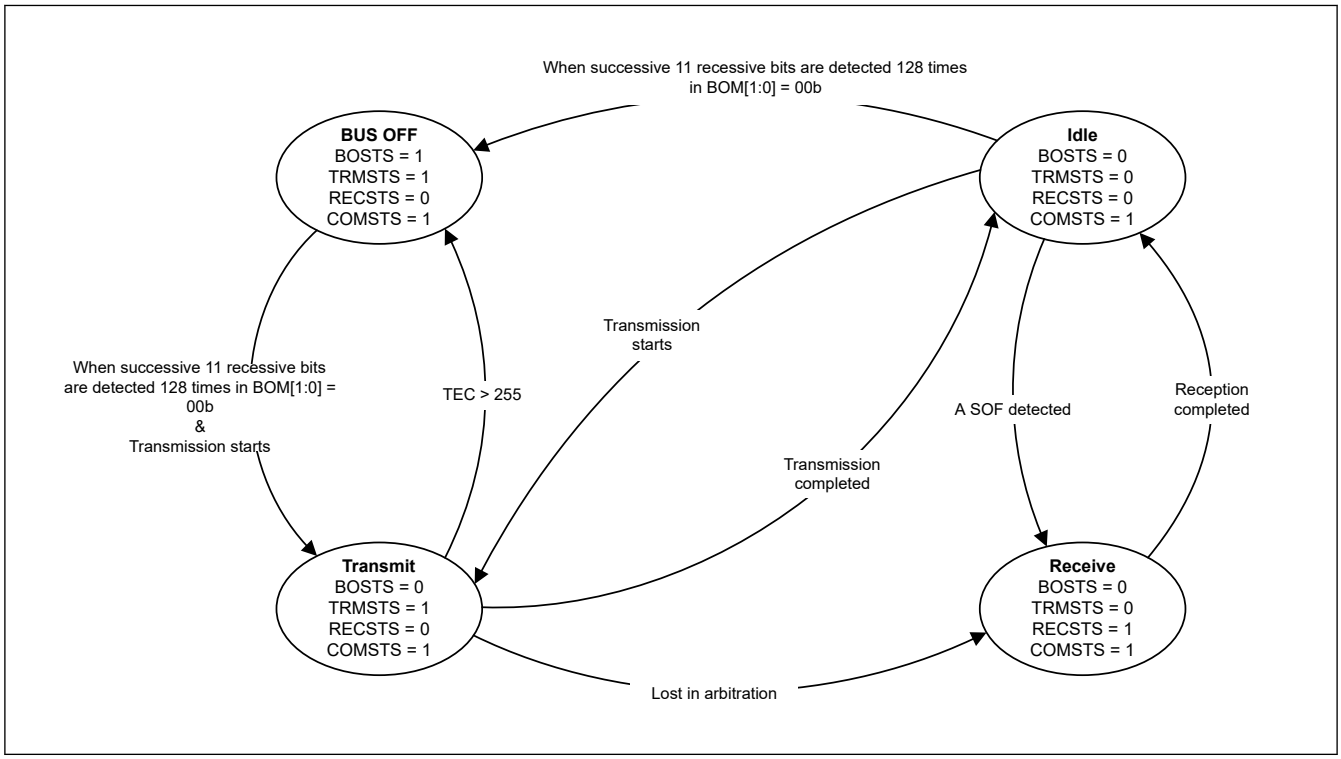


Figure 35.12 Sub-modes of CAN Channel Operation mode (only when BOM[1:0] = 00b)

35.4.3.5 CAN Channel Bus-Off State

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CAN Channel Operation mode from the bus-off state:

- CFDCnCTR.BOM[1:0] = 00b:

Bus-Off recovery is compliant to ISO 11898-1, namely the CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. TEC and REC counters are initialized to 0. The Bus-Off Recovery Flag CFDCnERFL.BORF is set in this case.

- CFDCnCTR.BOM[1:0] = 01b:
  The CAN channel changes the value of the CFDCnCTR.CHMDC[1:0] bits within the CAN Channel Control Register to 10b and switches immediately to Channel Halt mode automatically after entering bus-off state. TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDCnERFL.BORF is not set in this case.
- CFDCnCTR.BOM[1:0] = 10b:
  The CAN channel changes the value of the CFDCnCTR.CHMDC[1:0] bits within the CAN Channel Control Register to 10b as soon as it reaches bus-off state and enters Channel Halt mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDCnERFL.BORF is set in this case.
- CFDCnCTR.BOM[1:0] = 11b:
  Bus-off recovery is initiated but CAN channel can immediately enter Channel Halt mode when still in bus-off state if a request is made to enter Channel Halt mode.
  TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDCnERFL.BORF is not set.
  Without setting CFDCnCTR.CHMDC[1:0] = 10b and when 11 recessive bits is detected 128 times continuously, transition conditions become the same as CFDCnCTR.BOM[1:0] = 00b.

Note: If the recovery from bus-off occurs normally in Channel Halt mode (after waiting for 128 sequences of 11 consecutive recessive bits), and no halt request has been generated during this period, then the Bus-Off Recovery flag CFDCnERFL.BORF is set.

When software writes to the CFDCnCTR.CHMDC[1:0] bit at the same time as the CAN channel enters Halt mode (at the start of bus-off when CFDCnCTR.BOM[1:0] = 01b, or at the end of bus-off when CFDCnCTR.BOM[1:0] = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CFDCnCTR.CHMDC[1:0] bits to Channel Halt mode request is performed when the CFDCnCTR.CHMDC[1:0] bits value is previously 00b (Channel Operation mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting CFDCnCTR.RTBO to 1. The error state changes from bus-off state to integrating state with a maximum delay of 1 CAN bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The Bus-Off Recovery Flag is not set in this case, and the TEC and REC counters are initialized to 0.

Before setting CFDCnCTR.RTBO to 1, all pending transmissions from the TX message buffers, TX Queues and/or Common FIFO in TX or GW mode should be disabled.

The disable of the pending transmission message buffer, TX Queue or FIFO must be confirmed by the corresponding acknowledge flags.

For the TX message buffer, the acknowledge flags are the Transmission Result Flags (CFDTMSTSn.TMTRF[1:0]). For the TX Queue, it is the TX Queue Empty flag (CFDTXQSTSn.TXQEMP). For the FIFO, it is the FIFO Empty flag (CFDCFSTSn.CFEMP).

The CFDCnCTR.RTBO bit should be used for bus-off recovery only when CFDCnCTR.BOM[1:0] is set to 00b.

Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.

Table 35.18 shows the settings for the Bus-Off Entry flag CFDCnERFL.BOEF and the Bus-Off Recovery flag CFDCnERFL.BORF for the different configurations of CFDCnCTR.BOM[1:0].

Table 35.18 Behavior of Bus-off Entry and Recovery flags (1 of 2)

BOM[1:0]	BOEF bit setting	BORF bit setting
00b	Always (on entry to bus-off)	Always (on exit from bus-off)
00b CFDCnCTR.RTBO set to 1	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software sets CFDCnCTR.RTBO to 1
01b	Always (on entry to bus-off)	Never
10b	Always (on entry to bus-off)	Always (on exit from bus-off)

Table 35.18 Behavior of Bus-off Entry and Recovery flags (2 of 2)

BOM[1:0]	BOEF bit setting	BORF bit setting
11b	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software issues a Halt request

For an efficient software procedure, it is not necessary to wait for the bus-off recovery sequence to end.

It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in Figure 35.13.

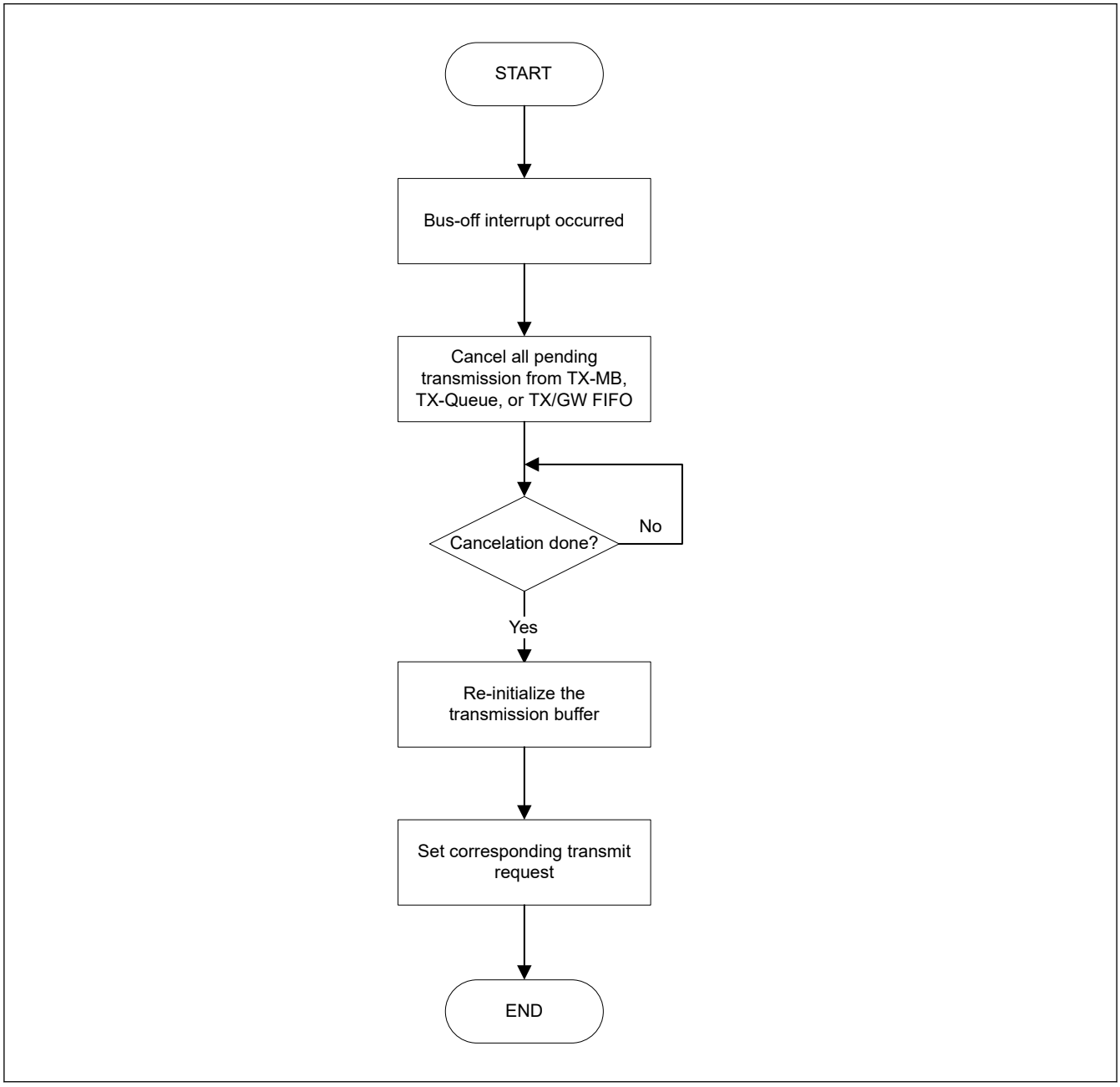


Figure 35.13 Transmission re-initialization during bus-off state

35.4.4 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:

- Changing the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers does not affect the Global Mode Control bits CFDGCTR.GMDC[1:0].

- Changing the Global Mode Control bits CFDGCTR.GMDC[1:0] affects the channel mode control as described in Table 35.19.

Table 35.19 Instruction between Global and Channel mode transition

Global mode change	Channel mode	Channel mode transition action
Sleep → Reset	Sleep	Channel remains in Sleep mode
Sleep → Halt	— (Global mode change not possible)	
Sleep → Operation	— (Global mode change not possible)	
Reset → Sleep	Sleep	Channel remains in Sleep mode
	Reset	Channel Sleep request bit is set automatically, channel transits to Sleep mode
Reset → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Reset → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Halt → Sleep	— (Global mode change not possible)	
Halt → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel transits to Reset mode
Halt → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
Operation → Sleep	— (Global mode change not possible)	
Operation → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel transits to Reset mode
	Operation	Channel mode control is set to Reset mode, channel transits to Reset mode
Operation → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
	Operation	Channel mode control is set to Halt mode, channel transits to Halt mode after communication finished

35.4.4.1 Timing of Global Mode Change

The transition time for the Global mode changes are shown in the following table.

From	To	Maximum transition time
GL_SLEEP	GL_RESET	3 peripheral clock cycles*2
GL_RESET	GL_SLEEP	3 peripheral clock cycles
GL_RESET	GL_HALT	10 peripheral clock cycles
GL_RESET	GL_OPERATION	10 peripheral clock cycles
GL_HALT	GL_RESET	2 CAN bit times
GL_HALT	GL_OPERATION	3 peripheral clock cycles
GL_OPERATION	GL_RESET	2 CAN bit times
GL_OPERATION	GL_HALT	3 CAN frames*1, *3

Note 1. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.

Note 2. Exit GL_SLEEP mode only when CFDGSTS.GRAMINIT is cleared.
Note 3. TQ, CAN frame, and CAN bits are related to the individual channels. For maximum transition time, the channel with the lowest baud rate must be used.

35.4.4.2 Timing of Channel Mode Change

The transition time for the Channel mode changes are shown in the following table.

From	To	Maximum transition time
CH_SLEEP	CH_RESET	3 peripheral clock cycles
CH_RESET	CH_SLEEP	3 peripheral clock cycles
CH_RESET	CH_HALT	3 CAN bit times
CH_RESET	CH_OPERATION	4 CAN bit times
CH_HALT	CH_RESET	2 CAN bit times
CH_HALT	CH_OPERATION	4 CAN bit times*2
CH_OPERATION	CH_RESET	2 CAN bit times
CH_OPERATION	CH_HALT	2 CAN frames*1 *3

Note 1. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CFDCnCTR.BOM[1:0] bits.
Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.
Note 3. In general, if the baud rate prescaler value CFDCnNCFG.NBRP is changed in CH_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the TQ clock, and new BRP value is captured when the counter reaches the value 0.

35.5 Initialization

Before joining CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data rate)
- Baud rate setting (nominal and data rate)
- CAN-FD setting
- Acceptance filter setting (configuration of Global Acceptance Filter List)
- Reception, Transmission and GW-FIFO setting
- CAN Operation mode setting

35.5.1 Initialization of CAN Clock, Bit Timing and Baud Rate

35.5.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the limitations that apply to the segment setting.

1. Each segment setting
   SS = Fixed to 1 TQ
   TSEG1 = Refer to (CFDCnNCFG) and (CFDCnDCFG)
   TSEG2 = Refer to (CFDCnNCFG) and (CFDCnDCFG)
   SJW = Refer to (CFDCnNCFG) and (CFDCnDCFG)
   SS + TSEG1 + TSEG2 = 5 to 49 TQs for data bit rate and 8 to 385 for nominal bit rate
2. Limitations on TSEG1, TSEG2 and SJW
   TSEG1 (N) > TSEG2 (N) ≥ SJW (N)
   TSEG1 (D) ≥ TSEG2 (D) ≥ SJW (D)
   When only classical frames are used, configure the bit fields TSEG1 and TSEG2 of CFDCnDCFG to valid values.
   Table 35.20 shows an example of how to set the bit timing to achieve the required Sample Point settings.

Table 35.20 Transition behavior in CAN Reset mode and Halt mode

1 Bit	Set value (TQ)				Sample Point*1 (%)
	SS	TSEG1	TSEG2	SJW	
5 TQ	1	2	2	1	60.00
8 TQ	1	4	3	1	62.50
	1	5	2	1	75.00
10 TQ	1	6	3	1	70.00
	1	7	2	1	80.00
12 TQ	1	8	3	1	75.00
	1	9	2	1	83.33
15 TQ	1	10	4	1	73.33
	1	11	3	1	80.00
16 TQ	1	10	5	1	68.75
	1	11	4	1	75.00
20 TQ	1	12	7	1	65.00
	1	13	6	1	70.00
24 TQ	1	15	8	1	66.66
	1	16	7	1	70.83
50 TQ	1	39	10	4	80.00

Note 1. Sample point (in case of 75%)

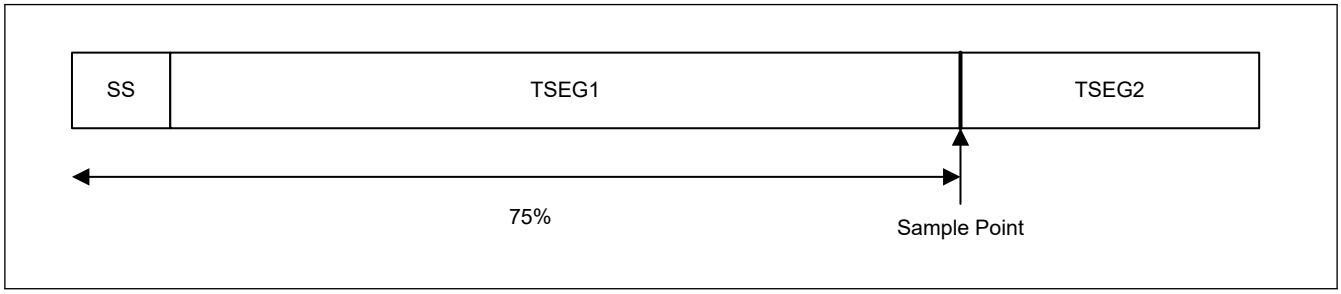


Figure 35.14 Sample point (in case of 75%)

35.5.1.2 CAN Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured individually for each channel using the related CFDCnNCFG and CFDCnDCFG registers.

Figure 35.15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (TQ), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the baud rate prescaler (nominal and data rate).

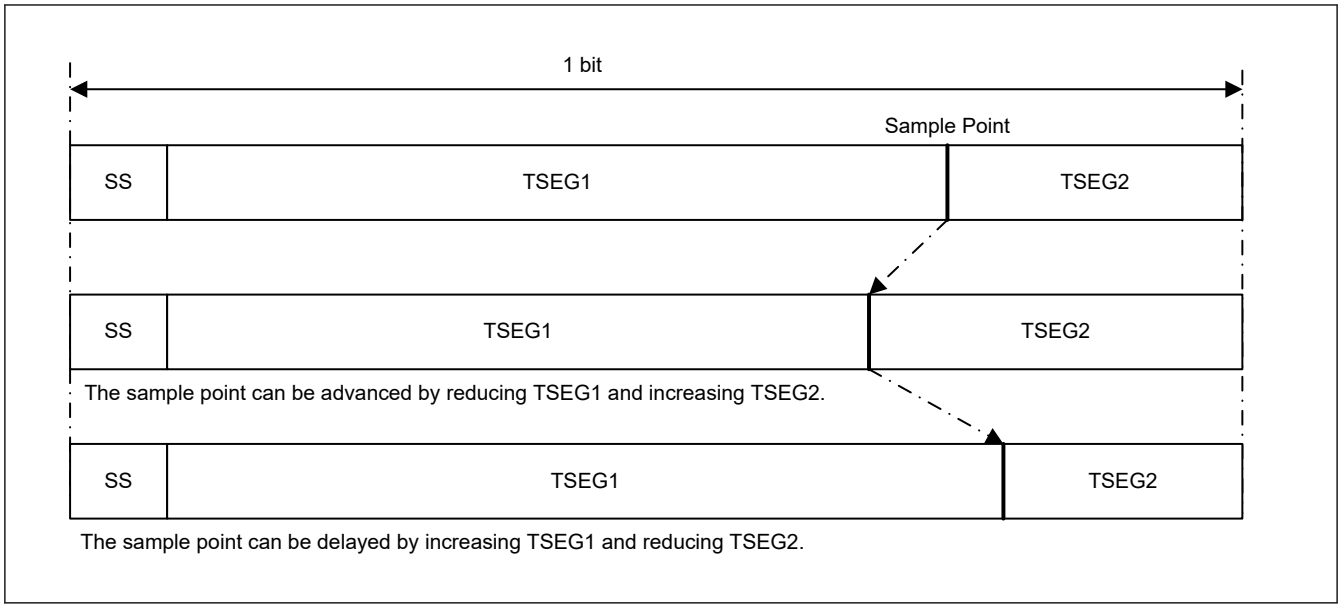


Figure 35.15 Segment composition of a bit and the sample point

1. SS: Synchronization Segment
   This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. This comprises of intermission, suspend transmission, bus idle, during bus idle, and all nodes that can start transmission.
2. TSEG1: Time Segment 1
   This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2
   This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.
4. SJW: Resynchronization Jump Width
   This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 35.15 shows only one symbolic sample point.

35.5.1.3 Baud Rate

Either PCLKM or PCLKCAN can be selected globally for all CAN channels as CAN communication clock.

The transfer speed is determined by the DLL clock, the divide-by-N value of the baud rate prescaler, and the number of TQs in one bit.

$$\text{baud rate} = \frac{\text{DLL\_Clock}}{(\text{number\_of\_time\_quanta\_per\_bit}) \times (\text{BRP} + 1)}$$

Figure 35.16 shows a block diagram of the circuit that generates the CAN channel system clock, Table 35.21 and Table 35.22 show baud rate examples.

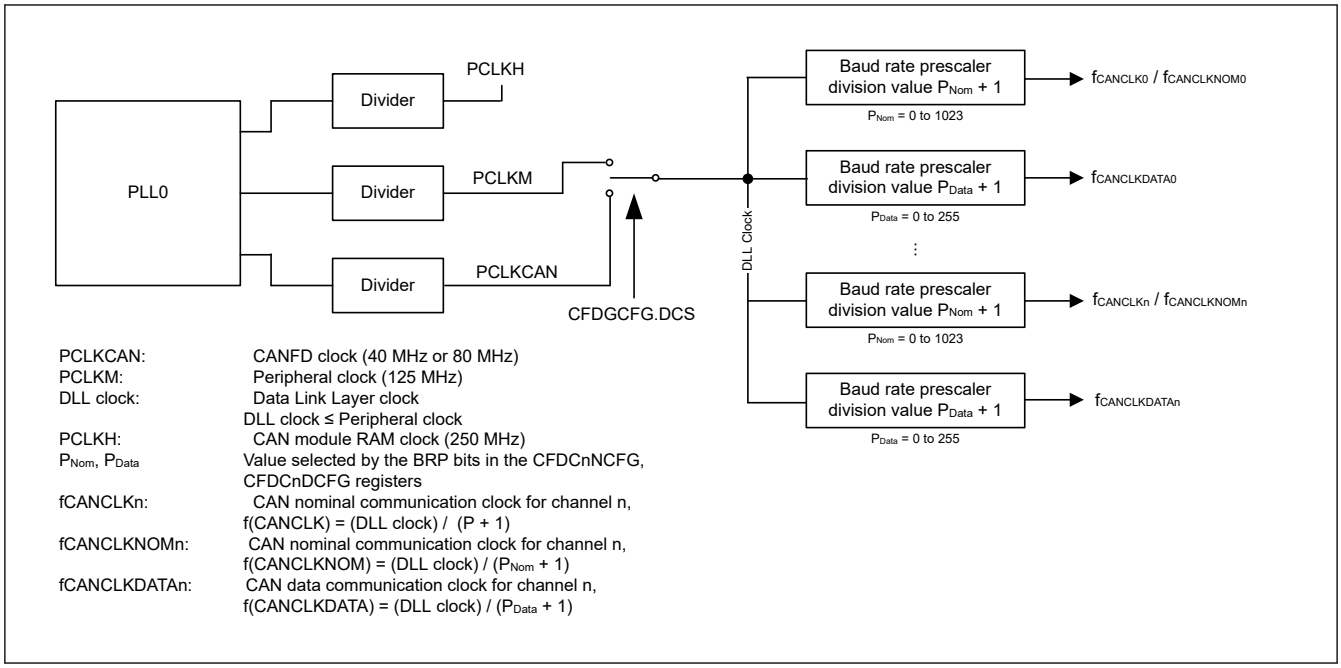


Figure 35.16 Block diagram of the circuit that generates the CAN channel communication clock

Table 35.21 Nominal baud rate calculation formula and example CAN communication configurations

Baud rate calculation formula	(DLL clock) (Baud rate prescaler divide-by-N value*1) × (Number of TQs in one bit)		
	80 MHz	**40 MHz**	**125 MHz**
1 Mbps	8 TQ (10) 20 TQ (4)	8 TQ (5) 20 TQ (2)	25 TQ (5)
500 Kbps	8 TQ (20) 20 TQ (8)	8 TQ (10) 20 TQ (4)	25 TQ (10)
250 Kbps	8 TQ (40) 20 TQ (16)	8 TQ (20) 20 TQ (8)	25 TQ (20)
125 Kbps	8 TQ (80) 20 TQ (32)	8 TQ (40) 20 TQ (16)	25 TQ (40)
83.3 Kbps	8 TQ (120) 12 TQ (80) 16 TQ (60) 24 TQ (40)	8 TQ (60) 12 TQ (40) 16 TQ (30) 24 TQ (20)	10 TQ (150) 12 TQ (125) 15 TQ (100) 20 TQ (75)
33.3 Kbps	8 TQ (300) 12 TQ (200) 16 TQ (150) 20 TQ (120) 24 TQ (100)	8 TQ (150) 12 TQ (100) 16 TQ (75) 20 TQ (60) 24 TQ (50)	10 TQ (375) 15 TQ (250) 25 TQ (150)

Note: Shown in ( ) are the baud rate prescaler divided-by-N values.
Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 to 1023), P: value selected by the BRP bits in the Channel Configuration Registers.

Table 35.22 Baud rate calculation example for nominal and data bit rate CAN communication configurations (1 of 2)

Baud rate calculation formula	(DLL clock) (Baud rate prescaler divide-by-N value*1) × (Number of TQs in one bit)		
	80 MHz	**40 MHz**	**125 MHz**
Nominal 1 Mbps Data 8 Mbps	80 TQ (1)	40 TQ (1)	125 TQ (1)
	10 TQ (1)	5 TQ (1)	Not possible

Table 35.22 Baud rate calculation example for nominal and data bit rate CAN communication configurations (2 of 2)

Baud rate calculation formula	(DLL clock) (Baud rate prescaler divide-by-N value*1) × (Number of TQs in one bit)		
	80 MHz	40 MHz	125 MHz
Nominal 1 Mbps Data 5 Mbps	80 TQ (1)	40 TQ (1)	125 TQ (1)
	16 TQ (1)	8 TQ (1)	25 TQ (1)
Nominal 500 Kbps Data 2 Mbps	160 TQ (1)	80 TQ (1)	250 TQ (1)
	40 TQ (1)	20 TQ (1)	Not possible

Note: Shown in ( ) are the baud rate prescaler divided-by-N values.
Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 to 1023), P: value selected by the BRP bits in the Channel Configuration Registers.

For optimum clock tolerance in networks using the FD frame format, the length of the time quantum should be the same in nominal bit time and in data bit time. This means CFDCnNCFG.NBRP = CFDCnDCFG.DBRP.

Additionally, if transceiver delay compensation is used, do not program the CFDCnDCFG.DBRP[7:0] bits to be greater than 1, as 1 means divide by 2.

35.5.1.4 Setting of CAN Clock, Bit Timing and Baud Rate

Figure 35.17 shows the procedure for setting the CAN clock and the baud rate for each channel.

These settings should be performed during Channel Reset mode (Configuration mode) for the CAN channels.

Before going to channel communication state, the baud rate must be configured, otherwise the mode does not switch correctly.

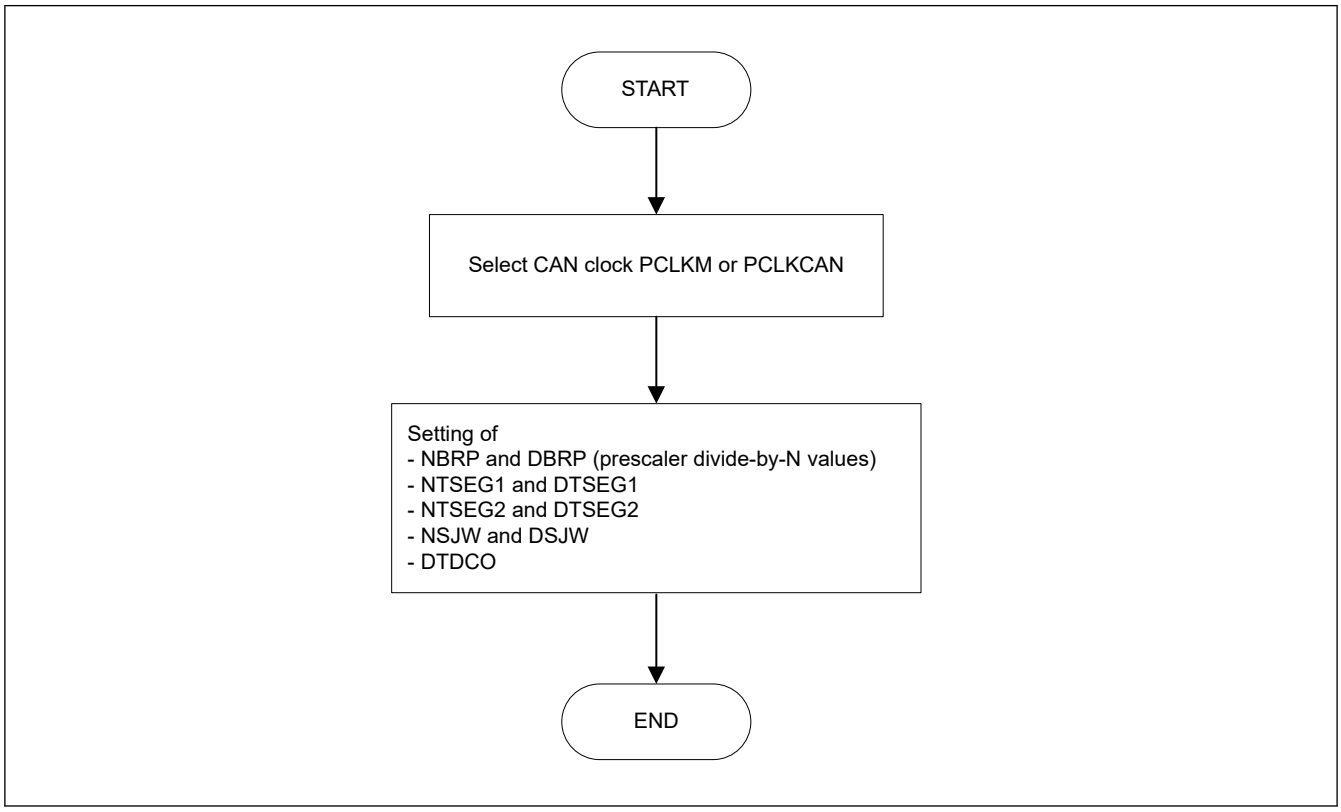


Figure 35.17 Procedure for setting the CAN bit timing and baud rate

35.5.1.5 Transmitter Delay Compensation

When a high baud rate is used such as 5 to 8 Mbps for the data phase, the transmitter delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CAN-FD frame. The TDC compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.

There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CAN-FD frames. This is derived from the Transceiver Delay Compensation Result bit (CFDCnFDSTS.TDCR) as shown in Figure 35.18.

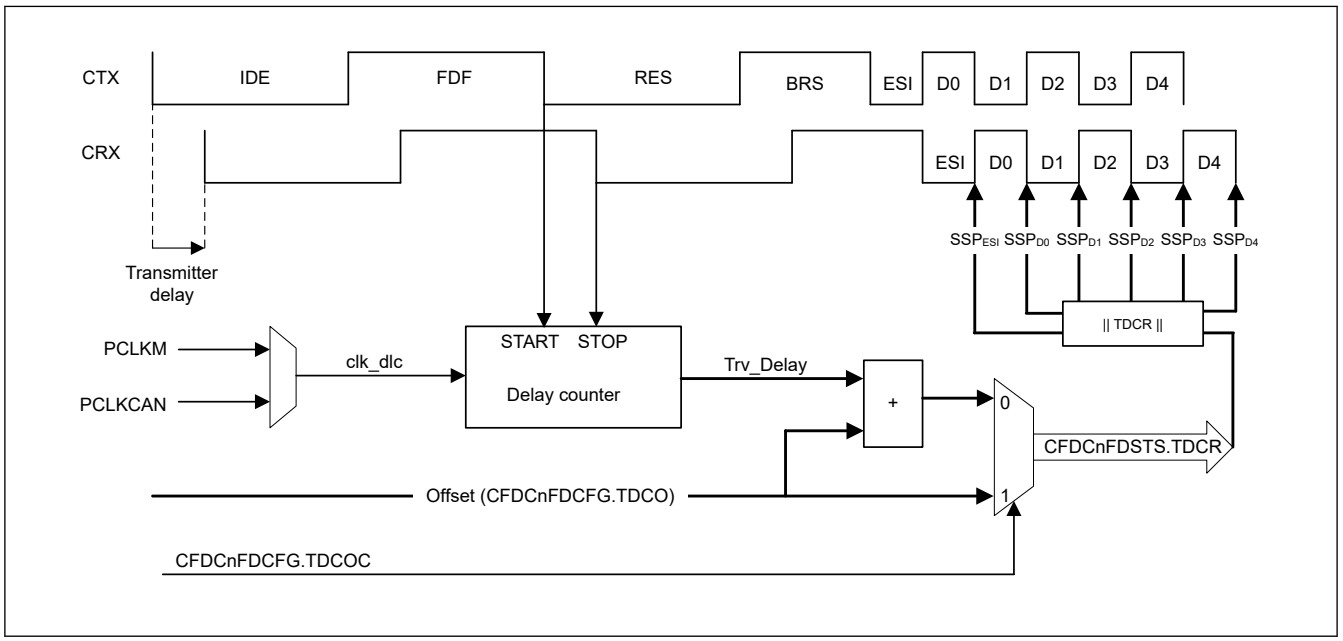


Figure 35.18 Transmitter delay compensation

The measured Trv_Delay is based on the number of clk_dlc clock cycles. The delay is counted up by 1 for each started clock until the dominant value is seen on CAN_RX. Figure 35.19 shows the measured result. Trv_Delay counted to maximum 127 with a clk_dlc clock.

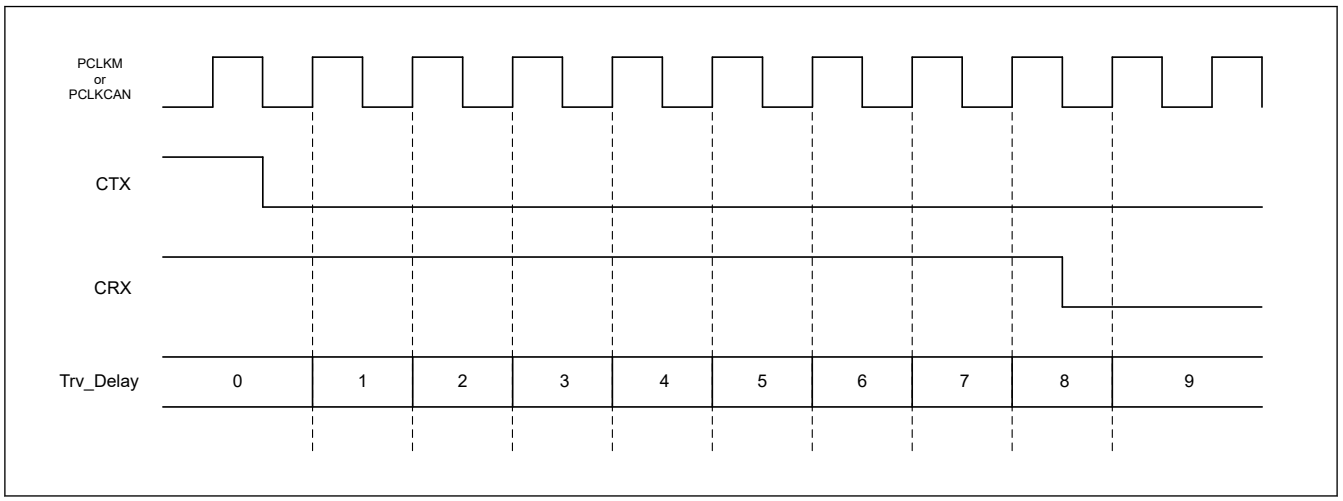


Figure 35.19 Trv_Delay measurement example

The SSP is calculated by taking the result from CFDCnFDSTS.TDCR and rounding the value down to the nearest integer number of data time quanta.

Figure 35.20 shows the positioning of the secondary sample point. When CFDCnFDCFG.TDCOC is equal to 0, the SSP is equal to the Trv_Delay (measured delay) + CFDCnFDCFG.TDCO, rounded down to the nearest integer number of time quanta. Usually, the TDCO value should have the size of (Sync Segment data + TSEG1 data) to position the SSP to a theoretical location of the sample point.

If the CFDCnFDCFG.TDCOC is equal to 1, the SSP is defined by CFDCnFDCFG.TDCO. If CFDCnDCFG.DBRP[7:0] is greater than 0, the value is also rounded down to the nearest integer number of time quanta.

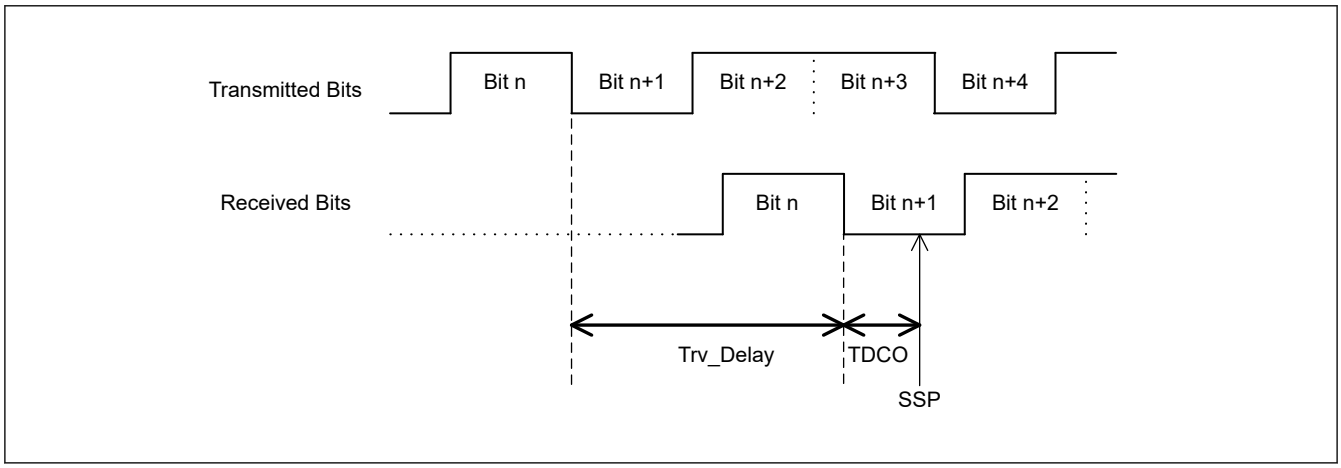


Figure 35.20 Position of the secondary sample point

The maximum delay (Trv_Delay + TDCO) which can be compensated by the CAN-FD module is (6 data bits - 2clk_dlc).

The ISO 11898-1 allows you to set different values for BRP_data and BRP_nom.

If different values are used for CFDCnNCFG.NBRP and CFDCnDCFG.DBRP, then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in Figure 35.21.

The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means CFDCnNCFG.NBRP = CFDCnDCFG.DBRP.

Different bit rates can be achieved by selecting different configuration values for the Time Segments. The nominal bit rate can be configured from 8 to 385 TQs and the data bit rate from 5 to 49 TQs.

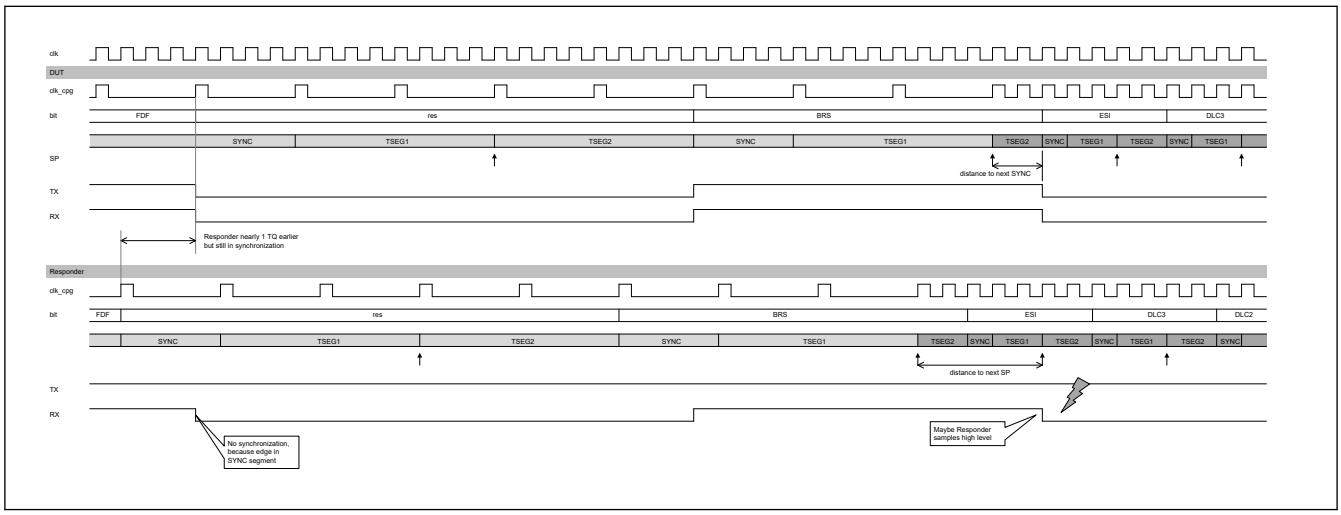


Figure 35.21 Loss of synchronization between 2 CAN nodes

The transmitter delay compensation measurement result is updated at the falling edge from FDF bit to RES bit when configured accordingly (CFDCnFDCFG.TDCE = 1, CFDCnFDCFG.TDCOC = 0).

Figure 35.22 shows the read flow to get the measured transmitter delay compensation result.

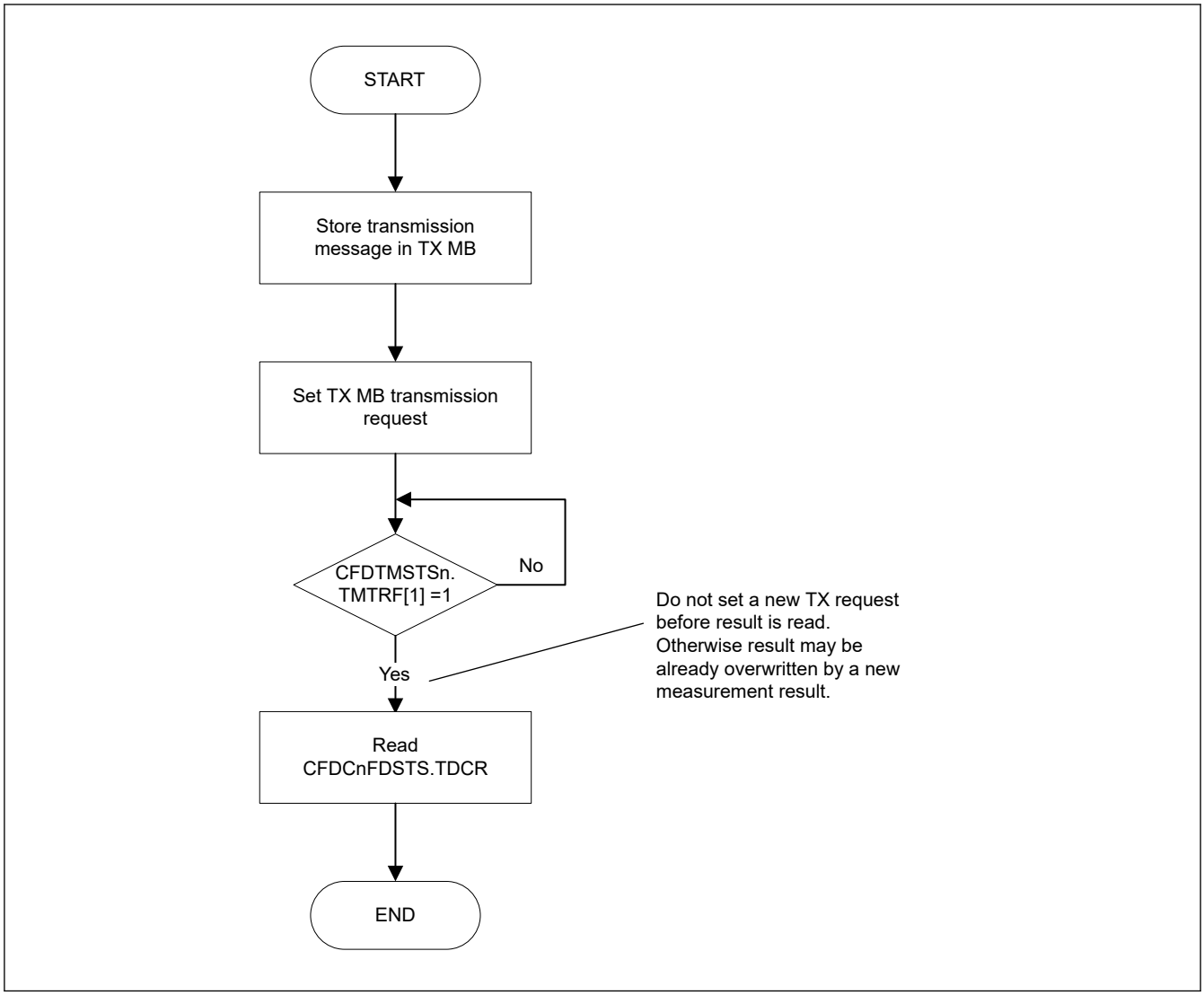


Figure 35.22 TDC result read flow

35.5.2 CAN Module Configuration after Hardware Reset

After a hardware reset (power on reset) or after setting and clearing the CFDGRSTC.SRST bit, the CAN-FD module enters Global Sleep mode automatically.

To enable configuration of the CAN-FD module, you must exit Sleep mode by clearing the Global Sleep Request bit CFDGCTR.GSLPR to 0.

After a hardware reset, the module starts RAM initialization, the CFDGSTS.GRAMINIT bit in the Global Status Register is set automatically to indicate that the CAN-FD logic is initializing the RAM.

After RAM initialization is complete, this bit is cleared automatically.

Do not access registers of the CAN-FD in either read or write until RAM initialization is complete and the CFDGSTS.GRAMINIT bit is cleared.

Before going to communication mode, the Global Acceptance Filter List and message FIFO buffers must be configured. In addition, each required CAN channel must be configured such as CAN bit timing. For this configuration, all required CAN channels must be released from Channel Sleep mode and must be configured for communication in Channel Reset mode (Configuration mode).

For this, all required CAN channels must be released from Channel Sleep mode and must be configured for communication in Channel Reset mode (Configuration mode).

Figure 35.23 shows the configuration procedure. For details about each step, see section 35.6. Acceptance Filtering Function using Global Acceptance Filter List (AFL), section 35.7. FIFO Buffers and Normal Message Buffer Configuration, section 35.8. Interrupt and DMA, and section 35.5.1.3. Baud Rate.

The CAN-FD module does not perform RAM initialization sequence after executing a software reset by setting CFDGRSTC.SRST.

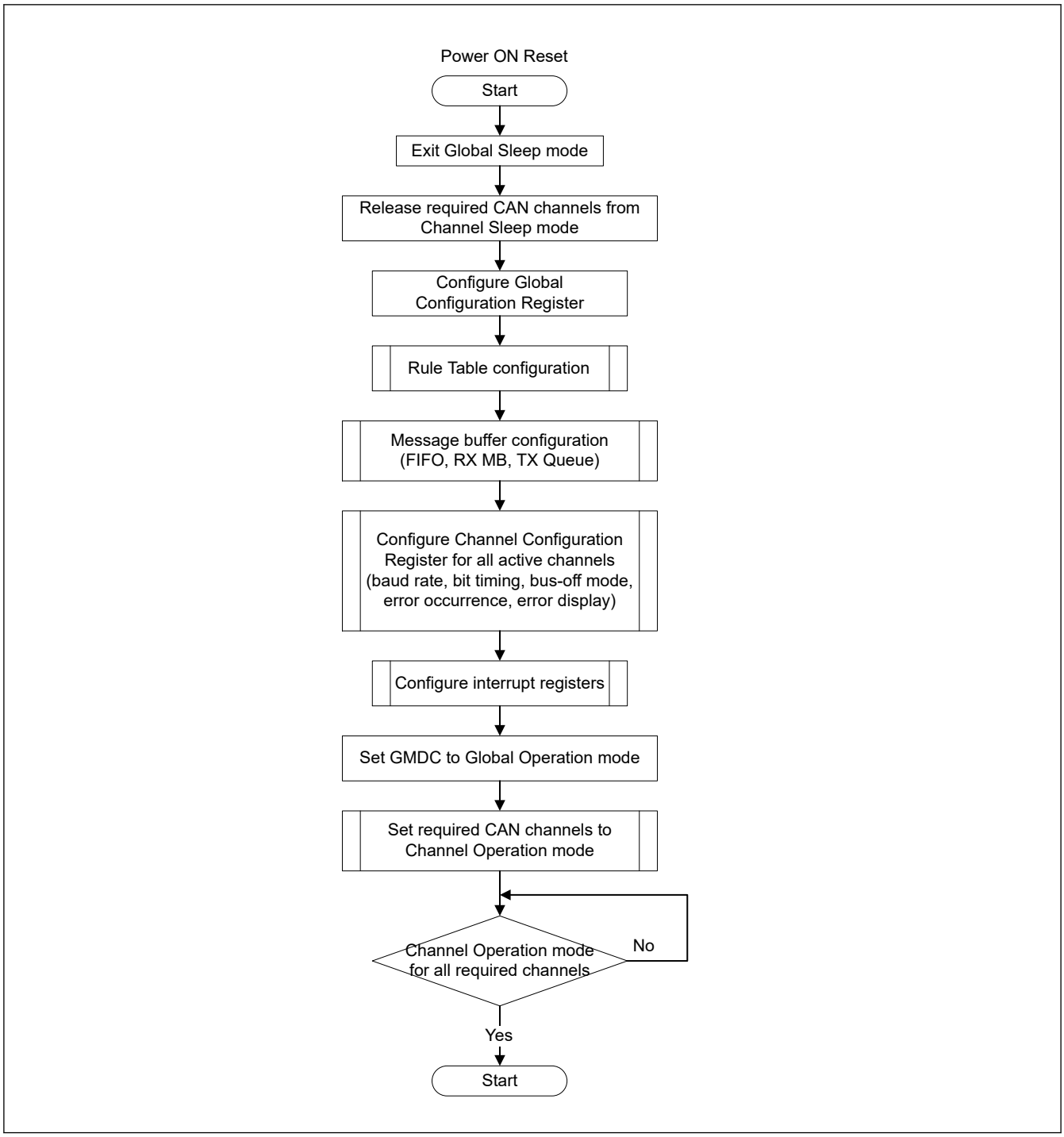


Figure 35.23 Configuration procedure after a hardware reset

35.6 Acceptance Filtering Function using Global Acceptance Filter List (AFL)

35.6.1 Overview

The CAN-FD module can handle message acceptance filtering for all channels with a global Acceptance Filter List (AFL). Each element of the AFL defines a filter rule for messages received on a specific channel.

The following actions are performed based on the AFL entries:

- Acceptance filtering based on received CAN identifier and masking
- DLC filtering based on received DLC value
- Message data payload according to the CFDGCFG.CMPOC bit
- Storage of accepted messages in the message buffer objects defined in the related AFL entry
- Attaching a 16-bit pointer to the stored messages defined in the related AFL entry, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages defined in the related AFL entry

The 2-channel CAN-FD module allows a maximum of 256 AFL entries across all channels with a maximum of 128 AFL entries per channel.

During the acceptance filtering process, each AFL entry in a channel is checked against the received message by the acceptance filter unit. The check starts from the lowest AFL entry number for this channel.

The AFL search stops when a match of the received identifier with a configured identifier/mask combination occurs or when the received identifier has been compared against all AFL entries defined for the related channel. If no match occurs, then the received message is rejected. No notification is given to the application in this case.

Additionally, an automatic DLC filtering is performed for each accepted message if DLC check is globally enabled. If the DLC value of the received message is equal to or higher than the configured DLC value in the matching AFL entry, the DLC check is passed.

If DLC replacement (CFDGCFG.DRE bit) is enabled, DLC value configured in the matching AFL entry is greater than 0x0 and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received on the CAN bus are not stored in the destination RXMB or FIFO Buffer. These additional data bytes are stored as 0x00 in the destination RXMB or FIFO Buffer.

If DLC replacement is enabled and DLC value of matching AFL entry is 0x0, then the received value of DLC is stored in the destination RX MB or FIFO Buffer.

If DLC replacement (the CFDGCFG.DRE bit) is disabled and DLC check passes, then the received value of DLC on the CAN bus is stored in the destination RXMB or FIFO buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received from the CAN bus are also stored in the destination RXMB or FIFO buffer.

If DLC value of the received message is less than the configured DLC value in the matching AFL entry, then DLC check fails. In this case, the received message is rejected and is not stored in any RXMB or FIFO buffer.

Additionally, DLC check failure is flagged by the DLC Error Flag in the Global Error Flag Register. If configured, an error interrupt is also generated. The DLC replacement configuration has no impact if the DLC check fails.

If a message has passed both acceptance filtering and DLC filtering, it is stored in a single reception message buffer and/or in FIFO buffers configured for reception or gateway function.

This message storage target information is also defined in the same AFL entry. Do not set a target at the AFL entry which is not configured.

Each accepted received message can be stored into a maximum of 8 different target destinations (single reception message buffer and/or FIFO buffers).

The programming of more than eight target destinations is not allowed. When more destinations are programmed then internal timing, a race condition can occur and received message may not be stored to the message RAM. Correct configuration of the numbers of target destination is the responsibility of the application.

Additional protection mechanism is made for the case when a received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS[2:0], CFDRFCCn.RFPLS[2:0] or CFDCFCCn.CFPLS[2:0]).

If CFDGCFG.CMPOC = 0, the message is completely rejected and is stored in the target destination. When CFDGCFG.CMPOC = 0 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS[2:0], CFDRFCCn.RFPLS[2:0] or CFDCFCCn.CFPLS[2:0]), the corresponding CFDFMSTS.RFXMLT[7:0] or CFDFMSTS.CFXMLT[5:0] bits are not set to 1, respectively.

When CFDGCFG.CMPOC = 1, the received data bytes greater than CFDRMNB.RMPLS is rejected. When CFDGCFG.CMPOC = 1 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS[2:0], CFDRFCCn.RFPLS[2:0] or CFDCFCCn.CFPLS[2:0]), the corresponding CFDFMSTS.RFXMLT[7:0] or CFDFMSTS.CFXMLT[5:0] bits are set to 1, respectively.

Depending on the CFDGCFG.DRE bit, the original received DLC or the DLC value configured at the AFL entry is stored.

Regardless of the CFDGCFG.CMPOC bit setting, CFDGERFL.CMPOF is set to 1 if a payload overflow condition is detected.

The DLC filtering is performed before the payload overflow function. So for one reception frame, only one flag can be set at the same time with CFDGERFL.DEF or CFDGERFL.CMPOF.

35.6.2 Allocation of AFL Entries to each CAN Channel

The number of AFL entries per channel can be configured using the dedicated field in the related Global Acceptance Filter Configuration Registers (see Figure 35.24).

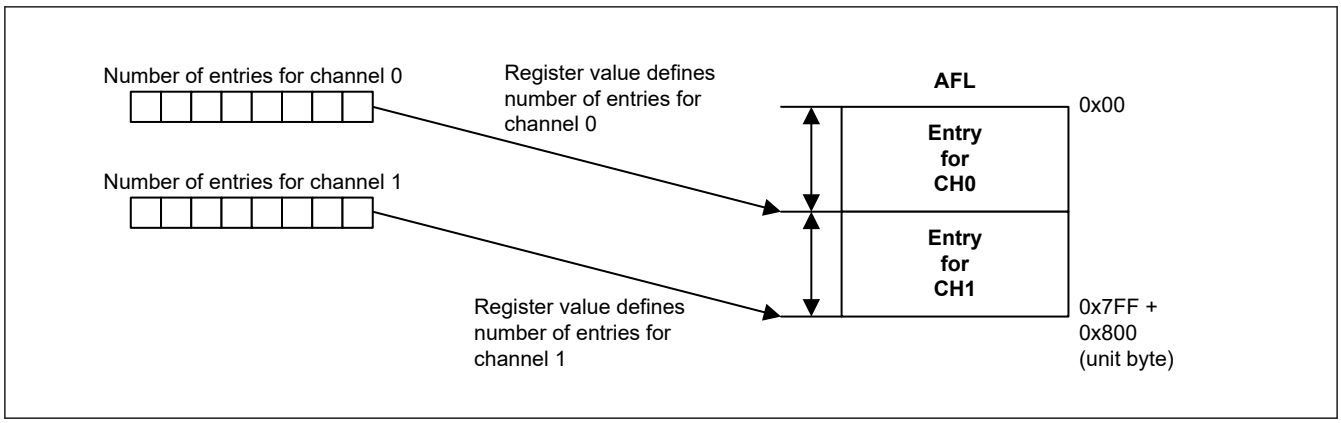


Figure 35.24 Configuration of AFL for each channel

The minimum number of entries for one channel is 0 (no entries defined for the channel) and the maximum number of entries for one channel is 128. The total number of entries for all channels should not exceed the maximum limit of (n + 1) × 128.

All entries are unique for a channel and overlapping or sharing of entries is not supported. Correct configuration of the AFL is the responsibility of the application.

The CAN-FD module does not flag errors related to the configuration of the AFL.

35.6.3 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.

Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier (11 bits for Standard Frame format, 29 bits for Extended Frame format):
  Acceptance filter unit checks the identifier field of the received message against the identifier field of each AFL entry (full 29 bits masking of identifier bits is possible, see information that follows).
- IDE bit:
  Acceptance filter unit checks the IDE bit of the received message against this bit and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, see the information that follows).

- RTR bit:
  Acceptance filter unit only accepts data frames (RTR = 0) or remote frames (RTR = 1) according to the setting of this bit (masking of RTR bit is possible, see the information that follows).
- Loopback Configuration bit:
  This bit can enable or disable the AFL entry depending on the Loopback Configuration or Mirror mode condition.
- Mask for Identifier bits (29 bits):
  Each bit in the identifier mask can mask the corresponding identifier bit in the AFL entry during acceptance filtering, see Figure 35.25.
- Mask for IDE bit:
  If this Mask bit masks the IDE bit of the AFL entry in both Standard Identifier and Extended Identifier format, messages can be accepted by this AFL entry. The identifier of the received message is compared against the Standard Identifier part of the AFL entry for Standard Identifier format messages and against the Extended Identifier part of the AFL entry for Extended Identifier format messages.
- Mask for RTR bit:
  If this Mask bit masks the RTR bit of the AFL entry in both frame formats, data frame and remote frame formats are accepted by this AFL entry.
- Pointer information (16 bits):
  This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added during message storage in the message buffer area and can be used by application as support function. The pointer information can be used for example to support PDU identifier allocation for the received message in AUTOSAR systems.
- Information label (2 bits):
  This 2-bit label is attached to a received message accepted by the related AFL entry. The label is added during message storage in the message buffer area and can be used by application as support function.
- DLC value for automatic DLC filtering:
  If the DLC value of the received message is equal or higher than the configured DLC value, the DLC check is passed. If the DLC value in this AFL entry is configured to 0, DLC filtering is effectively disabled for this entry (all accepted messages pass DLC filtering).

Each AFL entry contains the following information for the handling of received messages:

- Message buffer number of one single reception message buffer as target for received message storage
- Single reception message buffer enable bit to configure the single reception message buffer number to be valid or invalid, as target for received message storage
- FIFO direction pointer - each bit of the FIFO direction pointer configures a dedicated FIFO as possible target for a received message

Note: A message received on channel A can be routed to Common FIFO buffer of another channel. If this Common FIFO buffer is configured in Gateway mode, then the message stored in this Common FIFO Buffer is transmitted on that channel because Common FIFO buffer is associated with channel.

There is no hardware protection against such storage of message. Therefore, the FIFO direction pointer must be configured carefully.

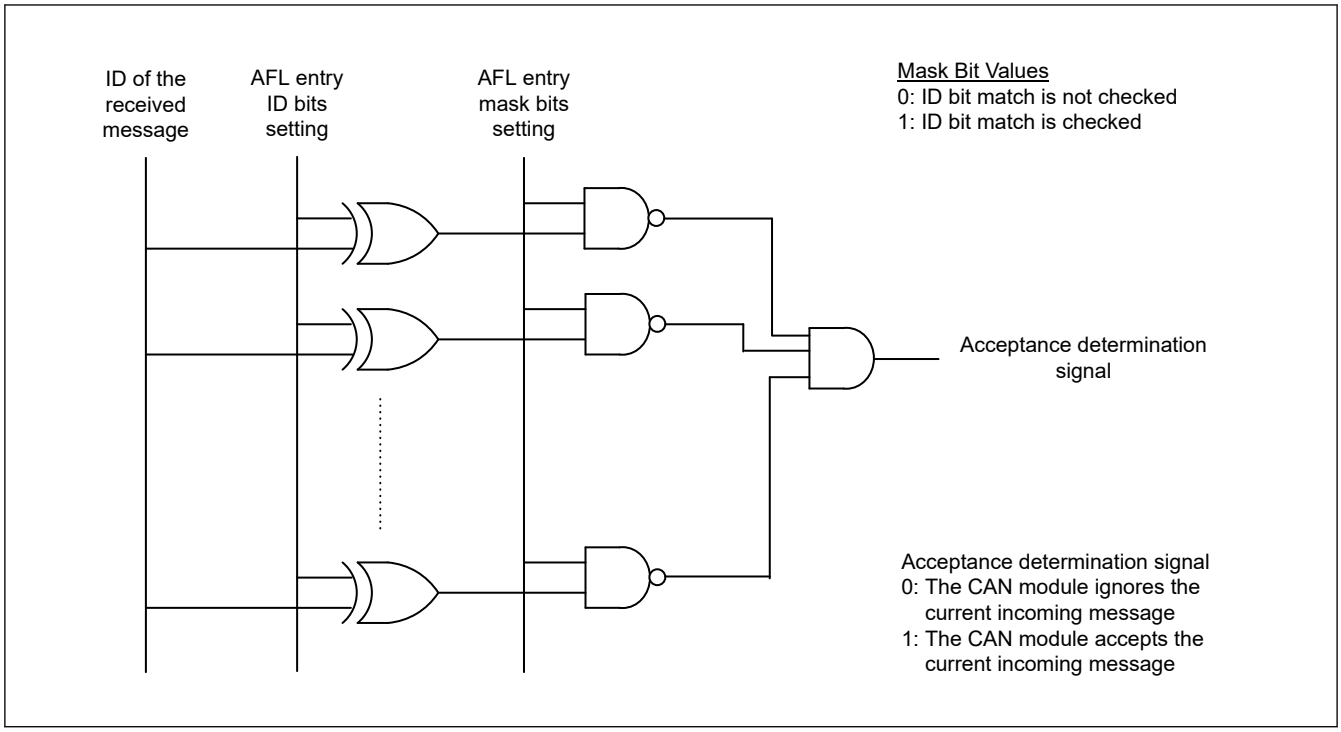


Figure 35.25 Acceptance function

35.6.4 Entering Entries in the AFL

Application software can enter one full entry into the AFL using the following registers:

- Global AFL ID Entry Register: Part 1 of the AFL entry
- Global AFL Mask Entry Register: Part 2 of the AFL entry
- Global AFL Pointer 0 Entry Register: Part 3 of the AFL entry
- Global AFL Pointer 1 Entry Register: Part 4 of the AFL entry

16 sets of these registers form a group of AFL entries. Each group can be accessed through a page mechanism. For the CAN-FD module, 2-channel version, 16 of these pages exist to allow access to the whole AFL range. The AFL should only be configured in CH_RESET or CH_HALT mode. Pages are linked to the AFL entries in the following way:

Table 35.23 Mapping between pages and AFL entries (1 of 2)

Page 0	Entry 0 – 15
Page 1	Entry 16 – 31
Page 2	Entry 32 – 47
Page 3	Entry 48 – 63
Page 4	Entry 64 – 79
Page 5	Entry 78 – 95
Page 6	Entry 96 – 111
Page 7	Entry 112 – 127
Page 8	Entry 128 – 143
Page 9	Entry 144 – 159
Page 10	Entry 160 – 175
Page 11	Entry 176 – 191
Page 12	Entry 192 – 207
Page 13	Entry 208 – 223

Table 35.23 Mapping between pages and AFL entries (2 of 2)

Page 14	Entry 224 – 239
Page 15	Entry 240 – 255

The selection of the AFL access page is done using the Global Acceptance Filter List Entry Control Register (CFDGAFLECTR) (see Figure 35.26). This register has the following fields:

- 4 bits to select the AFL page number
- 1 bit to enable or disable the AFL data access to prevent unwanted write access to the AFL

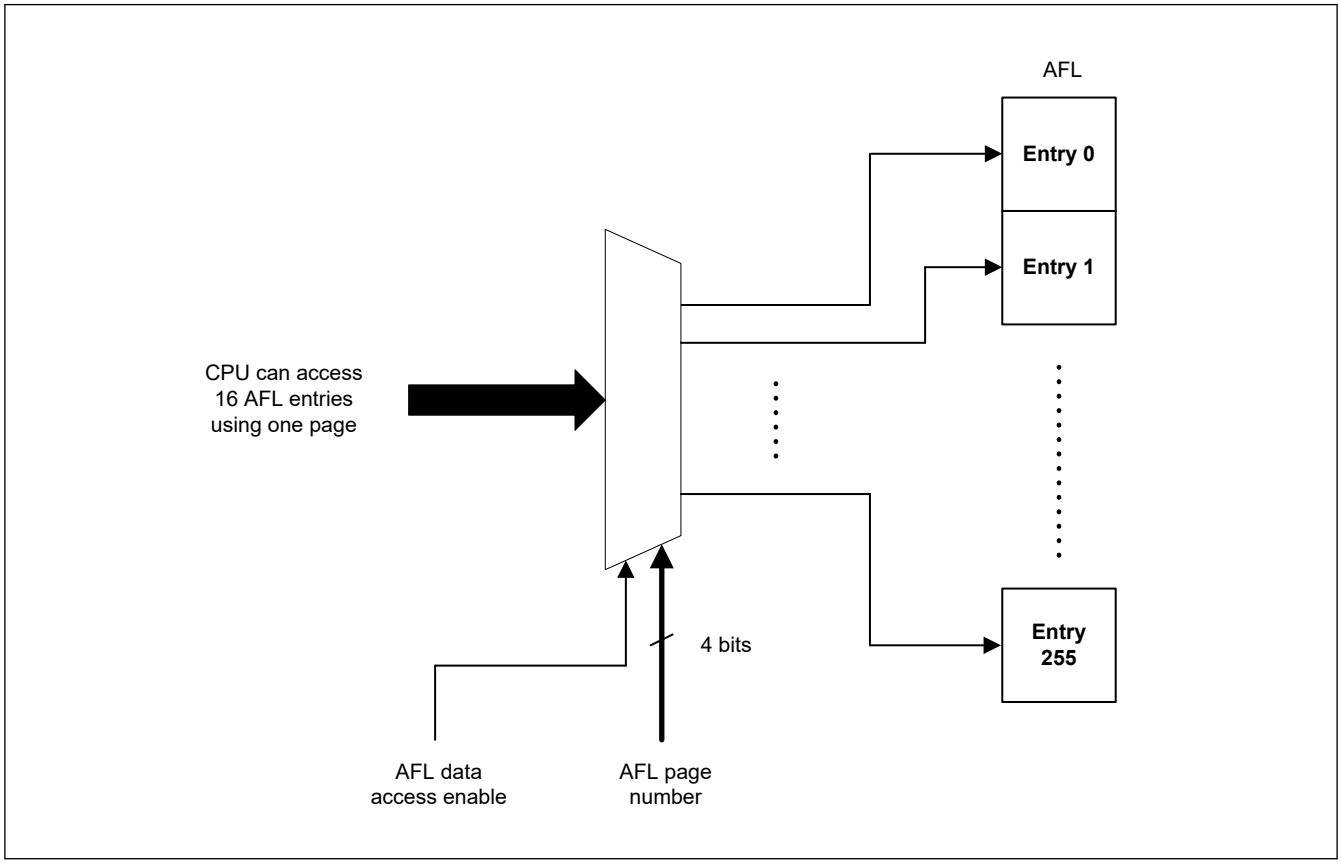


Figure 35.26 AFL page access

Application software should not write numbers higher than 0xF for the AFL page number.

Follow the configuration shown in Figure 35.27 to program the AFL.

After entering all entries in Configuration mode, locking of the AFL access should be performed to protect unwanted write access to the AFL.

Write protection is active during all Global modes (GL_RESET, GL_HALT, and GL_OPERATION) if the lock bit is set.

Read access to AFL is still possible during all Global modes even when AFL data access is disabled (consistency check of AFL contents is possible during run time).

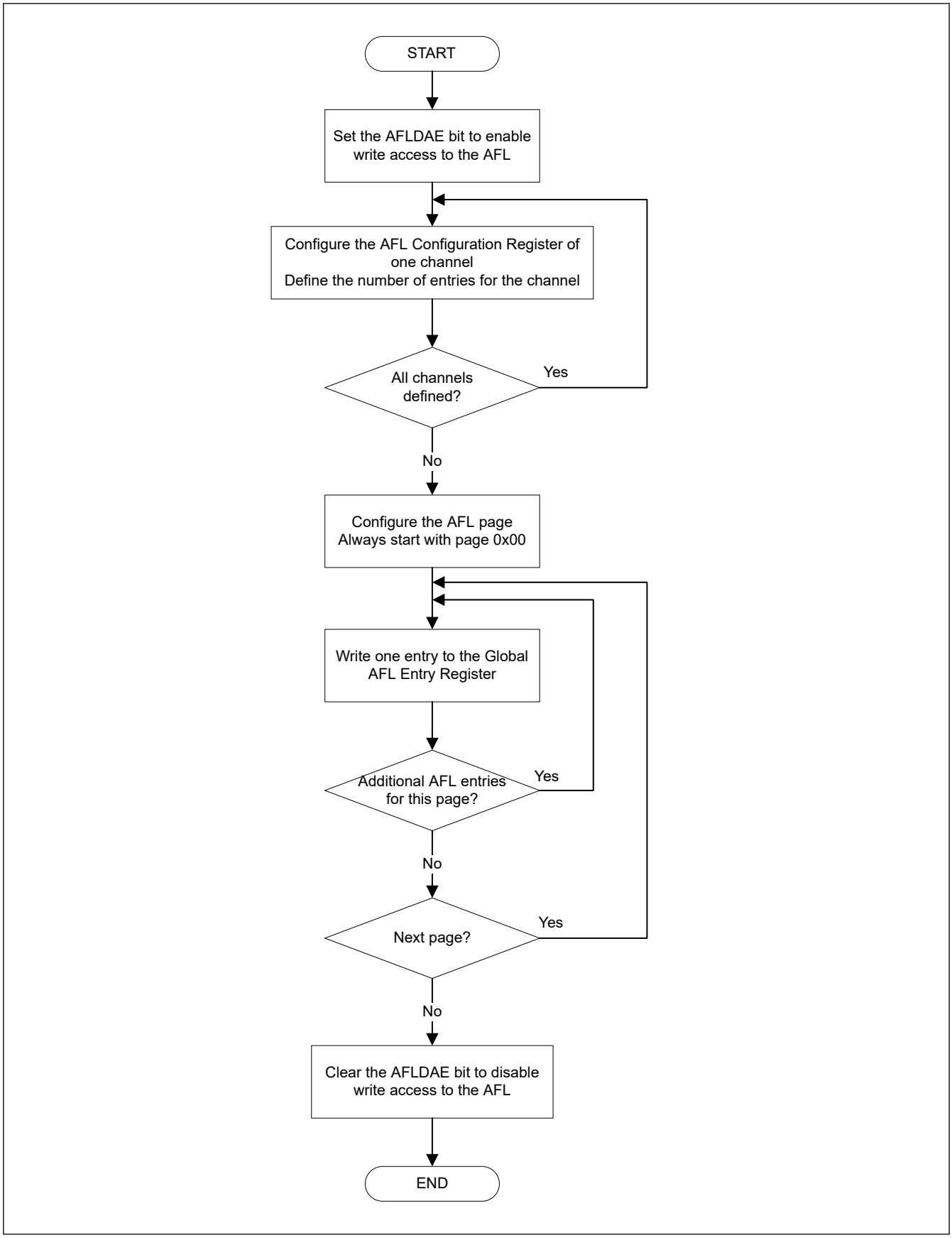


Figure 35.27 AFL configuration flow

35.6.5 Loopback Modes

If the Loopback Configuration bit is set, the AFL entry is only valid in loopback test mode (Self test mode 0 or Self test mode 1) or in mirror mode when receiving messages that were transmitted by the respective CAN channel itself.

The AFL entry is not valid for received messages in loopback mode transmitted by other CAN nodes on the bus. The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID respectively.

If the Loopback Configuration bit is 0, the AFL entry is only valid for:

- Received messages transmitted by other CAN nodes on the bus in normal (non-loopback mode) and mirror modes
- Received messages transmitted by other CAN nodes or the CAN channel itself in loopback test mode.

The mirror mode can be enabled with the CFDGCFG.MME bit in the Global Configuration Register. If CFDGCFG.MME bit is set, then a successfully transmitted message can be stored back in an RX message buffer or FIFO buffer if a matching entry is configured in the AFL for that channel.

The Loopback Configuration bit in the matching AFL entry must be set to store this frame.

If mirror mode and loopback test mode are configured at the same time, the loopback test mode behavior applies.

Table 35.24 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 35.24 Behavior of acceptance filter based on the loopback configuration setting in AFL entry

Mirror mode enable (MME Configuration bit)	Loopback in Test mode (Self test mode 0 or Self test mode 1)	Channel mode	Loopback Configuration bit in AFL entry	AFL entry
0	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Invalid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid
1	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Valid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

35.6.6 IDE Masking

When the GAFLIDEM bit is 0 in an AFL entry, the IDE bit configured in the AFL entry is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is based on the received IDE bit.

Consider the following example:

- The ID and Mask fields of an AFL entry x is configured as follows:
  - CFDGAFLID [x] = 0xC0553A20 → IDE = 1, RTR = 1, LLB = 0, ID[10:0] = 0x220 / ID[28:0] = 0x0553A20
  - CFDGAFLMn = 0x0000FFFF → IDEM = 0, RTRM = 0, IDM[10:0] = 0x7FF / IDM[28:0] = 0x0000FFFF
- The comparison result for the four different received IDs with AFL entry x is described as follows:

– If a frame with IDE = 0 and ID = 0x220 is received, this is considered as a match
– If a frame with IDE = 0 and ID = 0x320 is received, this is not a match
– If a frame with IDE = 1 and ID = 0x1FFF3A20 is received, this is considered as a match
– If a frame with IDE = 1 and ID = 0x08803220 is received, this is not a match.

35.7 FIFO Buffers and Normal Message Buffer Configuration

This section describes the process for configuring the number of RX message buffers, the FIFO buffers, and the flat TX message buffers in the CAN-FD module. The message buffers are mapped as shown in Figure 35.28.

The RX message buffers can be accessed with the RX Message Buffer Registers.

The RX FIFO buffers and the common FIFO buffers configured in RX mode, TX mode, or GW mode can only be accessed with the FIFO Access Registers.

If the common FIFO is configured in TX mode, you can only write data into the FIFO buffer using the FIFO Access registers.

If the common FIFO is configured in GW mode or RX mode, you can only read data from the FIFO Access Registers.

The TX message buffers can be accessed with the TX Message Buffer Registers.

If unused message buffer locations are read, the message buffer locations are read as unknown values.

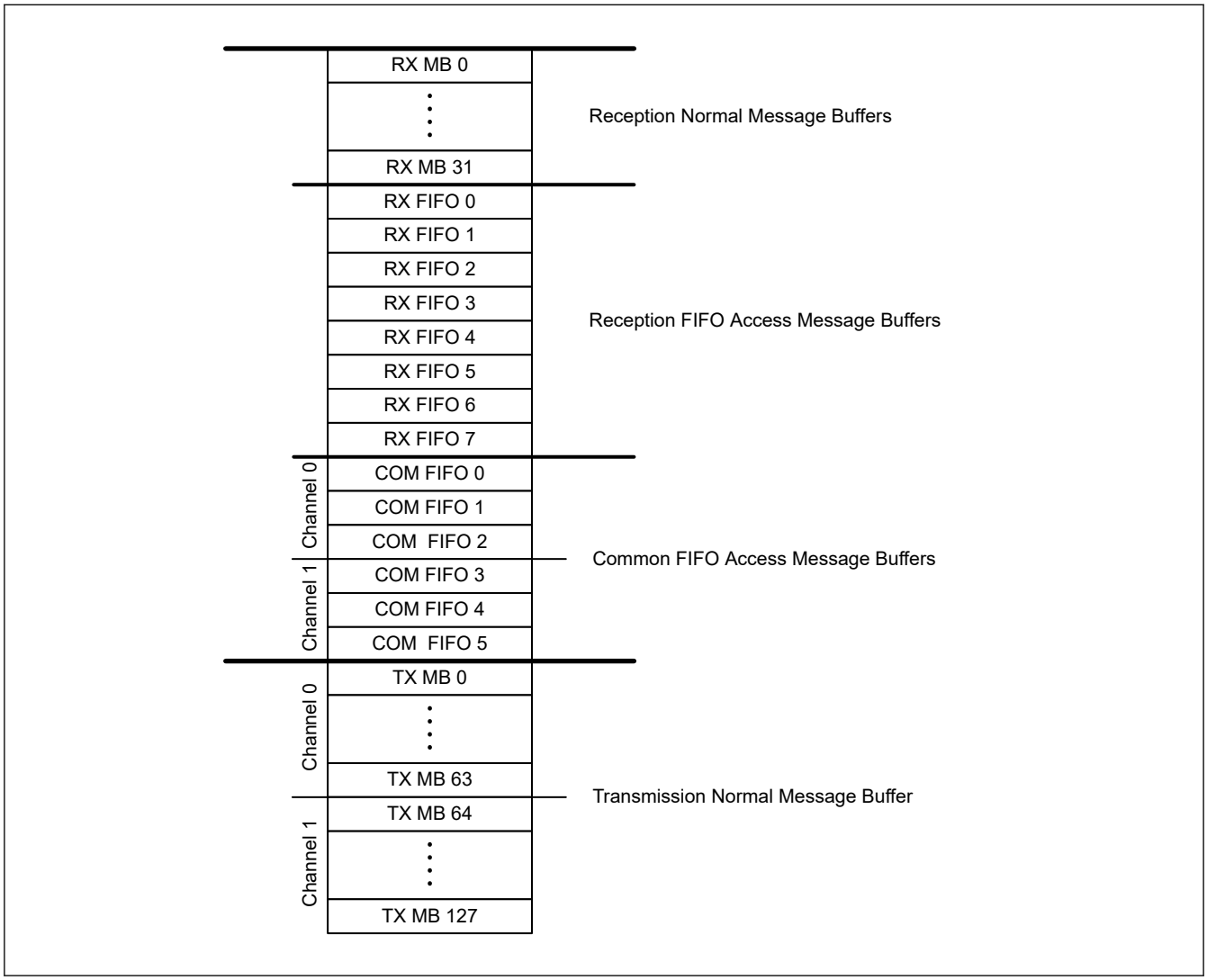


Figure 35.28 Message buffer configuration

35.7.1 Normal RX Message Buffers

In CAN-FD module, the frames received by various channels can be stored in normal RX message buffers based on the configuration of the AFL entries.

Additionally, the number of normal RX message buffers required in the system can be chosen up to a fixed maximum limit.

35.7.1.1 Normal RX Message Buffer Configuration

In CAN-FD module, the number of normal RX message buffers can be configured by writing to the RX Message Buffer Number Register.

The limiting values for the configuration of number of message buffers are:

- Minimum value = 0x00 (no normal RX MB)
- Maximum value = (16 × number of CAN channels) = 16 × 2 = 32 = 0x20 (32 flat RX MBs for 2 channels)

Do not use values outside these limits.

The AFL entries for routing the received messages to normal RX message buffers must be configured to match the requirements of the system.

The AFL entries must also be configured properly, and an AFL entry for normal RX message buffers should not exceed the number of message buffers configured in the RX Message Buffer Number Register.

Note: There is no internal check procedure provided in CAN-FD module against incorrect configuration of the AFL.

The data field size of the RX message buffer can be configured with the CFDRMNB.RMPLS[2:0] bits. The default size is 8 bytes and the maximum data payload size is 64 bytes.

When the receiving frame exceeds the data field size, then the acceptance depends on the configuration of CFDGCFG.CMPOC (message rejecting or data payload cut).

35.7.2 FIFO Buffer

The CAN-FD module provides a fixed number of FIFO buffers to support storage of frames for reception, transmission and gateway functions for various CAN channels.

The number of reception-only FIFO buffers is fixed to 8. However, 3 common FIFO buffers per channel can be configured to store messages for transmission, reception, or gateway function.

These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

- Size
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the TX FIFO or GW FIFO

When the receiving frame exceeds the data field size, the acceptance depends on the configuration of the CFDGCFG.CMPOC bit (message rejecting or data payload cut).

35.7.2.1 FIFO Buffers Configuration

In CAN-FD module, the FIFO buffers can be configured to match the system requirements.

The total number of FIFO buffers = 8 RX FIFO buffers + 6 common FIFO buffers = 14 FIFO buffers for 2 channels and message overwrite mechanism.

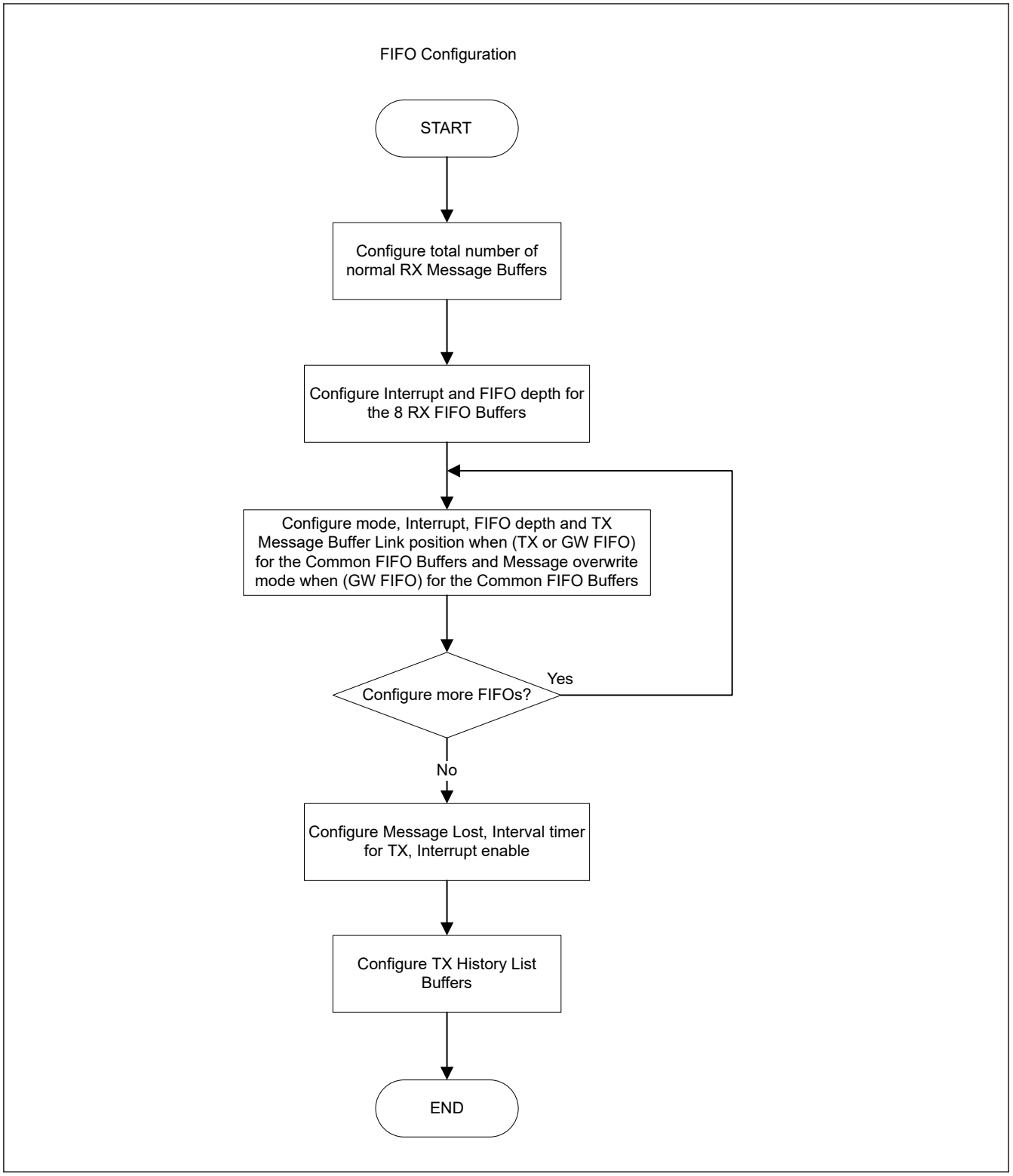


Figure 35.29 FIFO buffer configuration flow in CAN-FD module

As shown in Figure 35.29, the various FIFO buffers can be configured by writing to the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

For the 8 RX FIFO buffers, the following parameters can be configured:

- Interrupts
- FIFO depth
- FIFO payload data size

For the common FIFO buffers, the following parameters can be configured:

- Mode
- Interrupts FIFO depth
- FIFO payload data size
- FIFO TX link position

(1) FIFO Mode Configuration of Common FIFO Buffers

The mode of the common FIFO buffers can be configured by writing to the CFDCFCCn.CFM[1:0] bits in the Common FIFO Configuration/Control Registers. The possible modes of configuration for Common FIFO buffers are:

- 00b: RX mode (default mode after hardware reset)
- 01b: TX mode
- 10b: GW mode
- 11b: Reserved (do not write this value to the register bits)

Messages can only be read from the RX FIFO buffers and the Common FIFO buffers configured in RX mode. Messages are stored by the CAN module in these FIFO buffers based on the AFL entries.

Messages can be read and written into the Common FIFO buffers configured in TX mode. These messages are transmitted on the appropriate CAN channel.

Messages can only be read from the Common FIFO buffers configured in GW mode. However, the CPU read access has no impact on the read or write pointers. The pointers can only be incremented when a new message is stored in the FIFO buffer and decremented when a message is transmitted on the corresponding CAN channel by the CAN-FD module.

After a hardware reset, all the Common FIFO buffers are configured in RX mode by default. Only enable the FIFO buffers after configuring the Common FIFO buffers in the required modes.

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting the CFDCFCCEn.CFMOWM bit.

- When CFDCFCCEn.CFMOWM = 0:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded, and CFDCFSTSn.CFMLT bit is set to 1.
- When CFDCFCCEn.CFMOWM = 1:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message. The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message. The CFDCFSTSn.CFMOW bit is then set to 1, which notifies that the oldest message has been overwritten with the received message.
  In addition, when a CAN bus error or arbitration-lost for the transmitting message occurs in the transmit/receive FIFO buffer full, the transmitting message is lost and retransmission for the message is not performed. The read point is then moves to the next message automatically. Do not write to this bit when the CFDCFCCn.CFE bit is 1.

(2) FIFO TX Message Buffer Link Configuration

When the common FIFO is configured as TX or GW FIFO, the FIFO buffer must be linked to a normal TX message buffer to participate in the transmission scan of a CAN channel.

The link to a normal TX message buffer must be unique, for example the same TX message buffer cannot be shared between 2 or more common FIFO buffers.

Do not write data into a TX message buffer that is linked to a Common FIFO buffer. Also, the TX message buffer linked to a Common FIFO buffer should not be a part of the TX Queue.

The TX message buffer link of each Common FIFO buffer can be configured by writing to the CFDCFCCn.CFTML[4:0] bits in the Common FIFO Configuration/Control Registers. Available options for TX message buffer link configuration are:

- 0x00: TX Message Buffer 32
- 0x01: TX Message Buffer 33
  ⋮

- 0x1F: TX Message Buffer 63

(3) FIFO Depth Configuration

The depth of each FIFO buffer can be configured by writing to the CFDRFCCn.RFDC[2:0] bits and CFDCFCCn.CFDC[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/ Control Registers. The 8 available options for depth configuration are:

- 000b: 0 Messages (FIFO buffer cannot be enabled)
- 001b: 4 Messages
- 010b: 8 Messages
- 011b: 16 Messages
- 100b: 32 Messages
- 101b: 48 Messages
- 110b: 64 Messages
- 111b: 128 Messages

The RAM allocation for RX message buffers along with FIFO buffers is limited to (n + 1) × 256 messages. Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CAN-FD module logic does not check the validity of the configuration.

Note:
- If the FIFO depth of a common FIFO is 4 messages or more (CFDCFCCn.CFDC[2:0] > 000b), then the Common FIFO TX message buffer link is valid when the FIFO is disabled or enabled.
- If FIFO depth is 0 messages, then the Common FIFO TX message buffer link is not valid when the FIFO is disabled or enabled.

(4) FIFO Payload Size Configuration

The data size of each FIFO buffer can be configured by writing to the CFDRFCCn.RFPLS[2:0] bits and CFDCFCCn.CFPLS[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/ Control Registers. The eight available options for depth configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes

(5) FIFO Interrupt Configuration

The interrupt generation conditions for the FIFO buffers can be configured by writing to the CFDRFCCn.RFIM bit and the CFDCFCCn.CFIM bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The two available options are:

- 0:
  - RX FIFO mode: Interrupt generated when the Common FIFO counter reaches CFDRFCCn.RFIGCV/ CFDCFCCn.CFIGCV value
  - TX FIFO mode: Interrupt generated when the Common FIFO transmits the last message successfully
  - GW FIFO mode
    Frame RX: Interrupt generated when message counter increments and reaches the interrupt threshold value
    Frame TX: Interrupt generated when the last message is transmitted successfully from FIFO
- 1:

– RX FIFO mode: Interrupt generated at the end of storage of every received message
– TX FIFO mode: Interrupt generated for every successfully transmitted message
– GW FIFO mode
Frame RX: Interrupt generated when message is stored in the FIFO
Frame TX: Interrupt generated when message is successfully transmitted from the FIFO

If the Interrupt Mode bit is 0 for a RX FIFO, then interrupt is generated based on the configuration of the CFDRFCCn.RFIGCV[2:0] bits.

Similarly, if the Interrupt Mode bit is 0 for a Common FIFO configured in RX mode, then interrupt is generated based on the configuration of CFDCFCCn.CFIGCV[2:0] bits.

The eight available options for configuring the FIFO counter value for generation of an interrupt are:

- 000b: Interrupt generated when FIFO is 1/8th Full
- 001b: Interrupt generated when FIFO is 1/4th Full
- 010b: Interrupt generated when FIFO is 3/8th Full
- 011b: Interrupt generated when FIFO is 1/2 Full
- 100b: Interrupt generated when FIFO is 5/8th Full
- 101b: Interrupt generated when FIFO is 3/4th Full
- 110b: Interrupt generated when FIFO is 7/8th Full
- 111b: Interrupt generated when FIFO is Full

In this case, an interrupt is generated when the message count matches the configured value.

However, there are some limitations on the configuration of the CFDRFCCn.RFIGCV[2:0] and CFDCFCCn.CFIGCV[2:0] bits depending on the FDC[2:0] bits (FIFO Depth Configuration), see Table 35.25.

Table 35.25 FIFO interrupt generation counter and FIFO depth configuration

RFDC[2:0] (FDC[2:0])	RFIGCV[2:0] (CFGCV[2:0])							
	111b	110b	101b	100b	011b	010b	001b	000b
000b	Don't care (FIFO cannot be enabled)							
001b	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed
010b	Allowed							
011b	Allowed							
100b	Allowed							
101b	Allowed							
110b	Allowed							
111b	Allowed							

Common FIFO buffer can set an interrupt output at the completion time of transmitting one frame, or the completion of reception. In addition, Common FIFO and RX FIFO can set an interrupt output, when stored to the setup number (CFDC/RFDC) of FIFO stages.

35.7.2.2 FIFO Buffers Control

The FIFO interrupt must be enabled by setting any one of the following bits in the RX FIFO Configuration/Control Registers:

- CFDRFCCn.RFIE
- CFDRFCCn.RFFIE

In addition, the FIFO interrupt must be enabled by setting any one of the following bits in the Common FIFO Configuration/Control Registers:

- CFDCFCCn.CFRXIE

- CFDCFCCn.CFTXIE
- CFDCFCCEn.CFFIE
- CFDCFCCEn.CFOFRXIE
- CFDCFCCEn.CFOFTXIE

After configuration is complete, each FIFO can be enabled by setting the CFDRFCCn.RFE and CFDCFCCn.CFE bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers to allow transmission and reception of messages.

When CFDCFCCEn.CFBME = 1, it becomes FIFO buffering mode, send data is stored in the Common FIFO, and transmission is stopped. Transmission starts when CFDCFCCEn.CFBME = 0.

Do not write 1 from 0 for this bit when the CFDCFCCn.CFE bit is 1.

35.8 Interrupt and DMA

35.8.1 Interrupts

The CAN-FD module generates several interrupts. The interrupt output, which is connected to the Interrupt Controller Unit (ICU), can be controlled by the corresponding interrupt enable bit.

The status flag is set independent from this enable bit.

The channel transmission interrupt has an additional Status Flag register. These status bits are only set when the corresponding interrupt enables are set.

The Status Flag register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.

The interrupts in the CAN-FD module can be classified into 2 groups, global interrupts and channel interrupts:

- Global interrupts:
  The CAN-FD module can generate 2 global interrupts:
  1. One Global interrupt for successful reception into the 8 RX FIFO buffers
  2. One Global error interrupt
- Channel interrupts:
  Each channel of the CAN-FD module can generate 3 channel interrupts:
  1. Channel transmission
     Transmission completion from channel
     Transmission abort from channel
     Transmission from TX Queue for a channel
     Channel THL Interrupt
     Successful transmission from a Common FIFO in TX or GW mode for a channel
  2. Channel error interrupt
  3. Successful reception in a Common FIFO in RX or GW mode for a channel or successful routing in a TXQ

The interrupts are cleared when the corresponding flag bits are cleared or interrupt enable bits are cleared.

Table 35.26 provides an overview of interrupt.

Table 35.26 Overview of interrupt source

Parameter	Interrupt	Interrupt source	Interrupt clearing
Global Interrupts	Successful reception into at least one RX FIFO	Interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the interrupt flag of corresponding RX FIFO buffer for which interrupt is enabled
	FIFO full into at least one RX FIFO	FIFO Full Interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the FIFO Full Interrupt flags of corresponding RX FIFO buffer for which interrupt is enabled
	Global error	Any of the following: • DLC Error flag • Message Lost Status bit • Message Overwrite Status bit • TXQ Message Lost Status bit • TXQ Message Overwrite Status bit • TX History Entry Lost Status bit • CAN-FD Message Payload overflow flag	Clear all of: • DLC Error flag • Message Lost flags in all of the FIFO Status Registers • Message Overwrite flags in all of the Common FIFO Status Registers • Message Lost flags in all of the TXQ Status Registers • Message Overwrite Flags in all of the TXQ Status Registers • TX History List Entry Lost flag • CAN-FD Message Payload Overflow flag
Channel Transmission Interrupts	Channel n successful transmission	Any channel related TX MB Successful flag when interrupt is enabled Separate interrupts are provided for common FIFO buffers and TX Queue*1	Clear all channel related TX MB Result Status bits for which the interrupt is enabled
	Channel n abort	Any channel related TX MB Abort flag when interrupt is enabled Separate interrupts are provided for common FIFO buffers and TX Queue*1	Clear all channel related TX MB Result Status bits for which the interrupt is enabled globally
	Channel n transmission from TX Queue	Related channel TX Queue Interrupt flag	Clear related channel TX Queue Interrupt flag
	Channel n THL Interrupt	Channel n THL Interrupt Status flag	Clear the relevant THL Interrupt Status flag
	Channel n COM FIFO TX Interrupt	Interrupt flag for Common FIFOs in TX or GW mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in TX or GW mode belonging to the related channel
	Channel n COM FIFO One Frame TX Interrupt	One Frame Transmission Interrupt flag for Common FIFOs belonging to the related channel	Clear the One Frame Transmission Interrupt flags of Common FIFOs belonging to the related channel
	Channel n TXQ One Frame TX Interrupt	One Frame Transmission Interrupt flag for TXQs belonging to the related channel	Clear the One Frame Transmission Interrupt flags of TXQs belonging to the related channel
Channel COM RX FIFO Interrupt	Channel n COM FIFO RX Interrupt	Interrupt flag for Common FIFOs in RX or GW mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in RX or GW mode belonging to the related channel
	Channel n COM FIFO One Frame RX Interrupt	One Frame Reception Interrupt flag for Common FIFOs belonging to the related channel	Clear the One Frame Reception Interrupt flags of Common FIFOs belonging to the related channel
	Channel n COM FIFO Full Interrupt	FIFO Full Interrupt flag for Common FIFOs in RX or GW mode belonging to the related channel	Clear the FIFO Full Interrupt flags of Common FIFOs in RX or GW mode belonging to the related channel
	Channel n TXQ One Frame Routing Interrupt	One Frame Routing Interrupt flag for TXQs in GW mode belonging to the related channel	Clear the One Frame Routing Interrupt flags of TXQs in GW mode belonging to the related channel
	Channel n TXQ Full Interrupt	TXQ Full Interrupt flag for TXQs in GW mode belonging to the related channel	Clear the FIFO Full Interrupt flags of TXQs in GW mode belonging to the related channel
Channel Error Interrupt	Channel n Error	Any channel related error flag in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register	Clear all channel related error flags in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register

Note 1. These interrupts are only set for TX Message Buffers that do not belong to an enabled TX Queue and are not pointing to a common FIFO.

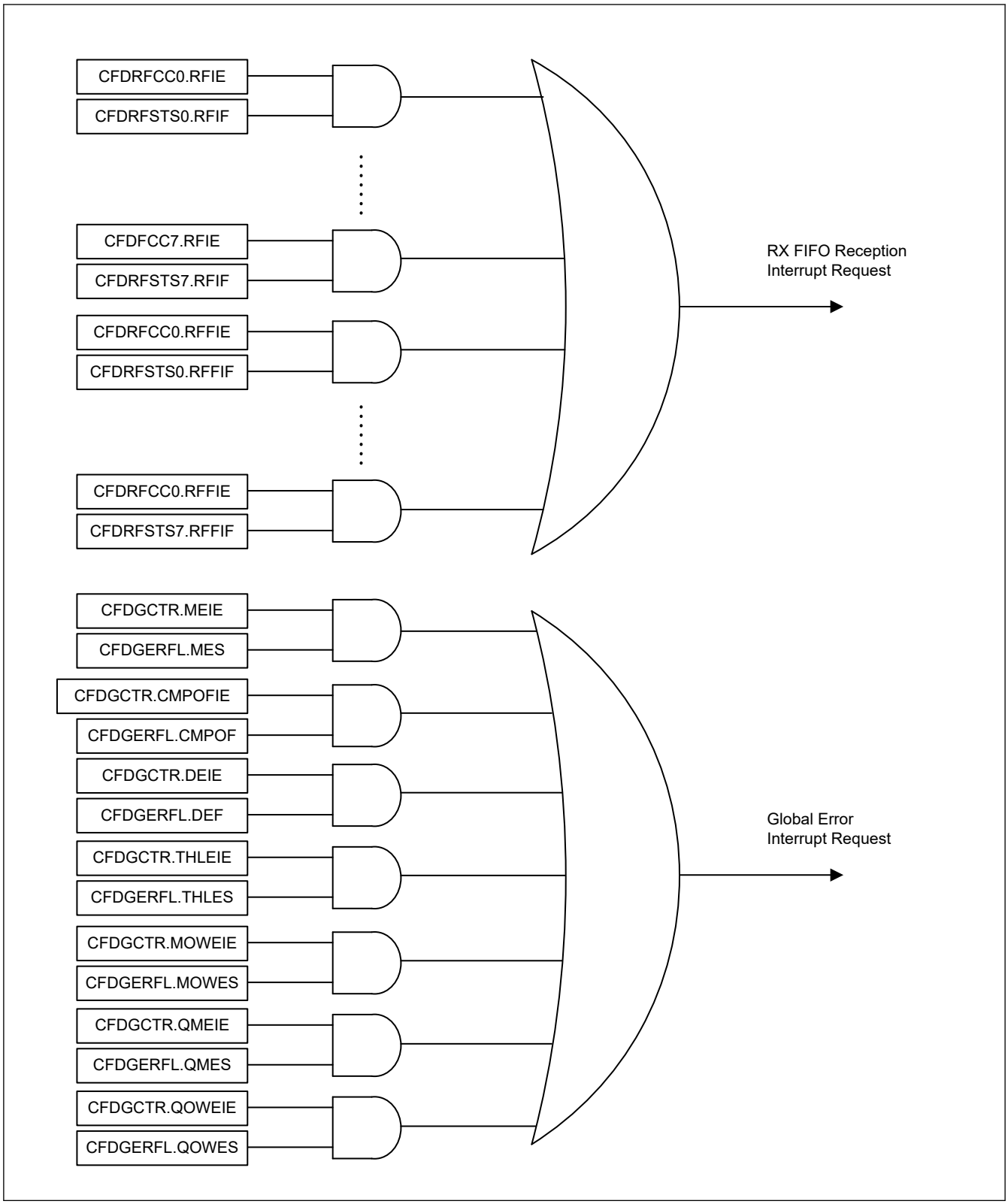


Figure 35.30 Block diagram of Global interrupt

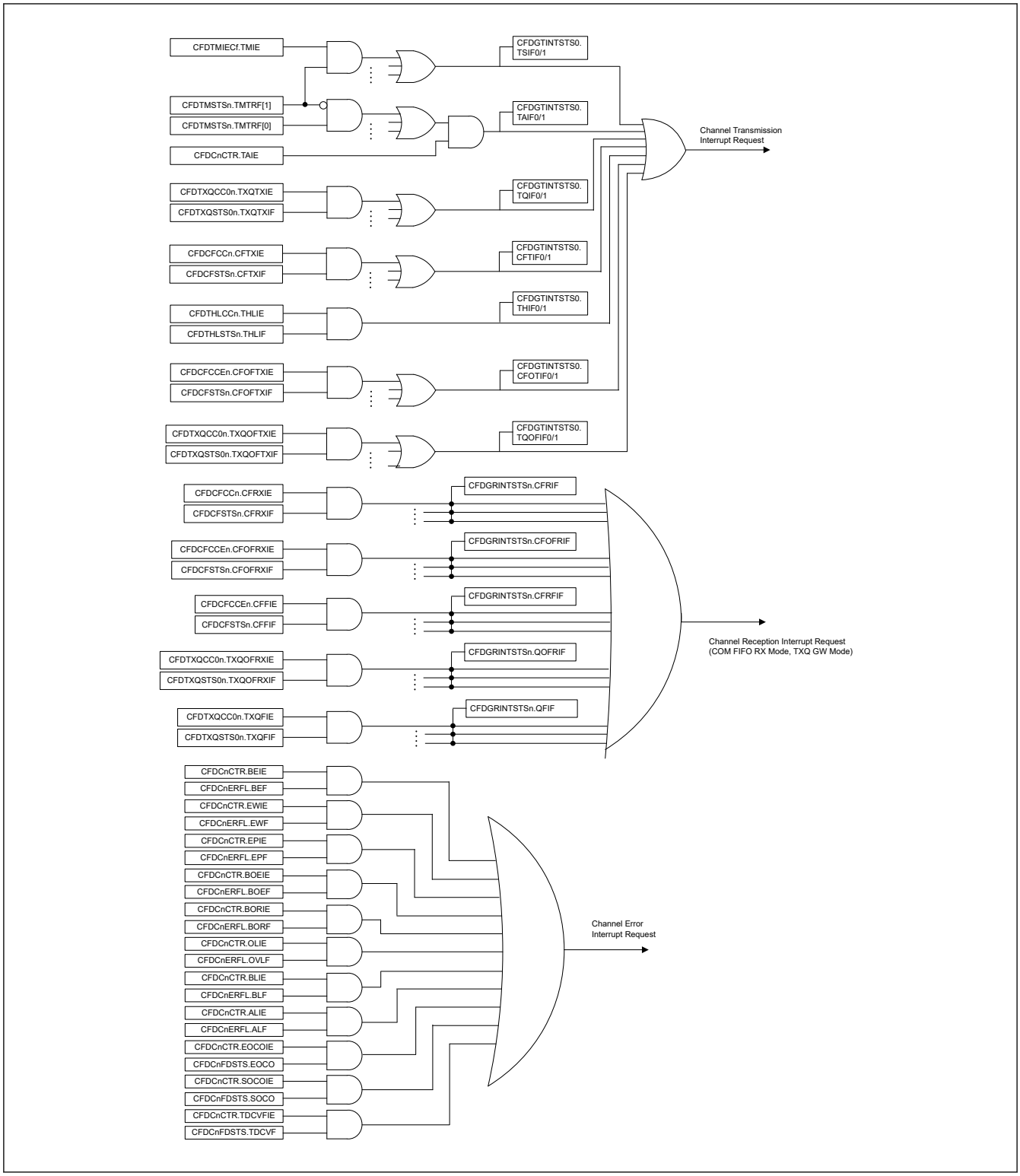


Figure 35.31 Block diagram of Channel interrupt

35.8.2 DMA Transfer

The CAN-FD module has some message buffer which can be associated with a DMA channel:

- Reception DMA
  - 8 RX FIFO Message Buffers
  - 2 Common FIFO Message Buffers

- Transmission DMA
  - 16 TXQ Message Buffers (TXQ0, TXQ3)
  - 2 Common FIFO Message Buffer

Figure 35.32 shows the potential DMA channels.

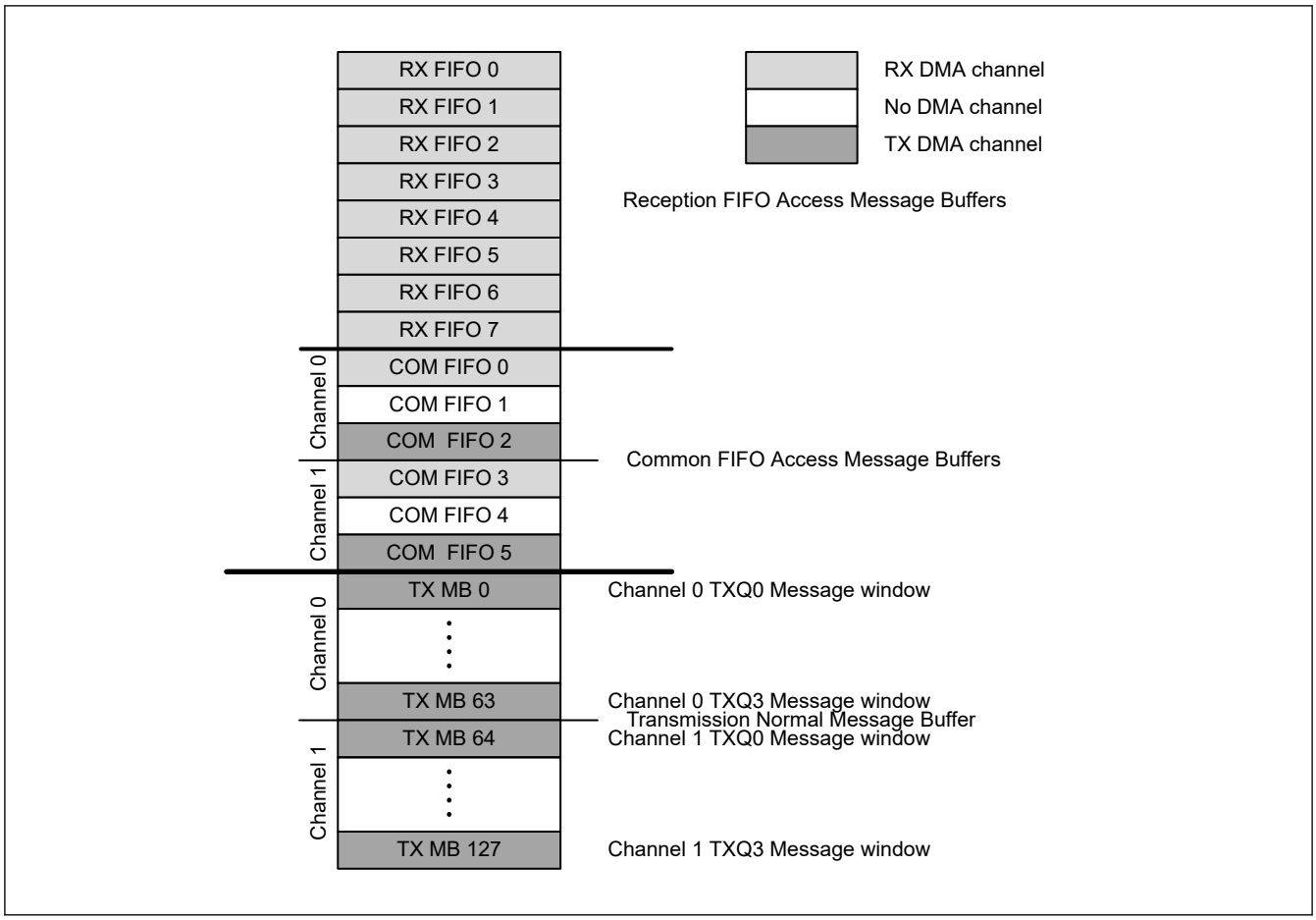


Figure 35.32 Message buffer connectable to DMA channel

A DMA channel transfer request is generated for each FIFO entry to the DMAC when the related CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE is set to 1 and the belonging FIFO is not empty.

Reception FIFO interrupt should be disabled for this particular FIFO (CFDRFCCn.RFIE or CFDCFCCn.CFRXIE).

Use the regular start address for the DMA access window address and add 0x8000 to the regular start address for the debugger access window. See Table 35.27.

Table 35.27 DMA channel access window address (1 of 2)

b = Message buffer component.index	MBCP	Register	p	Start address n = 0, 1
0 to number_of_RFMBCPs - 1	RFMBCPb	RFID	x	0x6000 + b × 0x0080
		RFPTR	x	0x6004 + b × 0x0080
		RFFDSTS	x	0x6008 + b × 0x0080
		RFDFp	0 to 15	0x600C + p × 0x0004 + b × 0x0080

Table 35.27 DMA channel access window address (2 of 2)

b = Message buffer component.index	MBCP	Register	p	Start address n = 0, 1
0 to number_of_CFMBCPs_per_channel - 1	CFMBCPb	CFID	x	0x6400 + b × 0x0080 + n × 0x0180
		CFPTR	x	0x6404 + b × 0x0080 + n × 0x0180
		CFFDCSTS	x	0x6408 + b × 0x0080 + n × 0x0180
		CFDFp	0 to 15	0x640C + p × 0x0004 + b × 0x0080 + n × 0x0180

DMA FIFO pointer decrement is done automatically by reading the last configured data payload byte (CFDRFCCn.RFPLS or CFDCFCCn.CFPLS).

Note: The DMA must read the exact length of the configured data payload size (CFDRFCCn.RFPLS or CFDCFCCn.CFPLS).

The software debugger must access outside of the regular SFR address range from 0xE000 to 0xEFFF.

Do not write to the FIFO and TXQ control registers when DMA is enabled.

The DMA enable of the particular DMA FIFO (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE) can be set at any time, the procedure in this section is a configuration flow for an initial setup.

When CFDCDTTCT.TQ0DMAE or CFDCDTTCT.TQ3DMAE or CFDCDTTCT.CFDMAE is set, the messages of the corresponding TXQ or Common FIFO can be handled by DMA controller.

Use the following procedure when the TXQ or the Common FIFO can be handled by DMA controller.

1. CPU checks the TXQ or the Common FIFO is not full.
2. When transmit data can be used, CPU sets this data to Common FIFO or TXQ.
   When using Common FIFO, transmit data is write in CFDCFID, CFDCFPTR, CFDCFFDCSTS, and CFDTMBCPb[i] registers.
   When using TXQ, transmit data is write in CFDTMID, CFDTMPTR, CFDTMFDCTR, and CFDTMDFp registers.
3. For Common FIFO, the common FIFO pointer is incremented automatically when DMA controller writes the last data payload byte configured by CFDCFCCn.CFPLS.
   For TXQ, if the data of 64 data payload is written, a TXQ pointer increases automatically. When payload data is less than 64 bytes, dummy data must be written in and 64 data payload size must be done.

   Note: Only 32-bit write-access can be possible on the DMA message handling.

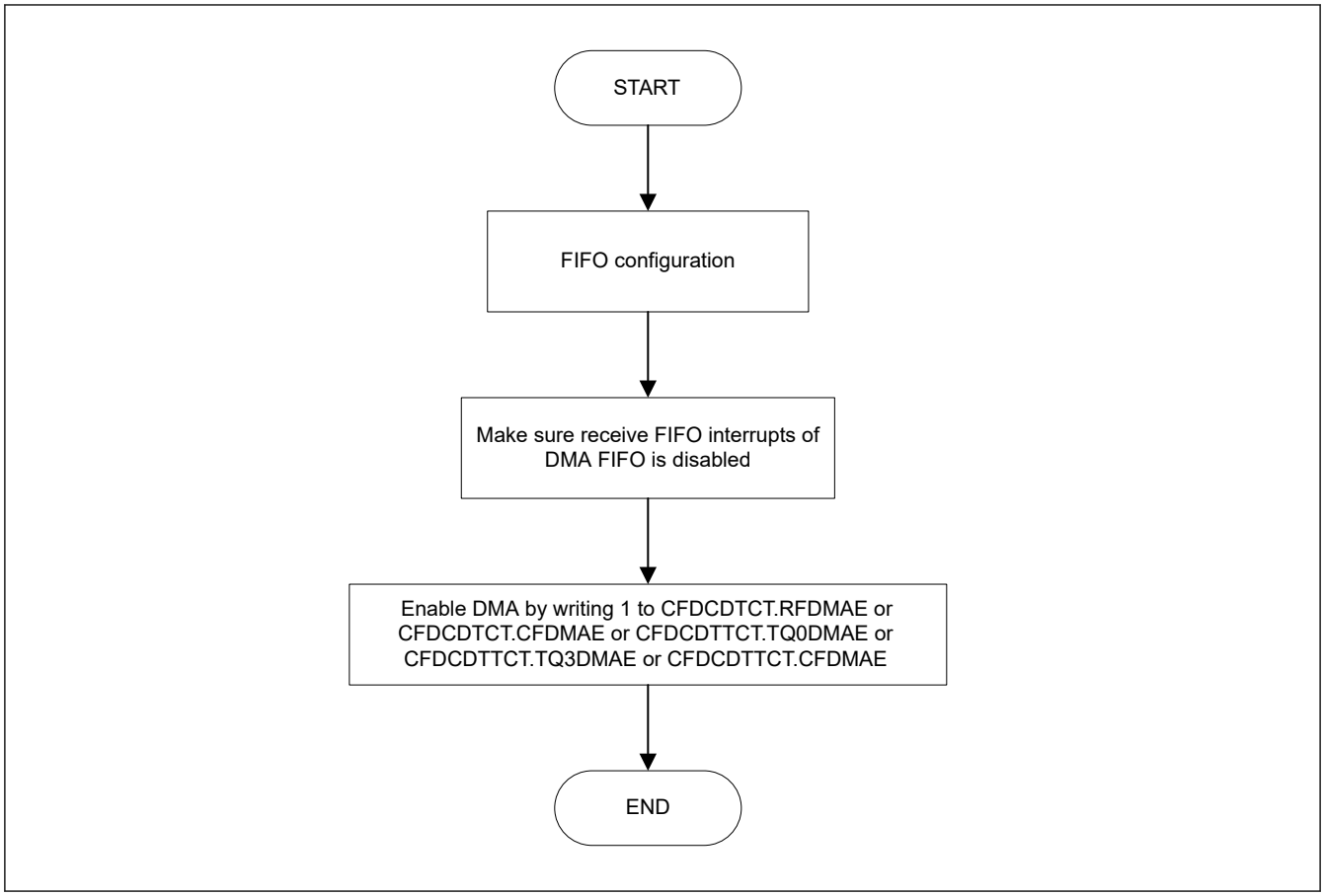


Figure 35.33 DMA enable flow

To disable a DMA transfer requested, disable the particular DMA enable bit (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE). If the disable is made during an ongoing transfer, then this must be completed first before further action can be taken. The transfer status can be identified by the CFDCDTSTS.RFDMASTS or CFDCDTSTS.CFDMASTS. For reference, see the flow in Figure 35.34. When the DMA is disabled, then consider what to do with the remaining or new incoming messages to this particular FIFO reception.

When the FIFO is not disabled, reception to the FIFO continues.

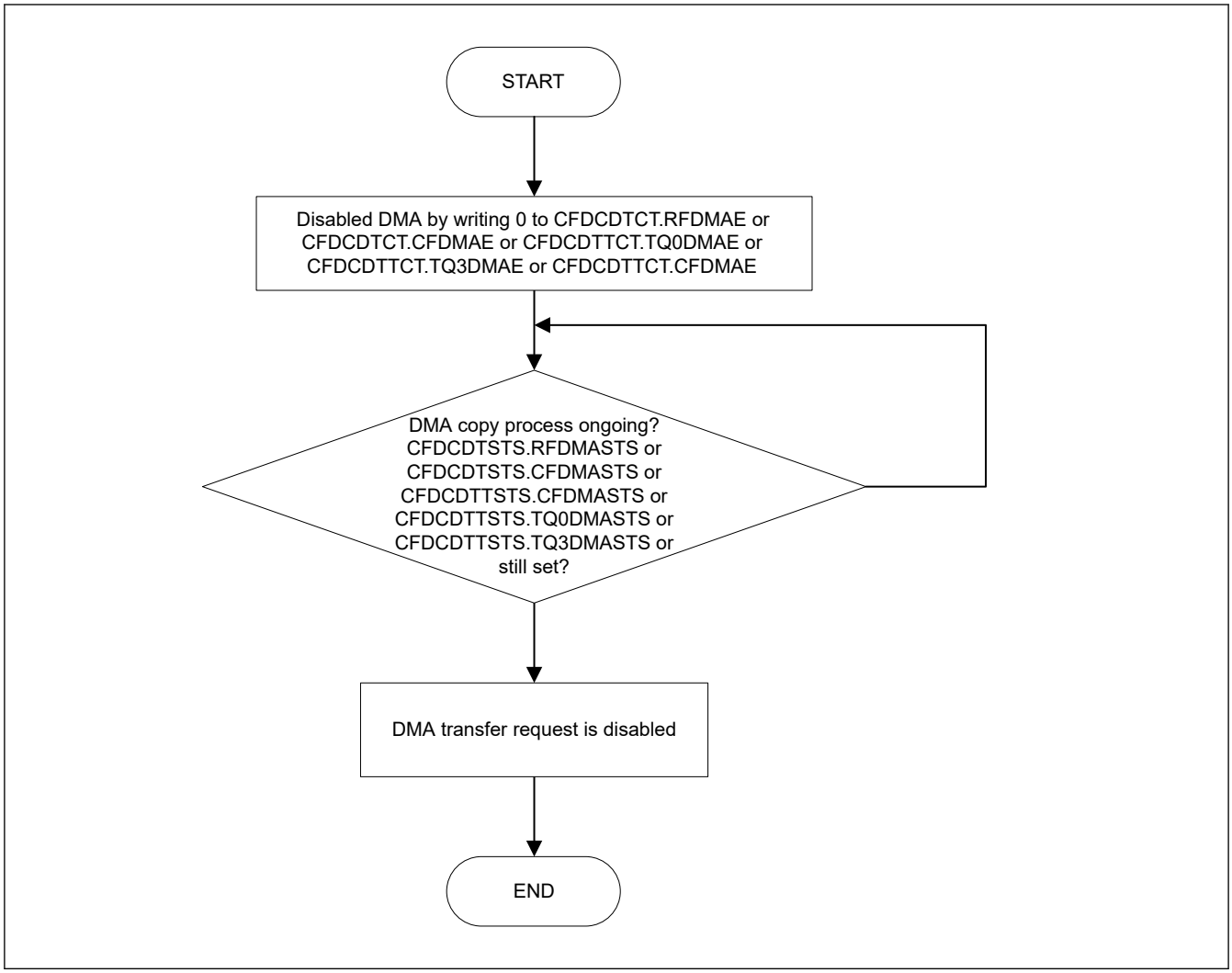


Figure 35.34 DMA disable flow

35.9 Reception and Transmission

35.9.1 Reception

FIFO buffers or Common FIFO buffers configured in RX mode or GW mode depending on the Acceptance Filter List entries are as follows:

- Up to 32 RX Message Buffers can be configured
- 8 RX FIFO buffers available
- Up to 6 Common FIFO buffers can be configured in RX mode or GW mode
- Up to 4 TX Queues can be configured in GW mode

35.9.1.1 Message Storage in RX Message Buffers

When a message is successfully received and stored in a RX Message Buffer, the corresponding New Data flag is set in the RX Message Buffer New Data Register.

The CAN message can be read from the corresponding RX Message Buffer.

If a new message is stored into a RX Message Buffer before the previous message in this message buffer can be read, then the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the RX Message Buffer. If such a loss of messages is not acceptable, then RX FIFO should be used to store related messages.

Note:
- Interrupts are not provided for the RX Message Buffers in the CAN-FD module and so the RX Message Buffer New Data Register should be accessed periodically to check if a new message has been stored in the RX Message Buffers.
- Unused data bytes are filled with 0x00 depending on the DLC value.

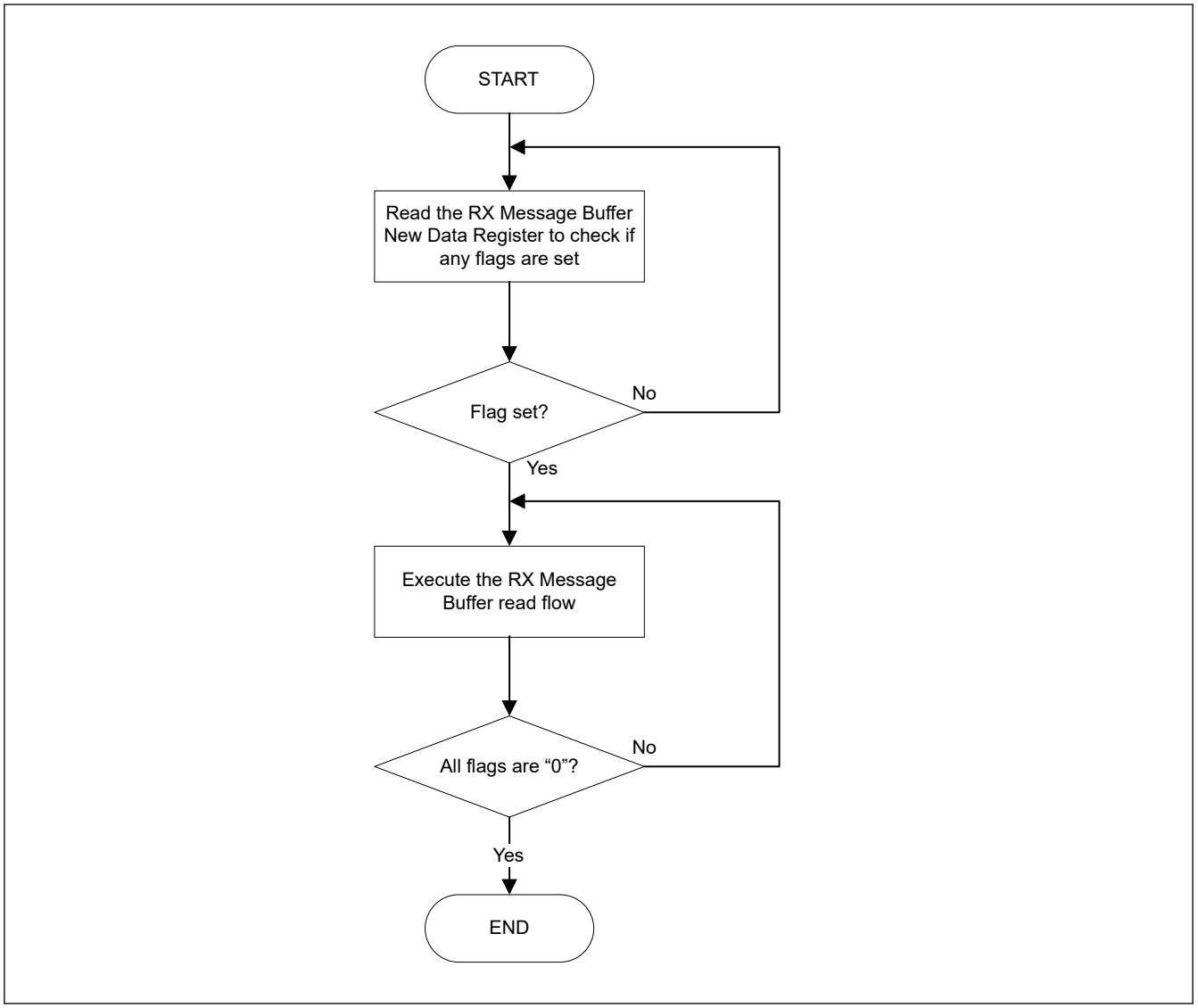


Figure 35.35 RX Message Buffer message access flow

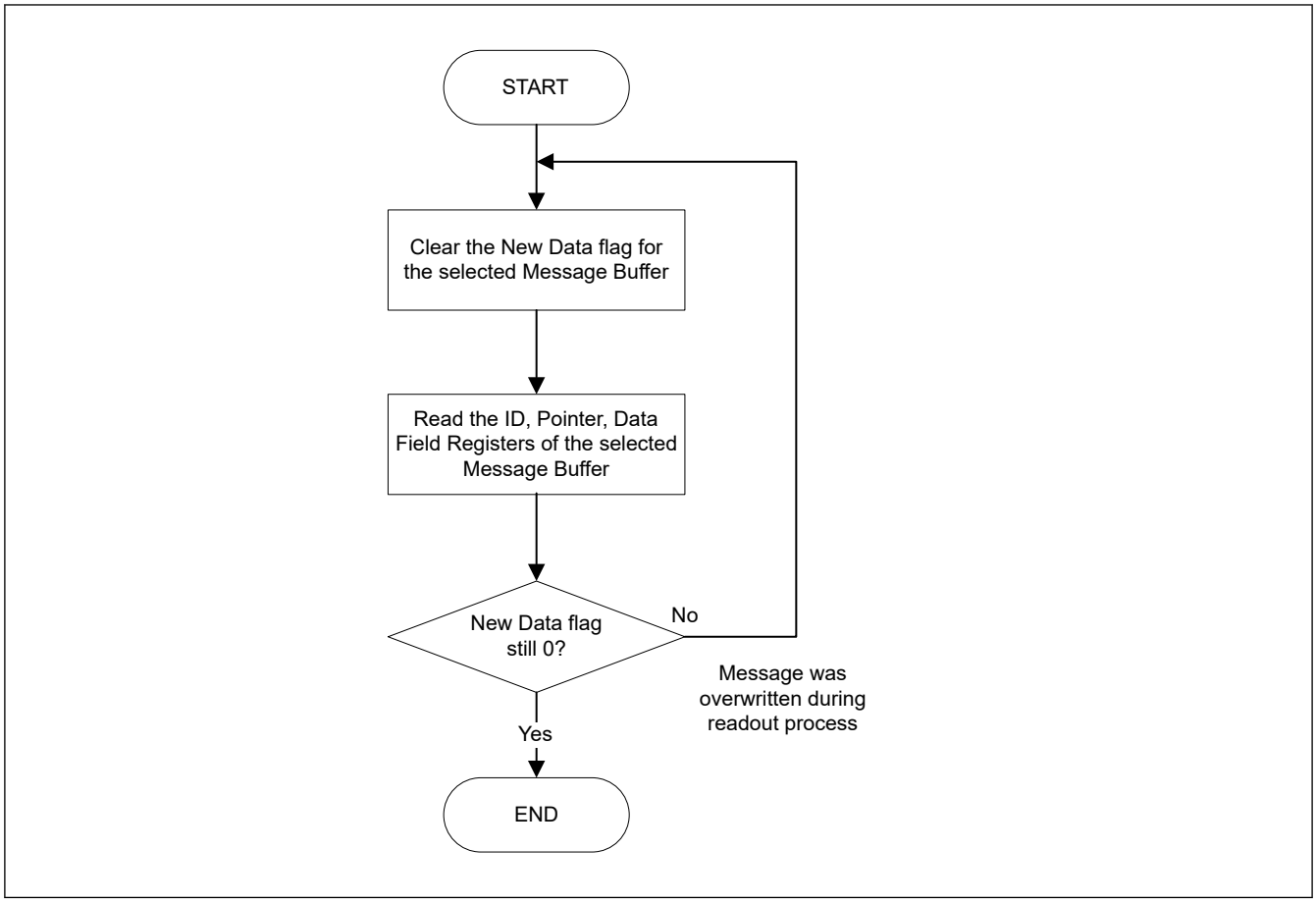


Figure 35.36 RX Message Buffer read flow

35.9.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to RX FIFO buffers or Common FIFO buffers configured in RX or GW mode should be configured based on system requirements.

The CFDGAFLP1n.GAFLFDP[31:0] field in the matching AFL entry selects the FIFO buffers to which the related reception message is stored.

When the received message is stored in one or more RX FIFO buffers or Common FIFO buffers configured in RX mode or GW mode, the message counter value is incremented in the corresponding RX FIFO Status Registers or Common FIFO Status Registers.

Depending on the configuration of the FIFO buffers, an interrupt may also be generated.

The message can be read from the corresponding FIFO Access registers.

Note: Because many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, the FIFO Full flag is set.

When the value 0xFF is written to the corresponding FIFO Pointer Control Register, the message count is decremented by 1.

Only write 0xFF to the FIFO Pointer Control register after reading the complete message from the FIFO Access registers of the corresponding FIFO.

When all the messages stored in the FIFO are read, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO Full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

An appropriate value can be configured as warning level to generate an interrupt before the FIFO full condition occurs to avoid loss of a message due to overrun condition.

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting CFDCFCCEn.CFMOWM bit.

- When CFDCFCCEn.CFMOWM = 0:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded and CFDCFSTSn.CFMLT bit is set to 1.
- When CFDCFCCEn.CFMOWM = 1:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message.
  The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message.
  The CFDCFSTSn.CFMOW bit is then set to 1, which notifies that the oldest message has been overwritten with the received message.
  In addition, if a CAN bus error or arbitration-lost for the transmitting message occurs in the transmit/receive FIFO buffer full, the transmitting message is lost and retransmission of the message is not performed. The read point moves to the next message automatically.

Do not write change for this bit when the CFDCFCCn.CFE bit is 1.

Common FIFO can set interrupt when:

- CAN frame reception is completed
- FIFO is in full status in RX mode or GW mode

Note:
- The message lost can be set only in RX or GW mode by CAN, the flag is not set when the CPU is overloading the FIFO buffers.
- When CFDGAFLP0n.GAFLSRDi (i = 0 to 2) is set and CFDTXQCCin.TXQGWE (i = 0 to 2, n = 0, 1) is also set, a receiving frame is stored in the target TXQ as send data by routing.

The RX FIFO Buffers and the Common FIFO Buffers configured in RX or GW mode can be disabled at any time by clearing the CFDRFCCn.RFE or CFDCFCCn.CFE bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

When the CFDRFCCn.RFE or CFDCFCCn.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO Buffers are lost and no further messages can be stored into the FIFO.

When the RX FIFO Buffers or Common FIFO Buffers configured in RX mode are assigned as DMA channel, software should not access the FIFO Access Register of this FIFO buffer or write 0xFF to the FIFO Pointer Control Register (CFDCFPCTRn.CFPC or CFDRFPCTRn.RFPC), because this could lead to unintended FIFO message decrement. The DMA channel controls the FIFO decrement automatically.

Note: If the interrupt flag is set for a FIFO Buffer and then the FIFO is disabled, the interrupt flag is not automatically cleared. The interrupt flag should be cleared before disabling the FIFO.

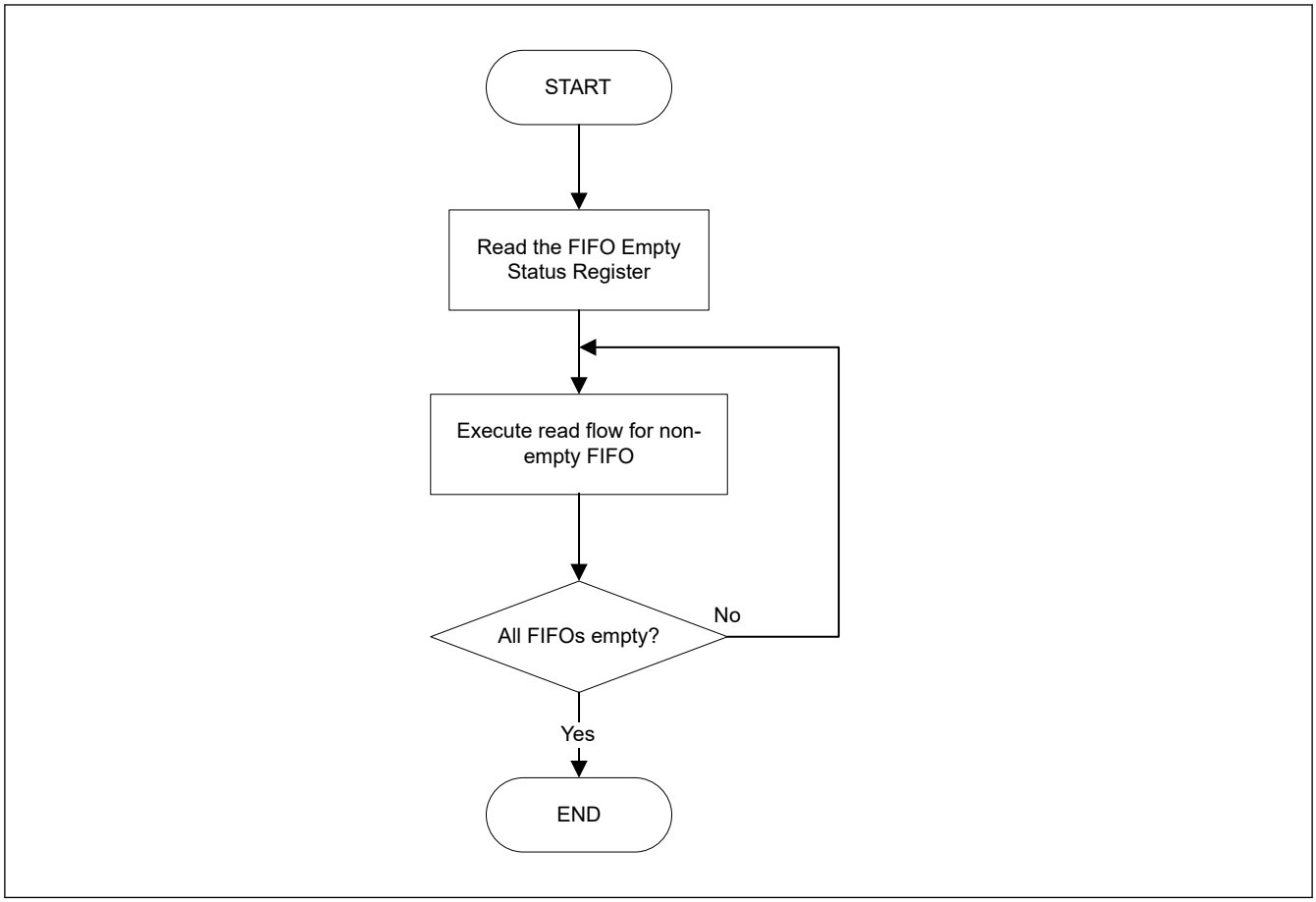


Figure 35.37 FIFO buffer message access flow (example for polling case)

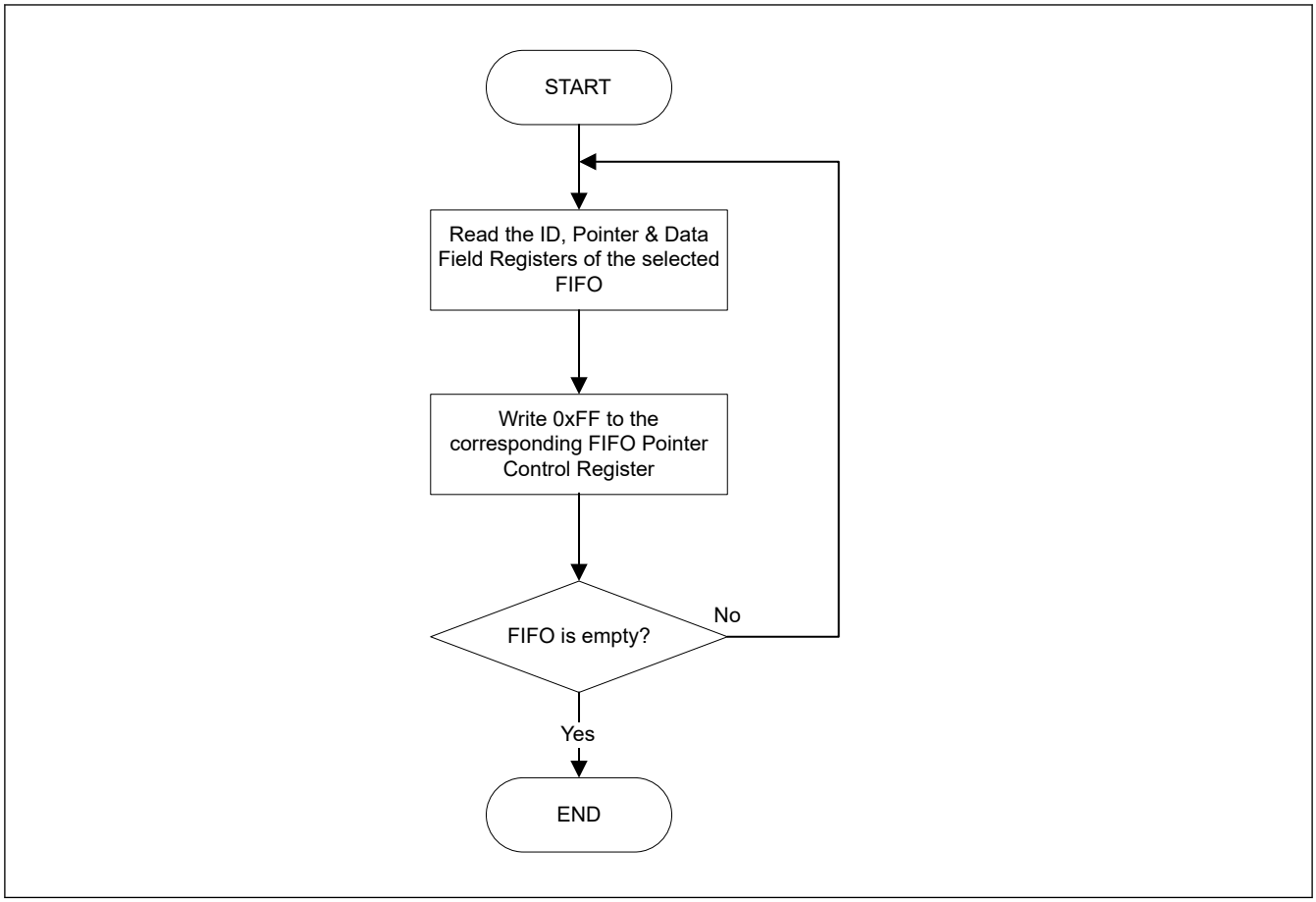


Figure 35.38 RX FIFO buffer read flow (example for polling case)

Note: When the next frame is received before clearing the completion interrupt flag of reception, the completion interrupt of reception is not set again.

Even if it clears an interruption flag after the completion processing of reception, the already received interrupt flag is not set.

It is necessary to perform the completion processing of reception even before the next completion of frame reception, and to clear an interruption flag.

When processing does not meet the deadline, after checking that receiving data is empty, interrupt flag is cleared and it checks that receiving data is empty again.

35.9.1.3 Timestamp

The Timestamp counter is a free-running counter that can be used to check reception time of an incoming message or transmission time of successful transmitted messages. The Timestamp counter value is captured based on the CFDGFDCFG.TSCCFG[1:0] configuration (at the sample point of start of frame, point in time when the frame is valid, or for CAN-FD frames also at the sample point of the RES bit). For reception, it is stored together with the message ID and Data into the target RX Message Buffer or RX/GW FIFO.

For transmit message, the Timestamp counter value is stored as part of the TX History List entry.

The counter can be clocked with the peripheral clock or with the CAN channel bit timing clock. The counter source clock can be configured with the CFDGCFG.TSSS bit of the Global Configuration Register. If it is 0, the peripheral clock is used. If it is 1, the selected CAN channel bit time clock is used.

The channel selection is done with the CFDGCFG.TSBTCS bit of the Global Configuration Register.

Care must be taken when using selected CAN channel bit time clock as the clock source. When entering Channel Halt mode or Channel Reset mode for this channel, the Timestamp counter is stopped. Therefore, for other CAN channels, the Timestamp counter value is also not updated.

If peripheral clock is selected as the Timestamp counter clock source, Channel modes do not influence the Timestamp counter function.

The source clock for the Timestamp counter can be divided by a factor defined by the CFDGCFG.TSP bits (Timestamp Prescaler) in the Global Configuration Register.

The Timestamp counter can be reset to 0x0000 with the CFDGCTR.TSRST bit (Timestamp Reset).

35.9.2 Transmission

There are several possible transmission configurations for each channel:

- Normal transmission
- FIFO transmission
- Gateway transmission
- TX Queue transmission
  A fixed number of transmission message buffers (64 TX message buffers) are dedicated for each channel. These message buffers are only used for transmission and cannot be configured for reception.
  Additionally transmission from TX Queue and/or Common FIFO in TX or GW mode can be configured in the following way (see Figure 35.39):
  - TX Queue: Up to 32 transmission message buffers for one channel can be grouped to form a TX Queue with a common access window.
    Upper transmission message buffers are used to form the TXQ1 or TXQ3.
    Lower transmission message buffers are used to form the TXQ0 or TXQ2.
    Transmission control and status registers of these transmission message buffers should not be used.
    One channel has four TX Queues. Each TXQ has an access window:
    - TXQ0 is transmission Message Buffer 0 of each channel
    - TXQ1 is transmission Message Buffer 31 of each channel
    - TXQ2 is transmission Message Buffer 32 of each channel
    - TXQ3 is transmission Message Buffer 63 of each channel

    When using TXQ1 and TXQ0 simultaneously, the sum of the depths of TXQ1 and TXQ0 should not exceed 32.
    When using TXQ3 and TXQ2 simultaneously, the sum of the depths of TXQ3 and TXQ2 should not exceed 32.
- Common FIFO (TX/GW mode): Each Common FIFO in TX or GW mode is linked to a dedicated channel.
  Each channel has a fixed number of three Common FIFOs assigned to it. Within the channel, a Common FIFO configured in TX or GW mode, can be freely linked (assigned) between 32 and 39 transmission message buffers (only one FIFO to one transmission message buffer).
  The Common FIFO buffer then replaces the transmission message buffer linked to it.
  Transmission control and status registers of these transmission message buffers should not be used.

See Figure 35.28 for information on Common FIFO buffer assignment to related channels.

Note: Common FIFO buffers should not be linked to TX message buffers that are already part of a TX Queue.

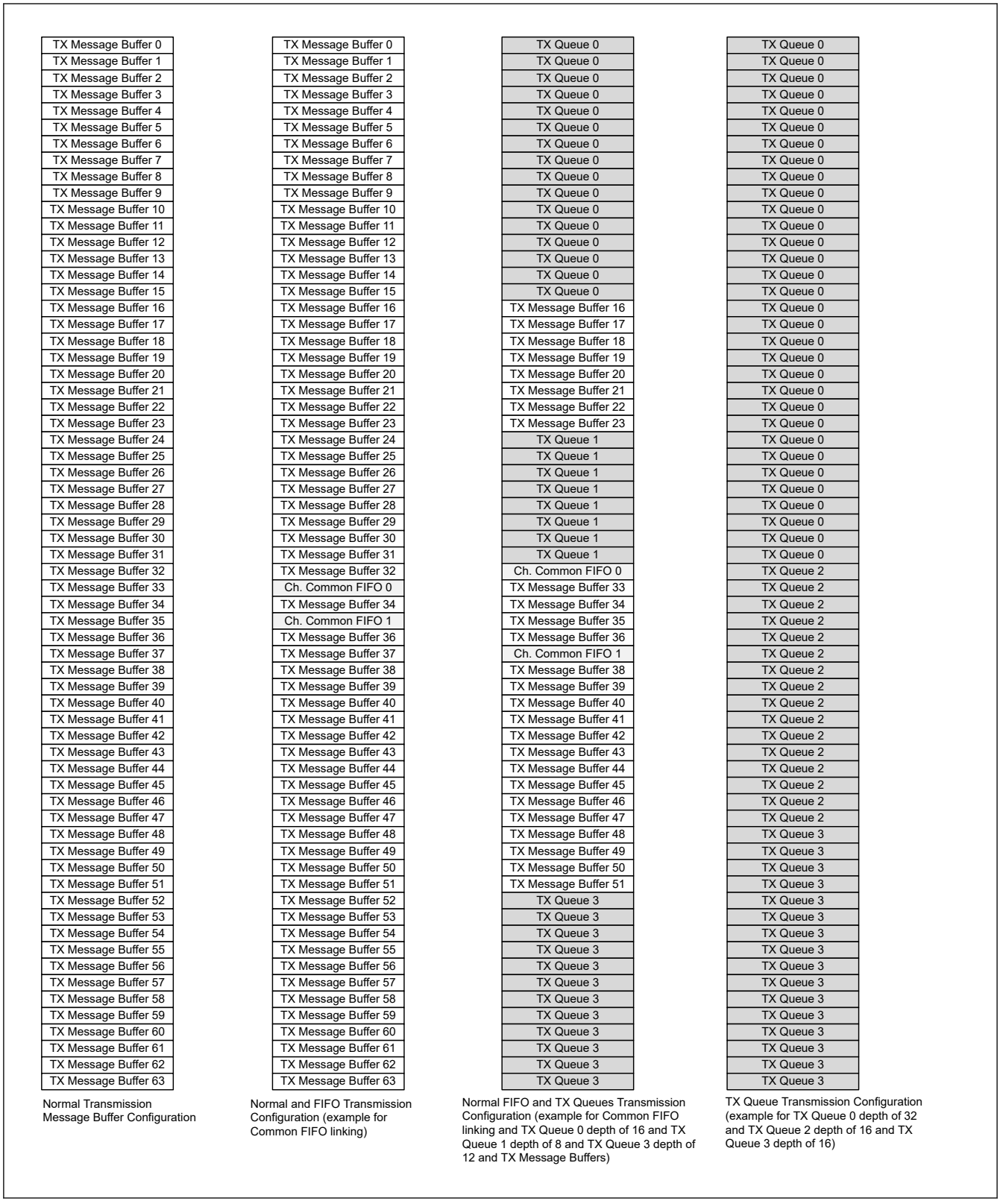


Figure 35.39 Channel transmission message buffer configuration

35.9.2.1 Transmission Priority

If two or more transmission message buffers of a channel are configured for transmission, the transmission priority in the CAN-FD module can be selected from the following two modes:

- CAN ID priority

- Message buffer number priority

The transmission priority mode is common for all message buffers and all CAN channels. It can be configured with the CFDGCFG.TPRI bit in the Global Configuration Register.

For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the TX message buffers linked to the Common FIFO buffers configured in TX mode or GW mode.

However, message buffer number priority should not be used if TX Queue is enabled.

For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All TX message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the TX message buffers linked to the Common FIFO buffers configured in TX mode or GW mode and includes the TX Queue message buffers.

If the ID of two or more message buffers is the same, the smaller message buffer number has higher priority for transmission.

Note: For Common FIFO buffers configured in TX mode or GW mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.

If the message is being transmitted from the FIFO, the next pending message within the same FIFO will be considered in the transmission arbitration.

In contrast to this, all transmission message buffers of a TX Queue participate in internal transmission arbitration.

Figure 35.40 shows the transmission configuration flow.

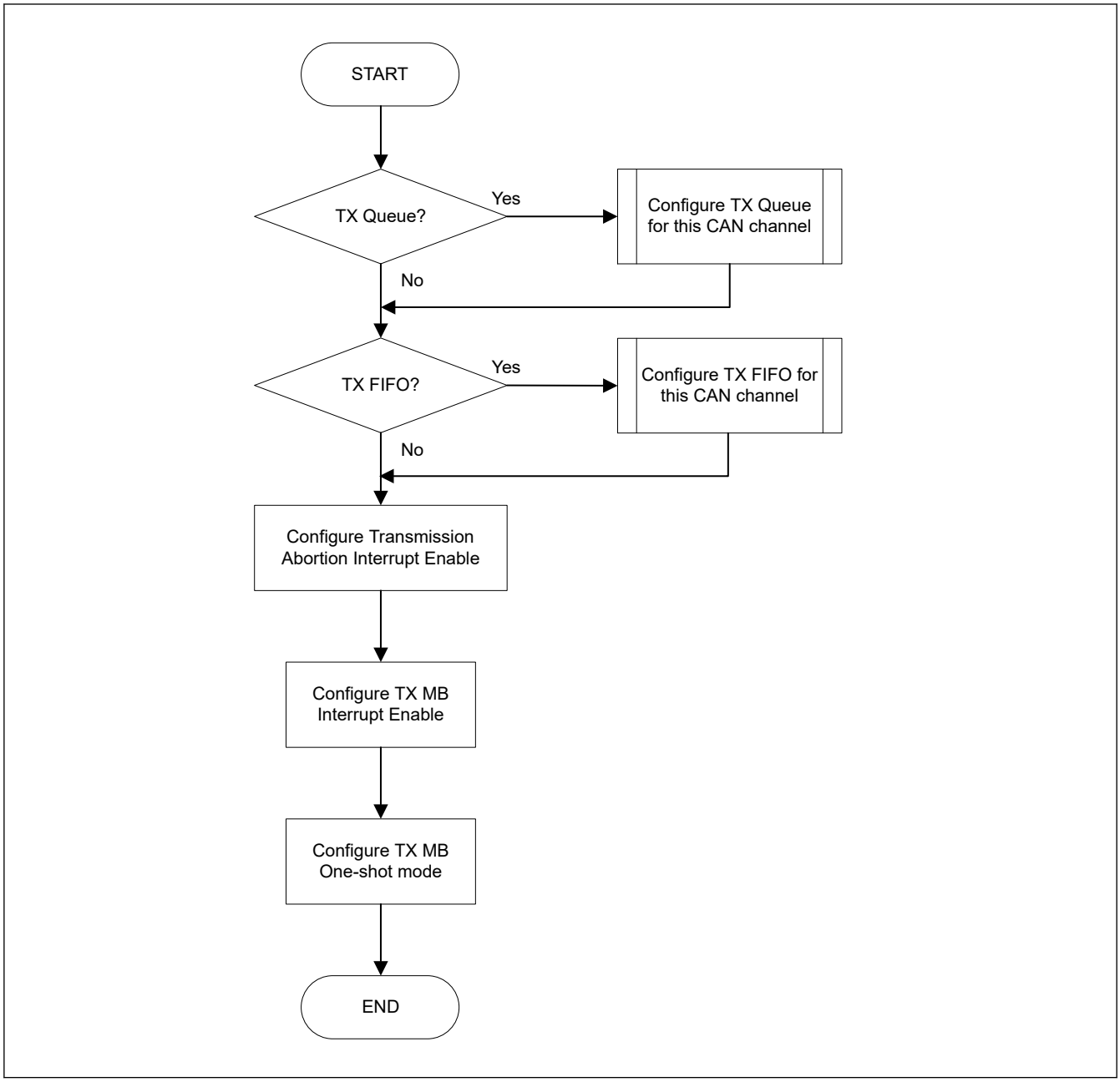


Figure 35.40 Transmission configuration flow

35.9.2.2 Normal Transmission

Each transmission message buffer has two modes of message transmission:

- Regular transmission mode
  If the message buffer is placed in regular transmission mode, the data frame or remote frame set in that message buffer can be transmitted.
  Completion of regular transmission can be checked through the related TX Message Buffer Transmission Result flag bits (CFDTMSTSn.TMTRF[1:0]) in the TX Message Buffer Status Registers. These bits are set to 10b or 11b when the regular transmission is successful.
  When arbitration is lost or an error occurs, message transmission will be attempted further if no transmission abort request is set for this transmission message buffer.
  New internal transmission arbitration for this channel is performed for all message buffers with transmission request.
- One-shot transmission mode
  When the CFDTMCn.TMOM bit of the TX Message Buffer Control Registers is set for a transmission message buffer, then the message buffer is placed in one-shot transmission mode and attempts to transmit a message only once.

Completion of one-shot transmission can be checked through the related TX Message Buffer Transmission Result flag bits (CFDTMSTSn.TMTRF[1:0]) in the TX Message Buffer Status Registers. The CFDTMSTSn.TMTRF[1:0] bits are set to 10b or 11b when the one-shot transmission is successful.
The CFDTMSTSn.TMTRF[1:0] bits are set to 01b when arbitration is lost or an error occurs during the transmission of the related message buffer.
Additional message transmission will not be attempted in this case.

The regular transmission request procedure after a configuration is shown in Figure 35.41.

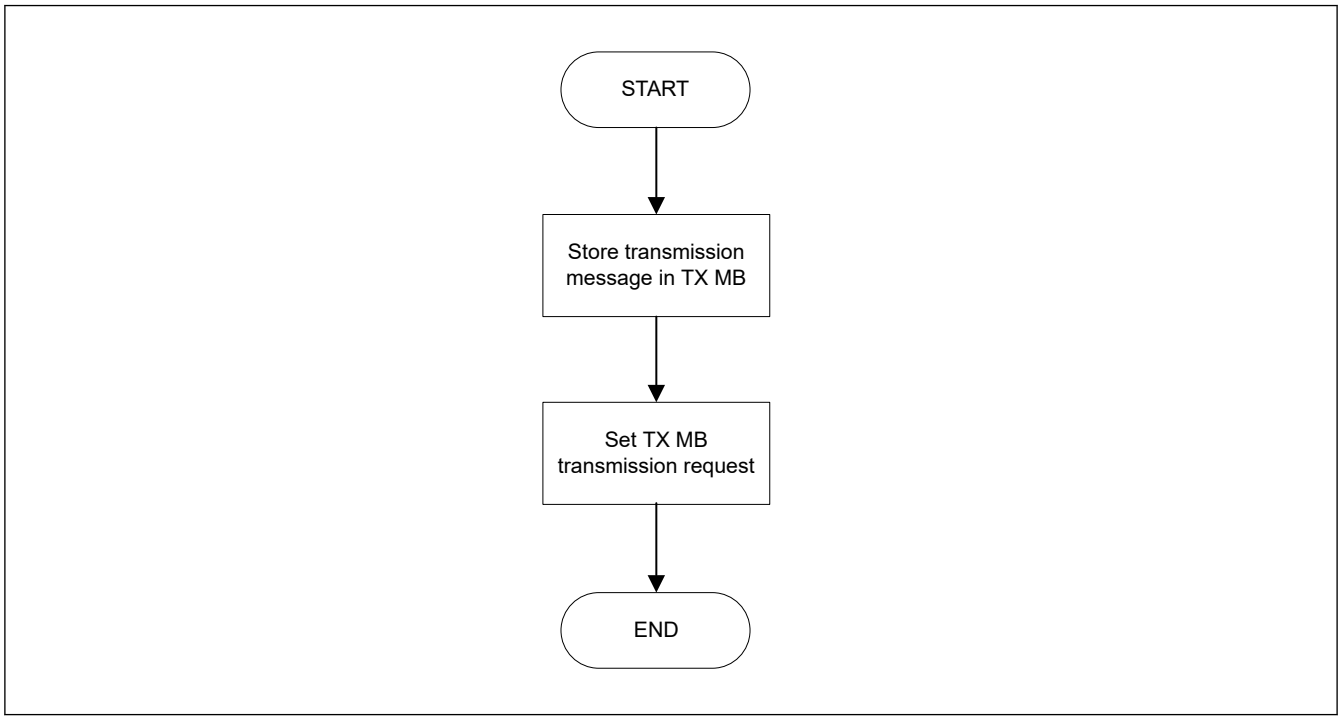


Figure 35.41 Transmission request procedure using normal TX message buffer mode

(1) TX Message Buffer Control Register Setting

Table 35.28 shows configuration of the normal CAN transmission mode.

Table 35.28 Configuration of CAN transmission mode

Transmission request CFDTMCn.TMTR	Transmission abortion request CFDTMCn.TMTAR	One Shot Enable CFDTMCn.TMOM	Communication activity
0	0	0	Message buffer is disabled
0	0	1	Message buffer is disabled
1	0	0	Configured as a transmission message buffer for a data frame or a remote frame
1	0	1	Configured as a one shot transmission message buffer for a data frame or a remote frame
1	1	0	Transmission abortion is requested
1	1	1	One shot transmission abortion is requested

The configuration bits can be configured in the TX Message Buffer Control Registers.

Figure 35.42 shows timings for successful transmission for two message buffers of one channel.

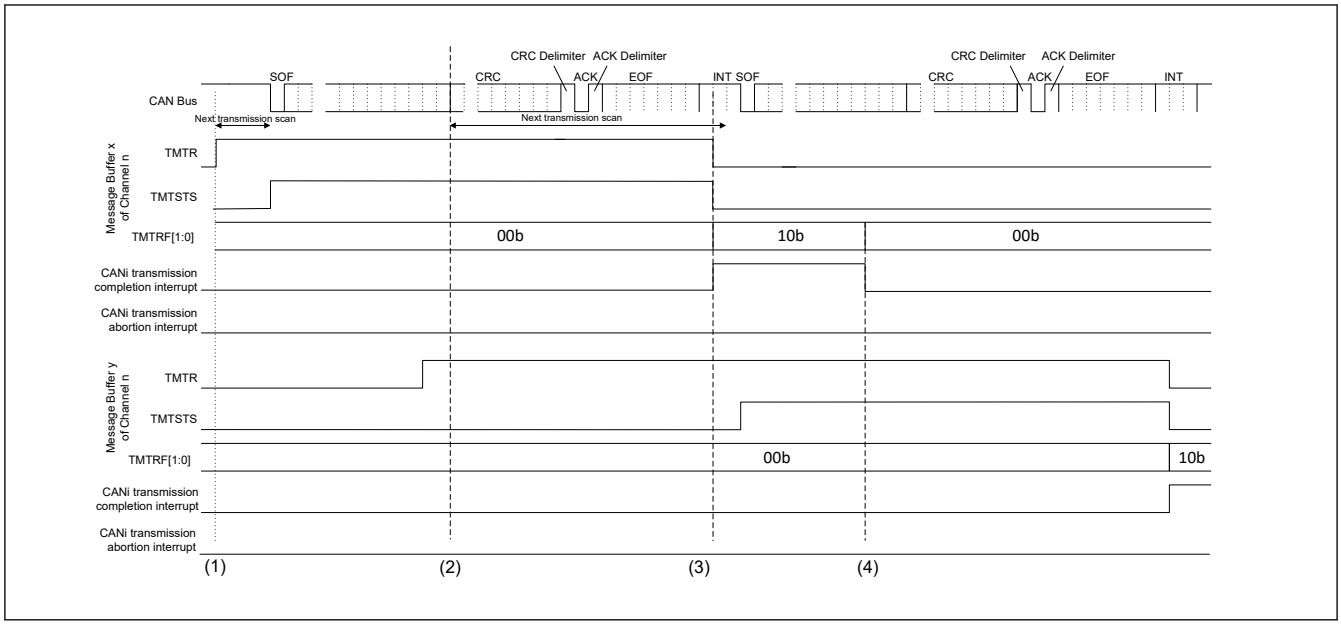


Figure 35.42 Timing of request and flag bits for successful transmission

1. If the CFDTMCn.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, message buffer scanning procedure starts to determine the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSn.TMTSTS bit in the related TX Message Buffer Status Registers is set (Transmitting/Transmitter), and CAN channel starts the transmission[*1].
2. On the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.
   The scan time can be delayed due to other transmission scan on other channels, but it will be finished before Intermission 3 to be able to continue transmission without any gaps.
3. If the message is successfully transmitted, the CFDTMSTSn.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 10b and CFDTMSTSn.TMTSTS and the CFDTMCn.TMTR bits are cleared.
   When the TMIE bits in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line the CFDTMSTSn.TMTRF flag bits must be cleared.
4. Before starting the next transmission, clear the CFDTMSTSn.TMTRF[1:0] bits. Load the next message in the transmission message buffer and set the CFDTMCn.TMTR bit again. CFDTMCn.TMTR bit cannot be set again before CFDTMSTSn.TMTRF[1:0] bits are cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs either during the transmission or following the loss of arbitration, then during Error Frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

Note: The setting point of CFDTMSTSn.TMTSTS is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID due to the synchronization logic implemented for the PLL bypass.

Figure 35.43 shows timings for transmission abort for two message buffers of one channel.

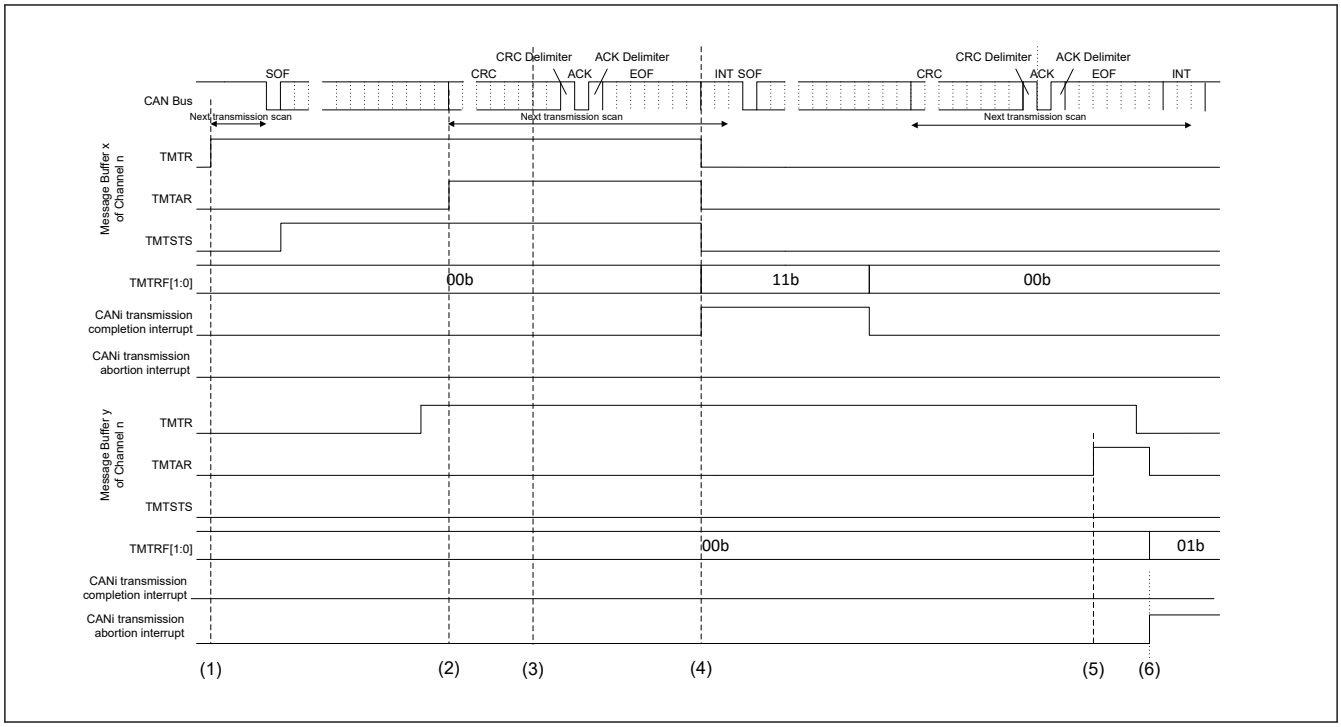


Figure 35.43 Timing of request and flag bits for transmission abort

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSn.TMTSTS bit is cleared.

1. If the CFDTMCn.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, message buffer scanning procedure starts to determine the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSn.TMTSTS bit in the TX Message Buffer Status Registers is set (Transmitting/Transmitter), and CAN channel starts the transmission[*1].
2. If the CFDTMCn.TMTAR bit is set when the related message buffer is already selected for transmission or currently transmitting, the message is not aborted, if no error occurs or arbitration is lost.
3. On the first CRC bit, the transmission scanning procedure starts for the next transmission. In this example timing diagram, message buffer y is not selected as the next transmission message buffer.
   The scan time can be delayed due to other transmission scan on other channels, but it will be finished before Intermission 3 to be able to continue transmission without any gaps.
4. If the message is successfully transmitted, the CFDTMSTSn.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 11b and CFDTMSTSn.TMTSTS and the CFDTMCn.TMTR bits are cleared.
   When the TMIE bits in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line, clear the CFDTMSTSn.TMTRF[1:0] bits.
5. Another CAN node is transmitting on the CAN bus (CFDTMSTSn.TMTSTS not set). If the CFDTMCn.TMTAR bit is set when the related channel is under transmission scan, the transmission request cannot be cleared.
6. After internal processing time, the transmission is aborted and the CFDTMSTSn.TMTRF[1:0] bits are set to 01b. If the message buffer is not transmitting or selected as the next transmission message buffer or under transmit scan, then the abort is immediately accepted and the corresponding CFDTMSTSn.TMTRF[1:0] bits in the TX Message Buffer Status Registers are set to 01b.
   In addition, CFDTMCn.TMTR, and CFDTMCn.TMTAR bits are cleared automatically.
   When the transmission abort interrupt enable TAIE bit of the related Channel Control Register is set, an interrupt is generated for successful transmission abort.
   To clear the related interrupt line, clear the CFDTMSTSn.TMTRF[1:0] bits.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSn.TMTSTS is cleared. The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit. If an error occurs, either during the transmission, or following the loss of arbitration, then during Error Frame, the transmission scanning procedure is performed again to search for the highest priority transmission Message Buffer.

35.9.2.3 TX FIFO or GW FIFO Transmission

Three common FIFO buffers are assigned to each channel. The three FIFO buffers can be linked to any normal TX Message Buffer position for this channel by the CFDCFCCn.CFTML[4:0] bits in the Common FIFO Configuration/Control Register if configured in TX or GW mode.

When the transmission scan starts and the FIFO buffer corresponding to this TX Message Buffer is enabled, the relevant message in the FIFO buffer participates in the transmission scan.

Configuration of a TX Message Buffer linked to a FIFO buffer configured in TX or GW mode should not be done.

(1) TX FIFO Operation

CAN messages can be written into the TX FIFO by writing to the corresponding FIFO Access registers.

When the value 0xFF is written into the corresponding FIFO Pointer Control Register, the message count of the related FIFO is incremented by 1.

Only write to the FIFO Pointer Control register after writing the complete message to the corresponding FIFO Access registers.

If the message count matches the FIFO Depth, the FIFO Full flag is set.

The oldest message in the TX FIFO is included in the scan for transmission by the corresponding CAN-FD module channel logic.

When a message is successfully transmitted from the TX FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the FIFO Empty flag is set.

The interrupt generation conditions for the TX FIFO buffers can be configured by configuring the CFDCFCCn.CFIM bit in the corresponding Common FIFO Configuration/Control Registers.

If CFDCFCCn.CFIM bit is 0, an interrupt is generated when the last message is successfully transmitted from the TX FIFO buffer.

If CFDCFCCn.CFIM bit is 1, an interrupt is generated for every successfully transmitted message from the TX FIFO buffer.

Common FIFO can set interrupt, when CAN frame transmitted is completed.

The Common FIFO buffers configured in TX mode can be disabled by clearing the CFDCFCCn.CFE bit in the Common FIFO Configuration/Control Registers. If this bit is cleared to 0, the FIFO Empty flag is set as follows:

- Immediately if the message from the TX FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX FIFO is already scheduled for transmission or already in transmission.

Note: The Common FIFO buffer is considered as disabled after clearing the CFDCFCCn.CFE bit only when the Empty flag is set for the corresponding Common FIFO Buffer.

Other possible messages pending from the TX FIFO are lost and their transmission must be requested again. Before CFDCFCCn.CFE is set again, ensure that CFDCFSTSn.CFEMP bit is set and that there is no pending abort from the TX FIFO.

When the CFDCFCCn.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after configuration is shown in Figure 35.44.

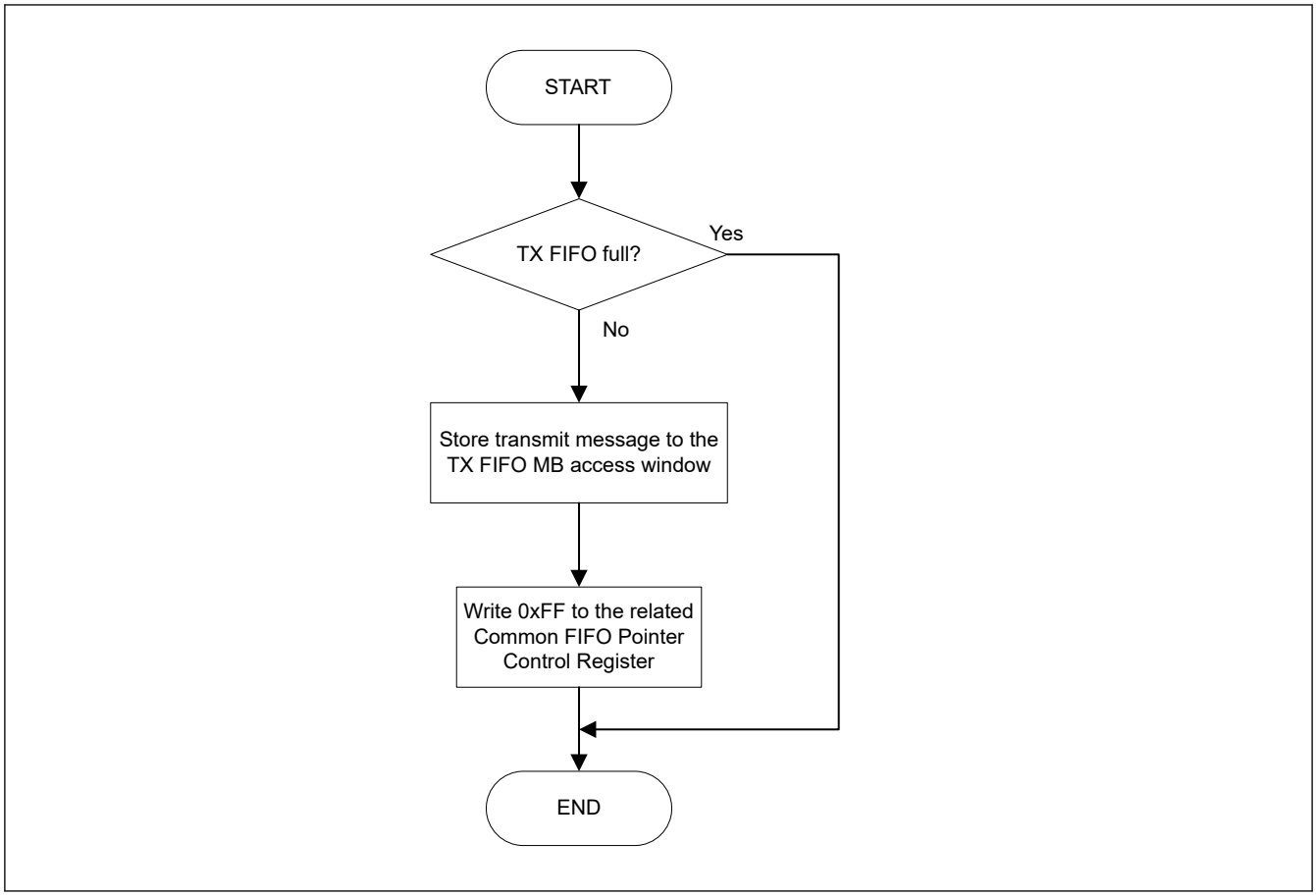


Figure 35.44 TX FIFO transmission request procedure

(2) GW FIFO Operation

The AFL entries for routing the received messages to GW FIFO buffers should be configured based on the requirements of the system. The matching AFL entry selects the GW FIFO buffer for storage of a received message on any of the CAN channels.

When a message is successfully received and stored in a GW FIFO buffer, the FIFO message count in the corresponding FIFO Status Register is incremented by 1.

If the message count matches the FIFO depth, then the FIFO Full flag is set.

The oldest message in the GW FIFO is included in the scan for transmission by the corresponding CAN-FD module channel logic.

When a message is successfully transmitted from the GW FIFO, the message count value is decremented by 1. When all the messages from the GW FIFO are transmitted, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting CFDCFCCEn.CFMOWM bit.

- When CFDCFCCEn.CFMOWM = 0:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded and CFDCFSTSn.CFMLT bit is set to 1.
- When CFDCFCCEn.CFMOWM = 1:
  When writing of data is required due to reception of a new message while a transmit/receive.
  FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message.
  The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message.

The CFDCFSTSn.CFMOW bit is then set to 1, which notifies that the oldest message is overwritten with the received message.
In addition, when a CAN bus error or arbitration-lost for the transmitting message occurs in transmit/receive FIFO buffer full, the transmitting message is lost and retransmission for the message is not performed. The read point moves to the next message automatically.

The interrupt generation conditions for the GW FIFO buffers can be configured by configuring the CFDCFCCn.CFIM bit in the corresponding Common FIFO Configuration/Control Registers.

If CFDCFCCn.CFIM bit is 0, then RX interrupt flag is set when FIFO counter increments and reaches value configured by CFDCFCCn.CFIGCV and the TX interrupt flag is set when FIFO transmits the last message successfully.

If CFDCFCCn.CFIM bit is 1, then RX interrupt flag is set at the end of storage of every received message and TX interrupt flag is set if a message is successfully transmitted from the FIFO.

Common FIFO can set interrupt when:

- CAN frame transmitted is completed
- CAN frame reception is completed
- FIFO is in full status in RX mode or GW mode

When CFDCFCCEn.CFBME = 1, it becomes FIFO buffering mode, send data is stored in Common FIFO, and transmission is stopped. Transmission will be started if it is set as CFDCFCCEn.CFBME = 0.

The Common FIFO buffers configured in GW mode can be disabled by clearing the CFDCFCCn.CFE bit in the Common FIFO Configuration/Control Register. If this bit is cleared, the GW FIFO becomes empty as follows:

- Immediately if the message from the GW FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the GW FIFO is already scheduled for transmission or already in transmission

Other possible messages pending from the GW FIFO are lost.

Before CFDCFCCn.CFE is set again, ensure that the CFDCFSTSn.CFEMP bit is set and that there is no pending abort from the GW FIFO.

When the CFDCFCCn.CFE bit is cleared and the CFDCFSTSn.CFEMP bit is set, the message read and write pointers of the GW FIFO are cleared and are no longer active. Therefore, all messages in the GW FIFO buffers are lost and no further message can be stored into the GW FIFO.

In applications intended to be used as CAN-to-CAN gateways, it is useful if the Error State Indication (ESI) information of the routing messages is not replaced by the sending node Error State Indication. For this, each channel has the control function register CFDCnFDCFG.ESIC to replace their own ESI information by the routing ESI information.

Note: If the sending node is error passive, the ESI bit is sent anyway as error passive (ESI = 1).

(3) Interval Timer for FIFO Transmission

For each Common FIFO in TX or GW mode, it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFDCFCCn.CFE bit is set.

When the Common FIFO in TX or GW mode is enabled, the first message is transmitted without considering this interval time.

The interval timer stops counting when:

- FIFO is disabled by clearing the CFDCFCCn.CFE bit
- CAN channel is in CH_RESET mode

The interval time is specified by the CFDCFCCn.CFITT value in the Common FIFO Configuration/Control Register and can be specified from 0 to 255 timer units.

The timer unit can be defined based on two different source clocks for the interval timer. To disable the interval timer for FIFO transmission, a value of 0 should be selected.

The timer source can be selected by the configuration bit CFITSS in the Common FIFO Configuration/Control Register. For the timer source the CAN bit timing clock of the FIFO related channel or a global reference clock can be selected.

If CAN channel bit time clock is configured as clock source and the CAN channel enters CH_HALT or CH_RESET or CH_SLEEP mode, the interval timer is stopped for that channel.

If peripheral clock is selected as interval timer clock source, then the interval timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.

The reference clock can be used to configure the interval time in fixed time units. It is based on the peripheral clock. The reference clock prescaler value CFDGCFG.ITRCP in the Global Configuration Register defines the relation between the peripheral clock frequency/period and the reference clock period.

See Table 35.29 for CFDGCFG.ITRCP configuration values to achieve different reference clock periods based on the peripheral clock frequency/period.

Table 35.29 Configuration example for the FIFO interval timer reference clock

Peripheral clock (PCLKM)	Reference clock		
	1 µs	100 µs	500 µs
125 MHz / 8 ns	125	12500	62500

Additionally, the reference clock resolution can be specified by the interval timer reference clock resolution value CFDCFCCn.CFITR in the Common FIFO Configuration/Control Register.

The interval time is based on the reference clock period multiplied by the configured value (× 1 or × 10).

The reference clock based interval timer can be used to follow the requirements of the ISO 15765-2 Separation Time. The whole range for the separation time from 100 µs to 127 ms can be covered.

The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).

When the interval time has elapsed, the next transmission request is raised by the related TX/GW FIFO. Therefore, the interval time defines the minimum time between two messages transmitted from one FIFO.

The next message is sent at earliest after this interval time.

Figure 35.45 shows an example timing of the internal processing.

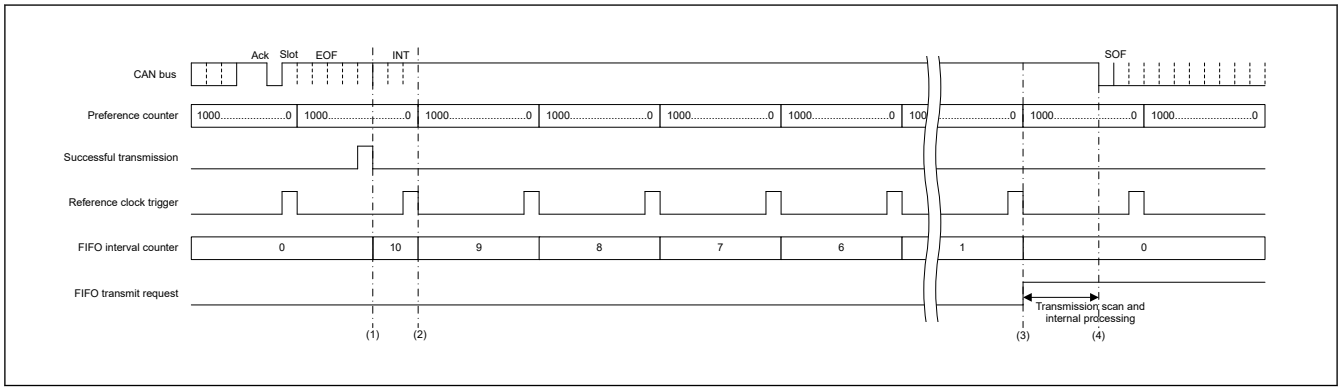


Figure 35.45 Example for interval processing time

The configuration for the above timing is as follows:

- Peripheral clock frequency = 125 MHz
- Interval timer reference clock (CFDGCFG.ITRCP) = 1250 times
- Reference clock due to the above settings = 10 µs
- Common FIFO interval timer source selection (CFDCFCCn.CFITSS) = 0
- Common FIFO interval timer resolution (CFDCFCCn.CFITR) = 0
- Common FIFO interval transmission time (CFDCFCCn.CFITT) = 10 times
- Theoretical message separation interval = 100 µs

1. Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore, the first interval is counting less or equal to one reference clock interval.
2. With the next reference clock trigger, the FIFO interval timer is decremented.
3. When the FIFO interval timer reached the value 0 the FIFO transmit request is set.
4. When the FIFO is selected for transmission, the transmission starts. Due to internal processing this usually takes less than 3 CAN bit time, between internal FIFO transmit request set 3 and actual transmission.

In the worst case when multi events such as reception scan, internal message routing, transmit scan on all channels occur, then it can take up to 432 peripheral clock cycles.

As shown in Figure 35.45, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure CFDCFCCn.CFITT to the required minimum value + 1.

If additional TX message buffers or TX/GW FIFOs are configured for transmission for the same channel, the real delay between two messages transmitted from a TX FIFO can be much longer than specified by the interval time due to higher priority message transmission from these TX message buffers or TX/GW FIFOs.

Figure 35.46 shows a block diagram of the FIFO interval time generation circuit.

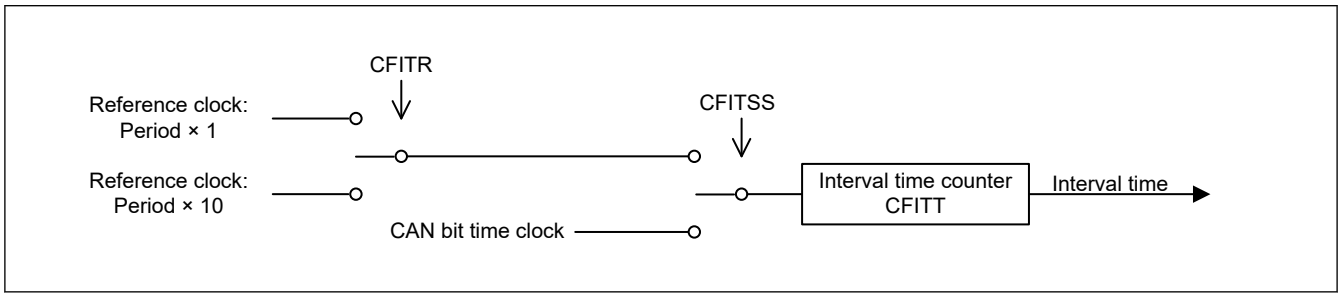


Figure 35.46 Block diagram of FIFO interval timer

35.9.2.4 TX Queue

Each enabled TX Queue for a specific channel consists of 3 to 32 TX message buffers, which are accessed by one access window. One channel has 4 TX Queues. One TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 0 as access window (referred to as TXQ0). The second TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 31 as access window (referred to as TXQ1). The third TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 32 as access window (referred to as TXQ2). The fourth TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 63 as access window (referred to as TXQ3).

All TXQ0, TXQ1, TXQ2, and TXQ3 messages enter the priority comparison for the transmission, which should be only ID Priority (CFDGCFG.TPRI = 0).

The registers for TXQ0 are CFDTXQCC0n, CFDTXQSTS0n, and CFDTXQPCTR0n.

The registers for TXQ1 are CFDTXQCC1n, CFDTXQSTS1n, and CFDTXQPCTR1n.

The registers for TXQ2 are CFDTXQCC2n, CFDTXQSTS2n, and CFDTXQPCTR2n.

The registers for TXQ3 are CFDTXQCC3n, CFDTXQSTS3n, and CFDTXQPCTR3n.

When access window TX Message Buffer No.63 (TXQ3) or TX Message Buffer No.32 (TXQ2) or TX Message Buffer No.31 (TXQ1) or TX Message Buffer No.0 (TXQ0) is used, refer to the related access registers TX Message Buffer ID Registers (CFDTMIDn), TX Message Buffer Pointer Registers (CFDTMPTRn), TX Message Buffer Data Field 0 Registers (CFDTMDF0_n), and TX Message Buffer Data Field 1 Registers (CFDTMDF1_n).

The depth of each TXQ0 buffer can be configured by writing to the CFDTXQCC0n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ0 can set from TXMB0 to TXMB31 as a queue buffer at the maximum.

The 31 available options for depth configuration are:

- 0x00: TX Queue disabled

- 0x01: Reserved
- 0x02: 3 messages
- ⋮
- 0x1D: 30 messages
- 0x1E: 31 messages
- 0x1F: 32 messages

The depth of each TXQ1 buffer can be configured by writing to the CFDTXQCC1n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ1 can set from TXMB31 to TXMB0 as a queue buffer at the maximum.

The 31 available options for depth configuration are:

- 0x00: TX Queue disabled
- 0x01: Reserved
- 0x02: 3 messages
- ⋮
- 0x1D: 30 messages
- 0x1E: 31 messages
- 0x1F: 32 messages

The depth of each TXQ2 buffer can be configured by writing to the CFDTXQCC2n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ2 can set from TXMB32 to TXMB63 as a queue buffer at the maximum.

The 31 available options for depth configuration are:

- 0x00: TX Queue disabled
- 0x01: Reserved
- 0x02: 3 messages
- ⋮
- 0x1D: 30 messages
- 0x1E: 31 messages
- 0x1F: 32 messages

The depth of each TXQ3 buffer can be configured by writing to the CFDTXQCC3n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ3 can set from TXMB63 to TXMB32 as a queue buffer at the maximum.

The 31 available options for depth configuration are:

- 0x00: TX Queue disabled
- 0x01: Reserved
- 0x02: 3 messages
- ⋮
- 0x1D: 30 messages
- 0x1E: 31 messages
- 0x1F: 32 messages

When using TXQ1 and TXQ0 simultaneously, the depth of TXQ is 32 or less in total.

When using TXQ3 and TXQ2 simultaneously, the depth of TXQ is 32 or less in total.

Do not access all the TX message buffers forming the TX Queue directly (except TX Message Buffer No. 63, TX Message Buffer No. 32, TX Message Buffer No. 31 and TX Message Buffer No. 0, which acts as TX Queue access window).

When CFDGAFLP0n.GAFLSRD i (i = 0 to 2) is set and the CFDTXQCCin.TXQGWE (i = 0 to 2, n = 0, 1) is also set, a receiving frame is stored in the target TXQ as send data by routing.

When CFDTXQCCn.TXQOWE bit is 1, the TX Queue is in TX Queue overwrite mode. If the message of the same ID is stored in TX Queue when a frame is received and it is stored in TX Queue, an old message is overwritten by a new message. Therefore, an old message is not transmitted. When the old message of the same ID is transmitting and a CAN bus error and an arbitration-lost occur, the message of old ID is not resent.

When using the function in GW mode and TX Queue overwrite mode, the depth of TXQ (CFDTXQCC0n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3. If it accesses by routing in gateway mode when a TXQ buffer is full, CFDTXQSTS.TXQMLT is set and send data is thrown away.

The function is valid for the standard ID frame and is invalid for the extended ID frame.

Explanation of operation of the TX Queue with same ID over-writing function in GW mode is shown in Figure 35.47.

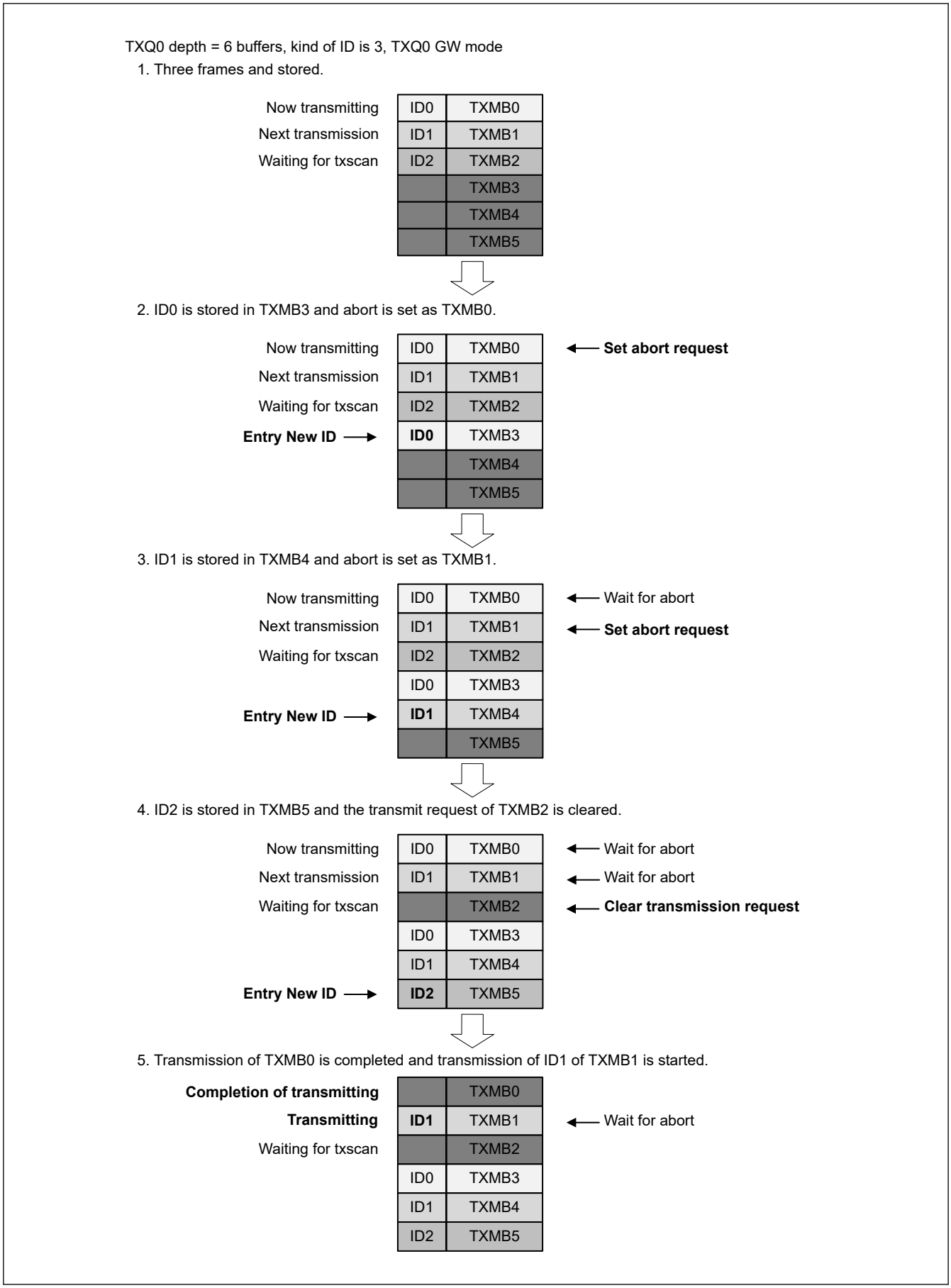


Figure 35.47 Operation of TXQ0 in Gateway mode

When a system writes in TXQ, a system should write in send data, after checking the state of TXQ.

Do not access or configure the related TX Message Buffer Control Registers.

The messages stored to the TX Queue access window are internally stored to a free buffer of the TX Queue.

When the buffer is full then no further access should be done to the queue, until it is no longer full. If it accesses by software writing in when the buffer of TXQ is full, send data is overwritten.

The TX Queue can be disabled by clearing the TXQE bit in the TX Queue Configuration/Control Register. If this bit is cleared, the TX Queue Empty flag is set as follows:

- Immediately if the message from the TX Queue is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX Queue is already scheduled for transmission or already in transmission.

Note: The TX Queue is disabled only when the Empty flag is set after clearing the TXQE bit for the corresponding TX Queue.

Other possible messages pending from the TX Queue are lost and their transmission must be requested again.

Before TXQE is set again, ensure that the CFDTXQSTSn.TXQEMP bit is set and that there is no pending abort from the TX Queue.

When the TXQE bit is cleared, all messages in the TX Queue buffers are lost and no further message should be stored into the TX Queue.

When a message has been stored to the TX Queue, 0xFF must be written into the TX Queue Pointer Control Register. This sets the transmit request automatically and changes the internal message buffer pointer to the next free message buffer location of the TX Queue.

Note: If two messages with the same identifier are stored in the TX Queue, then the order of transmission of these messages can be different from the order in which they were stored in the TX Queue.

To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same identifier is stored in the TX Queue. Or if TX queue overwrite mode is used, the frame of the same ID is rewritten on a new frame.

For the TX Queue a dedicated interrupt can be enabled by setting the TXQIE bit of the TX Queue Configuration/Control Register.

The interrupt mode can be configured with the CFDTXQCCn.TXQIM bit of the same register either to generate an interrupt for every transmitted message or for the last transmitted message.

The TX Queue transmission request procedure after configuration is shown in Figure 35.48.

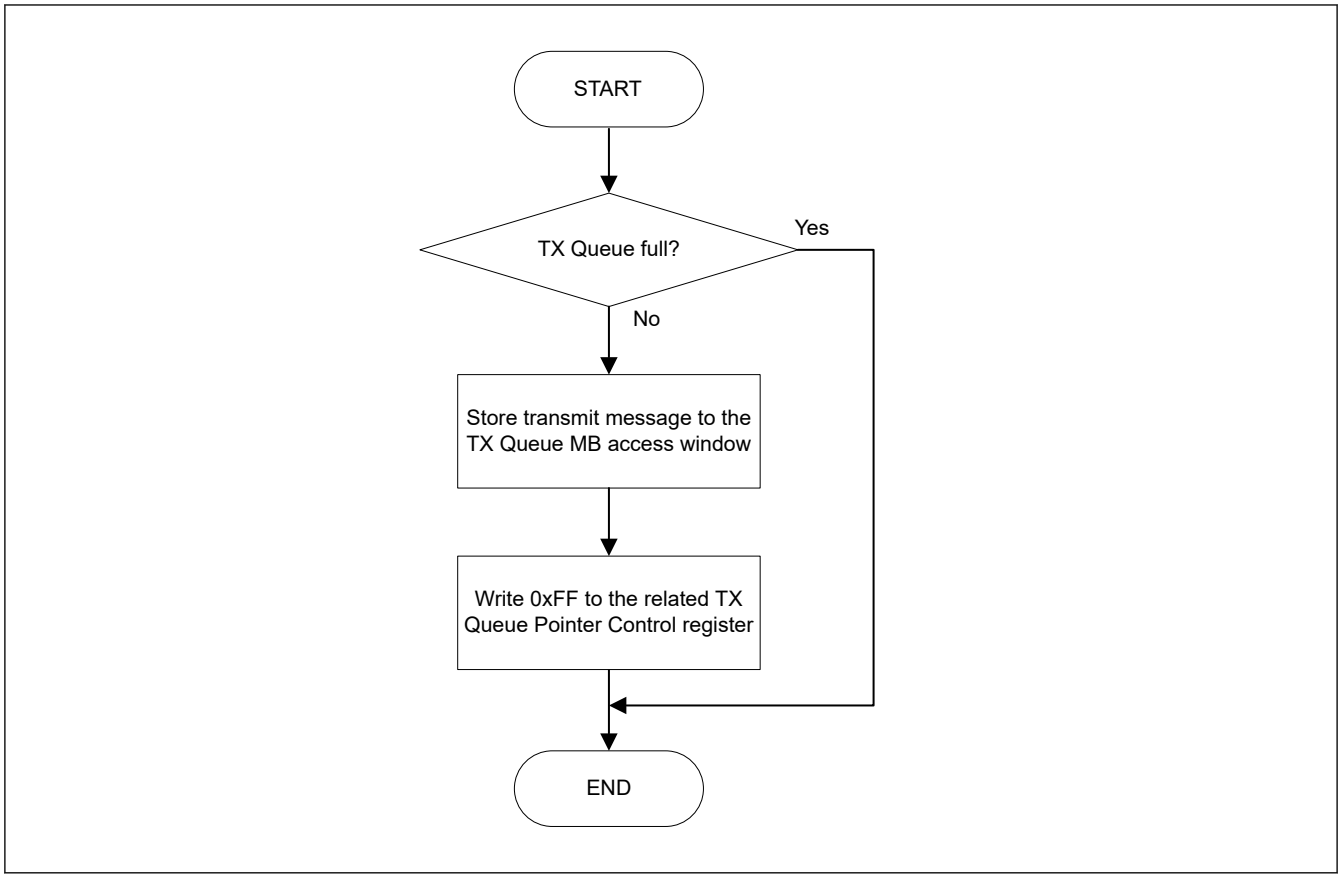


Figure 35.48 TX Queue transmission request

TXQ name	Access window	Range width	Direction	Hardware routing access point	CPU access point	DMA access point	Note
TXQ0	TXMB0	0, 3-32	TXMB0 → TXMB31	Yes	Yes	Yes	When using both TXQ0/1, the total number of stages is 32 or less
TXQ1	TXMB31	0, 3-32	TXMB31 → TXMB0	Yes	Yes	No	
TXQ2	TXMB32	0, 3-32	TXMB32 → TXMB63	Yes	Yes	No	When using both TXQ2/3, the total number of stages is 32 or less
TXQ3	TXMB63	0, 3-32	TXMB63 → TXMB32	No	Yes	Yes	

TXQ0 can use hardware routing, CPU access, and DMA access.

Hardware routing access, CPU access, and DMA access should not be used simultaneously. Choose one access method.

35.9.2.5 TX History List

The TX History List function records the information of the successfully transmitted message in the TX History List buffers for each CAN channel. Two TX History List buffers are provided for each CAN channel and each TX History List buffer can store up to 16 TX History List entries for a CAN channel.

The CFDTHLCCn.THLDTE bit of the TX History List Configuration/Control Register can be used to configure if only message information from TX FIFOs/TX Queue is stored or if all transmit message information from TX Queue, TX FIFO or normal TX message buffers should be stored in the TX History List for a CAN channel.

When a CFDTHLCCn.THLDGE bit is set up, the information on all the frames transmitted in GW mode is stored in TX History List.

Each transmit message can be individually configured for acceptance to the TX History List by the CFDCFID.THLEN bit in the Message Buffer Pointer Register.

The message information is stored to the TX History List Buffer of a CAN channel after the message is successfully transmitted on that CAN channel.

Storing to the List is not synchronized with the status of CFDTMSTSn.TMTRF[1:0] bits in the TX Message Buffer Status Register.

Due to internal processing, the storage to the List can happen with a delay after the successful transmission indication.

Storing the TX History List data can be recognized by the condition that the THLIF bit is set to 1 when the THLIE is configured to 1 or when the TX History List counter CFDTHLSTSn.THLMC[5:0] is increased.

The delay time is dependent on the number of channels due to internal processing.

- Maximum delay time from setting the CFDTMSTSn.TMTRF to storing the TX History List data is 224 peripheral bus clock cycles.

The History list records the following information of the transmitted message:

- Buffer type:
  - 001b: TX Message Buffer
  - 010b: TX FIFO
  - 100b: TX Queue
- Buffer number:
  TX Message Buffer, TX Queue Message Buffer or TX Message Buffer Link for the Common FIFO Buffer from which the transmission occurred. The number depends on the buffer type, see Table 35.30.
- Transmission ID:
  Transmission pointer stored in the transmission message
- Transmit timestamp:
  Message timestamp captured at the capture point as configured by CFDGFDCFG.TSCCFG.
- Transmission information label:
  Transmission information label stored in the transmission message.
- Transmit gateway buffer indication:
  For data transmitted from the gateway, CFDTHLACC0n.TGW bit is set to 1.

Table 35.30 TX History List buffer number entry

CFDTHLACC0n.BT[2:0] buffer type		
001b	**101b**	**100b**
TX Message Buffer	**TX FIFO**	**TX Queue**
TXMB0	Number shown corresponds to the common FIFO. TX Message Buffer Link CFTML of the related Common FIFO Configuration	Number shown corresponds to the message buffer belonging to the TX Queue for which the frame was transmitted.
TXMB1		
TXMB2		
TXMB3		
⋮		
TXMB30		
TXMB31		
TXMB32		
TXMB33		
TXMB34		
⋮		
TXMB61		
TXMB62		
TXMB63		

The Transmission ID entry is used to identify which message of a TX FIFO or TX Queue has been successfully transmitted because the TX FIFO or TX Queue number alone is not sufficient.

Therefore, a unique number can be attached to each transmission message stored in a TX FIFO or TX Queue. This unique identification number should be written to the CFDCFFDCSTSn.CFPTR[15:0] part of the Common FIFO Access Pointer Register for a TX FIFO or to the CFDTMFDCTRn.TMPTR[15:0] part of the TX Message Buffer Pointer Register of the TX Queue access window message buffer.

When the message is successfully transmitted, the identification number is then stored together with the other message related information to the TX History List and can be read with the Transmission ID (TID) of the TX History List Access Register.

Also for normal TX message buffers, the CFDTMFDCTRn.TMPTR[15:0] part of the TX Message Buffer Pointer Register is stored in the Transmission History List. Information label is the same.

Figure 35.49 shows a transmission preparation flow when TX History List is used.

Read access to the TX History List Access Register is done for every single entry.

After reading one entry, 0xFF must be written to the corresponding TX History List Pointer Control Register to be able to access the next entry until TX History List is empty.

Figure 35.50 shows an example flow for processing the TX History List information.

The TX History Lists have dedicated interrupts, which can be configured with the CFDTHLCCn.THLIM bit of the corresponding TX History List Configuration/Control Registers and enabled with the CFDTHLCCn.THLIE bit of the same registers, either to generate an interrupt when the history list reached a filling level of 75% or for every new TX History List entry.

An entry lost indication is flagged by the CFDTHLSTSn.THLELT bit in the TX History List Status Register.

Status of this bit is also shown by the THLES bit in the Global Error Flag Register.

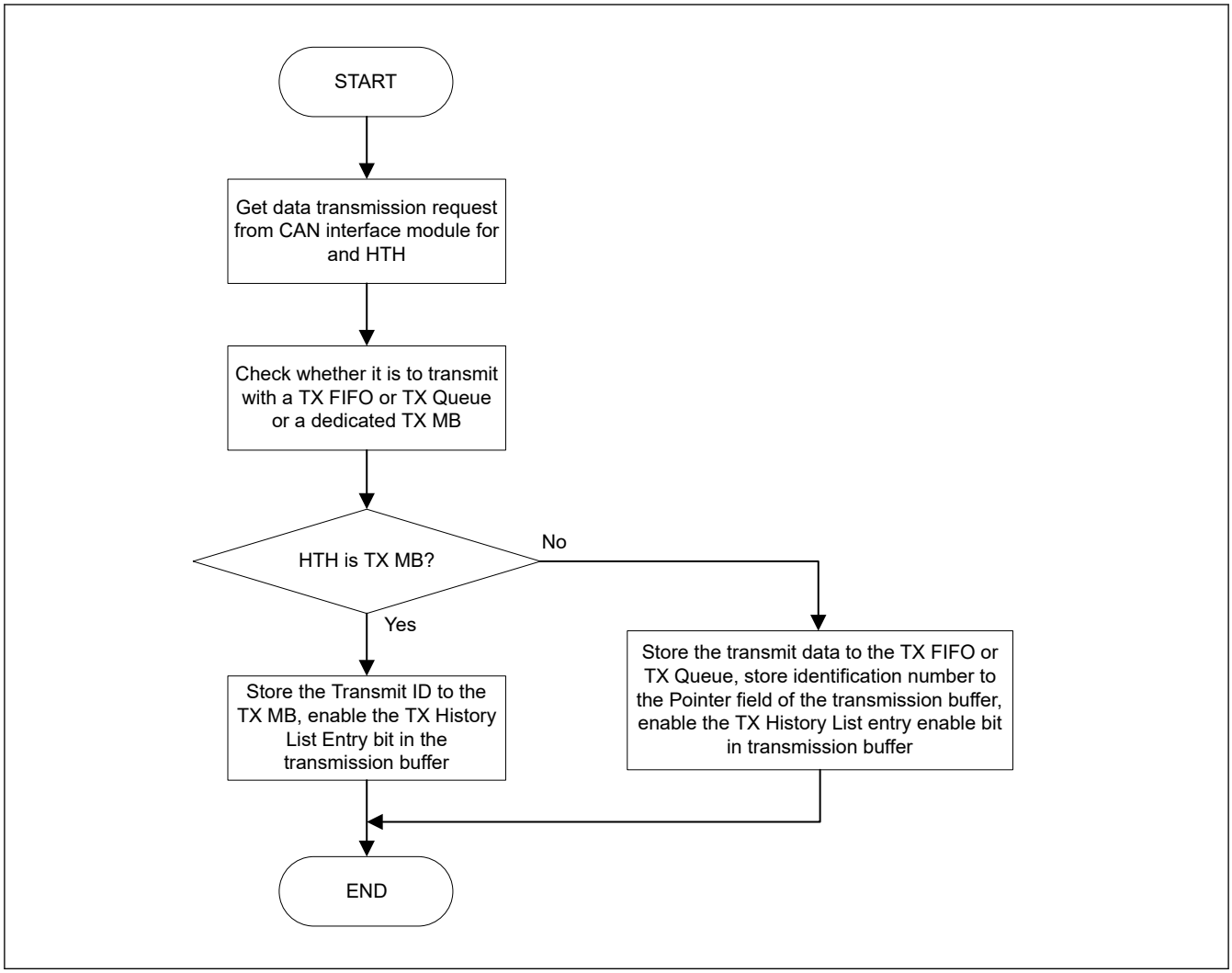


Figure 35.49 Transmission preparation flow when TX History List is used

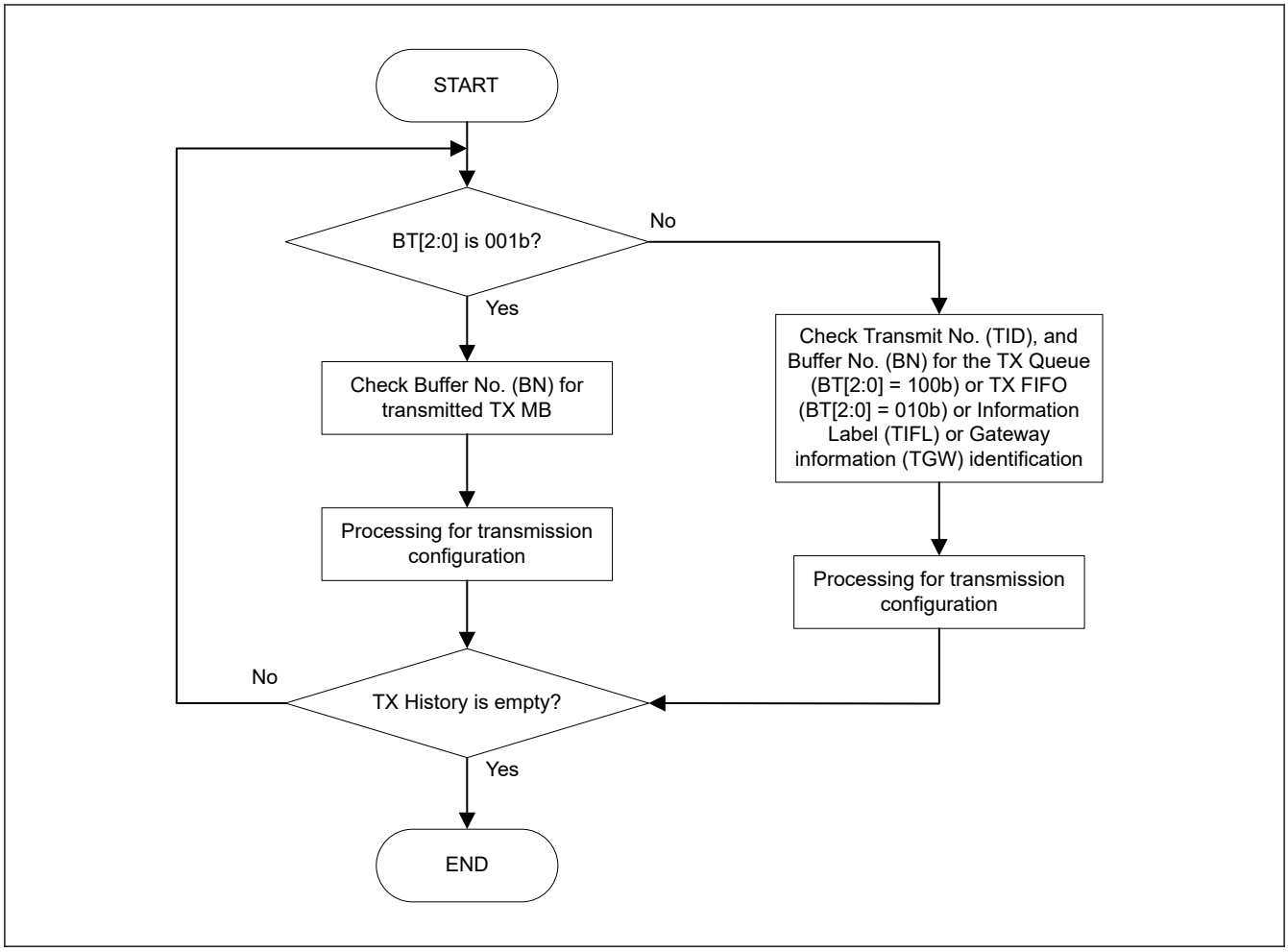


Figure 35.50 Example flow for processing TX History List information

35.9.2.6 TX Data Padding

If the data length code (DLC) of the transmitting message has a higher number of data bytes than the buffer size, then the data bytes beyond the restricted range are replaced by bytes with the value of CC HEX.

This can happen for Common FIFOs configured as (TX or GW) when the transmit message DLC is higher than CFDCFCCn.CFPLS.

This can also happen in FD-only mode, if a Classical Frame is configured with a DLC larger than 8.

35.10 Test Mode

The CAN-FD module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CAN-FD module in test modes.

All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combinations of the various test modes specified in this section.

The test modes can be broadly split into 2 groups:

- Channel specific test modes
- Global test modes

35.10.1 Channel Specific Test Modes

Each CAN channel can be configured into the following test modes:

- Basic test mode

- Listen-only mode
- Self-test mode 0 (External Loop Back mode)
- Self-test mode 1 (Internal Loop Back mode)
- Restricted operation mode

35.10.1.1 Basic Test Mode

The Basic test mode should be used when a particular test setting must be enabled other than when in Listen-Only and Self-test modes.

35.10.1.2 Listen-Only Mode

The ISO 11898-1 recommends an optional Bus-monitoring mode. In this mode, the CAN channel is able to receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit.

If the CAN engine is required to send a dominant bit (ACK bit, Overload flag, Active Error flag), the bit is routed internally so that the CAN engine monitors this as dominant. The external TX pin remains in recessive state.

This mode can be used for baud rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.

In this mode, it is not permitted to request transmission from any normal TX Message Buffer or TX-/GW-FIFO of this channel.

Note: If a message is stored in GW FIFO or Routing TXQ, ensure that the transmitting channel is not in Listen-only mode so that transmission is not requested for this channel from the GW FIFO or Routing TXQ.

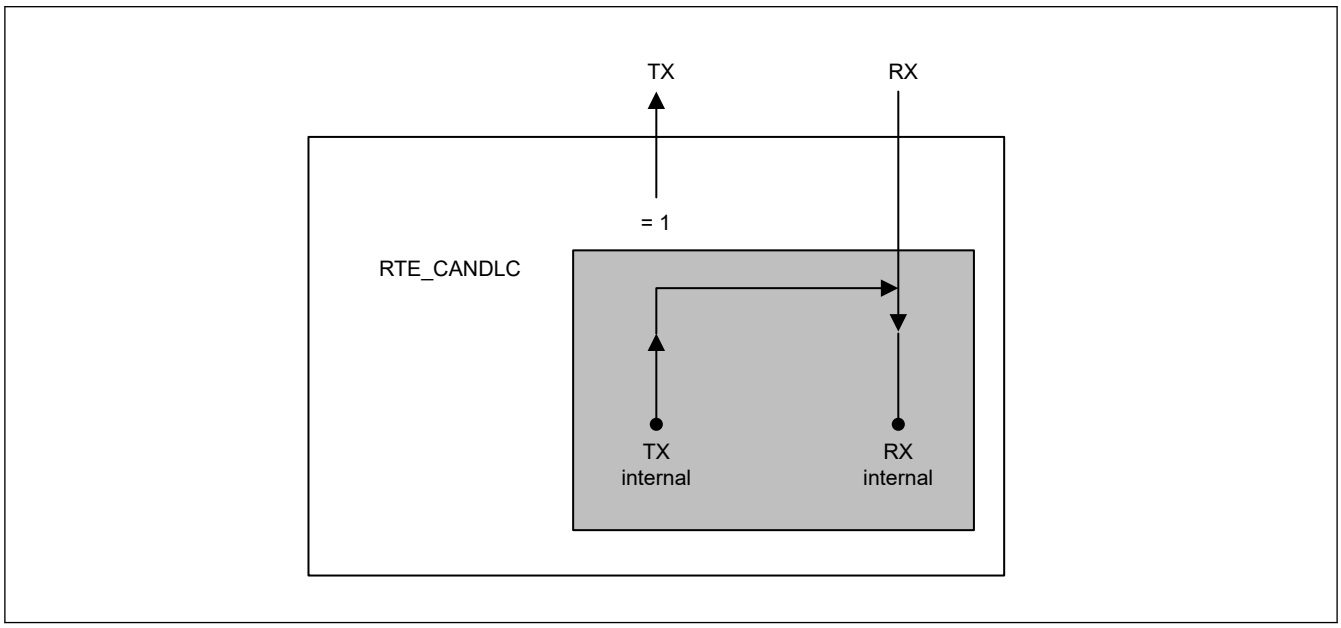


Figure 35.51 Listen-only mode

35.10.1.3 Self-Test Mode 0 (External Loop Back Mode)

In Self-test mode 0, the CAN engine treats its own transmitted messages as received messages through the CAN transceiver and can store them into its receive message buffers.

To be independent from external stimulation, the engine generates its own Acknowledge bit.

This test can be used for CAN transceiver tests.

The RX/TX pins should be connected to the transceiver.

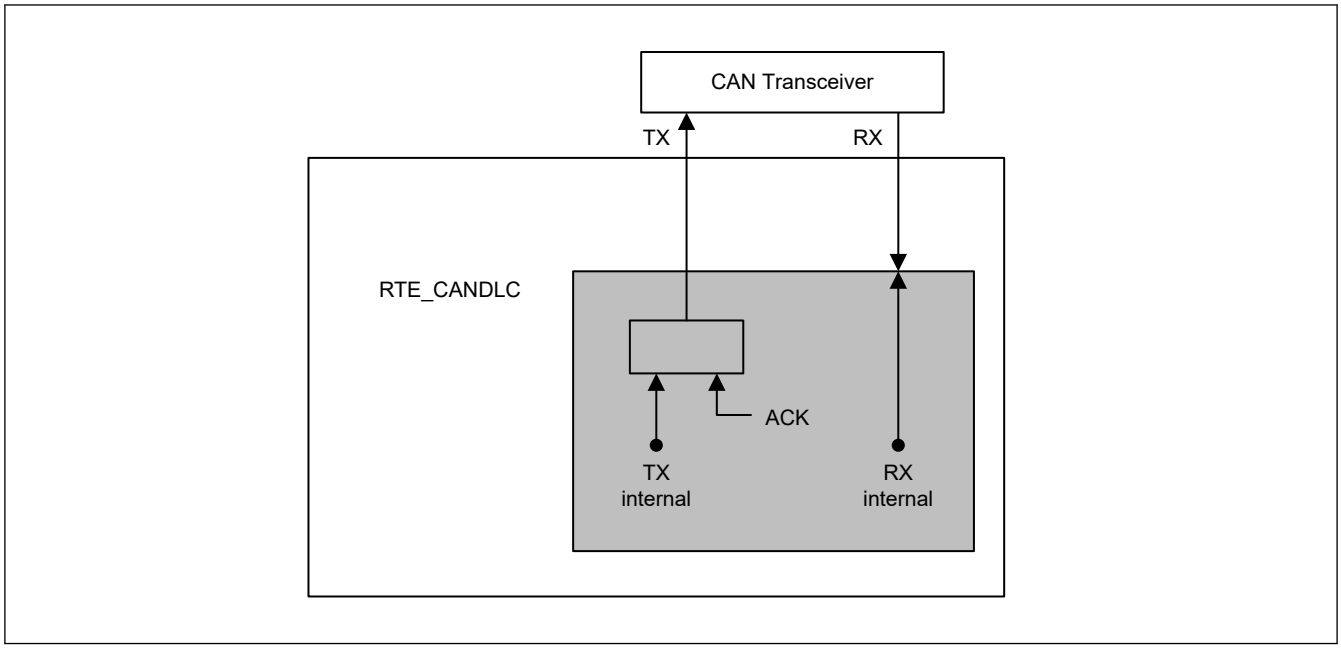


Figure 35.52 Self-test mode 0 (External Loop Back mode)

35.10.1.4 Self-Test Mode 1 (Internal Loop Back Mode)

In Self-test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent from external stimulation, the CAN engine generates its own Acknowledge bit. In this mode the CAN engine performs an internal feedback from TX internal to RX internal. The actual value of the external RX input is disregarded by the CAN engine.

The external TX pin outputs only recessive bits.

The RX/TX pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the internal CAN bus communication line.

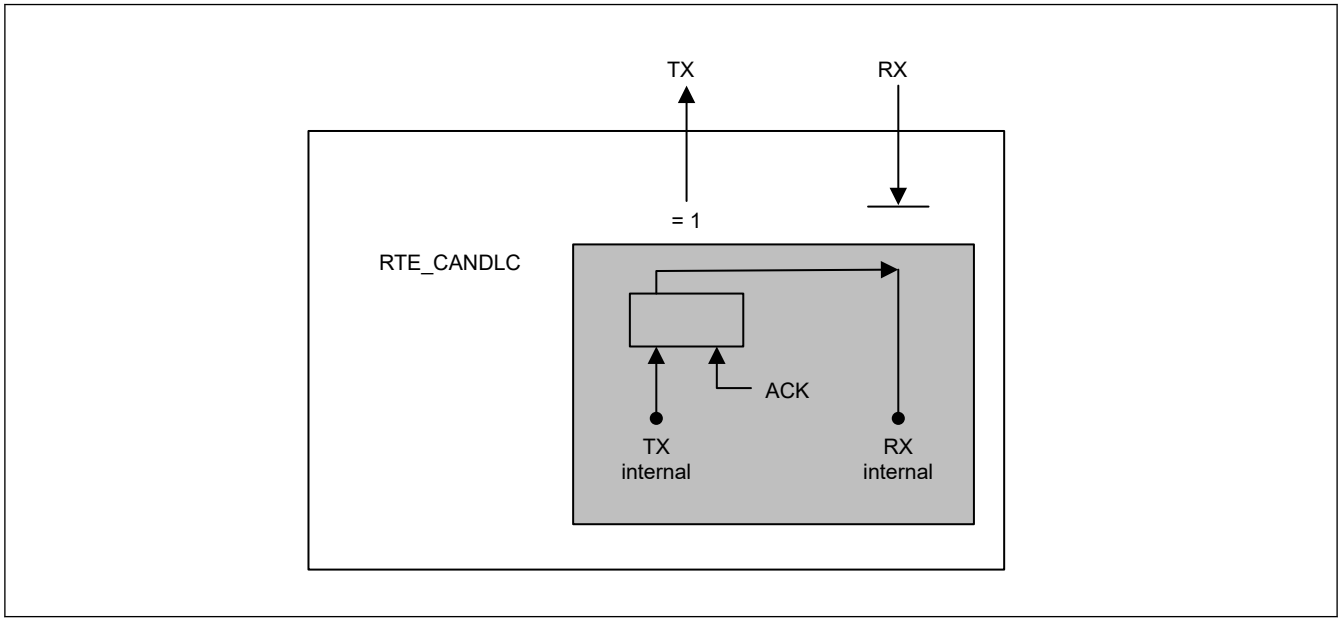


Figure 35.53 Self-test mode 1 (Internal Loop Back mode)

35.10.1.5 Restricted Operation Mode

In Restricted operation mode, the CAN node is able to receive valid data and remote frames that generates the Acknowledge bit.

Active error and overload frames cannot be transmitted, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.

Additionally, the Receive and Transmit Error Counter (REC and TEC) are frozen independently from the occurrence of errors.

The mode is specified as in ISO 11898-1. However, it is permitted to set any requested transmit.

35.10.2 Global Test Modes

The CAN-FD module can be configured into the following test modes:

- RAM test mode
- Internal CAN bus communication mode
- CRC error test

These test modes are protected by a special software procedure to enable the mode. This software procedure enables write access to the test mode by a specific unlock key, the related unlock key can be seen in the following table.

Test mode	Unlock key 1	Unlock key 2
RAM test mode	0x7575	0x8A8A

If the software sequence of the two consecutive unlock key write accesses (half-word or word accesses) is interrupted by any other write access to the SFR or if incorrect data is written to the Global Unlock Key Register then the corresponding test mode cannot be set and the sequence must be restarted.

After the two unlock key write accesses, the next write access should be to set the corresponding Test Mode Enable bit. If this is not followed, the unlock mechanism resets and the Test Mode Enable bit cannot be set. At this time, the unlock sequence must be restarted.

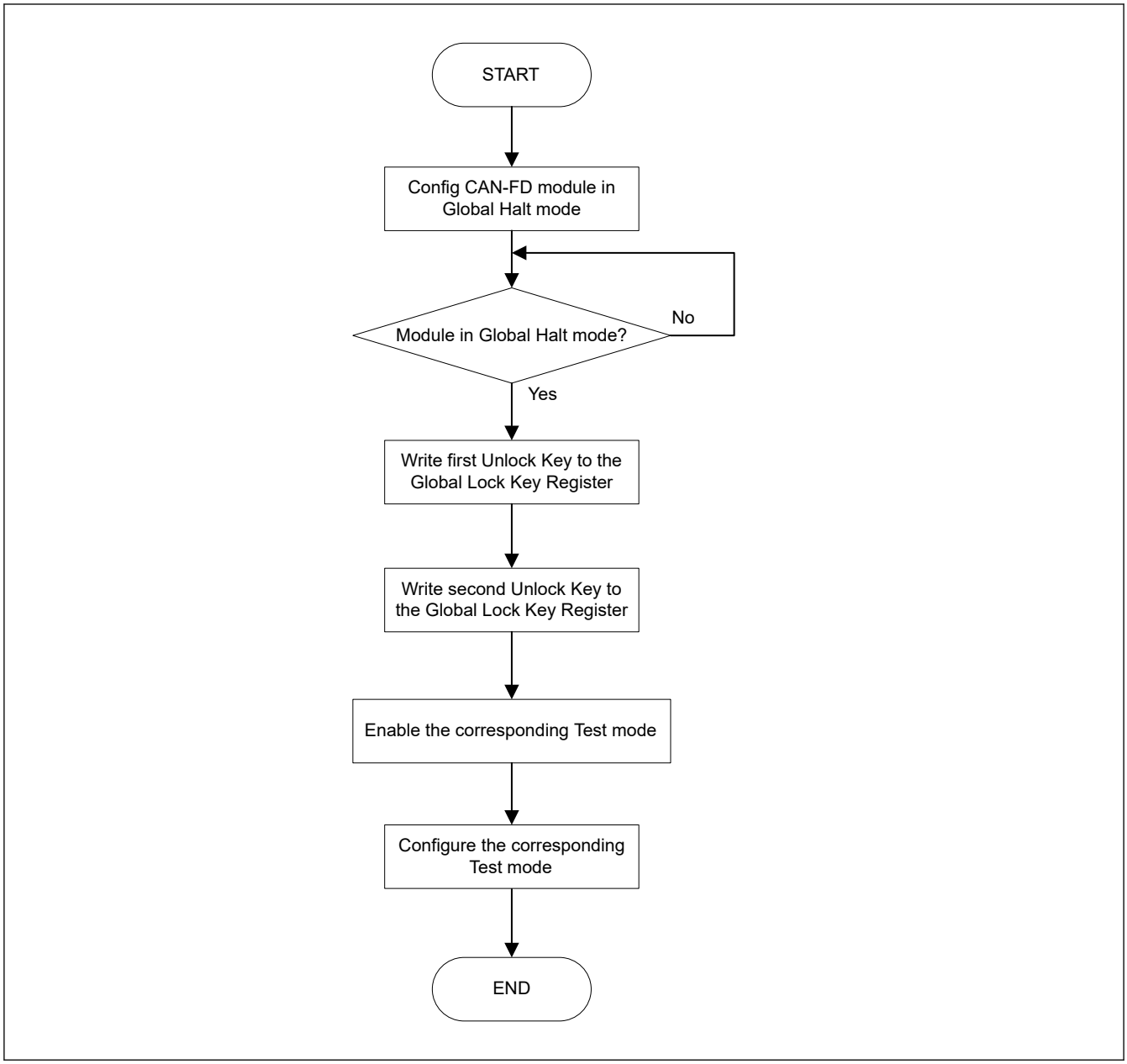


Figure 35.54 Unlock software protection routine

35.10.2.1 RAM Test Mode

The CAN-FD module can be configured in RAM test mode by setting the CFDGTSTCTR.RTME bit in the Global Test Control Register when the corresponding lock key is previously written. This is a special test mode, in which, the complete RAM area can be accessed.

In this mode, the RAM area is split into number of pages (pn) of 256 bytes each. Which can be accessed with the CFDRPGACCn register.

The page should be selected for read/write access by writing to the CFDGTSTCFG.RTMPS[9:0] bits in the Global Test Control Register. Data can then be read from or written into the RAM Test Page Access Registers.

Figure 35.55 shows the structure of the pages in the RAM when performing a RAM test mode.

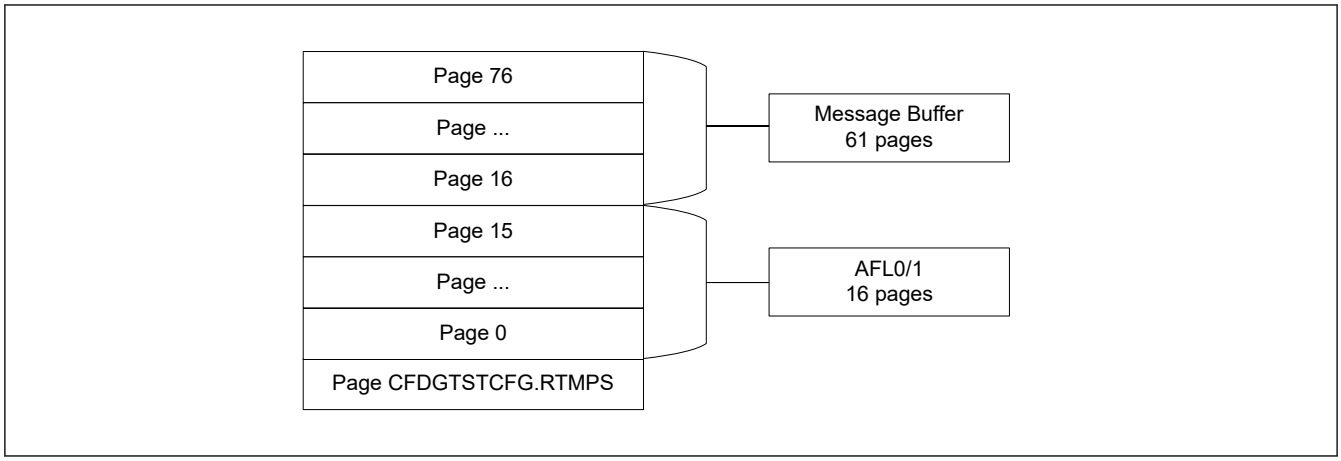


Figure 35.55 RAM page structure

The total available RAM size for a 2-CAN channel version is 4096 bytes for the AFL RAM and 15424 bytes for the Message Buffer RAM.

AFL RAM0/1 can treat RAM test mode as one RAM.

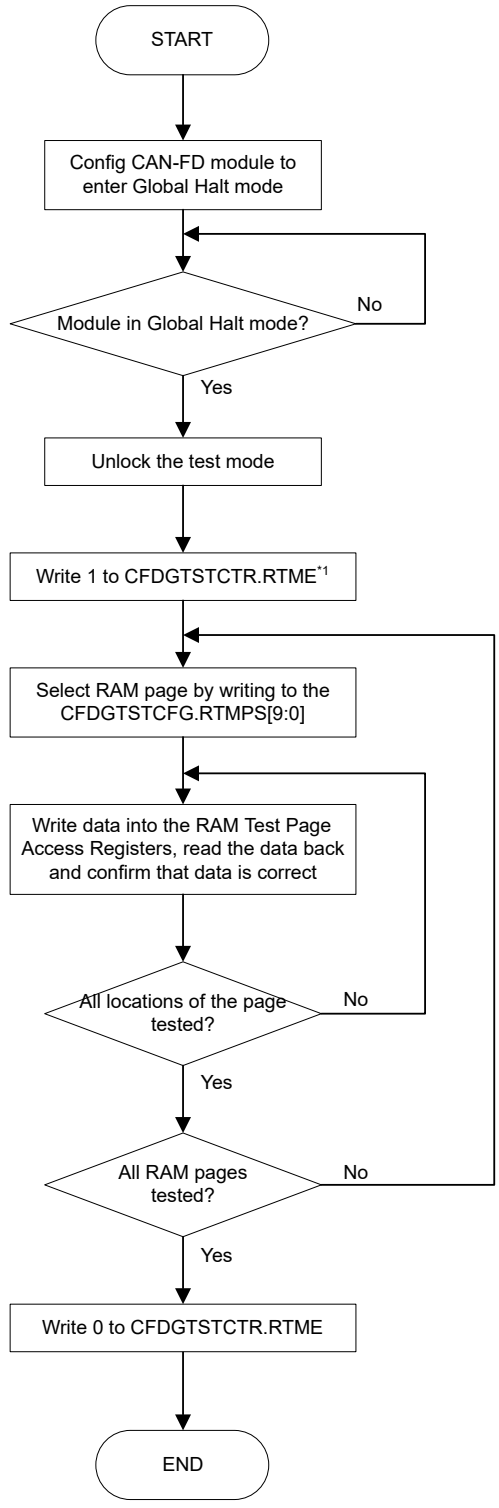
The number of pages (pn) and CFDGTSTCFG.RTMPS[9:0] values for the AFL and MB RAMs are calculated in the following way:

pn = ceil (total RAM size in bytes / number of bytes per page)

- AFL RAM:
  pn = ceil (4096 / 256) = 16 pages
  CFDGTSTCFG.RTMPS[9:0] = 0 to 15 (0x00F) inclusive
- MB RAM:
  pn = ceil (15424 / 256) = 61 pages
  CFDGTSTCFG.RTMPS[9:0] = 16 to 76 (0x04C) inclusive

Do not access more than 64 bytes of RAM on the last page (RTMPS[9:0] = 0x04C).

Figure 35.56 shows the software flow for RAM test mode.



Note 1: Change into the following status before changing to RAMTEST.

- Cancel of a request of transmission
- Disable of all the FIFO and TXQ
- Clear of the receiving flag of a receiving buffer

Figure 35.56 Software flow of RAM test mode

To exit this test mode, clear the CFDGTSTCTR.RTME bit. The CFDGTSTCTR.RTME bit is cleared by writing 0 to it.

The CFDGTSTCTR.RTME bit is cleared automatically when the CAN-FD module enters Global Reset mode from the test mode.

35.10.2.2 Internal CAN Bus Communication Mode

The CAN-FD module can be configured in Internal CAN bus communication test mode by setting the CFDGTSTCTR.ICBCTME bit in the Global Test Control Register. This is a special test mode, in which the CAN channels can be connected together internally to generate a CAN cluster within the CAN-FD module.

Only use the following sequence to enter Internal CAN bus communication test mode:

1. Configure all channels in Halt mode and check that all channels have entered Global Halt mode.
2. Write data into the Global Test Configuration Register to select the channels participating in the Internal CAN bus communication test.
3. Set the CFDGTSTCTR.ICBCTME bit of the Global Test Control Register.
4. Check that CFDGTSTCTR.ICBCTME bit is set in the Global Test Control Register.

In this mode, the TXD outputs of the channels participating (configured) in Internal CAN bus communication mode are connected together using AND gate. The output of the AND gate is connected to the RXD inputs of all participating channels to create a CAN cluster within the CAN-FD module. The channels are isolated from the external CAN bus while the CAN-FD module is in this test mode.

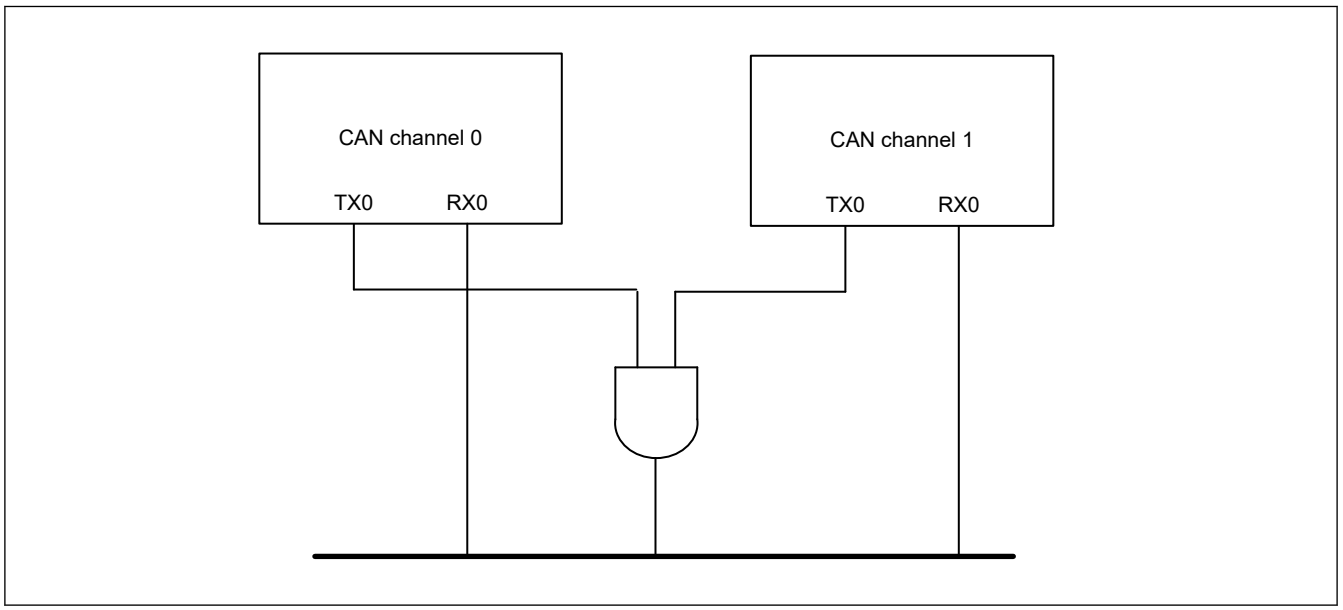


Figure 35.57 Internal CAN bus connections

The AFL, Flat RX message buffers, FIFO buffers, Flat TX message buffers, and various registers can now be configured as normal to start communication between channels.

The channels not participating in Internal CAN bus should only be configured in Halt mode.

35.10.2.3 CRC Error Test

After the CAN-FD module has been configured in Internal CAN bus communication test mode, use the following sequence to perform CRC error testing. In the following sequence, channel x is the reference transmitter CAN-FD module and channel y is the receiver CAN-FD module where (x, y = [0,…, n] and $x \neq y$):

1. Configure channel x node to transmit one reference message.
2. Set the CFDCyCTR.CRCT bit to 1, in order to invert the first bit of the incoming bit stream from channel x.
3. Set the CFDTMCx.TMTR.
4. Read either CFDCyERFL.CRCREG or CFDCyFDCRC.CRCREG (depending on the received frame type: Classical or FD). The value should be different from the received CRC value of the reference message from channel x.

5. Check that CFDCyERFL.CERR is 1.

As the CRC generator logic is shared for RX and TX there is no need to create a separate TX CRC error test.

35.11 Bus Traffic Measurement

The idle time of the CAN bus can be measured using PCLKM or PCLKCAN. Bus traffic can be calculated based on the measurement results.

35.11.1 How to count the CAN Bus Idle Time

Figure 35.58 shows the concept of measuring the idle time of the CAN bus.

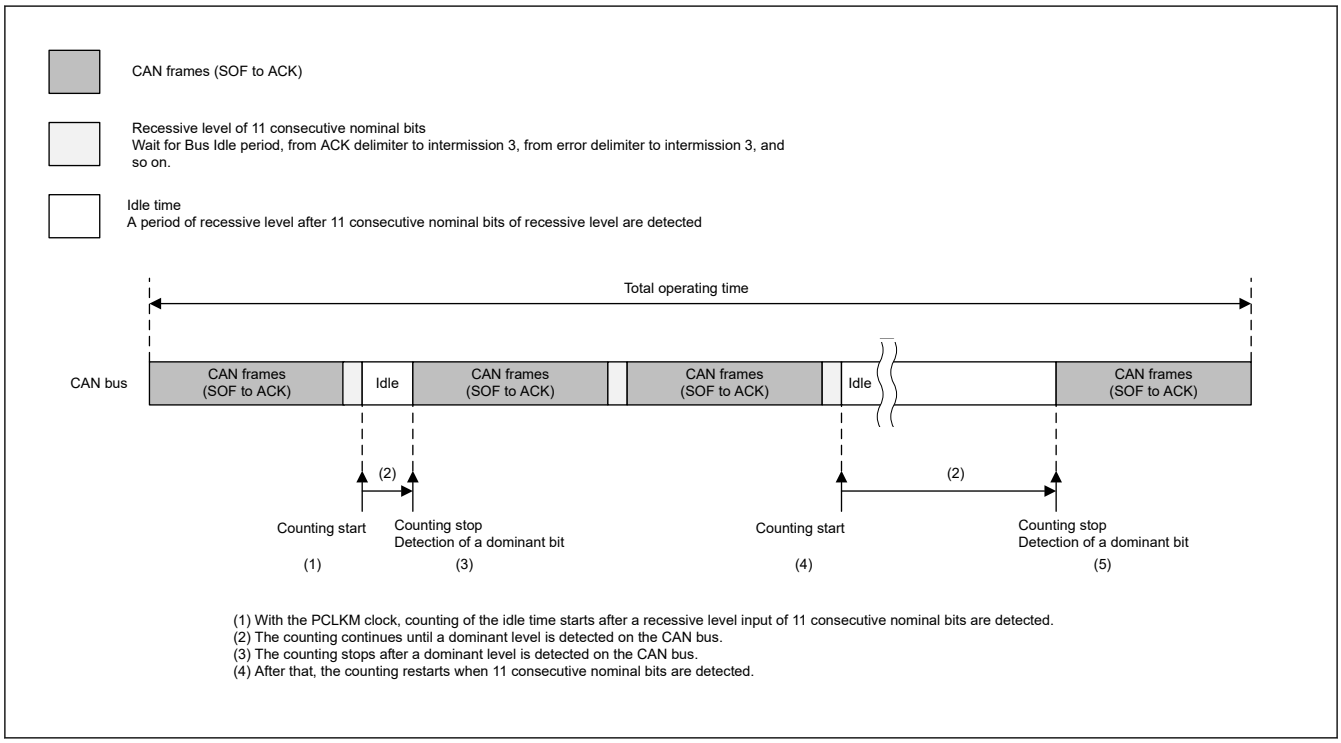


Figure 35.58 How to measure idle time of the CAN bus

35.11.2 Operations and Measurement Procedure

The following procedure is for measuring the idle time of the CAN bus.

1. The channel to be measured transits to operation mode.
2. Write 1 to CFDCnBLCT.BLCE bit to set the measuring counter to operating mode.
3. Write 1 to CFDCnBLCT.BLCLD bit to clear the counter register.
4. Detect a recessive level input of 11 consecutive nominal bits.
5. Start counting the bus idle time.
6. Detect a dominant level.
7. The counting stops.
8. Detect a recessive level input of 11 consecutive nominal bits.
9. The counting starts.
10. Write 1 to CFDCnBLCT.BLCLD bit to clear the counter register and simultaneously load the counter value to CFDCnBLSTS.
11. Read the value of CFDCnBLSTS.

To stop the measurement counter, write 0 to CFDCnBLCT.BLCE bit.

To initialize the counter, write 1 to CFDCnBLCT.BLCLD bit.

This measurement is enabled when the channel to be measured is in Operation mode.

When the relevant channels are in reset mode, the counter does not operate.

Also, accurate measurements are not available in test mode.

Write 1 to CFDCnBLCT.BLCLD bit to clear the counter register and simultaneously load the value of the counter to CFDCnBLSTS.

The lower 3 bits of CFDCnBLSTS are fixed to 0.

Based on the values of the counter, software can calculate the CAN bus traffic according to the following formulas:

$$\frac{(\text{Total operating time} - \text{Total idle time})}{\text{Total operating time}} = \frac{\text{Bus operating time}}{\text{Total operating time}} = \text{Bus usage ratio}$$

- Total idle time: a value read from CFDCnBLSTS × a clock cycle of PCLKM
- Total operating time: a setting interval of CFDCnBLCT.BLCLD bit

Example: Below is a calculation example under the following conditions:

- Conditions: nominal bit rate = 1 Mbps
- PCLKCAN clock = 40 MHz (= 25 ns)
- A setting interval of CFDCnBLCT.BLCLD bit = cycle of 1 ms
- A read value of CFDCnBLSTS register = 0x4E20

$$\frac{(\text{Total operating time} - \text{Total idle time})}{\text{Total operating time}} = \frac{(1000000\ \text{ns} - 20000 \times 25\ \text{ns})}{1000000\ \text{ns}} = 50\%$$

35.12 Flexible CAN Mode

This is a mode in which it is possible to connect the CAN modules of 2 channels to a single CAN driver.

The pair of Channel in flexible CAN mode is as follows:

When CFDGFCMC.FLXC0 bit is set, Channel 0 and Channel 1 of a CAN-FD module are Flexible CAN mode.

The Channel 1 uses TX/RX terminal of Channel 0, and the TX/RX terminal of Channel 1 cannot use.

In Flexible CAN mode, each channel performs communication processing independently. However, when one of the channels transmits, the other channel will not return an acknowledge bit.

Note: When operating in Flexible CAN mode, the error counters (TEC/REC) of the two CAN nodes are not synchronized with each other.

Flexible transmission buffer assignment configured in CFDGFTBAC register and Flexible CAN mode configured in CFDGFCMC register should not be used simultaneously.

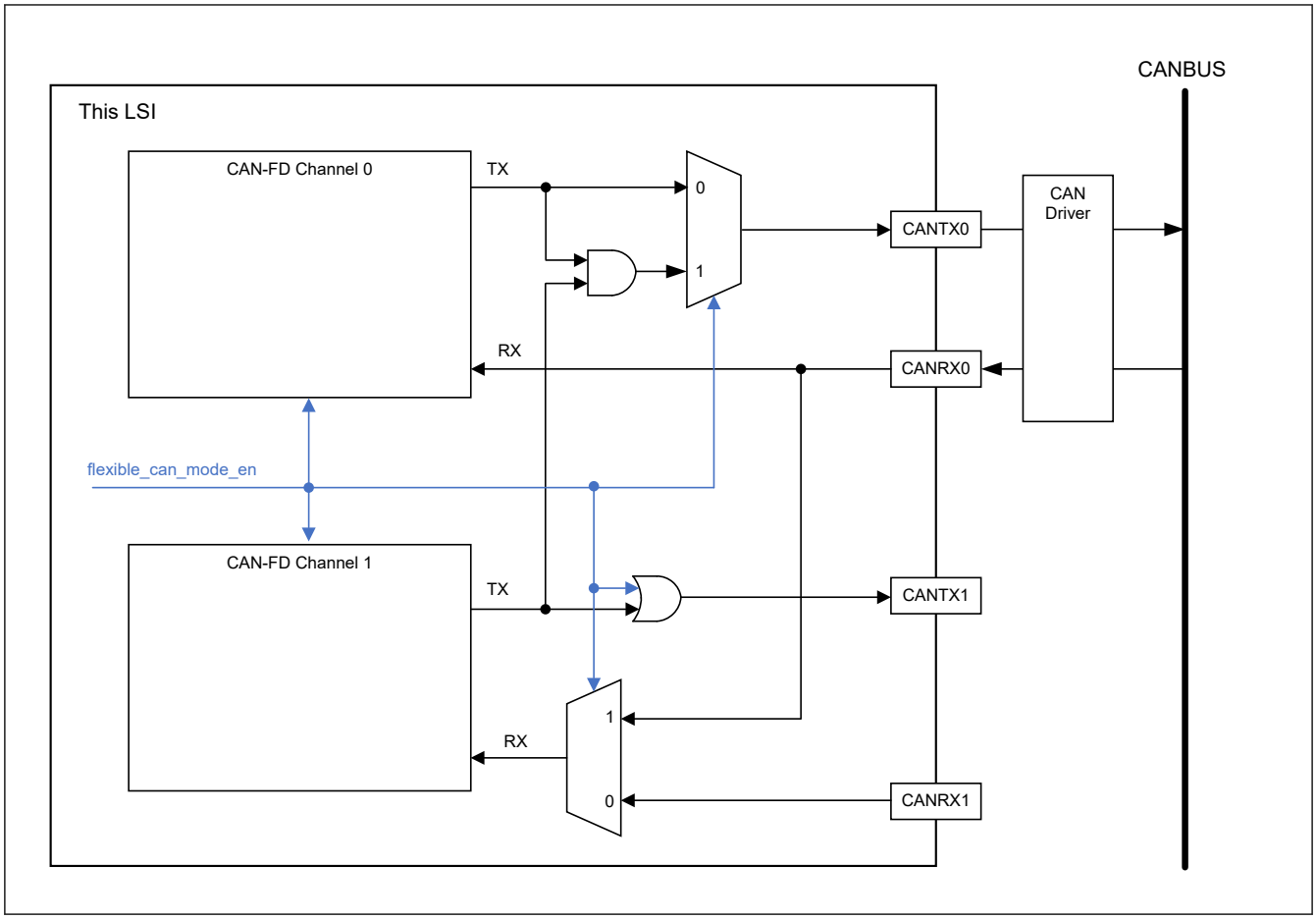


Figure 35.59 Diagram of the Flexible CAN

35.13 Flexible Transmission Buffer Assignment

Each Channel has 64 transmit buffers by exclusive use.

In order to correspond to an insufficient case by a TXMB, this Channel can rent a maximum of 32 TXBMs from the next Channel.

The buffer which can be rented becomes every four units by 4 to 32 buffers.

When CFDGFTBAC.FLXMB0 bit is set, flexible transmission buffer assignment is performed between Channel 0 and Channel 1.

Flexible transmission buffer assignment configured in CFDGFTBAC register and Flexible CAN mode configured in CFDGFCMC register should not be used simultaneously.

Interrupt of the rented buffer is output to interrupt of the rented channel.

When using TXQ by the rented buffer, TXQ should only set within the rented range.

The TXMB lent out operates in the mode of the channel to be used. For example, when the channel 1 is in a reset mode, using rented TXMB and transmission of the channel 0 is possible. Moreover, the rented TXMB is affected by the transmitting status, TX SCAN PROCESS, or transmitting abortion of the rented channel.

In the case of message buffer number priority mode, the priority of TXMB0 is high and the priority of TXMB95 (case which rented 32 buffers) becomes low. Priority of rented TXMB becomes low.

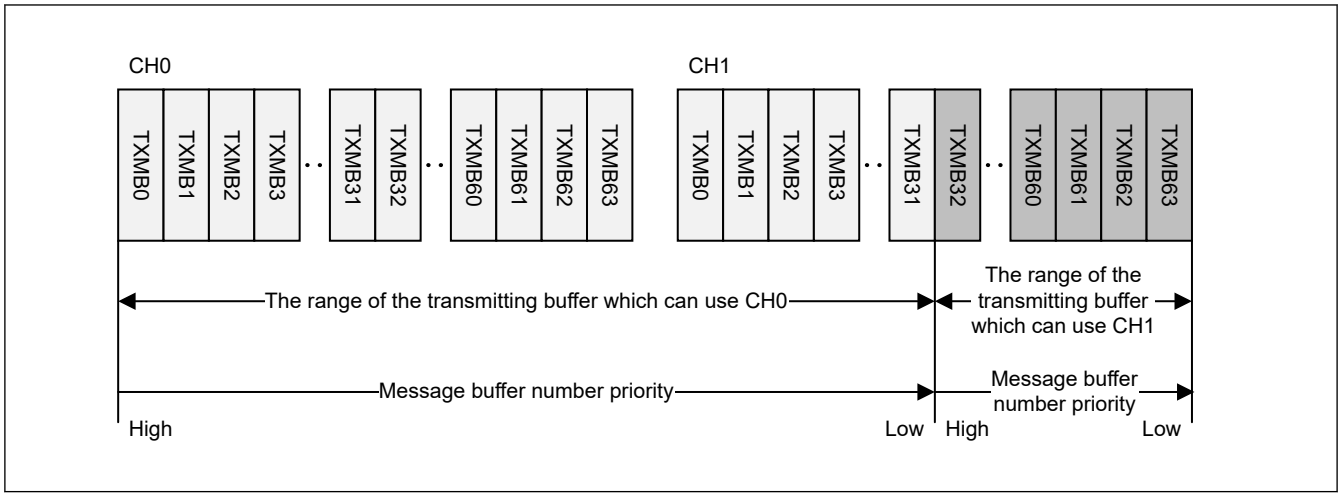


Figure 35.60 The priority of message buffer number priority mode (Example of CH0: 96 buffers, CH1: 32 buffers)

The example of a rental between channel 0 and channel 1 is shown below.

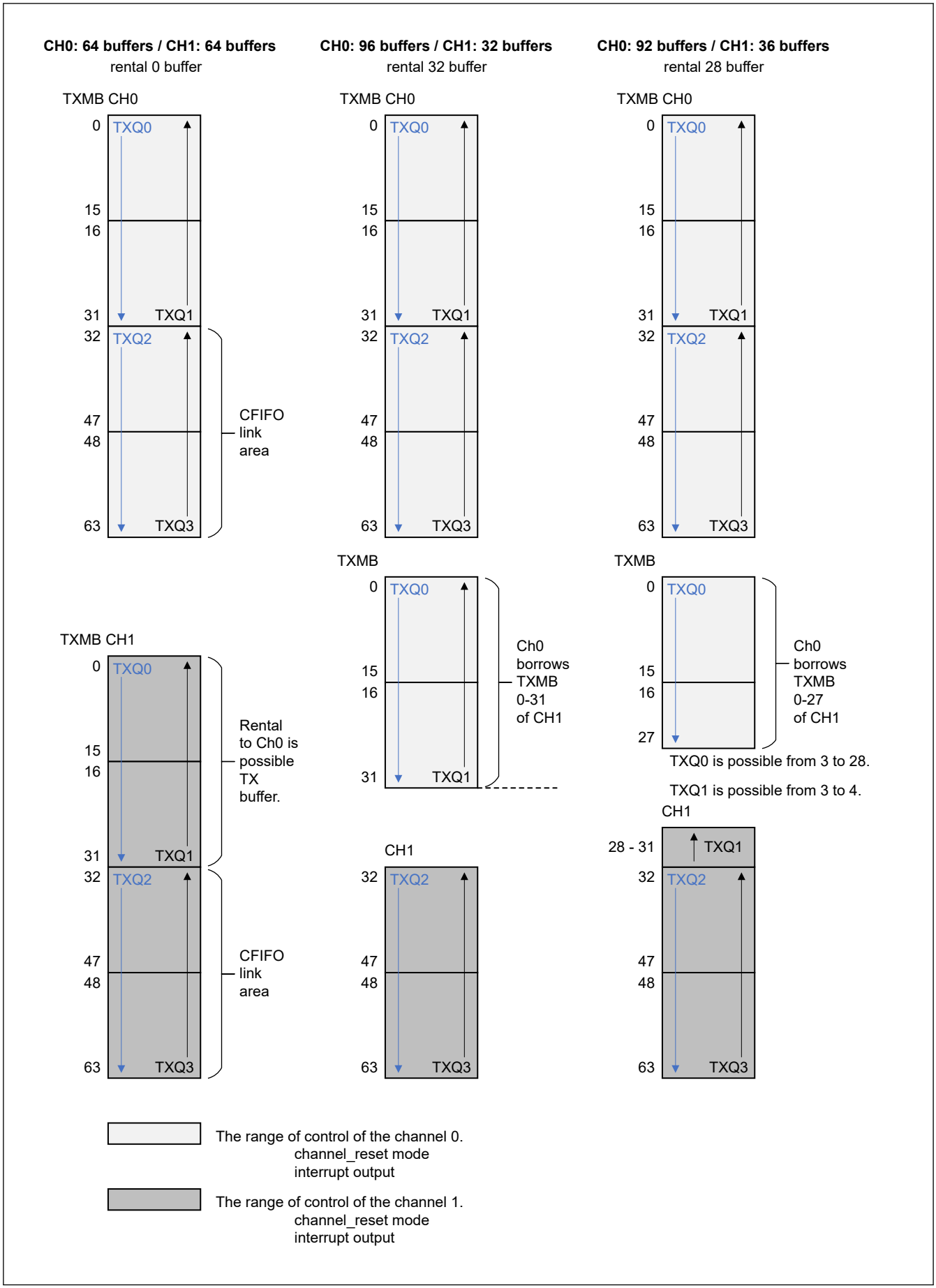


Figure 35.61 Flexible transmission buffer assignment (0, 32, 28 TXMB rental)

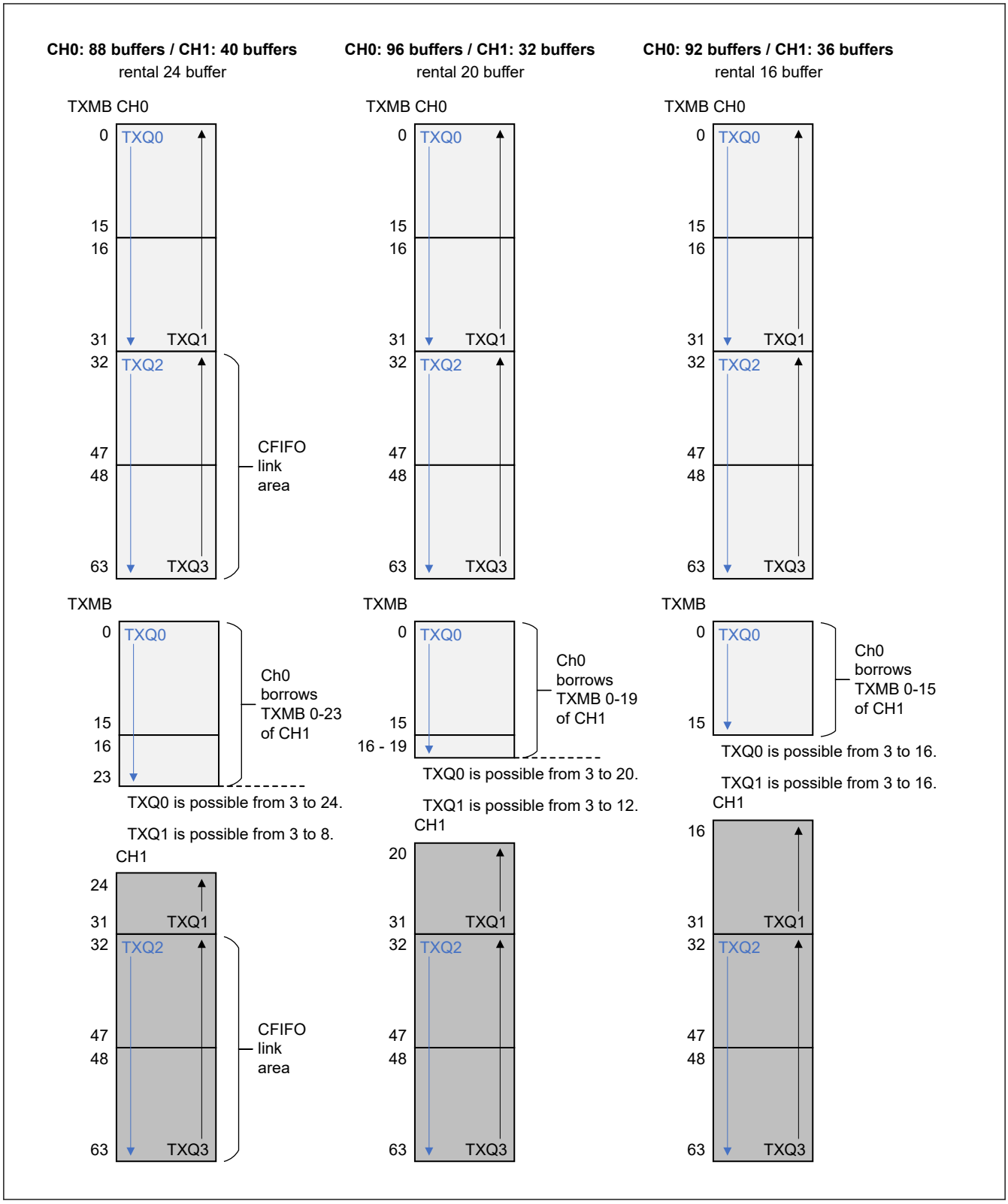


Figure 35.62 Flexible transmission buffer assignment (24, 20, 16 TXMB rental)

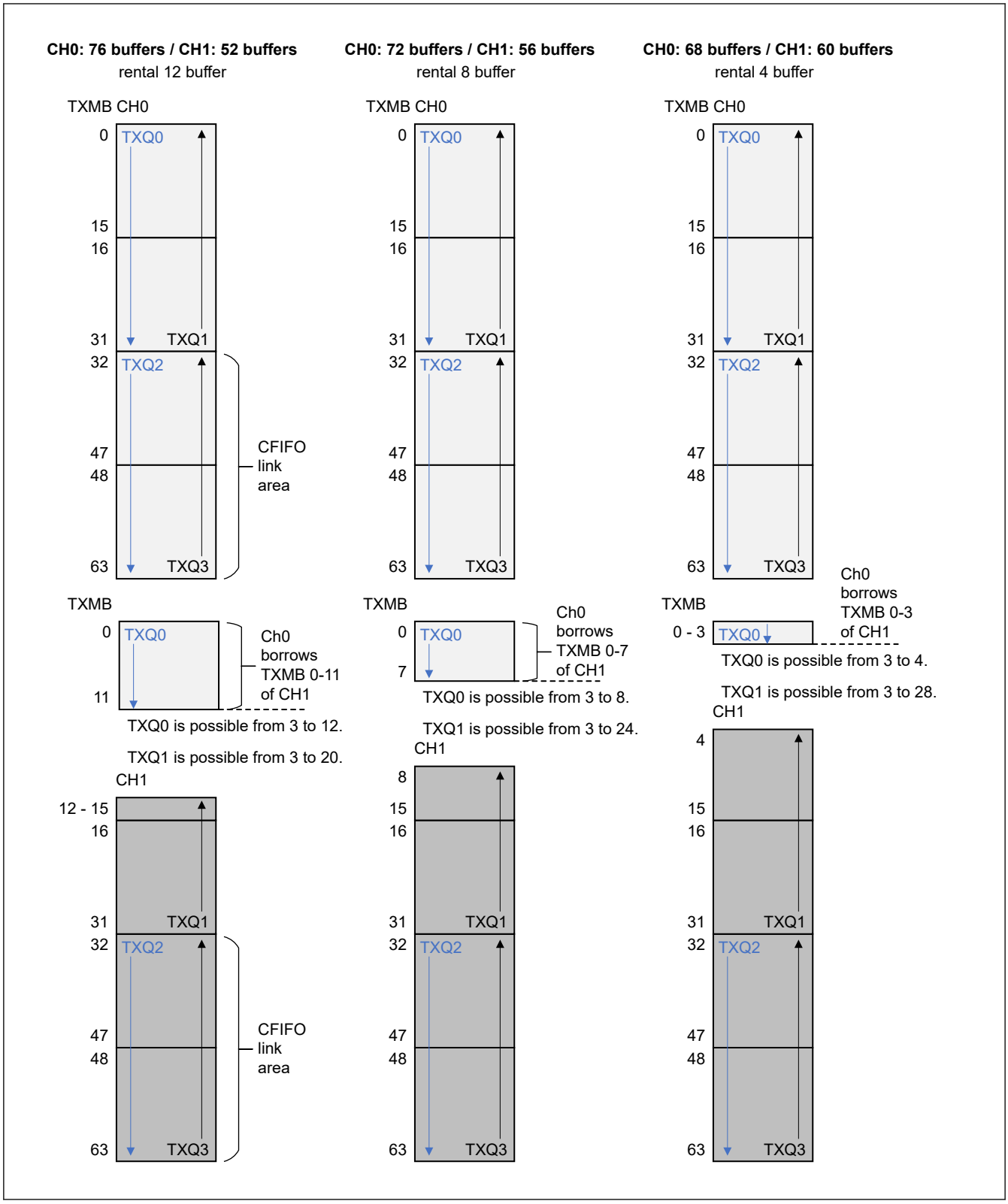


Figure 35.63 Flexible transmission buffer assignment (12, 8, 4 TXMB rental)

36. Serial Peripheral Interface (SPI)

36.1 Overview

The SPI supports serial communications for transmit-receive, transmit only, and receive only. The SPI is provided with a function for high-speed serial communication with multiple processors and peripheral devices.

36.1.1 Features

Table 36.1 lists the specifications of the SPI, and Figure 36.1 shows a block diagram of the SPI.

Table 36.1 SPI specifications (1 of 2)

Item	Description
Number of channels	4 channels (ch0 to ch2 in NON-SAFETY and ch3 in SAFETY)
Transfer functions	• SPI serial communication (4-wire) and clock synchronous (3-wire) serial communication are possible by using the MOSI (Master Out Slave In), MISO (Master In Slave Out), SSL (Slave Select), and RSPCK (SPI Clock) signals. • Transmit-only operation is available • Receive-only operation is available • Serial communication is possible in master mode and slave mode • RSPCK polarity switching • RSPCK phase switching
Data format	• MSB first or LSB first selectable. • Transfer bit length selectable from 4 to 32 bits • 32 bit x 4 stage FIFO transmit and receive buffers • Up to four frames transferable in one round of transmission or reception (each frame consisting of up to 32 bits) • Byte swap operating function • Transmit/receive data inversion
Bit rate	• In master mode, the on-chip baud rate generator divides the PCLKSPIn to generate RSPCK. A division ratio of 2 to 4096 is settable. • In slave mode, an external input clock is used as a serial clock. The maximum frequency = PCLKSPIn/2 (High width: 1 PCLKSPIn cycles, low width: 1 PCLKSPIn cycles)
Buffer configuration	• Transmit buffer and receive buffer are configured independently.
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection • Parity error detection

Table 36.1 SPI specifications (2 of 2)

Item	Description
SSL control function	[Motorola SPI mode] ● Four SSL pins (SSLn0 to SSLn3) each channel ● In single master mode, SSLn0 to SSLn3 pins are output. ● In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused ● In slave mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins unused ● Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Configurable delay between frames in burst transfer ● Function for changing SSL polarity [TI SSP mode] ● Four SSL pins (SSL0 to SSL3) each channel ● In single master mode, SSLn0 to SSLn3 pins are output. ● In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused ● In slave mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins unused ● Controllable delay from SSL output assertion to SSL output negation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Controllable delay from RSPCK stop to Output disable (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Configurable delay between frames in burst transfer ● Function for changing SSL polarity
Communication Protocol	● Motorola SPI ● TI SSP (Synchronous Serial Protocol)
Synchronization bypass function	● Synchronization circuit can be bypassed using bus clock (PCLKM) as operation clock (PCLKSPIn)
Control in master transfer	[Motorola SPI mode] ● Transfers of up to eight commands each can be executed sequentially in looped execution ● For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, LSB first/MSB first, burst, RSPCK delay, SSL negation delay, next-access delay ● Transfers can be initiated by writing to the transmit buffer ● MOSI signal value specifiable in SSL negation ● RSPCK auto-stop function [TI SSP mode] ● Transfers of up to eight commands each can be executed sequentially in looped execution ● For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, LSB first/MSB first, burst, RSPCK delay, SSL negation delay (Output disable delay), next-access delay ● Transfers can be initiated by writing to the transmit buffer ● MOSI signal value specifiable in SSL negation ● RSPCK auto-stop function
Interrupt sources	● Maskable Interrupt sources SPI receive buffer full / Receive data ready interrupt SPI transmit buffer empty interrupt SPI communication end interrupt SPI error interrupt (mode fault, overrun, parity error, receive data ready) SPI idle interrupt (SPI idle)
Others	● Switching between CMOS output and open-drain output ● SPI disable (initialization) function ● Loopback mode function
Module-stop function	Module-stop state can be set to reduce power consumption*1

Note 1. Module stop condition: It depends on the setting higher than this module. This module is not reset, and the clock stops on module stop condition. It contributes to low power consumption, because the clock stops.

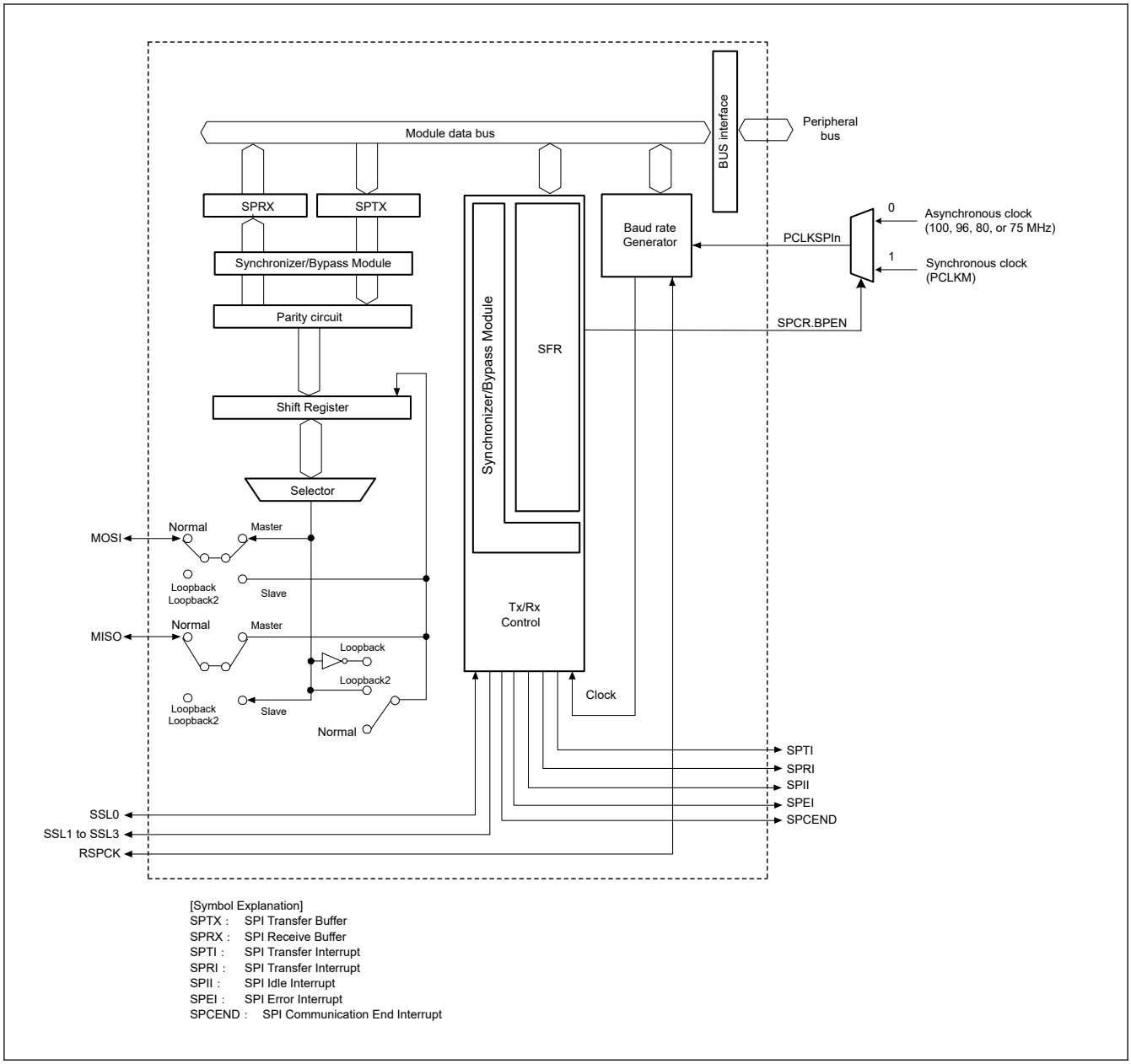


Figure 36.1 SPI block diagram

The SPI automatically switches the SSL0 pin input and output directions. The SSL0 pin is switched to output when single master is set, and it is switched to input when multi-master or slave is set. Furthermore, the SPI automatically switches the input and output directions of pins RSPCK, MOSI, MISO, SSL1 to SSL3 according to the master/slave setting and SPI operation (4-wire)/clock synchronous operation (3-wire) SSL0 input level. (See section 36.4.2. SPI Pin Control.)

Table 36.2 lists the I/O pins used in the SPI.

Table 36.2 SPI pin configuration

Channel	Pin name	I/O	Function
SPIn (n = 0 to 3)	SPI_RSPCKn	I/O	Clock I/O
	SPI_MOSIn	I/O	Master transmit data I/O
	SPI_MISOn	I/O	Slave transmit data I/O
	SPI_SSLn0	I/O	Slave selection signal I/O
	SPI_SSLn1	Output	Slave selection signal output
	SPI_SSLn2	Output	Slave selection signal output
	SPI_SSLn3	Output	Slave selection signal output

Note: For convention, pin names within this chapter are written without "SPI_" notation except this table.

36.2 Register Map

Table 36.3 SPI register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x8000_7000 + 0x0400 × n (n = 0 to 2) 0x8100_7000 (n = 3)	SPDR/SPDR_HA/ SPDR_BY	SPI Data Register	—
0x8000_7004 + 0x0400 × n (n = 0 to 2) 0x8100_7004 (n = 3)	SPCKD	SPI Clock Delay Register	—
0x8000_7005 + 0x0400 × n (n = 0 to 2) 0x8100_7005 (n = 3)	SSLND	SPI Slave Select Negation Delay Register	—
0x8000_7006 + 0x0400 × n (n = 0 to 2) 0x8100_7006 (n = 3)	SPND	SPI Next-Access Delay Register	—
0x8000_7007 + 0x0400 × n (n = 0 to 2) 0x8100_7007 (n = 3)	MRCKD	SPI Clock Digital control Register for Master Receive	—
0x8000_7008 + 0x0400 × n (n = 0 to 2) 0x8100_7008 (n = 3)	SPCR	SPI Control Register	—
0x8000_700C + 0x0400 × n (n = 0 to 2) 0x8100_700C (n = 3)	SPCRRM	SPI Control Register for Master Receive only	—
0x8000_700D + 0x0400 × n (n = 0 to 2) 0x8100_700D (n = 3)	SPDRCR	SPI Control Register for Received Data Ready Detection	—
0x8000_700E + 0x0400 × n (n = 0 to 2) 0x8100_700E (n = 3)	SPPCR	SPI Pin Control Register	—
0x8000_700F + 0x0400 × n (n = 0 to 2) 0x8100_700F (n = 3)	SPCR2	SPI Control Register 2	—
0x8000_7010 + 0x0400 × n (n = 0 to 2) 0x8100_7010 (n = 3)	SSLP	SPI Slave Select Polarity Register	—
0x8000_7011 + 0x0400 × n (n = 0 to 2) 0x8100_7011 (n = 3)	SPBR	SPI Bit Rate Register	—
0x8000_7013 + 0x0400 × n (n = 0 to 2) 0x8100_7013 (n = 3)	SPSCR	SPI Sequence Control Register	—
0x8000_7014 + 0x0400 × n + 0x04 × m (n = 0 to 2, m = 0 to 7) 0x8100_7014 + 0x04 × m (n = 3, m = 0 to 7)	SPCMDm	SPI Command Register m (m = 0 to 7)	—
0x8000_7040 + 0x0400 × n (n = 0 to 2) 0x8100_7040 (n = 3)	SPDCR	SPI Data Control Register	—
0x8000_7044 + 0x0400 × n (n = 0 to 2) 0x8100_7044 (n = 3)	SPDCR2	SPI Data Control Register 2	—
0x8000_7051 + 0x0400 × n (n = 0 to 2) 0x8100_7051 (n = 3)	SPSSR	SPI Sequence Status Register	—
0x8000_7052 + 0x0400 × n (n = 0 to 2) 0x8100_7052 (n = 3)	SPSR	SPI Status Register	—

Table 36.3 SPI register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x8000_7058 + 0x0400 × n (n = 0 to 2) 0x8100_7058 (n = 3)	SPTFSR	SPI Transfer FIFO Status Register	—
0x8000_705C + 0x0400 × n (n = 0 to 2) 0x8100_705C (n = 3)	SPRFSR	SPI Receive FIFO Status Register	—
0x8000_7060 + 0x0400 × n (n = 0 to 2) 0x8100_7060 (n = 3)	SPPSR	SPI Poling Register	—
0x8000_706A + 0x0400 × n (n = 0 to 2) 0x8100_706A (n = 3)	SPSRC	SPI Status Clear Register	—
0x8000_706C + 0x0400 × n (n = 0 to 2) 0x8100_706C (n = 3)	SPFCR	SPI FIFO Clear Register	—

Table 36.4 SPI related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0 (NONSAFETY)	—	MSTPCRB.MSTPCRB04	SLVACCCTL1.SPI0_SL
1 (NONSAFETY)	—	MSTPCRB.MSTPCRB05	SLVACCCTL1.SPI1_SL
2 (NONSAFETY)	—	MSTPCRB.MSTPCRB06	SLVACCCTL1.SPI2_SL
3 (SAFETY)	—	MSTPCRG.MSTPCRG02	SLVACCCTL2.SPI3_SL

36.3 Register Descriptions

36.3.1 SPDR/SPDR_HA/SPDR_BY : SPI Data Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

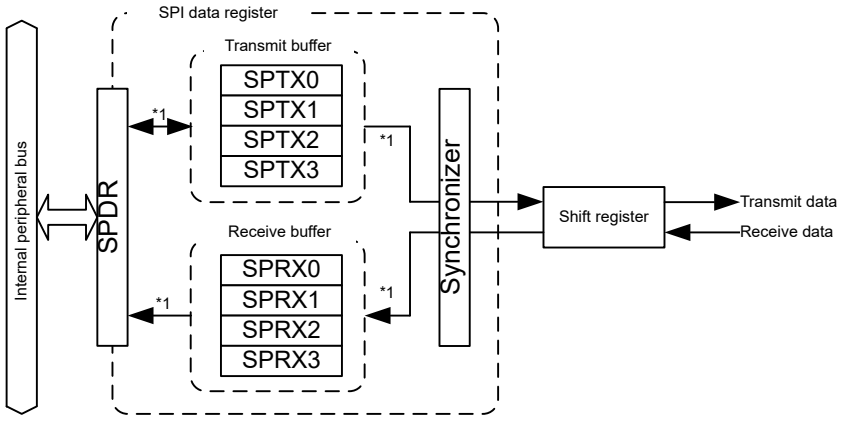
Offset address: 0x00

Bit position:	31																													0
Bit field:	SPD[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SPD[31:0]	The SPI data register (SPDR) is used to store SPI's transmit data and receive data. Transmit buffers and receive buffers independently function.	R/W

SPDR is the interface with the buffers that hold data for transmission and reception by the SPI.

Figure 36.2 shows the structure of SPDR.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 36.2 Structure of SPDR

32 bit × 4 stage transmit FIFO and 32 bit × 4 receive FIFO are provided. These 8 stage FIFO are mapped to one address in the SPDR. Transmit buffers (SPTXn, n = 0 to 3) can be written by writing data to SPDR to transmit written data.

Upon completion of receiving data, receive buffers store received data. When an overrun error occurs, data in the receive buffer is not updated.

Bus Interface

The SPI data register has 32 bit × 4 stage transmit FIFO and 32 bit × 4 receive FIFO (32 bytes in total). These 32 bytes are mapped to the 4-byte space of SPDR. Write transmit data from the LSB. Received data is stored from the LSB.

SPDR register write operation and read operation are described below.

(a) Write

A transmit buffer write pointer is provided for transmit buffers. When data is written to SPDR, the pointer automatically switches to the next buffer. The following illustrates the structure of the transmit buffer bus interface (write).

Figure 36.3 shows the configuration of the bus interface with the transmission buffer in the case of writing to SPDR.

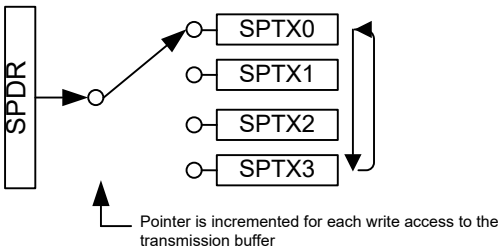


Figure 36.3 Structure of SPDR (write)

The transmit buffer (SPTX0 to SPTX3) switching order:

SPTX0→SPTX1→SPTX2→SPTX3→SPTX0→SPTX1→…

When writing transmit data to transmit buffers (SPTXn), write transmit data of frames +1 specified by the Transmission FIFO threshold setting bits of SPI data control register 2 (SPDCR2.TTRG[1:0]) while an SPI transmit buffer empty interrupt is present (SPSR.SPTEF flag = 1). Writing to the transmit buffer (SPTXn, n = 0 to 3) in the state where there is no empty stage in the transmit FIFO does not update the buffer value.

(b) Read

Values can be read from receive buffers (SPRXn, n = 0 to 3) or transmit buffers (SPTXn, n = 0 to 3) by reading the SPDR register. Reading a receive buffer or reading a transmit buffer can be selected by the SPI receive data or transmit data select bit (SPDCR.SPRDTD).

The SPDR register is read according to the independent receive buffer read pointer and the transmit buffer read pointer.

The following illustrates the structure of the receive buffer and transmit buffer bus interface (read).

Figure 36.4 shows the configuration of the bus interface with the receive and transmission buffers in the case of reading from SPDR.

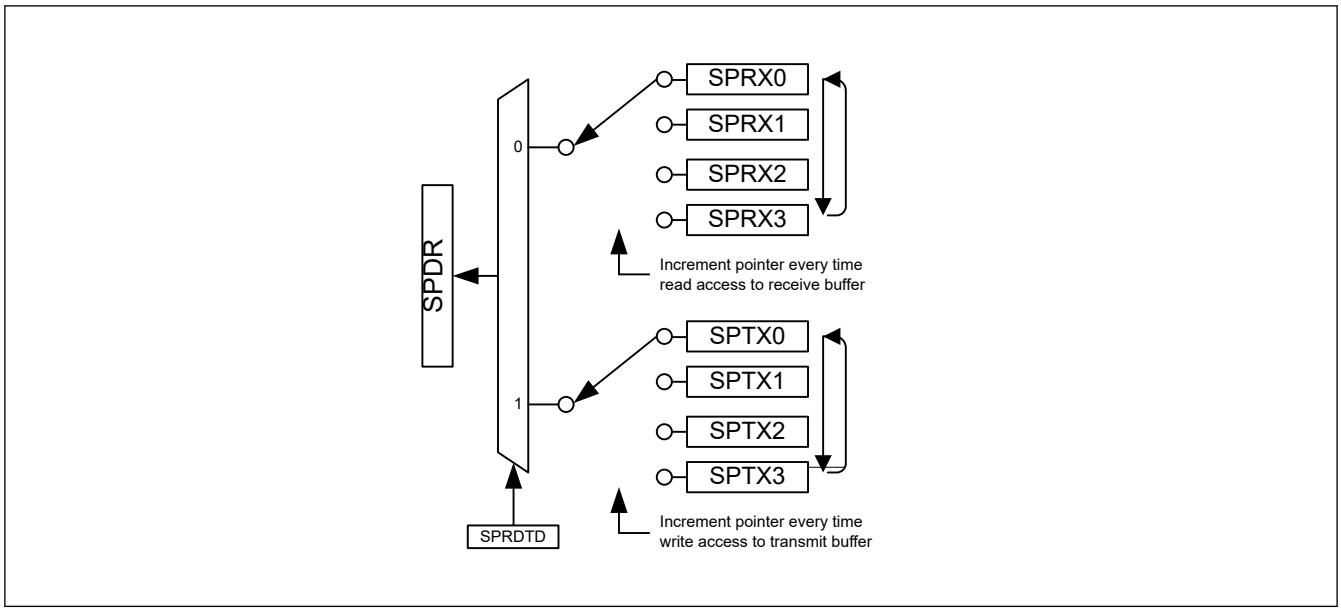


Figure 36.4 Structure of SPDR (read)

When a receive buffer is read, the receive buffer read pointer automatically switches to the next buffer. The receive buffer read pointer switches in the same order as the transmit buffer write pointer.

The transmit buffer read pointer is updated during the SPDR write access, but it is not updated during the transmit buffer read access. When a transmit buffer is read, the value written to SPDR last can be read.

36.3.2 SPCKD : SPI Clock Delay Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SCKDL[2:0]	RSPCK Delay Setting 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

[In the Motorola-SPI case]

The SPCKD register is used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while SPCMD.SCKDEN = 1. If SPCKD is modified while SPCR.MSTR = 1 and SPCR.SPE = 1, subsequent operation is not guaranteed.

[In the TI-SSP case]

The SPCKD register is used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while SPCMD.SCKDEN = 1. Also that is used to set the period until the SSL signal is negated. If SPCKD is modified while SPCR.MSTR = 1 and SPCR.SPE = 1, subsequent operation is not guaranteed.

SCKDL[2:0] bits (RSPCK Delay Setting)

These bits are used to set the RSPCK delay value when SPCMD.SCKDEN = 1.

To use the SPI in slave mode, set SCKDL[2:0] bits to 000b.

36.3.3 SSLND : SPI Slave Select Negation Delay Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SLNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SLNDL[2:0]	SSL Negation Delay Bits This is the synchronization delay for replacing the RSPCK input with PCLKSPIn. RSPCK Delay Setting 0 0 0: 1 RSPCK [Master mode] 1 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 0 0 1: 2 RSPCK [Master mode] 2 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 0 1 0: 3 RSPCK [Master mode] 3 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 0 1 1: 4 RSPCK [Master mode] 4 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 1 0 0: 5 RSPCK [Master mode] 5 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 1 0 1: 6 RSPCK [Master mode] 6 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 1 1 0: 7 RSPCK [Master mode] 7 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 1 1 1: 8 RSPCK [Master mode] 8 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode]	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

[In the Motorola-SPI case]

The SSLND register is used to set the period (SSL negation delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the SSL signal while SPCMD.SCKDEN = 1. If SSLND is modified while SPCR.MSTR = 1 and SPCR.SPE = 1, subsequent operation is not guaranteed.

[In the TI-SSP case]

The SSLND register is used to set the period (Output disable delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the Output enable signal while SPCMD.SCKDEN = 1. Also, that is used to set the period from when the SPI in slave mode detects the last RSPCK edge of serial transfer to when the Output enable signal is negated. If SSLND is modified while the SPE bit in SPCR.SPE = 1, subsequent operation is not guaranteed.

SLNDL[2:0] bits (SSL Negation Delay Bits)

[In the Motorola-SPI case]

These bits are used to set the SSL negation delay value when SPCMD.SLNDEN = 1.

To use the SPI in slave mode except TI-SSP, set SLNDL[2:0] bits to 000b.

[In the TI-SSP case]

These bits are used to set the Output disable delay value when SPCMD.SLNDEN = 1.

36.3.4 SPND : SPI Next-Access Delay Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SPNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SPNDL[2:0]	SPI Next-Access Delay Bits RSPCK Delay Setting 0 0 0: 1 RSPCK + 5 PCLKSPIn 0 0 1: 2 RSPCK + 5 PCLKSPIn 0 1 0: 3 RSPCK + 5 PCLKSPIn 0 1 1: 4 RSPCK + 5 PCLKSPIn 1 0 0: 5 RSPCK + 5 PCLKSPIn 1 0 1: 6 RSPCK+ 5 PCLKSPIn 1 1 0: 7 RSPCK + 5 PCLKSPIn 1 1 1: 8 RSPCK + 5 PCLKSPIn	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The SPND register is used to set the SSL signal inactive period (next-access delay) after completion of serial transfer while SPCMD.SCKDEN = 1. If SPND is modified while SPCR.MSTR = 1 and SPCR.SPE = 1, subsequent operation is not guaranteed.

SPNDL[2:0] bits (SPI Next-Access Delay Bits)

These bits are used to set the next-access delay value when SPCMD.SCKDEN = 1. To use the SPI in slave mode, set SPNDL[2:0] bits to 000b.

36.3.5 MRCKD : SPI ClocK Digital control Register for Master Receive

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ARST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ARST[2:0]	Receive Sampling Timing Adjustment Bits The sampling of the serial data (MISO) input in master mode can be adjusted by delaying 0 to 7 PCLKSPIn from the center of the bit. 0 0 0: 0 PCLKSPIn delay 0 0 1: 1 PCLKSPIn delay 0 1 0: 2 PCLKSPIn delay 0 1 1: 3 PCLKSPIn delay 1 0 0: 4 PCLKSPIn delay 1 0 1: 5 PCLKSPIn delay 1 1 0: 6 PCLKSPIn delay 1 1 1: 7 PCLKSPIn delay	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The MRCKD register is a register for adjusting the reception sampling timing in master mode. If the ARST[2:0] bits are rewritten while SPCR.SPE = 1, subsequent operations are not guaranteed.

ARST[2:0] bits (Receive Sampling Timing Adjustment Bits)

These bits are used to select the amount of reception sampling timing adjustment. The sampling timing can be adjusted by delaying 0 to 7 PCLKSPIn from the center of the bit.

Delay setting exceeding the dividing ratio is prohibited.

This bit setting is valid only when the loopback function is not used (when SPPCR.SPLP = 0, when MRCLK is selected (SPCR.SPSCKSEL = 1), and SPPCR.SPLP2 = 0) and in master mode. For details, see SPSCKSEL bit of section 36.3.6. SPCR : SPI Control Register.

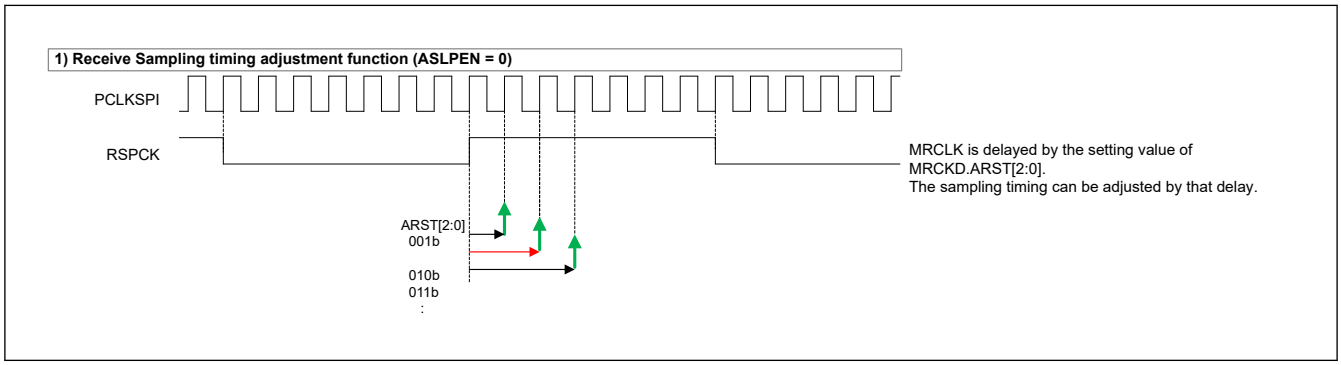


Figure 36.5 Description about receive sampling timing adjustment function

36.3.6 SPCR : SPI Control Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BPEN	MSTR	TXMD[1:0]		—	—	SPFRF	SPMS	—	—	CENDIE	SPTIE	SPDRES	SPIIE	SPRIE	SPEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MODFEN	BFDS	SCKASE	PTE	—	SPOE	SPPE	SPSCKSEL	—	—	—	—	—	—	SPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPE	SPI Function Enable 0: SPI function is disabled. 1: SPI function is enabled.	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	SPSCKSEL	SPI Master Receive Clock Select 0: MRIOCLK (Adjust with Analog Delay (SPCR2.SPSCKDL)) 1: MRCLK (Adjust with Digital Delay (MRCKD.ARST))	R/W
8	SPPE	Parity Enable 0: A parity bit is not added to transmit data. Received-data parity check is not performed. 1: A parity bit is added to transmit data. Received-data parity check is performed.	R/W
9	SPOE	Parity Mode 0: Even parity is used for transmission and reception. 1: Odd parity is used for transmission and reception.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	PTE	Parity Self-Diagnosis Enable 0: Parity circuit self-diagnosis function is disabled. 1: Parity circuit self-diagnosis function is enabled.	R/W
12	SCKASE	RSPCK Auto-Stop Function Enable 0: RSPCK auto-stop function is disabled. 1: RSPCK auto-stop function is enabled.	R/W
13	BFDS	Between Burst Transfer Frames Delay Select 0: Delay (RSPCK delay, SSL negation delay and next-access delay) between frames is inserted in burst transfer. 1: Delay between frames is not inserted in burst transfer.	R/W
14	MODFEN	Mode Fault Error Detection Enable 0: Mode fault error detection is disabled. 1: Mode fault error detection is enabled.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	SPEIE	SPI Error Interrupt Enable 0: SPI error interrupt request is disabled. 1: SPI error interrupt request is enabled.	R/W
17	SPRIE	SPI Receive Buffer Full Interrupt Enable 0: SPI receive buffer full interrupt request is disabled. 1: SPI receive buffer full interrupt request is enabled.	R/W
18	SPIIE	SPI Idle Interrupt Enable 0: Idle interrupt request is disabled. 1: Idle interrupt request is enabled.	R/W
19	SPDRES	SPI Receive Data Ready Error Select 0: Receive data full interrupt 1: Error interrupt	R/W
20	SPTIE	SPI Transmit Buffer Empty Interrupt Enable 0: SPI transmit buffer empty interrupt request is disabled. 1: SPI transmit buffer empty interrupt request is enabled.	R/W
21	CENDIE	SPI Communication End Interrupt Enable 0: Communication end interrupt request is disabled. 1: Communication end interrupt request is enabled.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	SPMS	SPI Function Enable 0: SPI operation (4-wire) 1: Clock synchronous operation (3-wire)	R/W
25	SPFRF	SPI Frame Format Select When SPMS = 1 (clock synchronous operation (3-wire)), this bit setting is invalid. 0: Motorola-SPI 1: TI-SSP	R/W

Bit	Symbol	Function	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
29:28	TXMD[1:0]	Communication Mode Select 0 0: Transmit-Receive 0 1: Transmit only 1 x: Receive only	R/W
30	MSTR	SPI Master/Slave Mode Select 0: Slave mode 1: Master mode	R/W
31	BPEN	Synchronization Circuit Bypass Enable 0: Non-Bypass 1: Bypass	R/W

The SPCR register is used to set operating mode of the SPI. If the set MSTR, TXMD[1:0], SPFRF, SPMS, MODFEN, BFDS, SCKASE, PTE, SPOE, SPPE, SPSCKSEL, and BPEN bit value is modified while SPCR.SPE = 1, subsequent operation is not guaranteed.

SPE bit (SPI Function Enable)

The SPE bit is used to enable or disable SPI functions. Setting this bit to 1 enables SPI functions. When SPSR.MODF = 1, the SPCR.SPE bit is cleared to 0 and the SPCR.SPE bit cannot be set to 1 until the MODF flag is cleared to 0. (See section 36.4.10. Error Detection.) Setting the SPE bit to 0 disables SPI functions and initializes a part of module functions. (See section 36.4.12. SPI Initialization.)

SPSCKSEL bit (SPI Master Receive Clock Select)

The SPSCKSEL bit selects the clock used to sample the received data in master mode. By setting this bit, the sampling timing of the MISO can be adjusted. This bit setting is valid only when the loopback function is not used (when SPPCR.SPLP = 0 and SPPCR.SPLP2 = 0) and in master mode.

Figure 36.6 shows a Master Receive Clock Delay configuration. By selecting this bit, you can select MRIOCLK (adjust with analog delay (SPCR2.SPSCKDL)) or MRCLK (adjust with digital delay (MRCKD.ARST)).

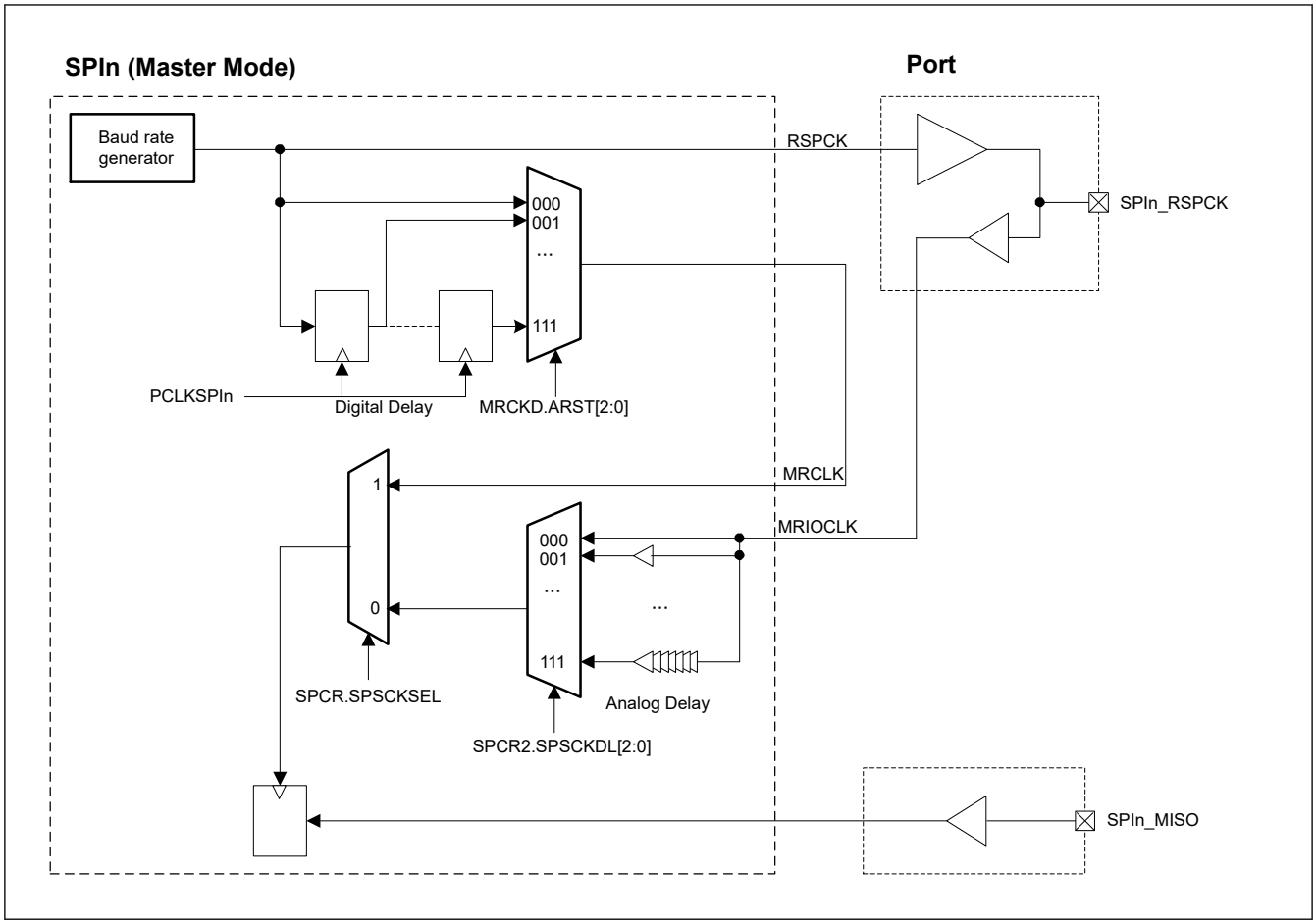


Figure 36.6 Master receive clock delay configuration (Master mode)

SPPE bit (Parity Enable)

The SPPE bit is used to enable or disable the parity function.

SPOE bit (Parity Mode)

The SPOE bit is used to specify even parity or odd parity.

In even parity mode, the parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an even number. In the same way, in odd parity mode, a parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an odd number. The SPOE bit is valid only when the SPCR.SPPE is set to 1.

PTE bit (Parity Self-Diagnosis Enable)

The PTE bit is used to enable or disable self-diagnosis of the parity circuit to confirm that the parity function is normal.

SCKASE bit (RSPCK Auto-Stop Function Enable)

The SCKASE bit is used to enable or disable the RSPCK auto-stop function. When this function is enabled, the RSPCK clock stops immediately before an overrun error occurs during data reception in master mode. For details, see section 36.4.10.1. Overrun Error.

BFDS bit (Between Burst Transfer Frames Delay Select)

The BFDS bit controls whether insert the delay time between the burst transfer frames.

Valid in the master mode (SPCR.MSTR = 1) for frames with the SPCMDm.SSLKP bit set to 1.

This bit shold be set to 0 in slave mode. The usage of SSL delay control between transfer frames is shown as below.

For details, see (4) Burst Transfer in section 36.4.13.2. Slave Mode Operation.

(1) Non-burst transmits

(2) Burst transmit with delay between frames

(2-1) From the 1st frame to the last previous frame

(2-2) The last frame

(3) Burst transmit with no delay between frames

(3-1) From the 1st frame to the last previous frame

(3-2) The last frame

Table 36.5 Usage of SSL delay control between transfer frames (master mode)

	SPCMDm SSLKP bit	SPCR BFDS bit	SSL delay control register*1 (RSPCK clock delay, SSL negation delay, next access delay)
(1)	0	0	Any given value. You can control each delay value according to setting for RSPCK clock delay, SSL negation delay, and next access delay.
(2-1)	1	0	
(2-2)	0	0	
(3-1)	1	1	Any given value. But delay is inserted only below. • RSPCK clock delay of the 1st frame • SSL negation delay and next access delay of the last frame
(3-2)	0	1	

Note 1. Whether the setting value of following bits are valid or not depends on the setting value of the SPCMD.SPNDEN bit. (See section 36.3.14. SPCMDm : SPI Command Register m (m = 0 to 7).)
The SPCKD.SCKDL[2:0] bits: RSPCK delay
The SSLND.SLNDL[2:0] bits: SSL negate delay
The SPND.SPNDL[2:0] bits: Next access delay

< Setting / operation example > (Motorola SPI, BFDS = 1 Case)

SPCMD0.SSLKP = 1 → Burst transfer / no interframe delay between 0 and 1 (SSL keep active)

SPCMD1.SSLKP = 1 → Burst transfer / no interframe delay between 1 and 2 (SSL keep active)

SPCMD2.SSLKP = 1 → Burst transfer / no interframe delay between 2 and 3 (SSL keep active)

SPCMD3.SSLKP = 1 → Burst transfer / no interframe delay between 3 and 4 (SSL keep active)

SPCMD4.SSLKP = 0 → do not Burst Transfer, and once in-activate SSL.

(BFDS setting is invalid because it does not Burst Transfer.)

SPCMD5.SSLKP = 1 → Burst transfer / no interframe delay between 5 and 6 (SSL keep active)

SPCMD6.SSLKP = 1 → Burst transfer / no interframe delay between 6 and 7 (SSL keep active)

SPCMD7.SSLKP = 0 → do not Burst Transfer, and once in-activate SSL.

(BFDS setting is invalid because it does not Burst Transfer.)

MODFEN bit (Mode Fault Error Detection Enable)

The MODFEN bit is used to enable or disable detection of a mode fault error. (See section 36.4.10. Error Detection.) The SPI determines SSL0 pin input or output direction according to the combination of the MODFEN and MSTR bits. (See section 36.4.2. SPI Pin Control.)

SPEIE bit (SPI Error Interrupt Enable)

The SPEIE bit is used to enable or disable an SPI error interrupt request when the SPI detects a mode fault error or an underrun error and sets the MODF flag in the SPI status register (SPSR) to 1, when the SPI detects an overrun error and sets the OVRF flag in SPSR to 1, or when the SPI detects a parity error and sets the PERF flag in SPSR to 1. (See section 36.4.10. Error Detection.)

SPRIE bit (SPI Receive Buffer Full Interrupt Enable)

The SPRIE bit is used to enable or disable a receive buffer full interrupt request of the SPI.

SPIIE bit (SPI Idle Interrupt Enable)

The SPIIE bit is used to enable or disable an idle interrupt request of the SPI after the SPI detects the idle state and sets the IDLNF flag in the SPI status register (SPSR) to 0.

SPDRES bit (SPI Receive Data Ready Error Select)

When a receive data ready is detected (SPSR.SPDRF = 1), select whether to use SPRI interrupt request or SPEI interrupt request.

SPTIE bit (SPI Transmit Buffer Empty Interrupt Enable)

The SPTIE bit is used to enable or disable a transmit buffer empty interrupt request of the SPI.

A transmit buffer empty interrupt request at the beginning of transmission is generated by setting the SPE bit to 1 simultaneously when or after the SPTIE bit is set to 1. Note that a transmit buffer empty interrupt is generated while the SPTIE bit is 1 even though SPI functions are disabled (SPE bit = 0).

CENDIE bit (SPI Communication End Interrupt Enable)

The CENDIE bit controls generation of a communication end interrupt request.

SPMS bit (SPI Function Enable)

The SPMS bit is used to select SPI operation (4-wire) or clock synchronous operation (3-wire).

For clock synchronous operation, the SSL pin is not used but three pins RSPCK, MOSI, and MISO are used for communication. When SPMS = 1 (clock synchronous operation (3-wire)), the setting of the SPFRF bit is invalid.

To perform clock synchronous operation in master mode (SPCR.MSTR = 1), the SPCMD.CPHA bit can be set to 0 or 1. To perform clock synchronous operation in slave mode (SPCR.MSTR = 0), set the CPHA bit to 1. If this bit is set to 0 for clock synchronous operation in slave mode (SPCR.MSTR = 0), subsequent operation is not guaranteed.

The communication status according to the settings of SPCR.MSTR, SPCR.TXMD[1:0], SPCR.SPFRF, and SPCR.SPMS as follows.

Table 36.6 SPI communication status

SPCR MSTR	SPCR TXMD[1]	SPCR TXMD[0]	SPCR SPFRF	SPCR SPMS	Communication Status	Communication Status No.
1	0	0	0	0	Transmit-Receive Master / Motorola SPI / SPI operation (4-wire)	1-(1)
1	0	0	1	0	Transmit-Receive Master / TI-SSP / SPI operation (4-wire)	1-(2)
1	0	0	—	1	Transmit-Receive Master / Clock synchronous operation (3-wire)	1-(3)
1	0	1	0	0	Transmit only Master / Motorola SPI / SPI operation (4-wire)	1-(4)
1	0	1	1	0	Transmit only Master / TI-SSP / SPI operation (4-wire)	1-(5)
1	0	1	—	1	Transmit only Master / Clock synchronous operation (3-wire)	1-(6)
1	1	—	0	0	Receive only Master / Motorola SPI / SPI operation (4-wire)	1-(7)
1	1	—	1	0	Receive only Master / TI-SSP / SPI operation (4-wire)	1-(8)
1	1	—	—	1	Receive only Master / Clock synchronous operation (3-wire)	1-(9)
0	0	0	0	0	Transmit-Receive Slave / Motorola SPI / SPI operation (4-wire) (default)	0-(1)
0	0	0	1	0	Transmit-Receive Slave / TI-SSP / SPI operation (4-wire)	0-(2)
0	0	0	—	1	Transmit-Receive Slave / Clock synchronous operation (3-wire)	0-(3)
0	0	1	0	0	Transmit only Slave / Motorola SPI / SPI operation (4-wire)	0-(4)
0	0	1	1	0	Transmit only Slave / TI-SSP / SPI operation (4-wire)	0-(5)
0	0	1	—	1	Transmit only Slave / Clock synchronous operation (3-wire)	0-(6)
0	1	—	0	0	Receive only Slave / Motorola SPI / SPI operation (4-wire)	0-(7)
0	1	—	1	0	Receive only Slave / TI-SSP / SPI operation (4-wire)	0-(8)
0	1	—	—	1	Receive only Slave / Clock synchronous operation (3-wire)	0-(9)

SPFRF bit (SPI Frame Format Select)

The SPFRF bit selects the communication protocol.

The format of the SPI terminal (RSPCK, SSL0 to 3) can be set according to the set communication protocol.

When SPMS = 1 (clock synchronous operation (3-wire)), this bit is invalid because SSL is not used.

TXMD[1:0] bits (Communication Mode Select)

The TXMD[1:0] bits are used to select the transmit-receive, transmit-only, and receive-only serial communication.

TXMD[1:0] = 01b

→ Transmit-only is performed without reception.

→ Receive buffer full interrupt request cannot be used.

TXMD[1] = 1b

→ Receive-only is performed without transmission.

→ Transmit buffer empty interrupt request cannot be used.

For details, see section 36.4.6. Communications Operating Mode.

MSTR bit (SPI Master/Slave Mode Select)

The MSTR bit is used to select master mode or slave mode of the SPI. The SPI determines input/output directions of pins RSPCK, MOSI, MISO, and SSL1 to SSL3 according to the MSTR bit setting.

BPEN bit (Synchronization Circuit Bypass Enable)

When BPEN = 1, bus clock (PCLKM) is used as operation clock (PCLKSPIn) and synchronization circuit is bypassed.

36.3.7 SPCRRM : SPI Control Register for Master Receive only

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RMSTTG	RMEDTG	—	RMFM[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	RMFM[4:0]	Frame processing count setting in Master Receive only 0x00: This function is not used.*1 0x01: Automatically stop communication after processing 1 received frame. ⋮ 0x1F: Automatically stop communication after processing 31 received frames.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	RMEDTG	1: Receive End (writable only when in Master Receive) Reading value is always 0.	W
7	RMSTTG	1: Receive Start (writable only when in Master Receive) Reading value is always 0.	W

Note 1. See (9) Software Processing Flow in section 36.4.13.1. Master Mode Operation.

The SPCRRM register controls the start and completion of communication for master mode reception only operation. If the RMFM[4:0] bits are rewritten while SPCR.SPE = 1, subsequent operations are not guaranteed.

RMFM[4:0] bits (Frame processing count setting in Master Receive only)

The number of received frames can be adjusted when operating in master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication operation mode select bits (SPCR.TXMD[1:0]) are 10b.

Only the start bit in master mode reception automatically stops communication after "starts frame processing according to the value set in this bit" after reception starts.

RMEDTG bit (1: Receive End (writable only when in Master Receive))

This bit is used to end reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD[1:0]) are 10b.

RMSTTG bit (1: Receive Start (writable only when in Master Receive))

This bit is used to start reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD[1:0]) are 10b.

Writing 1 to this bit during reception is not accepted. Write again after reception is completed.

36.3.8 SPDRCR : SPI Control Register for Received Data Ready Detection

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPDRC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SPDRC[7:0]	SPI Receive Data Ready Detect Adjustment 0x00: Disable receive data ready detection function 0x01: Performs reception data ready judgment after 1 PCLKSPIn ⋮ 0xFF: Performs reception data ready judgment after 255 PCLKSPIn	R/W

The SPDRCR register is used to set the SPI receive data ready detection function. If the set value is changed while SPCR.SPE = 1, subsequent operations are not guaranteed.

SPDRC[7:0] bits (SPI Receive Data Ready Detect Adjustment)

The receive data ready detection function can be disabled or, if used, the period until detection can be set from 1 to 255 PCLKSPIn.

The value set in the SPDRC[7:0] bits is used to 1 set the SPDRF flag. For details, see the description of SPDRF in section 36.3.18. SPSR : SPI Status Register.

36.3.9 SPPCR : SPI Pin Control Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x0E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MOIFE	MOIFV	—	SPOM	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPLP	SPI Loopback 0: Normal mode 1: Loopback mode (data is inverted for transmission)	R/W

Bit	Symbol	Function	R/W
1	SPLP2	SPI Loopback 2 0: Normal mode 1: Loopback mode (data is not inverted for transmission)	R/W
2	SPOM	SPI Output Pin Mode 0: CMOS output 1: Open-drain output	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	MOIFV	MOSI Idle Fixed Value 0: The fixed value of MOSI idle = 0. 1: The fixed value of MOSI idle = 1.	R/W
5	MOIFE	MOSI Idle Value Fixing Enable 0: The MOSI output value is the last data of previous transfer. 1: The MOSI output value is the set MOIFV bit value.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The SPPCR register is used to set pin mode of the SPI. If SPPCR is modified with the SPCR.SPE bit = 1, subsequent operation is not guaranteed.

SPLP bit (SPI Loopback)

When the SPLP bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when SPCR.MSTR = 1) or shuts down the route between the MOSI pin and the shift register, inverts the input route value in the shift register, and then connects the route to the output route (when SPCR.MSTR = 0) (loopback mode).

SPLP2 bit (SPI Loopback 2)

When the SPLP2 bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when SPCR.MSTR = 1) or shuts down the route between MOSI pin and the shift register and then connects the route to the output route without inverting the input route value in the shift register (when SPCR.MSTR = 0) (loopback mode). If this bit is set to 1 together with the SPLP bit, setting this bit takes precedence.

SPOM bit (SPI Output Pin Mode)

This bit is used to select CMOS output pin or open drain output pin as SPI's output pins.

MOIFV bit (MOSI Idle Fixed Value)

This bit is used to select the MOSI pin output value during the SSL negation period (including SSL retention period in burst transfer) when SPCR.MOIFE = 1 in master mode.

MOIFE bit (MOSI Idle Value Fixing Enable)

This bit is used for the SPI in master mode to fix the MOSI output value during the SSL negation period (including SSL retention period in burst transfer). When SPCR.MOIFE = 0, the SPI outputs the last data of the previous serial transfer to MOSI during the SSL negation period. When SPCR.MOIFE = 1, the SPI outputs the fixed MOIFV bit value to MOSI.

36.3.10 SPCR2 : SPI Control Register 2

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x0F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SPSCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SPSCKDL[2:0]	SPI Master Receive Clock Analog Delay Valid only in master mode 0 0 0: No delay 0 0 1: 1.1 ns (max) 0 1 0: 2.2 ns (max) 0 1 1: 3.3 ns (max) 1 0 0: 4.4 ns (max) 1 0 1: 5.5 ns (max) 1 1 0: 6.6 ns (max) 1 1 1: 7.7 ns (max)	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The SPCR2 register is used to set the analog delay of the master receive clock. If the value set in this register is changed while SPCR.SPE = 1, subsequent operations are not guaranteed.

SPSCKDL[2:0] bits (SPI Master Receive Clock Analog Delay)

These bits set the analog delay for MRIOCLK input to SPI in master mode. By setting these bits, the sampling timing of the received data can be adjusted. For details, see the SPSCKSEL bit of section 36.3.6. SPCR : SPI Control Register.

36.3.11 SSLP : SPI Slave Select Polarity Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSL0P	SSL0 Signal Polarity Setting 0: In the Motorola-SPI case, the SSL0 signal is active low (0). In the TI-SSP case, the SSL0 signal is active high (1). 1: In the Motorola-SPI case, the SSL0 signal is active high (1). In the TI-SSP case, the SSL0 signal is active low (0).	R/W
1	SSL1P	SSL1 Signal Polarity Setting 0: In the Motorola-SPI case, the SSL1 signal is active low (0). In the TI-SSP case, the SSL1 signal is active high (1). 1: In the Motorola-SPI case, the SSL1 signal is active high (1). In the TI-SSP case, the SSL1 signal is active low (0).	R/W
2	SSL2P	SSL2 Signal Polarity Setting 0: In the Motorola-SPI case, the SSL2 signal is active low (0). In the TI-SSP case, the SSL2 signal is active high (1). 1: In the Motorola-SPI case, the SSL2 signal is active high (1). In the TI-SSP case, the SSL2 signal is active low (0).	R/W
3	SSL3P	SSL3 Signal Polarity Setting 0: In the Motorola-SPI case, the SSL3 signal is active low (0). In the TI-SSP case, the SSL3 signal is active high (1). 1: In the Motorola-SPI case, the SSL3 signal is active high (1). In the TI-SSP case, the SSL3 signal is active low (0).	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

The SSLP register is used to set the polarity of SSL0 to SSL3 signals of the SPI. If any of these SSLP bits is modified with the SPCR.SPE bit = 1, subsequent operation is not guaranteed.

SSLnP bits (SSLn Signal Polarity Setting)

These bits are used to specify the polarity of SSL signals. The set SSLnP bit (n = 3 to 0) values indicate the active polarity of SSLn signals.

SSL0 is different from SSL1, SSL2, and SSL3. In slave or multi-master mode, it functions as an input.

For details, see section 36.4.3.2. Single Master/Single Slave (This LSI = Slave) and section 36.4.3.5. Multi-Master/Multi-Slave (with This LSI Acting as Master) .

36.3.12 SPBR : SPI Bit Rate Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPR[7:0]							
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	SPR[7:0]	The SPBR register is used to set the bit rate in master mode. If SPBR is modified while SPCR.MSTR = 1 and SPCR.SPE = 1, subsequent operation is not guaranteed.	R/W

When the SPI is used in slave mode, the bit rate depends on the input clock bit rate regardless of the SPBR.BRDV setting. (Specify a bit rate that meets electrical characteristics.)

The bit rate is determined by a combination of the set SPBR and BRDV[1:0] in the SPCMDm register (SPCMD0 to SPCMD7).

The bit rate is calculated by the following expression, where "n" is the set SPBR value (0 to 255) and "N" is the set BRDV[1:0] bits value (0 to 3).

$$\text{Bit rate} = \frac{f(\text{PCLKSPIn})}{2 \times (n+1) \times 2^{N}}$$

Table 36.7 shows an example of correspondence between bit rates and set values of SPBR and BRDV[1:0].

Table 36.7 Corresponding between bit rates and set values (examples)

SPBR (n)	BRDV[1:0] bits (N)	Division ratio	Bit rate				
			PCLKSPIn = 75 MHz	PCLKSPIn = 80 MHz	PCLKSPIn = 96 MHz	PCLKSPIn = 100 MHz	PCLKSPIn = 125 MHz
0	0	2	37.5 Mbps	40.00 Mbps	48.00 Mbps	50.00 Mbps	No support
1	0	4	18.75 Mbps	20.00 Mbps	24.00 Mbps	25.00 Mbps	31.25 Mbps
2	0	6	12.5 Mbps	13.33 Mbps	16.00 Mbps	16.67 Mbps	20.83 Mbps
3	0	8	9.38 Mbps	10.00 Mbps	12.00 Mbps	12.50 Mbps	15.63 Mbps
4	0	10	7.50 Mbps	8.00 Mbps	9.60 Mbps	10.00 Mbps	12.50 Mbps
5	0	12	6.25 Mbps	6.67 Mbps	8.00 Mbps	8.33 Mbps	10.42 Mbps
5	1	24	3.13 Mbps	3.33 Mbps	4.00 Mbps	4.17 Mbps	5.21 Mbps
5	2	48	1.56 Mbps	1.67 Mbps	2.00 Mbps	2.08 Mbps	2.60 Mbps
5	3	96	781 kbps	833 kbps	1.00 Mbps	1.04 Mbps	1.30 Mbps
255	3	4096	18.3 kbps	19.5 kbps	23.4 kbps	24.4 kbps	30.5 kbps

36.3.13 SPSCR : SPI Sequence Control Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x13

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SPSLN[2:0]	SPI Sequence Length Specification The order in which the SPCMD0 to SPCMD07 registers are to be referenced is changed in accordance with the sequence length that is set in these bits. The relationship among the setting of these bits, the sequence length, and the SPCMD0 to SPCMD7 register numbers referenced by SPI is as follows. However, the SPI in slave mode always refers to SPCMD0. 0 0 0: Sequence Length = 1 (0→0→⋯) 0 0 1: Sequence Length = 2 (0→1→0→⋯) 0 1 0: Sequence Length = 3 (0→1→2→0→⋯) 0 1 1: Sequence Length = 4 (0→1→2→3→0→⋯) 1 0 0: Sequence Length = 5 (0→1→2→3→4→0→⋯) 1 0 1: Sequence Length = 6 (0→1→2→3→4→5→0→⋯) 1 1 0: Sequence Length = 7 (0→1→2→3→4→5→6→0→⋯) 1 1 1: Sequence Length = 8 (0→1→2→3→4→5→6→7→0→⋯)	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The SPSCR register is used to set the sequence length for the SPI to perform master operation.

Before modifying the SPSLN[2:0] bits in SPSCR while SPCR.MSTR = 1 and SPCR.SPE = 1, confirm that SPSR.IDLNF = 0.

SPSLN[2:0] bits (SPI Sequence Length Specification)

These bits are used to set the sequence length for the SPI in master mode to perform sequence operation.

According to the sequence length specified by SPSLN[2:0] bits, the SPI in master mode changes the SPCMDm registers (SPCMD0 to SPCMD7) to be referenced and the reference sequence. For details, see (3) Sequence Control.

The SPI in slave mode always references SPCMD0.

36.3.14 SPCMDm : SPI Command Register m (m = 0 to 7)

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x14 + 0x04 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	SSLA[1:0]		—	—	—	SPB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCKDEN	SLNDEN	SPNDEN	LSBF	—	—	—	—	SSLKP	—	—	—	BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPHA	RSPCK Phase This register is valid only for Motorola-SPI cases. (In the TI-SSP case, it is fixed at 1). 0: Data is sampled at an odd edge and changes at an even edge. 1: Data changes at an odd edge and is sampled at an even edge.	R/W
1	CPOL	RSPCK Polarity 0: RSPCK in idle state is 0. 1: RSPCK in idle state is 1.	R/W
3:2	BRDV[1:0]	Bit Rate Division 0 0: The base bit rate is selected. 0 1: Two-divided base bit rate is selected. 1 0: Four-divided base bit rate is selected. 1 1: Eight-divided base bit rate is selected.	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	SSLKP	SSL Signal Level Hold 0: All SSL signals are negated at the end of transfer. 1: SSL signal level is held after the transfer ends until the next access starts.	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	LSBF	SPI LSB First 0: MSB first 1: LSB first	R/W
13	SPNDEN	SPI Next-Access Delay Enable 0: Next-access delay is 1 RSPCK + 5 PCLKSPIn 1: Next-access delay is the set value of the SPI next-access delay register (SPND).	R/W
14	SLNDEN	SSL Negation Delay Setting Enable 0: [Master] SSL negation delay is 1 RSPCK. [Slave in the TI-SSP] SSL negation delay is 1 PCLKSPIn + 2~3 PCLKSPIn (Synchronization delay) 1: SSL negation delay is the set value of the slave select negation delay register (SSLND).	R/W
15	SCKDEN	RSPCK Delay Setting Enable 0: RSPCK delay is 1 RSPCK. [In the Motorola-SPI case] RSPCK delay is 0 RSPCK. [In the TI-SSP case] 1: RSPCK delay is the set value of the RSPCK delay register (SPCKD).	R/W
20:16	SPB[4:0]	SPI Data Length 0x00: Setting prohibited 0x01: Setting prohibited 0x02: Setting prohibited 0x03: 4 bits 0x04: 5 bits 0x05: 6 bits ⋮ 0x1E: 31 bits 0x1F: 32 bits	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
25:24	SSLA[1:0]	SSL Signal Assertion 0 0: SSL0 0 1: SSL1 1 0: SSL2 1 1: SSL3	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The SPI has eight SPI command registers (SPCMD0 to SPCMD7) that are used to set the transfer format of the SPI in master mode. Furthermore, some bits in SPCMD0 are used to set the transfer format of the SPI in slave mode.

The SPI in master mode sequentially references SPCMD0 to SPCMD7 according to the setting of SPSCR.SPSLN[2:0], and then performs serial transfer specified in the referenced SPCMD.

SPCMD should be set while the transmit buffer is empty (next-transfer data is not set) and data which refers to SPCMD has not been written.

The SPCMD referenced by the SPI in master mode is indicated by SPCP[2:0] in the SPSSR register. If SPCMD0 is modified while the SPI in slave mode is enabled (SPCR.SPE = 1), subsequent operation is not guaranteed.

CPHA bit (RSPCK Phase)

This bit is used to set the RSPCK phase of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK phase must be set for both modules.

When SPCR.SPMS = 0 and SPCR.SPFRF = 1 (in TI SSP mode), setting CPHA = 0 is invalid.

This register is valid only for Motorola-SPI cases.

CPOL bit (RSPCK Polarity)

This bit is used to set the RSPCK polarity of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK polarity must be set for both modules.

BRDV[1:0] bits (Bit Rate Division)

This register is used to determine the bit rate with a combination of the set values of the BRDV[1:0] bits and the SPBR register. The set SPBR value determines the base bit rate. The set BRDV[1:0] bits value is used to select undivided, 2-divided, 4-divided, or 8-divided base bit rate. SPCMD0 to SPCMD7 enable setting of different BRDV[1:0] values. This makes it possible to perform serial transfer with a different bit rate for each command.

SSLKP bit (SSL Signal Level Hold)

This bit is used to set whether to hold or negate the SSL signal level of the current command during a period from the SSL negation timing for the current command to the SSL assertion timing for the next command when the SPI in master mode performs serial transfer. Setting this bit to 1 enables burst transfer in SPI operation master mode. For details, see (4) Burst Transfer in section 36.4.13.1. Master Mode Operation.

To use the SPI in slave mode, set SSLKP bit to 0.

LSBF bit (SPI LSB First)

This bit is used to set the data format of the SPI in master mode or slave mode to MSB first or LSB first.

SPNDEN bit (SPI Next-Access Delay Enable)

This bit is used to set the period from SSL negation to SSL assertion in SPI master mode. When SPNDEN = 0, the SPI sets the next-access delay to 1 RSPCK + 5 PCLKSPIn. When SPNDEN = 1, the SPI inserts the next-access delay in accordance with the SPND register setting.

To use the SPI in slave mode, set SPNDEN bit to 0.

SLNDEN bit (SSL Negation Delay Setting Enable)

[In the Motorola-SPI case]

This bit is used to set the period from RSPCK oscillation stop to SSL negation in SPI master mode. When SLNDEN = 0, the SPI sets the SSL negation delay to 1 RSPCK. When SLNDEN = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the SSLND register setting.

To use the SPI in slave mode, set SLNDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from RSPCK oscillation stop to output disable in SPI master mode, or last edge of RSPCK to output disable in slave mode. When the SLNDEN bit is 0, the SSL negate delay is 1 RSPCK in master mode and 1 PCLKSPIn + 2~3 PCLKSPIn (Synchronization delay) in slave mode. When SLNDEN = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the SSLND register setting.

When using SPI in slave mode except TI SSP setting, set the SLNDEN bit to 0.

SCKDEN bit (RSPCK Delay Setting Enable)

[In the Motorola-SPI case]

This bit is used to set the period (RSPCK delay) after the SPI in master mode activates the SSL signal until it oscillates RSPCK. When SCKDEN = 0, the SPI sets the RSPCK delay to 1 RSPCK. When SCKDEN = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the SPCKD register setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from the start of assertion of the SSL signal to the RSPCK oscillation (RSPCK delay) and the period of the SSL signal to negation by the SPI in master mode. When SCKDEN = 0, the SPI does not sets the RSPCK delay. When SCKDEN = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the SPCKD register setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

SPB[4:0] bits (SPI Data Length)

These bits are used to set the transfer data length of the SPI in master mode or slave mode.

SSLA[1:0] bits (SSL Signal Assertion)

These bits are used to control SSL signal assertion for the SPI in master mode to perform serial transfer. The set SSLA[1:0] bits value controls assertion of the SSL3 to SSL0 signals. The signal polarity when the SSL signal is asserted depends on the set value of the SSLP register. When SSLA[1:0] bits are set to 00b in multi-master mode, serial transfer is performed with all SSL signals negated (because SSL0 is input).

To use the SPI in slave mode, set SSLA[1:0] bits to 00b.

36.3.15 SPDCR : SPI Data Control Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x40

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPFC[1:0]		—	—	—	SINV	SPRDTD	SLSEL[1:0]		BYSW
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BYSW	Byte Swap Operating Mode Select 0: Byte swap is disabled 1: Byte swap is enabled	R/W
2:1	SLSEL[1:0]	SSL Pin Output Select See Table 36.8.	R/W
3	SPRDTD	SPI Receive Data or Transmit Data Selection 0: The SPDR reads the receive buffer 1: The SPDR reads the transmit buffer	R/W
4	SINV	Serial data invert 0: Not invert serial data 1: Invert serial data	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SPFC[1:0]	Frame Count Number of Frames Specification 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

The SPDCR register controls the data format.

If the value set in this register is changed while SPCR.SPE = 1, subsequent operations are not guaranteed.

BYSW bit (Byte Swap Operating Mode Select)

It is a setting bit, that is to swap a transmit/receive data in byte units. A data after byte swap is different by a data length (setting of SPCMD.SPB[4:0]).

When byte swap, A data length (setting of SPB[4:0]) must be set to 32 bits or 16 bits. Other case of data length (for example, 4 to 15, 17 to 31 bit length), byte swap is not guaranteed. For the arrangement of data before and after swapping data lengths of 32 bits and 16 bits, see section 36.4.4. Data Format and section 36.4.4.5. Byte swap reception.

When the parity function set to valid, the behavior is not guaranteed.

SLSEL[1:0] bits (SSL Pin Output Select)

The SLSEL[1:0] bits are used to select SSL output or I/O as an SSL pin function in master mode.

Table 36.8 SSL pin output selection

SSLn pin	SLSEL[1:0] = 00b	SLSEL[1:0] = 01b	SLSEL[1:0] = 10b	SLSEL[1:0] = 11b
SSL0	Output/Input	Output/Input	Output/Input	Setting prohibited
SSL1	Output	I/O	Output	
SSL2, SSL3	Output	I/O	I/O	

Note: Input or output of SSL0 is determined by the SPCR.MODFEN bit. For details, see Table 36.9.

SPRDTD bit (SPI Receive Data or Transmit Data Selection)

The SPRDTD bit is used to select receive buffer or transmit buffer from which the SPI data register (SPDR) value is read.

When the transmit buffer is read, the value written to SPDR the last time is read.

SINV bit (Serial data invert)

The SINV bit is used to invert transmit data and receive data.

When the SINV bit is set to 1, transmit buffer (SPTX) data is inverted to invert transmit data and receive data, and then the inverted data is stored in the receive buffer (SPRX). The parity bit is the value corresponding to the inverted transmission/reception data.

SPFC[1:0] bits (Frame Count)

The SPFC[1:0] bits are used for the condition to set the CENDF flag in slave receive only mode. For details on the CENDF flag setting conditions, see section 36.3.18. SPSR : SPI Status Register.

The SPFC[1:0] bits are invalid except in the slave receive only mode.

36.3.16 SPDCR2 : SPI Data Control Register 2

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x44

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TTRG[1:0]		—	—	—	—	—	—	RTRG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RTRG[1:0]	Receive FIFO threshold setting When using DMA, set the threshold to 0 for RTRG and TTRG bits. 0 0: Threshold 0 0 1: Threshold 1 1 0: Threshold 2 1 1: Threshold 3	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
9:8	TTRG[1:0]	Transmission FIFO threshold setting When using DMA, set the threshold to 0 for RTRG and TTRG bits. 0 0: Threshold 0 0 1: Threshold 1 1 0: Threshold 2 1 1: Threshold 3	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

The SPDCR2 register controls the FIFO threshold. If the value set in this register is changed while SPCR.SPE = 1, subsequent operations are not guaranteed.

RTRG[1:0] bits (Receive FIFO threshold setting)

Set the receive FIFO threshold.

When the number of data stored in the receive FIFO > the number of frames set by RTRG[1:0], the receive buffer full flag is set.

TTRG[1:0] bits (Transmission FIFO threshold setting)

Set the transmit FIFO threshold.

When the number of empty stages in the transmit FIFO > the number of frames set in TTRG[1:0], the transmit buffer empty flag is set.

36.3.17 SPSSR : SPI Sequence Status Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x51

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SPCP[2:0]	SPI Command Pointer 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
3	—	This bit is read as 0.	R
6:4	SPECM[2:0]	SPI Error Command 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
7	—	This bit is read as 0.	R

The SPSSR register indicates the sequence control status when the SPI performs master operation. Write access to SPSSR is ignored.

SPCP[2:0] bits (SPI Command Pointer)

These bits show the SPCMDm registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the current pointer in the SPI sequence control. For details about the SPI sequence control, see section 36.4.13.1. Master Mode Operation.

SPECM[2:0] bits (SPI Error Command)

These bits show the SPCMDm registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the command pointer (SPCP[2:0] bits) when an error was detected in the SPI sequence control. The SPI updates the SPECM[2:0] bits value only when an error is detected. When no error is present (SPSR.OVRF = 0, SPSR.MODF = 0, SPSR.PERF = 0), the SPECM[2:0] bits value has no meaning. For the SPI's error detection function, see section 36.4.10. Error Detection. For the SPI's sequence control, see section 36.4.13.1. Master Mode Operation.

36.3.18 SPSR : SPI Status Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x52

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPRF	CENDF	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF	SPDRF	—	—	—	—	—	—	—
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0.	R
7	SPDRF	SPI Receive Data Ready Flag 0: Receive data ready not detected 1: Receive data ready detected	R
8	OVRF	Overrun Error Flag 0: No overrun error is present. 1: An overrun error is present.	R
9	IDLNF	SPI Idle Flag 0: The SPI is in the idle state. 1: The SPI is in the transfer state.	R
10	MODF	Mode Fault Error Flag 0: Neither mode fault error nor underrun error is present. 1: A mode fault error or underrun error is present.	R
11	PERF	Parity Error Flag 0: No parity error is present. 1: A parity error is present.	R
12	UDRF	Underrun Error Flag This bit indicates error status in combination with the MODF flag. The UDRF bit is valid when MODF flag is 1. 0: Mode fault error occurred (MODF = 1) 1: Underrun error occurred (MODF = 1)	R
13	SPTEF	SPI Transmit Buffer Empty Flag 0: The number of empty stages in the transmit FIFO ≤ the value set in SPDCR2.TTRG 1: The number of empty stages in the transmit FIFO > the value set in SPDCR2.TTRG	R
14	CENDF	Communication End Flag 0: The SPI is not communicating or communicating. 1: The SPI communication completed.	R
15	SPRF	SPI Receive Buffer Full Flag 0: The number of data stored in the receive FIFO ≤ number of frames set by the SPDCR2.RTRG bit. 1: The number of data stored in the receive FIFO > number of frames set by the SPDCR2.RTRG bit.	R

SPDRF bit (SPI Receive Data Ready Flag)

During communication (SPCR.SPE = 1), a certain period of time has elapsed while the number of data stored in the reception FIFO ≤ the reception FIFO threshold.

This bit is set to 0 when the reception operation is not performed (SPCR.TXMD[1:0] = 01b).

[Setting (to 1) condition]

All the following two conditions are met:

- SPDRCR.SPDRC[7:0] ≠ 0x00
- After the receive FIFO has been written, when the number of data stored in the receive FIFO ≤ the receive FIFO threshold and the value set by SPDRC[7:0] has elapsed

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.SPDRFC bit

OVRF bit (Overrun Error Flag)

This flag indicates whether an overrun error is present. When the RSPCK clock auto-stop function is enabled (SPCR.SCKASE bit = 1) in master mode (SPCR.MSTR bit = 1), no overrun error occurs and, therefore, this flag is not set to 1. For details, see section 36.4.10.1. Overrun Error.

[Setting (to 1) condition]

When serial transfer is completed in one of the following two conditions with data stored in the receive FIFO for the number of FIFO stages.

- The SPCR.TXMD[1:0] bits = 00b. (transmit-receive mode)
- The SPCR.TXMD[1:0] bits = 10b. (receive only)

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.OVRFC bit

IDLNF bit (SPI Idle Flag)

This flag indicates transfer status of the SPI.

[Setting (to 1) condition]

[Transmit-Receive, Transmit-only Master mode]

- None of the [Transmit-Receive, Transmit-only Master mode] in [Clearing (to 0) conditions]

[Receive-only Master mode]

- When 1 is written to SPCRRM.RMSTTG bit

[Slave mode]

- SPCR.SPE = 1 (SPI function enabled)

[Clearing (to 0) conditions]

Communication status: 1-(1) to (6) * For details of communication status, see Table 36.6.

[Transmit-Receive, Transmit-only Master mode]

Any of the following two conditions is met:

- SPCR.SPE = 0 (SPI initialization)
- All the following three conditions are met:
  - The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
  - SPSSR.SPCP[2:0] = 000b (at the beginning of sequence control)
  - The operation completed until the next access delay (the master main state machine has transitioned to the idle state)

[Receive-only Master mode]

Communication status: 1-(7) to (9)

Any of the following two conditions is met:

- SPCR.SPE = 0 (SPI initialization)
- Any of the following three conditions is met:
  - When RMFM[4:0] = 0x00, after writing 1 to RMEDTG, operation completed until the next access delay (the master main state machine has transitioned to the idle state)
  - When RMFM[4:0] ≠ 0x00, after writing 1 to RMEDTG, operation completed until the next access delay (the master main state machine has transitioned to the idle state)
  - When RMFM[4:0] ≠ 0x00, the operation is completed up to the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

[Slave mode]

Communication status: 0 - (1) to (9)

- SPCR.SPE = 0 (SPI initialization)

MODF bit (Mode Fault Error Flag)

This flag indicates whether a mode fault error or an underrun error is present. The UDRF flag allows you to see which error (mode fault error or underrun error) has occurred.

[Setting (to 1) condition]

[Multi-master mode]

- The SSL0 pin input level becomes active level while the SPCR.MSTR bit = 1 (master mode) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.

[Slave, Motorola-SPI mode]

Any of the following two conditions is met:

- The SSL0 pin is negated before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 0 (Motorola-SPI) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Slave, TI-SSP mode]

Any of the following two conditions is met:

- The SSL0 pin is asserted before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 1 (TI-SSP) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

The SSL signal active level depends on the SSLP.SSLiP bits (SSL signal polarity bits).

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.MODFC bit

PERF bit (Parity Error Flag)

This flag indicates whether a parity error is present.

[Setting (to 1) condition]

When the serial transfer ends and a parity error is detected with the SPCR.SPPE bit set to 1 under any of the following two conditions:

- The SPCR.TXMD[1:0] bits = 00b (transmit-receive)
- The SPCR.TXMD[1:0] bits = 10b (receive-only)

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.PERFC bit

UDRF bit (Underrun Error Flag)

This flag indicates that a mode fault error or an underrun error is present.

[Setting (to 1) condition]

- Serial transfer is started before transmit data output becomes ready while the SPCR.MSTR bit = 0 (slave mode) and the SPCR.TXMD[1:0] bits = 00b or 01b (transmit-receive or transmit-only) and the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.UDRFC bit

SPTEF bit (SPI Transmit Buffer Empty Flag)

This flag indicates the transmit buffer (SPTX) status in the SPI data register (SPDR).

[Setting (to 1) condition]

Any of the following three conditions is met:

- The SPCR.SPE bit is set to 0 (SPI initialization)
- When the number of empty transmission FIFO stages > SPDCR2.TTRG[1:0]
- When 1 is written to SPFCR.SPFRST

[Clearing (to 0) condition]

Any of the following two conditions is met:

- At the time of final access when transmission data is written to SPDR (SPTXn, n = 0 to 2) in one processing routine using DMAC
- When 1 is written to the SPSRC.SPTEFC bit

Writing a value to the SPDR register is enabled only while the SPTEF flag = 1. If a value is written to the SPDR register while the SPTEF flag = 0, transmit buffer data is not updated.

CENDF bit (Communication End Flag)

This flag indicates communication end status of SPI. It turns 1 at communication end and turns 0 at starting next communication.

[Setting (to 1) condition]

Transmit-Receive / Transmit-only Master mode

Communication status: 1 - (1) to (6). For details of communication status, see Table 36.6.

The following three conditions are met:

- The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
- The SPSSR.SPCP[2:0] are 000b (it means the head of the sequential control)
- Operation completed until the next access delay (the master main state machine has transitioned to the idle state)

Receive-only Master mode

Communication status: 1 - (7) to (9)

Any of the following three conditions is met:

- When RMFM[4:0] = 0x00, after writing 1 to RMEDTG, operation completed until the next access delay (the master main state machine has transitioned to the idle state)

- When RMFM[4:0] ≠ 0x00, after writing 1 to RMEDTG, operation completed until the next access delay (the master main state machine has transitioned to the idle state)
- When RMFM[4:0] ≠ 0x00, the operation is completed up to the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

Transmit-receive / transmit-only slave, Motorola-SPI mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0).

Communication status: 0 - (1), (4)

The following three conditions are met:

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty (it means SPI does not do serial transfer)
- SSL0 was negated

Transmit-receive / transmit-only slave, TI-SSP mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0).

Communication status: 0 - (2), (5)

The following three conditions are met:

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty (it means SPI does not do serial transfer)
- When the SSL negate delay is completed

Transmit-receive / transmit only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1)

Communication status: 0 - (3), (6)

The following three conditions are met:

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty (it means SPI does not do serial transfer)
- The last even edge of RSPCK of the frame was detected (when the SPCMD.CPHA bit is 1)

Receive only slave, Motorola-SPI mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0).

Communication status: 0 - (7)

The following condition is met:

- SSL0 input was negated after getting frames for SPDCR.SPFC set value in the receive buffer

Receive only slave, TI-SSP mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0).

Communication status: 0 - (8)

The following condition is met:

- SSL0 negate delay is completed after getting frames for SPDCR.SPFC set value in the receive buffer

Receive only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1).

Communication status: 0 - (9)

The following condition is met:

- The last even edge of RSPCK of the last frame received for SPFC set value was detected (when the SPCMD.CPHA bit is 1)

[Clearing (to 0) condition]

Transmit-Receive / Transmit-only Master mode

Communication status: 1 - (1) to (6)

Any of the following two conditions is met:

- The next transmit data was written to the transmit buffer (SPTX)

- When 1 is written to the SPSRC.CENDFC bit

Receive-only Master mode

Communication status: 1 - (7) to (9)

Any of the following two conditions is met:

- When 1 is written to the SPCRRM.RMSTTG bit with SPCR.SPE = 1
- When 1 is written to the SPSRC.CENDFC bit

Transmit-receive / transmit only slave mode

Communication status: 0 - (1) to (6)

Satisfy one of following conditions:

- The next transmit data is written to the transmit buffer (SPTX)
- When 1 is written to the SPSRC.CENDFC bit

Receive-only slave mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0).

Communication status: 0 - (7) to (8)

Satisfy one of following conditions:

- SSL0 assertion of next data is detected
- When 1 is written to the SPSRC.CENDFC bit

Receive-only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1).

Communication status: 0 - (9)

Satisfy one of following conditions:

- The first edge of RSPCK of the next data is detected
- When 1 is written to the SPSRC.CENDFC bit

SPRF bit (SPI Receive Buffer Full Flag)

This flag indicates the receive buffer (SPRX) status in the SPDR register.

[Setting (to 1) condition]

When the number of data stored in the receive FIFO > the number of frames set in the SPDCR2.RTRG[1:0] bits in Transmit-Receive, Receive-only modes. However, the SPRF flag does not change from 0 to 1 while the OVRF flag = 1. (See section 36.4.10. Error Detection.)

[Clearing (to 0) condition]

Any of the following three conditions is met:

- At the last access when read data is read from SPDR (SPRXn, n = 0 to 2) in one processing routine using DMAC
- When 1 is written to the SPSRC.SPRFC bit
- When 1 is written to the SPFCR.SPFRST bit

36.3.19 SPTFSR : SPI Transfer FIFO Status Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x58

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TFDN[2:0]		
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	TFDN[2:0]	Transmit FIFO data empty stage number 0 0 0: Number of empty stages 0 0 0 1: Number of empty stages 1 0 1 0: Number of empty stages 2 0 1 1: Number of empty stages 3 1 0 0: Number of empty stages 4	R
7:3	—	These bits are read as 0.	R

Note: TFDN[2:0] is initialized by SPCR.SPE = 0.

36.3.20 SPRFSR : SPI Receive FIFO Status Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x5C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	RFDN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RFDN[2:0]	Receive FIFO data store stage number 0 0 0: Number of store stages 0 0 0 1: Number of store stages 1 0 1 0: Number of store stages 2 0 1 1: Number of store stages 3 1 0 0: Number of store stages 4	R
7:3	—	These bits are read as 0.	R

Note: RFDN[2:0] is initialized by SPCR.SPE = 0.

36.3.21 SPPSR : SPI Poling Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x60

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEPS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPEPS	SPI Polling Status 0: SPCR.SPE is 0 1: SPCR SPE is 1	R
31:1	—	These bits are read as 0.	R

Note: This bit shows a result of SPCR.SPE after synchronization by the operation clock (PCLKSPIn).

36.3.22 SPSRC : SPI Status Clear Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x6A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPRFC	CENDFC	SPTEFC	UDRFC	PERFC	MODFC	—	OVRFC	SPDRFC	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0.	R
7	SPDRFC	SPI Receive Data Ready Flag Clear The SPI receive data ready flag can be cleared by writing 1. When read, 0 is read.	W
8	OVRFC	Overrun Error Flag Clear By writing 1, the Overrun Error Flag can be cleared. Reading value is always 0.	W
9	—	This bit is read as 0.	R
10	MODFC	Mode Fault Error Flag Clear By writing 1, the Mode Fault Error Flag can be cleared. Reading value is always 0. Before setting MODFC and UDRFC, make sure that SPSR.MODF and UDRF are set to 1.	W
11	PERFC	Parity Error Flag Clear By writing 1, the Parity Error Flag can be cleared. Reading value is always 0.	W
12	UDRFC	Underrun Error Flag Clear By writing 1, the Underrun Error Flag can be cleared. Reading value is always 0. When clearing the UDRF flag, clear the MODF flag at the same time (MODFC = 1).	W
13	SPTEFC	SPI Transmit Buffer Empty Flag Clear By writing 1, the SPI Transmit Buffer Empty Flag can be cleared. Reading value is always 0.	W
14	CENDFC	Communication End Flag Clear By writing 1, the Communication End Flag can be cleared. Reading value is always 0.	W
15	SPRFC	SPI Receive Buffer Full Flag Clear By writing 1, the SPI Receive Buffer Full Flag can be cleared. Reading value is always 0.	W

The SPSRC register clears the status flag (SPSR) that indicates the operating status of SPI.

36.3.23 SPFCR : SPI FIFO Clear Register

Base address: SPIn = 0x8000_7000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_7000

Offset address: 0x6C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SPFRST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPFRST	SPI FIFO clear By writing 1, the pointer in the FIFO and the stored data are initialized. Reading value is always 0.	W
7:1	—	The write value should be 0.	W

The SPFCR register is used to clear the FIFO.

If SPFCR is rewritten while SPCR.SPE = 1, subsequent operations are not guaranteed.

36.4 Operation

In this section, a word "serial transfer period" is used as a word that means a period from the start of valid data drive to latching of the final valid data.

36.4.1 Overview of SPI Operations

The SPI can transfer data in the following five modes.

- Slave mode (SPI operation)
- Single master mode (SPI operation)
- Multi-master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation)

SPI modes can be set by the MSTR, MODFEN, SPMS and SPFRF bits in the SPCR register.

Table 36.9 lists the relationship between SPI modes and SPCR settings, and outline of each mode.

Table 36.9 Relationship between SPI modes and SPCR settings and description of each mode (1 of 2)

Mode	Slave (SPI operation)	Single-Master (SPI operation)	Multi-Master (SPI operation)	Slave (Clock synchronous operation)	Master (Clock synchronous operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
SPFRF bit setting	valid	valid	valid	In-valid	In-valid
RSPCK signal	Input	Output	Output/Hi-Z	Input	Output
MOSI signal	Input	Output	Output/Hi-Z	Input	Output
MISO signal	Output/Hi-Z	Input	Input	Output	Input
SSL0 signal	Input	Output	Input	Hi-Z (not used)	Hi-Z (not used)
SSL1 to SSL3 signals	Hi-Z (not used)	Output	Output/Hi-Z	Hi-Z (not used)	Hi-Z (not used)
Output pin mode	CMOS/open drain	CMOS/open drain	CMOS/open drain	CMOS/open drain	CMOS/open drain
SSL polarity modification function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLKSPIn/2	Up to PCLKSPIn/2	Up to PCLKSPIn/2	Up to PCLKSPIn/2	Up to PCLKSPIn/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	2 types				
Clock phase	2 types*5	2 types*5	2 types*5	One (CPHA = 1)	2 types
First transfer bit	MSB/LSB				
Transfer data length	4 to 32 bits				
Burst transfer	Enabled (CPHA = 1)	Enabled (CPHA = 0, 1)	Enabled (CPHA = 0, 1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported*6	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported

Table 36.9 Relationship between SPI modes and SPCR settings and description of each mode (2 of 2)

Mode	Slave (SPI operation)	Single-Master (SPI operation)	Multi-Master (SPI operation)	Slave (Clock synchronous operation)	Master (Clock synchronous operation)
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer empty interrupt request or transmit buffer write when SPTEF = 1	Transmit buffer empty interrupt request or transmit buffer write when SPTEF = 1	RSPCK oscillation	Transmit buffer empty interrupt request or transmit buffer write when SPTEF = 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmission buffer empty detection	Supported*4	Supported	Supported	Supported*4	Supported
Reception buffer full detection	Supported*1				
Overrun error detection	Supported*1	Supported*1 *3	Supported*1 *3	Supported*1	Supported*1 *3
Parity error detection	Supported*1 *2				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported*4	Not supported	Not supported	Supported*4	Not supported

Note 1. When SPI is transmit-master mode or transmit-slave mode (see Table 36.6), none of receive buffer full error, overrun error, and parity error is detected.
Note 2. When the SPCR.SPPE bit is 0, parity error is not detected.
Note 3. When the SPCR.SCKASE bit is 1, overrun error is not detected.
Note 4. When SPI is receive only slave mode (see Table 36.6), none of transmit buffer empty and underrun error is detected.
Note 5. CPHA = 0 is invalid in TI SSP mode. (Even if it is set, the operation is the same as when CPHA = 1.)
Note 6. Available only in TI SSP mode.

36.4.2 SPI Pin Control

The SPI automatically switches pin directions and output modes according to the settings of the SPCR.MSTR, SPCR.MODFEN and SPCR.SPMS bits and the SPPCR.SPOM bit.

When the SPPCR.SPOM bit is set to 0, each output becomes CMOS output, when the SPOM bit is set to 1, each output becomes open drain output.

Table 36.10 shows the relationship between pin state and set values of each bit.

For details of OE function, see section 36.4.5. Transfer Format (Frame Format).

Table 36.10 Relationship between pin states and set control bit values (1 of 2)

Mode	Pin	Pin State*1	
		SPOM = 0	SPOM = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL3	CMOS output	Open-drain output
	MOSI	CMOS output	Open-drain output
	MISO	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCK*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSL0	Input	Input
	SSL1 to SSL3*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSI*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISO	Input	Input

Table 36.10 Relationship between pin states and set control bit values (2 of 2)

Mode	Pin	Pin State*1	
		SPOM = 0	SPOM = 1
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCK	Input	Input
	SSL0	Input	Input
	SSL1 to SSL3*4	Hi-Z (not used)	Hi-Z (not used)
	MOSI	Input	Input
	MISO*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL3*4	Hi-Z (not used)	Hi-Z (not used)
	MOSI	CMOS output	Open-drain output
	MISO	Input	Input
Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCK	Input	Input
	SSL0 to SSL3*4	Hi-Z (not used)	Hi-Z (not used)
	MOSI	Input	Input
	MISO	CMOS output	Open-drain output

Note 1. The set SPI value is not applied to multi-function pins for which the SPI function is not selected.
Note 2. When SSL0 is at the active level, the pin state is Hi-Z. Motorola-SPI: When SSL0 is active level, the pin state becomes Hi-Z. TI-SSP: The terminal status becomes Hi-Z until SSL0 is asserted after SPCR.SPE = 1 and communication completed.
Note 3. Motorola-SPI: When SSL0 is inactive level or when SPCR.SPE = 0, the pin state becomes Hi-Z. TI-SSP: When SSL0 is except the communication period or when SPCR.SPE = 0 (assertion after SPE = 1 and communication completed), the pin status changes to Hi-Z.
Note 4. These pins are available for use as I/O port pins.

The SPI in single master mode (SPI operation) or multi-master mode (SPI operation) determines the MOSI signal value in the SSL negation period (including the SSL hold period in burst transfer) according to the settings of the SPPCR.MOIFE and SPPCR.MOIFV bits, as listed in Table 36.11.

Table 36.11 Determining the MOSI signal value in the SSL negation period

MOIFE bit	MOIFV bit	MOSI signal value during SSL negation period
0	0, 1	Final data from previous transfer
1	0	Always low
1	1	Always high

36.4.3 SPI System Configuration Examples

This configuration example describes that 0 level of SSL signals is active level. When connecting and using in a multi-slave or multi-master mode, the transfer format of the connected device should be unified to either Motorola-SPI or TISSP.

36.4.3.1 Single Master/Single Slave (This LSI = Master)

Figure 36.7 shows an example of single master/single slave SPI system configuration where this LSI is used as a master. In the single master/single slave configuration, the SSL0 to SSL3 output signals of this LSI (master) are not used.

SSL input signals of the SPI slave are fixed to 0 level to always select the SPI slave. When SPCMD.CPHA = 0, some slave devices cannot fix SSL signals to active level in the relevant transfer format. If the SSL signal level cannot be fixed, connect the SSL output of this LSI to the SSL input of the slave device.

This LSI (master) always drives the RSPCK and MOSI pins. The SPI slave always drives the MISO pin.

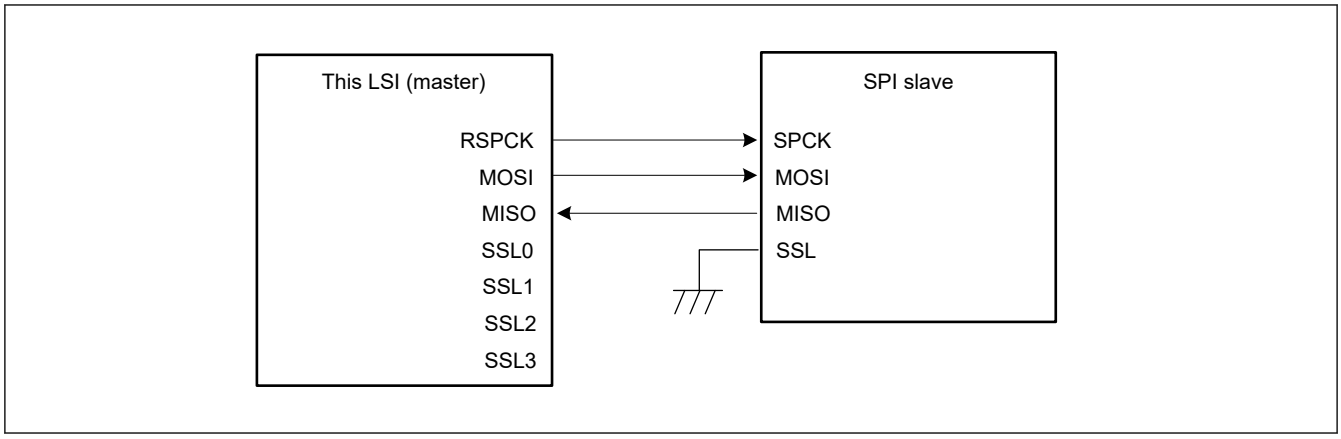


Figure 36.7 Single-master/single-slave configuration example (this LSI = master)

36.4.3.2 Single Master/Single Slave (This LSI = Slave)

Figure 36.8 shows an example of single master/single slave SPI system configuration where this LSI is used as a slave. When this LSI is used as a slave, the SSL0 pin is used as SSL input. The SPI master always drives the RSPCK and MOSI pins. This LSI (slave) always drives the MISO pin. When the SSL0 level is inactive, the pin state becomes Hi-Z.

In the single slave configuration with the SPCMD.CPHA bit set to 1, set the SPCR.SPFRF bit to 0, and set the SPMS bit to 0. There is the SSL0 input level of this LSI (slave) is fixed to 0 so that this LSI (slave) can be always selected and serial transfer can also be performed (Figure 36.9). However, the communication end interrupt does not output when SSL0 input was fixed as Figure 36.9.

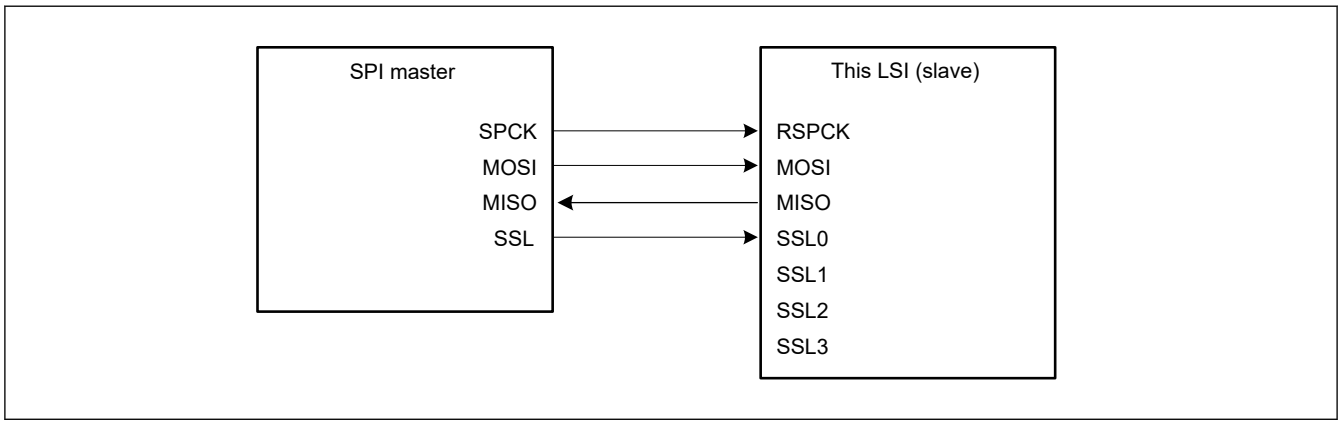


Figure 36.8 Single-master/single-slave configuration example (this LSI = slave, CPHA = 0)

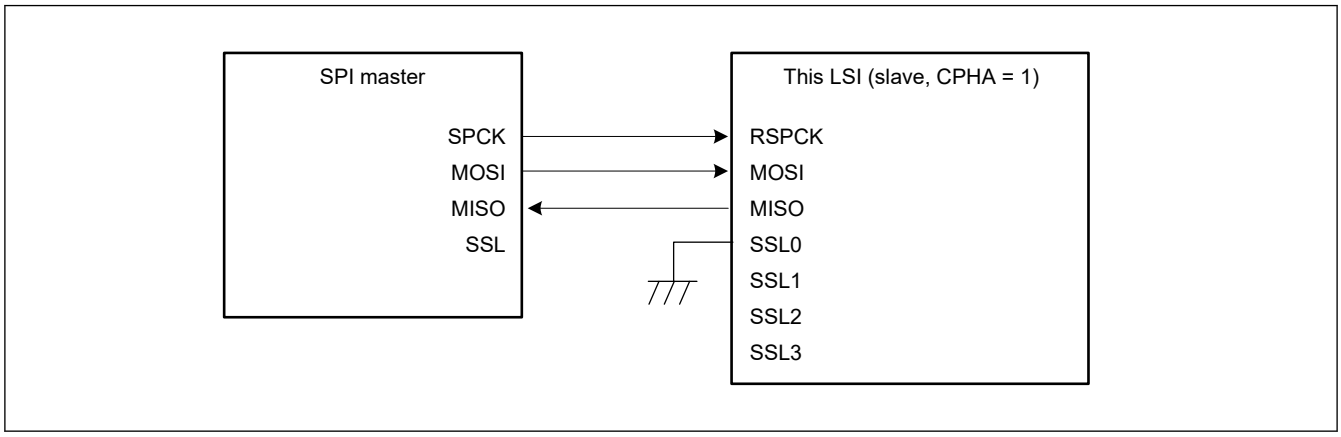


Figure 36.9 Single-master/single-slave configuration example (this LSI = slave, CPHA = 1)

36.4.3.3 Single Master/Multi-Slave (This LSI = Master)

Figure 36.10 shows an example of single master/multi-slave SPI system configuration where this LSI is used as a master. In the example in Figure 36.10, the SPI system is comprised of this LSI (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCK output and MOSI output pins of the master are connected to the RSPCK input and MOSI input pins of SPI slave 0 to SPI slave 3. All of the MISO output pins of SPI slave 0 to SPI slave 3 are connected to the MISO input pin of the master. The SSL0 to SSL3 output pins of this LSI (master) are connected to the SSL input pin of SPI slave 0 to SPI slave 3.

The master always drives RSPCK, MOSI, and SSL0 to SSL3 pins. The slave (out of SPI slave 0 to SPI slave 3) where 0 level is input to SSL input pin drives the MISO pin.

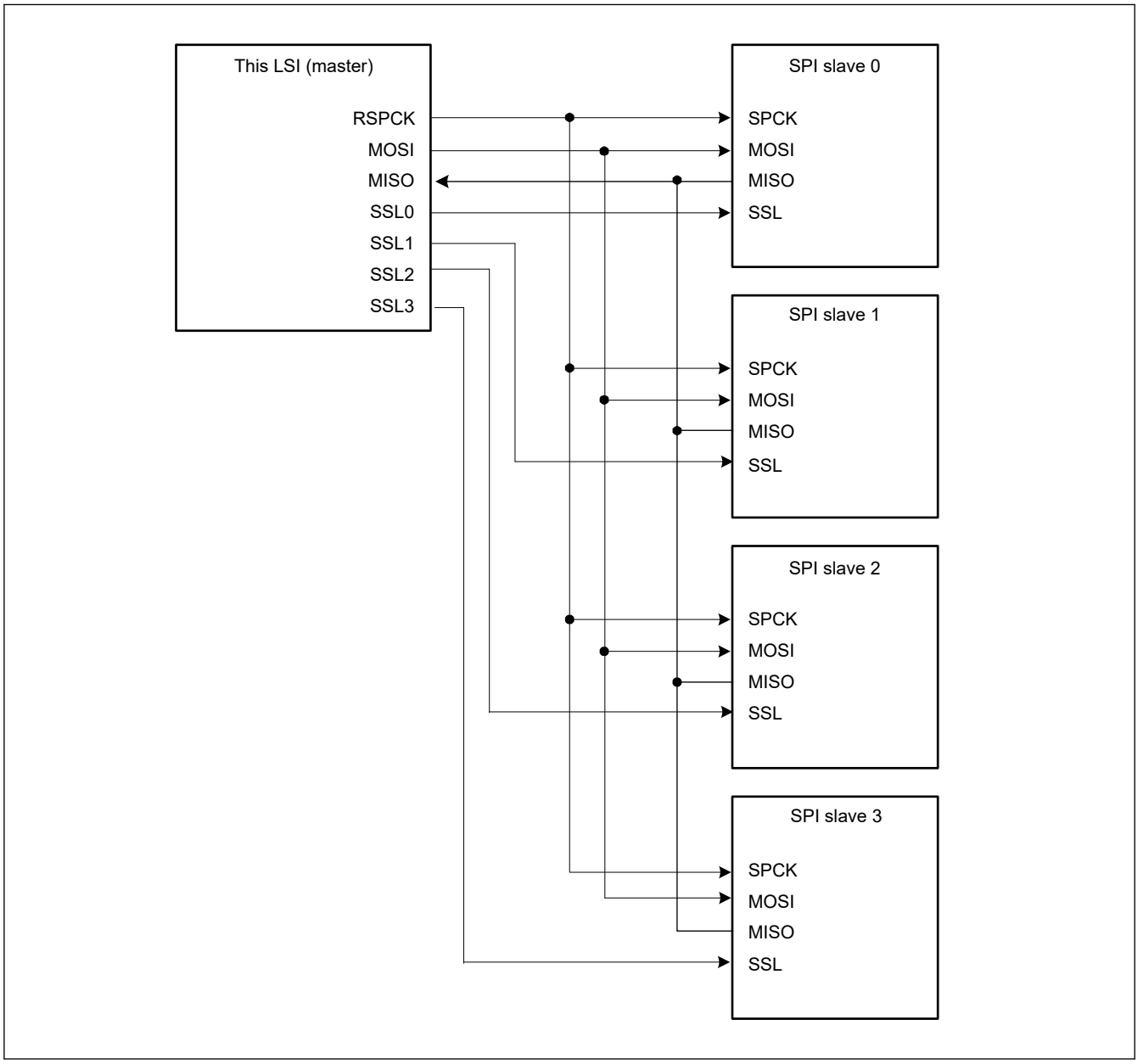


Figure 36.10 Single-master/multi-slave configuration example (this LSI = master)

36.4.3.4 Single Master/Multi-Slave (with This LSI Acting as Slave)

Figure 36.11 shows an example of single master/multi-slave SPI system configuration where this LSI is used as a slave. In the example in Figure 36.11, the SPI system consists of the SPI master and two these LSIs (slave X, slave Y).

The RSPCK output and MOSI output pins of the SPI master are connected to the RSPCK input and MOSI input pins of the LSIs (slave X, slave Y). The MISO output pin of the LSIs (slave X, slave Y) is connected to the MISO input pin of the SPI

master. The SSLX output and SSLY output pins of the SPI master are connected to the SSL0 input pin of the LSIs (slave X, slave Y).

The SPI master always drives the RSPCK, MOSI, SSLX, and SSLY pins. The slave X or slave Y (this LSI) where 0 level is input to the SSL0 input pin drives the MISO pin.

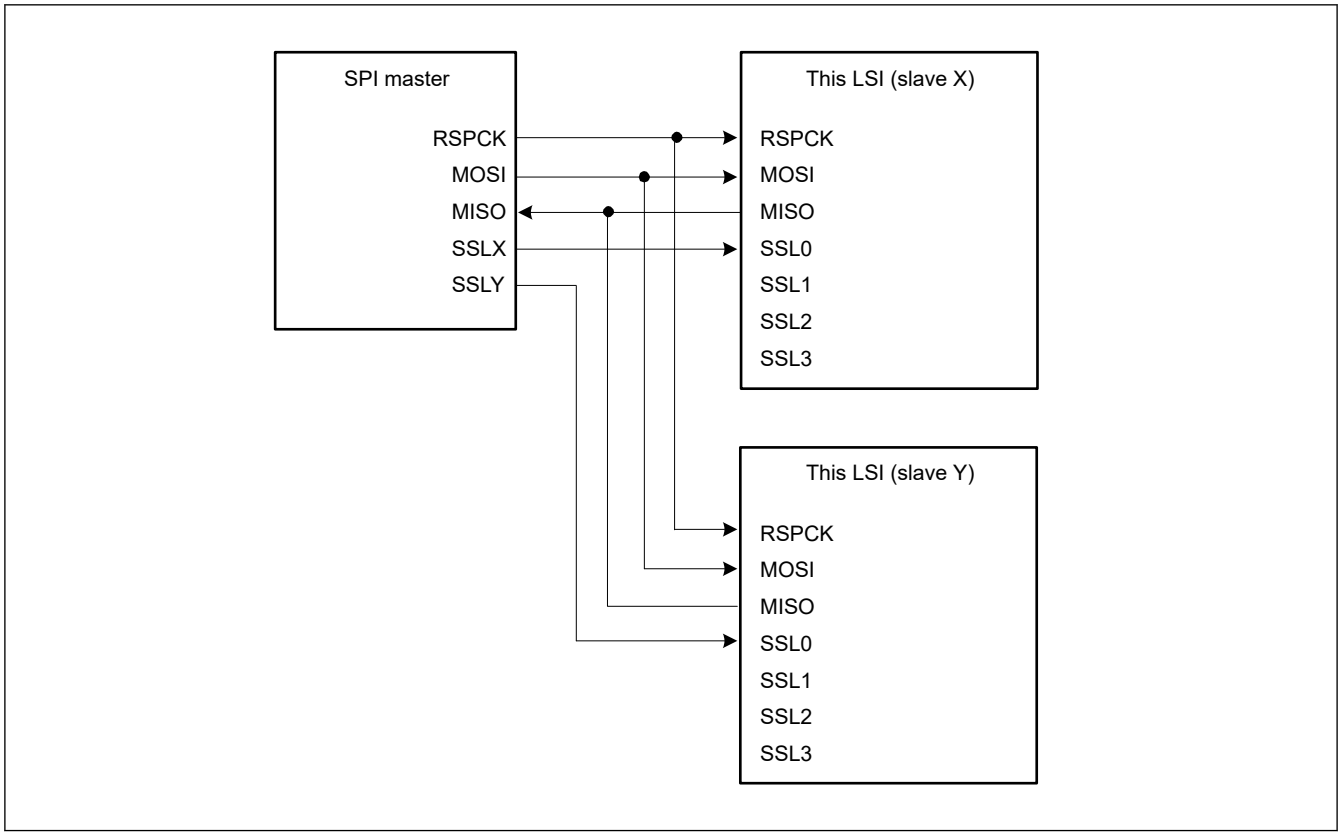


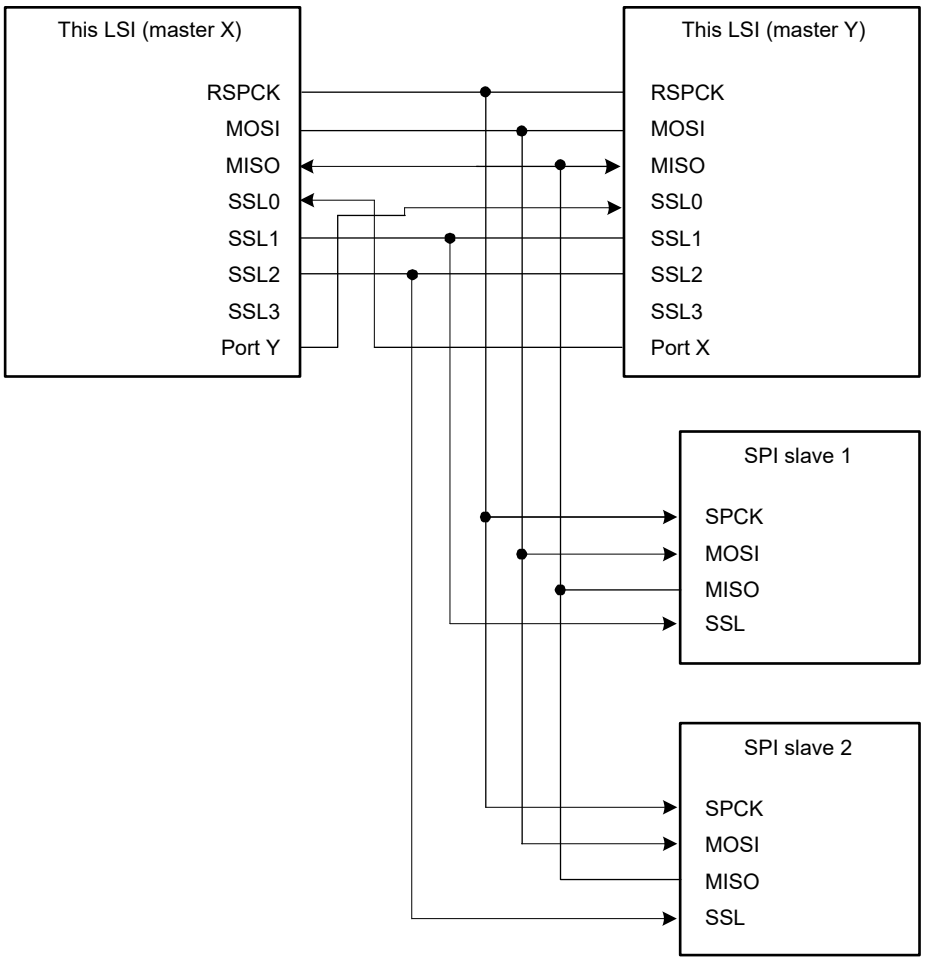
Figure 36.11 Single-master/multi-slave configuration example (this LSI = slave)

36.4.3.5 Multi-Master/Multi-Slave (with This LSI Acting as Master)

Figure 36.12 shows an example of multi-master/multi-slave SPI system configuration where this LSI is used as a master. In the example in Figure 36.12, the SPI system consists of two these LSIs (master X, master Y) and two SPI slaves (SPI slave 1, SPI slave 2).

The RSPCK output and MOSI output pins of these LSIs (master X, master Y) are connected to the RSPCK input and MOSI input pins of SPI slave 1 and SPI slave 2. The MISO output pin of SPI slave 1 and SPI slave 2 is connected to the MISO input pin of these LSIs (master X, master Y). The Port Y (general port) output pin of this LSI (master X) is connected to the SSL0 input pin of this LSI (master Y). The Port X (general port) output pin of this LSI (master Y) is connected to the SSL0 input pin of this LSI (master X). The SSL1 output and SSL2 output pins of these LSIs (master X, master Y) are connected to the SSL input pin of SPI slave 1 and SPI slave 2. In this configuration example, SSL3 output pins of this LSI are not used because the system can be configured only with SSL0 input pin and SSL1 output and SSL2 output pins for connecting slaves.

While the SSL0 input level is 1, this LSI drives the RSPCK, MOSI, SSL1, and SSL2 pins. While the SSL0 input level is 0, this LSI detects a mode fault error and changes the RSPCK, MOSI, SSL1, SSL2 pin levels to Hi-Z to release the SPI bus mastership for another master. SPI slave 1 or SPI slave 2 where 0 level is input to the SSL input pin drives the MISO pin.



Note: When setting TI SSP, enter the following levels for port X and port Y.
- Start of communication: the value of SSLP.SSL0P of the other master.
- End of communication: the inverted value of SSLP.SSL0P of the other master.

Figure 36.12 Multi-master/multi-slave configuration example (this LSI = master)

36.4.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (This LSI = Master)

Figure 36.13 shows an example of master (clock synchronous operation)/slave (clock synchronous operation) SPI system configuration where this LSI is used as a master. In this configuration, the SSL0 to SSL3 output pins of this LSI (master) are not used.

This LSI (master) always drives the RSPCK and MOSI pins. The SPI slave always drives the MISO pin.

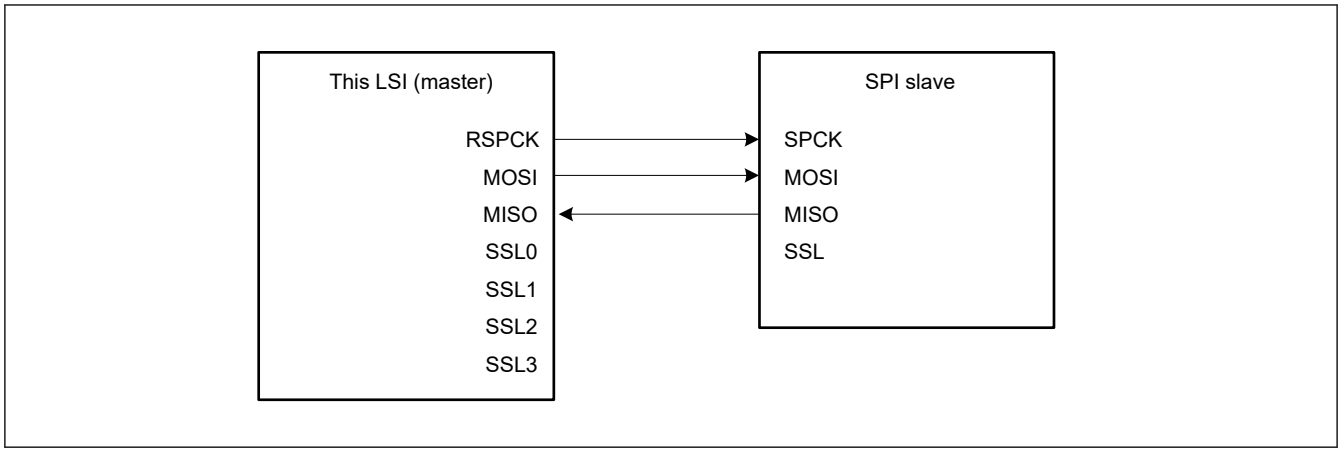


Figure 36.13 Master (clock synchronous operation)/slave (clock synchronous operation) configuration example (this LSI = master)

36.4.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (This LSI = Slave)

Figure 36.14 shows an example of master (clock synchronous operation)/slave (clock synchronous operation) SPI system configuration where this LSI is used as a slave. When this LSI is used as a slave (clock synchronous operation), this LSI (slave) always drives the MISO pin and the SPI master always drives the RSPCK and MOSI pins.

Only in the single slave configuration with the SPCMD.CPHA bit set to 1, this LSI (slave) can perform serial transfer.

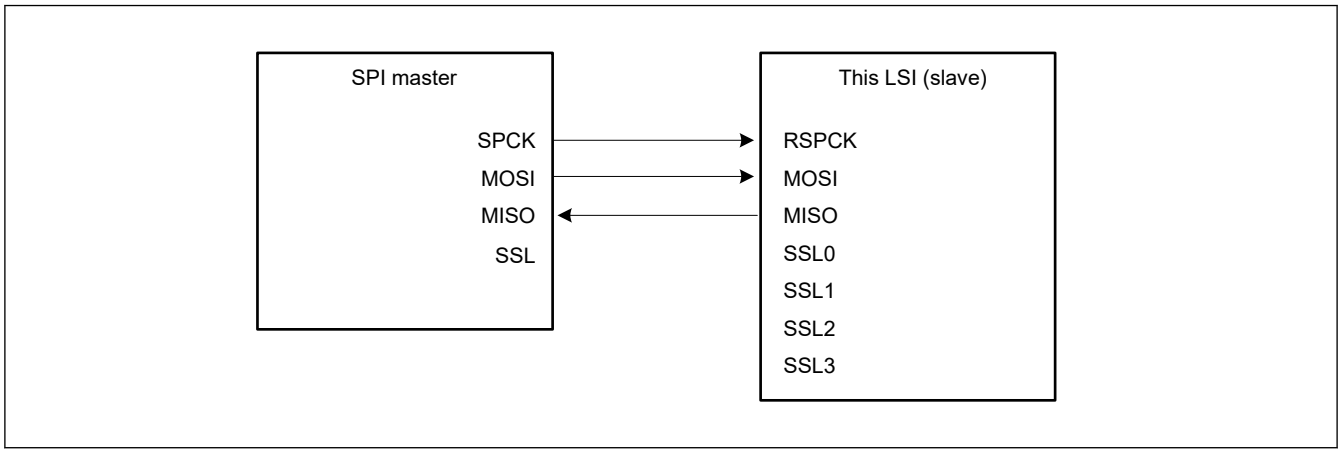


Figure 36.14 Master (clock synchronous operation)/slave (clock synchronous operation) configuration example (this LSI = slave, CPHA = 1)

36.4.4 Data Format

Data format of the SPI depends on the set values of the SPCMD register and the SPCR.SPPE bit. The SPI handles data of the set data length from the LSB in the SPDR register as transfer data regardless of MSB first or LSB first.

36.4.4.1 Data Format of One Frame

The data format of one frame for transmission and reception is shown below.

(1) With Parity Disabled (SPPE=0)

When the parity function is disabled, data of the bit length specified by SPCMDm.SPB[4:0] is transmitted and received.

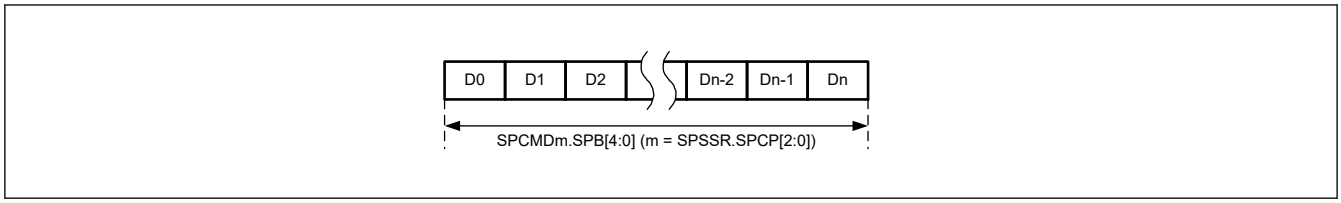


Figure 36.15 Outline of data format (when parity function is disabled)

(2) With Parity Enabled (SPPE = 1)

When the parity function is enabled, data of the bit length specified by SPCMDm.SPB[4:0] is transmitted and received. However, the final bit is used as a parity bit.

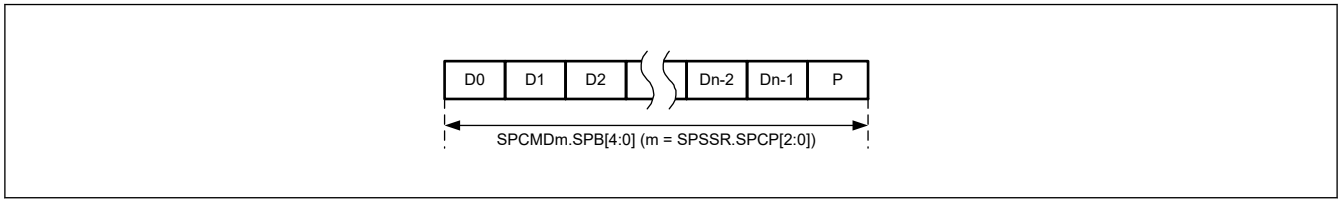


Figure 36.16 Outline of data format (when parity function is enabled)

36.4.4.2 When Parity is Disabled (SPPE = 0)

When the parity function is disabled, transmit buffer data is copied to the shift register without processing it. The following describes the relationship between SPDR, and shift register by using a combination of MSB first/LSB first and bit length.

(1) MSB First Transfer (32-Bit Data)

Figure 36.17 shows operations of the SPDR register and the shift register when the SPI performs 32-bit MSB-first transfer with the parity function disabled.

In transmission, T31 to T00 in the transmit buffer are copied to the shift register. Bit values in the shift register are shifted and transmitted in the order of T31 → T30 → … → T00 as transmit data.

In reception, received data is stored in bit 0 of the shift register, and received data is shifted in units of data. When necessary RSPCK cycles are input and data is stored from R31 to R00, the shift register value is copied to the receive buffer.

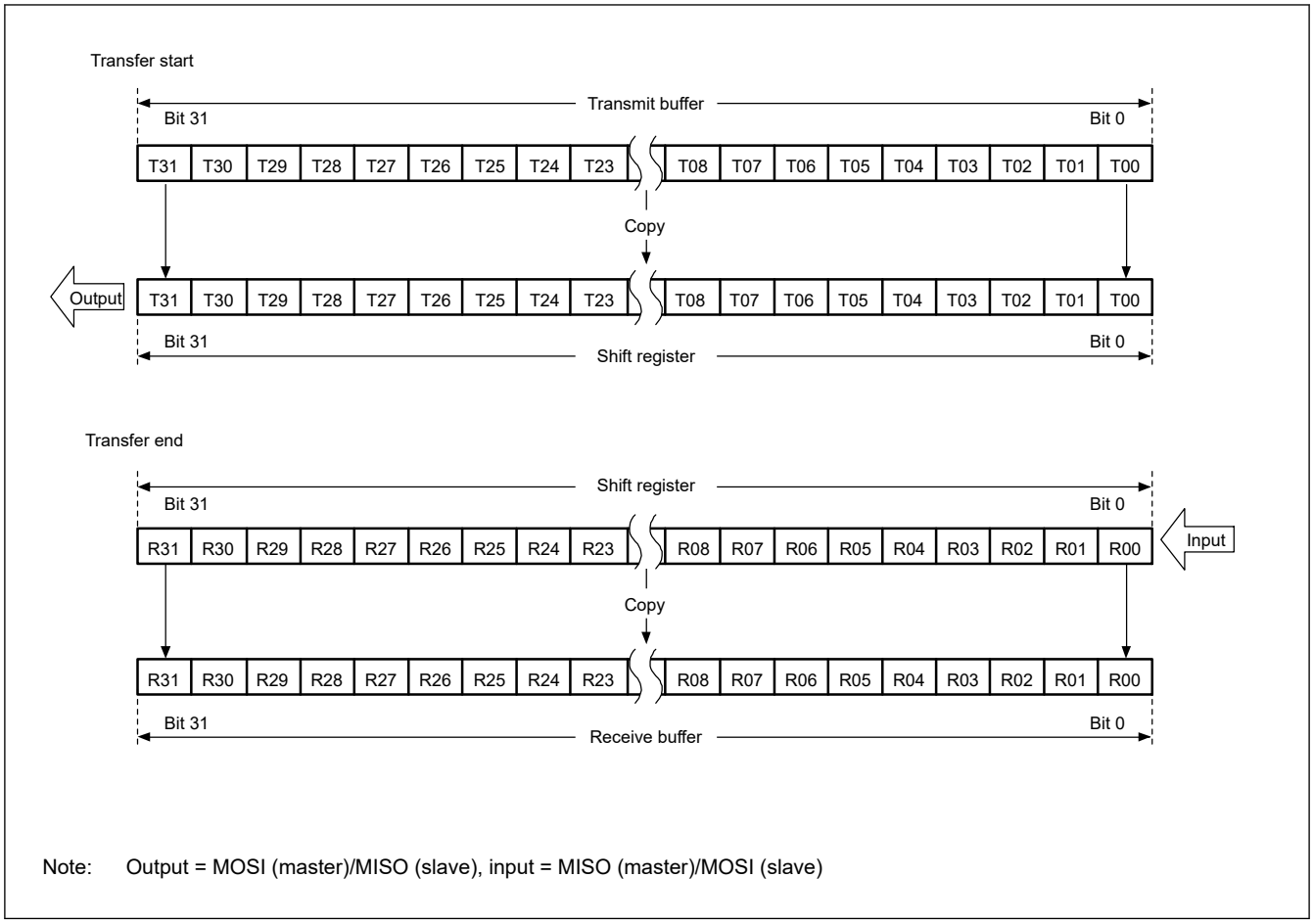


Figure 36.17 MSB first transfer (32-bit data, parity disabled)

(2) MSB First Transfer (24-Bit Data)

As an example of MSB-first transfer of data other than 32 bits with the parity function disabled, Figure 36.18 shows operations of the SPDR register and the shift register when the SPI performs 24-bit data transfer.

In transmission, lower 24 bits (T23 to T00) in the transmit buffer are copied to the shift register. Bit values in the shift register are shifted and transmitted in the order of T23 → T22 → … → T00 as transmit data.

In reception, received data is stored in bit 0 of the shift register, and received data is shifted in units of data. When necessary RSPCK cycles are input and data is stored from R23 to R00, the shift register value is copied to the receive buffer.

Figure 36.18 MSB first transfer (24-bit data, parity disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 36.19 shows operations of the SPDR register and the shift register when the SPI performs 32-bit LSB-first transfer with the parity function disabled.

In transmission, bit values of the transmit buffer (T31 to T00) are reversed in bit units and are copied to the shift register in the order of T00 to T31. Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … → T31 as transmit data.

In reception, the first received data is stored in bit 0 of the shift register, and received data is shifted in units of data.

When necessary RSPCK cycles are input and data is stored from R00 to R31, bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of R31 to R00.

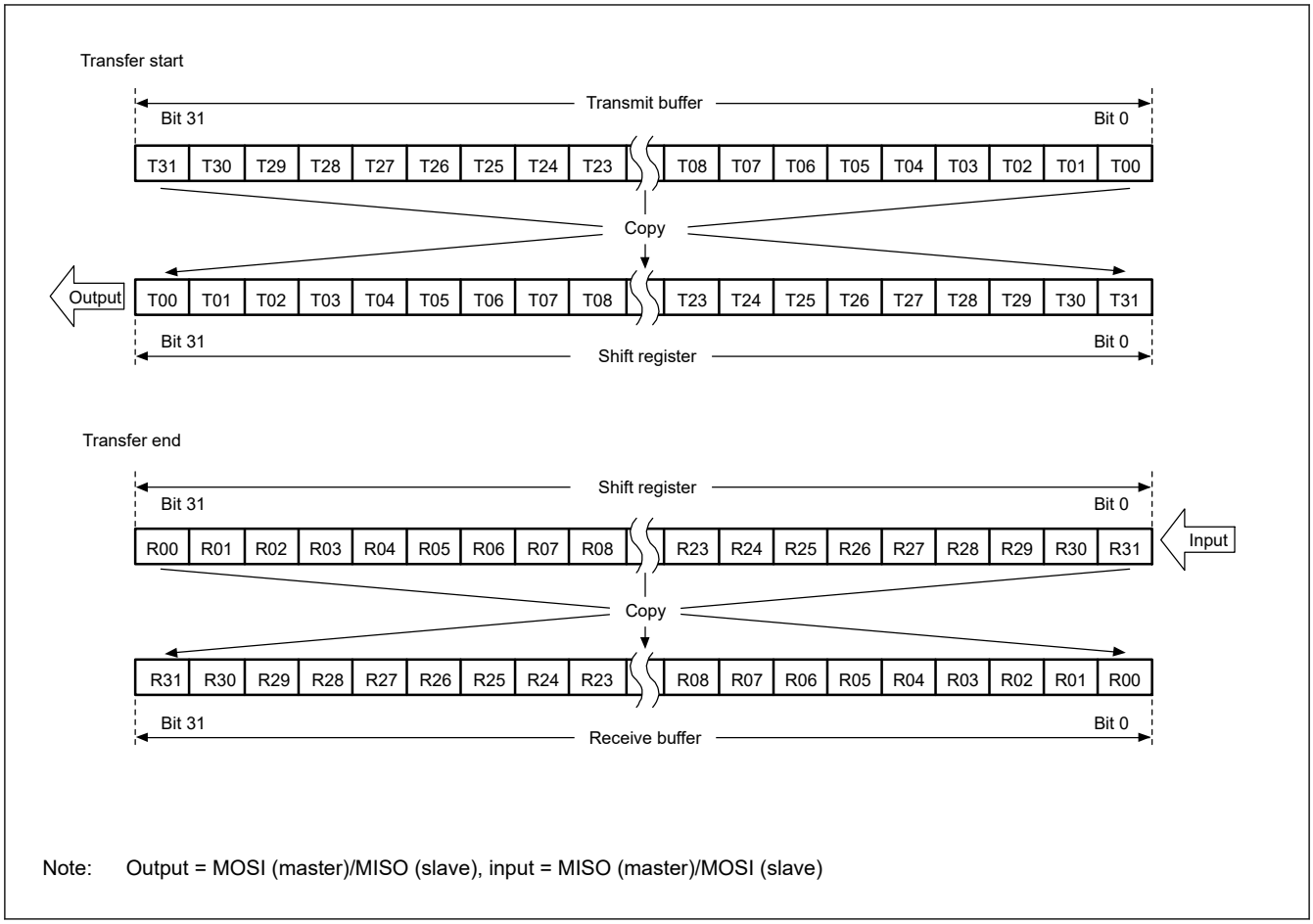


Figure 36.19 LSB first transfer (32-bit data, parity disabled)

(4) LSB First Transfer (24-Bit Data)

As an example of LSB-first transfer of data other than 32 bits with the parity function disabled, Figure 36.20 shows operations of the SPDR register and the shift register when the SPI performs 24-bit data transfer.

In transmission, lower 24-bit values of the transmit buffer (T23 to T00) are reversed to the order from T00 to T23 in bit units and are copied to the shift register. Bit values in the shift register are shifted and transmitted in the order of T00→ T01 → … → T23 as transmit data.

In reception, received data is stored in bit 8 of the shift register, and received data is shifted in units of data. When necessary RSPCK cycles are input and data is stored from R00 to R23, bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of R23 (bit 23) to R00 (bit 0).

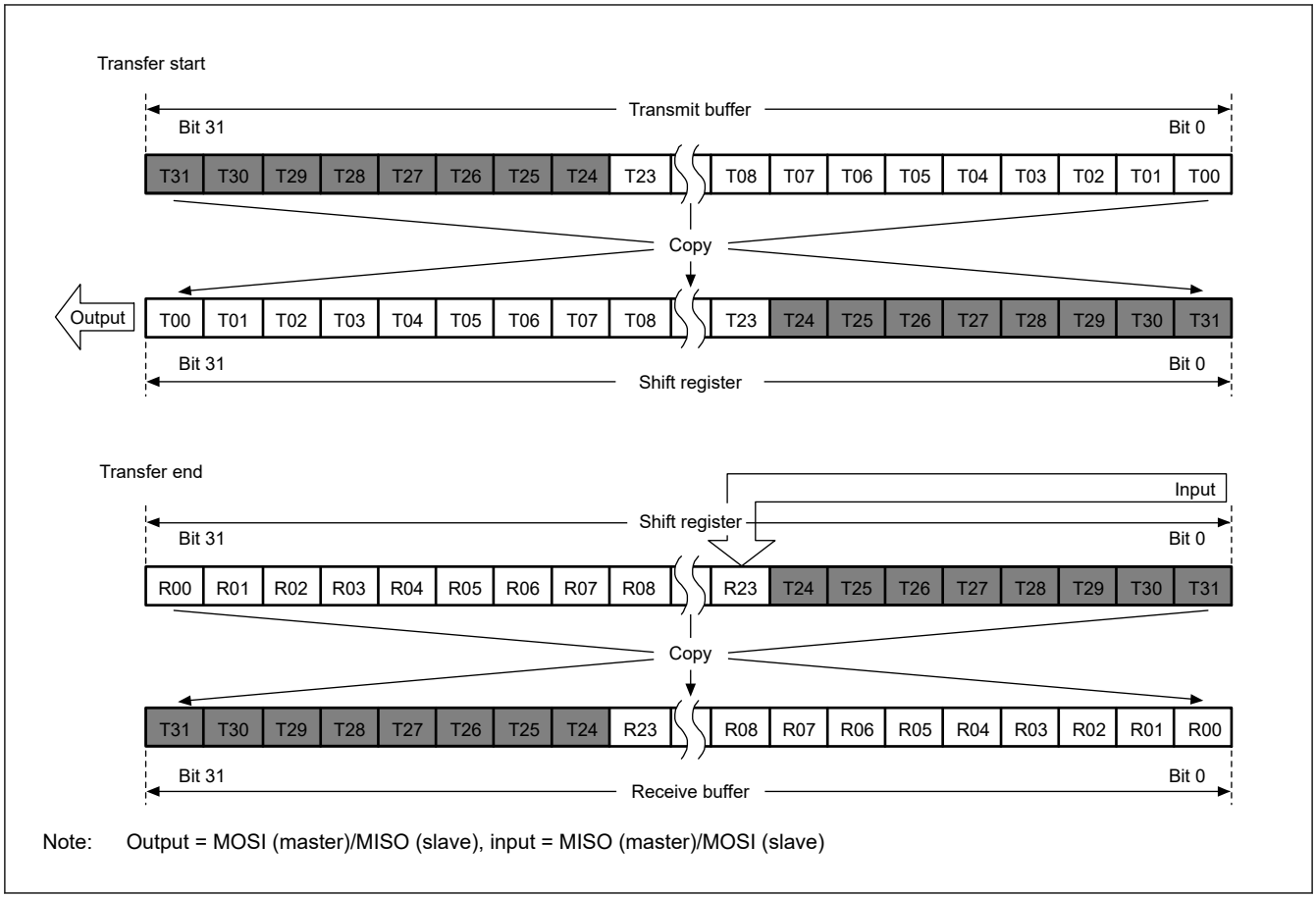


Figure 36.20 LSB first transfer (24-bit data, parity disabled)

36.4.4.3 When Parity is Enabled (SPPE = 1)

When the parity function is enabled, the LSB of transmit data or received data is converted to a parity bit. The parity bit value is calculated and converted by the hardware.

(1) MSB First Transfer (32-Bit Data)

Figure 36.21 shows operations of the SPDR register and the shift register when the SPI performs 32-bit MSB-first transfer with the parity function enabled.

In transmission, the parity bit (P) value is calculated first from the data value T31 to T01, then the final bit T00 is replaced with the calculated P value, and then transmit data is copied to the shift register. Transmit data is transmitted in the order of T31 → T30 → … → T01 → P.

In reception, received data is stored in bit 0 of the shift register, and is then shifted in units of data. When necessary RSPCK cycles are input and received data is stored from R31 to P, the shift register value is copied to the receive buffer, and at the same time, R31 to P data is checked to determine whether a parity error is present.

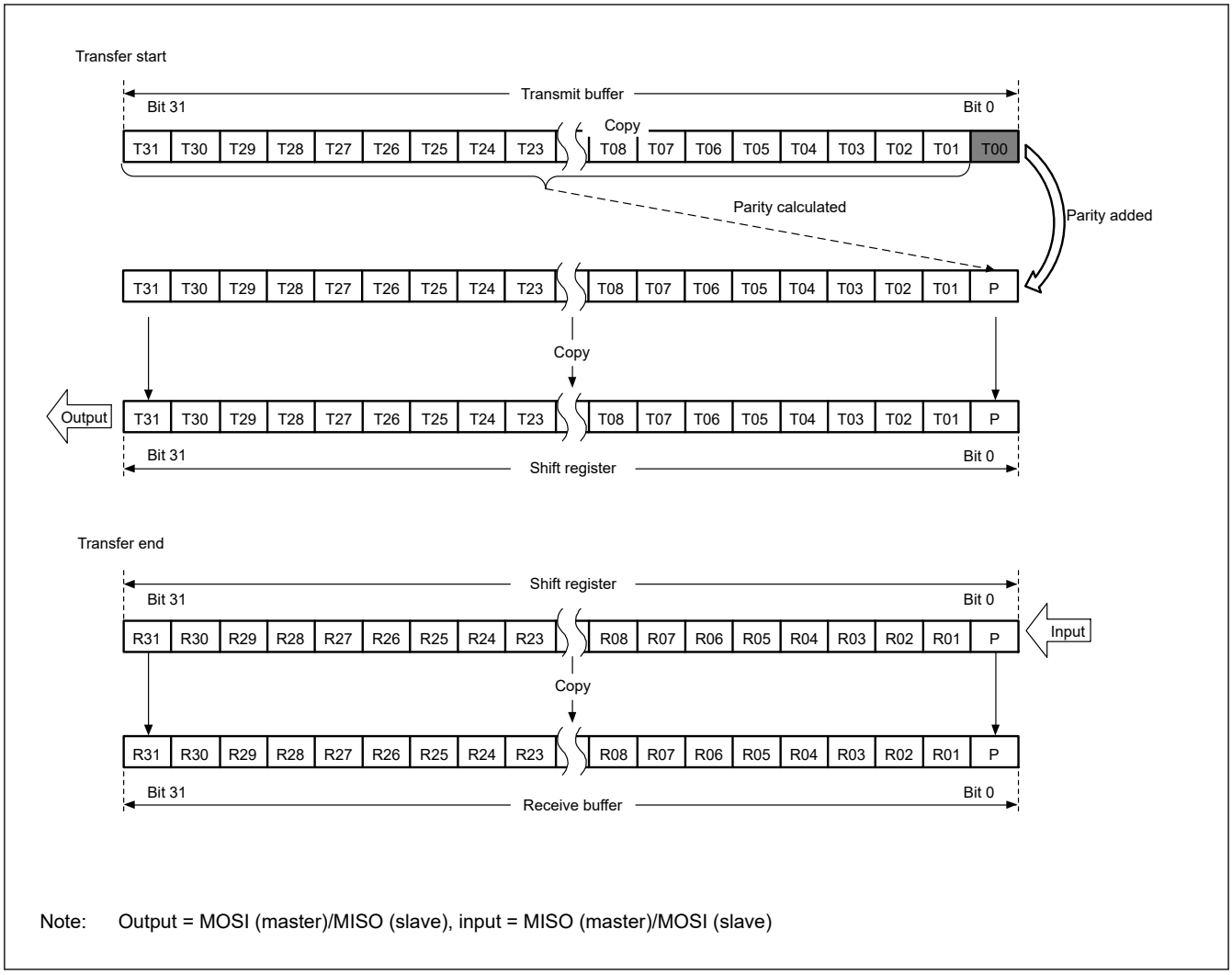


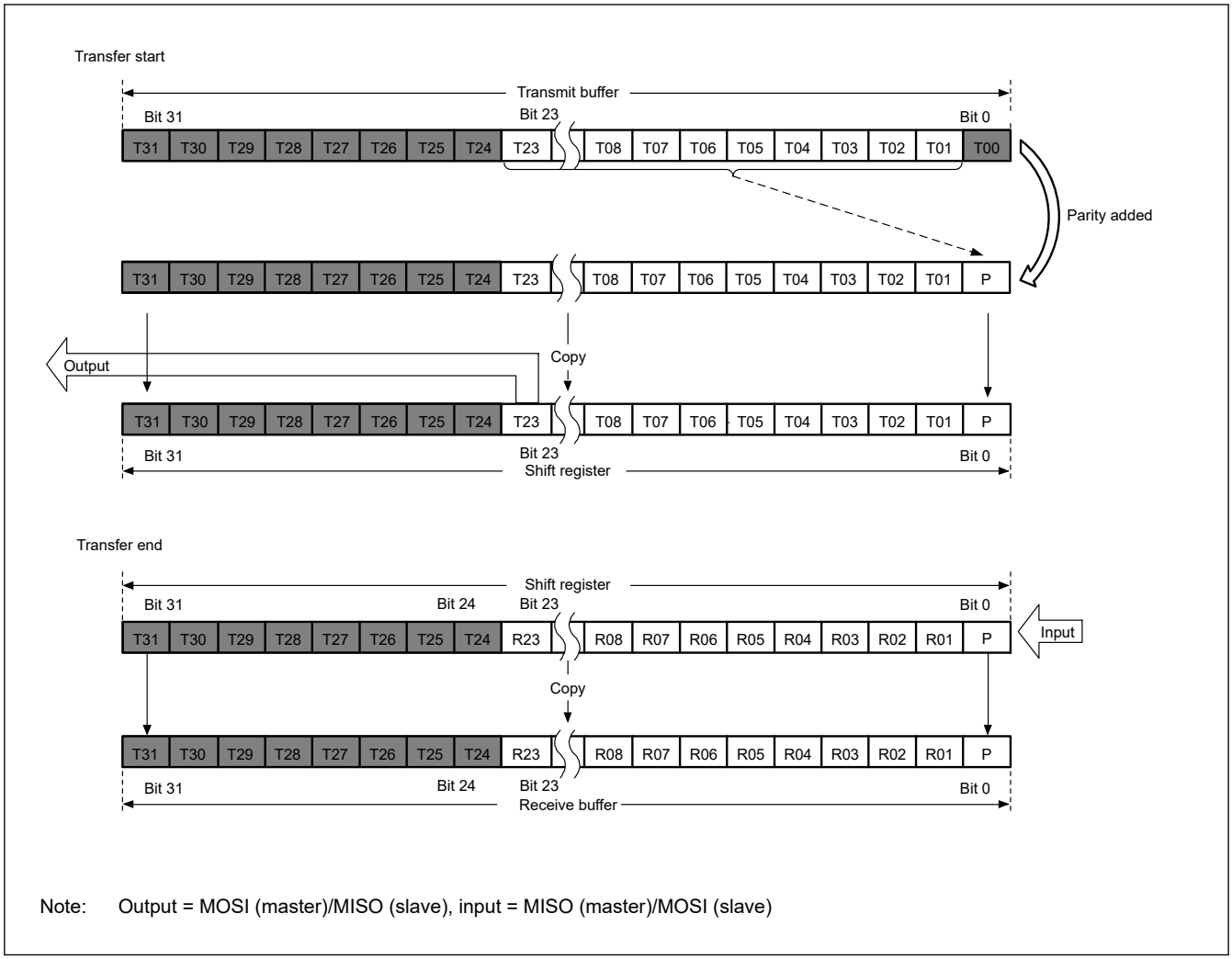
Figure 36.21 MSB first transfer (32-bit data, parity enabled)

(2) MSB First Transfer (24-Bit Data)

As an example of MSB-first transfer of data other than 32 bits with the parity function enabled, Figure 36.22 shows operations of the SPDR register and the shift register when the SPI performs 24-bit data transfer.

In transmission, the parity bit (P) value is calculated first from the data value T23 to T01, then the final bit T00 is replaced with the calculated P value, and then transmit data is copied to the shift register. Transmit data is transmitted in the order of T23 → T22 → … → T01 → P.

In reception, received data is stored in bit 0 of the shift register, and is then shifted in units of data. When necessary RSPCK cycles are input and received data is stored from R23 to P, the shift register value is copied to the receive buffer, and at the same time, R23 to P data is checked to determine whether a parity error is present.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

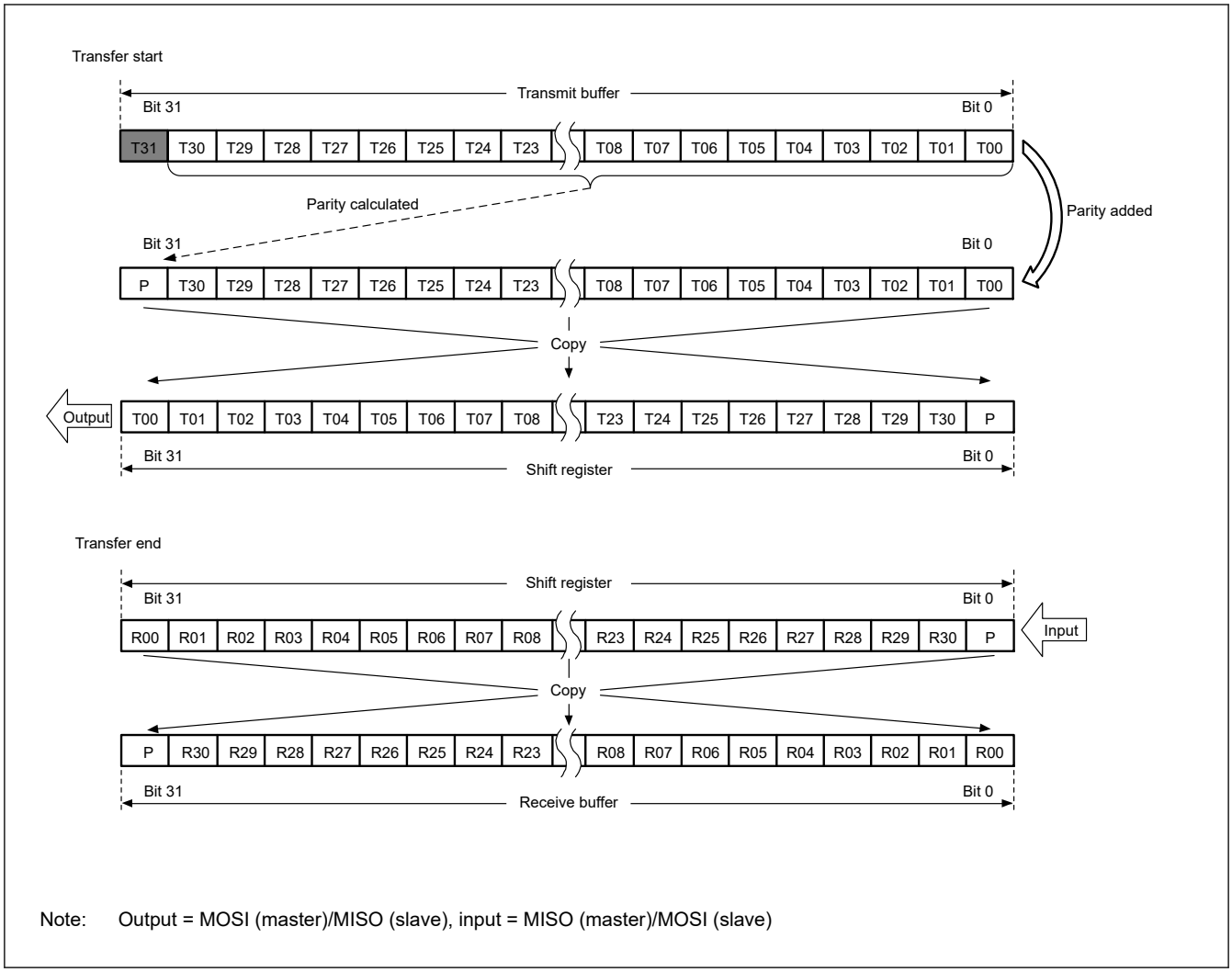
Figure 36.22 MSB first transfer (24-bit data, parity enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 36.23 shows operations of the SPDR register and the shift register when the SPI performs 32-bit LSB-first transfer with the parity function enabled.

In transmission, the parity bit (P) value is calculated first from the data value T30 to T00, then the final bit T31 is replaced with the calculated P value, and then transmit data is copied to the shift register. Transmit data is transmitted in the order of T00 → T01 → … → T30 → P.

In reception, received data is stored in bit 0 of the shift register, and is then shifted in units of data. When necessary RSPCK cycles are input and received data is stored from R00 to P, bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of P to R00, and at the same time, R00 to P data is checked to determine whether a parity error is present.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

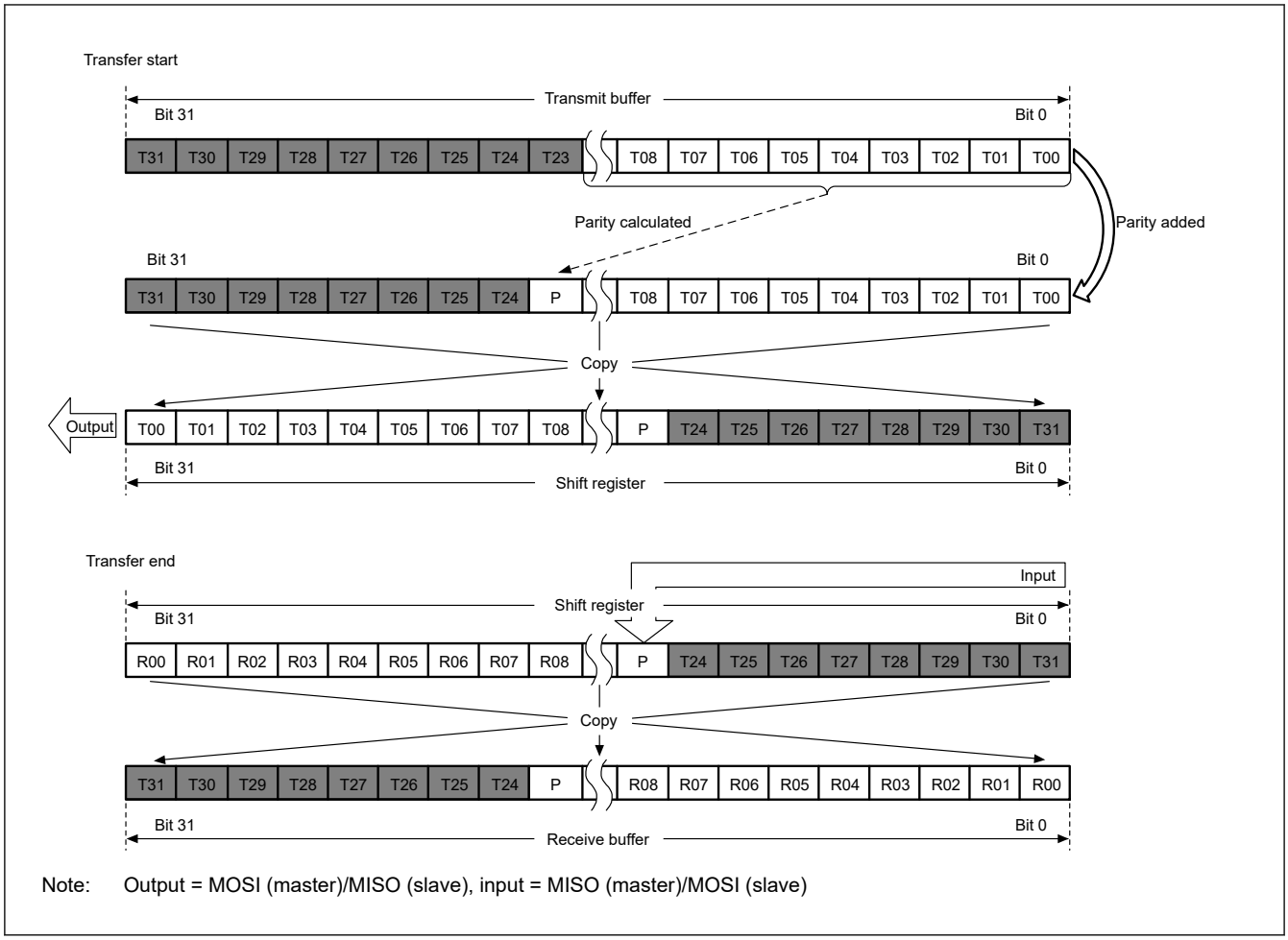
Figure 36.23 LSB first transfer (32-bit data, parity enabled)

(4) LSB First Transfer (24-Bit Data)

As an example of LSB-first transfer of data other than 32 bits with the parity function enabled, Figure 36.24 shows operations of the SPDR register and the shift register when the SPI performs 24-bit data transfer.

In transmission, the parity bit (P) value is calculated first from the data value T22 to T00, then the final bit T23 is replaced with the calculated P value, and then transmit data is copied to the shift register. Transmit data is transmitted in the order of T00→ T01 → … → T22 → P.

In reception, received data is stored in bit 8 of the shift register, and is then shifted in units of data. When necessary RSPCK cycles are input and received data is stored from R00 to P, bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of P (bit 23) to R00 (bit 0), and at the same time, R00 to P data is checked to determine whether a parity error is present.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 36.24 LSB first transfer (24-bit data, parity enabled)

36.4.4.4 Byte swap transmission

When byte swapping is enabled, the data in the transmission buffer, swapped in 8-bit units, is copied to the shift register.

(1) 32-bit data length

Figure 36.25 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 32-bit data length, using a combination of MSB/LSB first and with/without byte swap.

1. **MSB First Transfer (When the byte swap is disabled.)**
   Data (Byte3 [T31 to T24] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register.
   Bit values in the shift register are shifted and transmitted in the order of T31 → T30 → … → T00 as transmit data.
2. **MSB First Transfer (When the byte swap is enabled.)**
   Byte values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte3 [T07 to T00] to Byte0 [T31 to T24].
   Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → … → T00 → T15 → T14→ … → T08 → T23 → T22 → … → T16 → T31 → T30 → … → T24 as transmit data.
3. **LSB-first transfer (When the byte swap is disabled.)**
   Bit values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte3 [T24 to T31].
   Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … → T31 as transmit data.
4. **LSB-first transfer (When the byte swap is enabled.)**
   Bit values of each byte of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte3 [T24 to T31] to Byte0 [T00 to T07].
   Bit values in the shift register are shifted and transmitted in the order of T24 → T25 → … → T31 → T16 → T17 → … → T23 → T08 → T09 → … → T15 → T00 → T01 → … → T07 as transmit data.

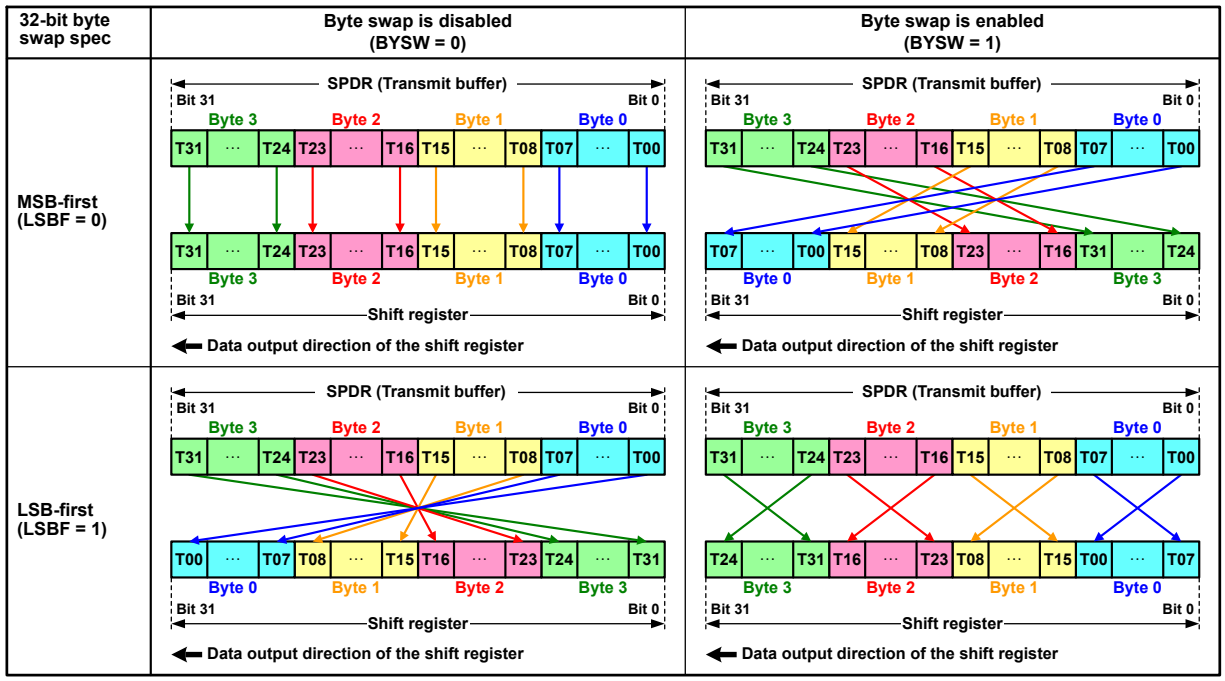


Figure 36.25 Byte swap with MSB/LSB transfer (32 bit)

(2) 16-bit data length

Figure 36.26 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 16-bit data length, using a combination of MSB/LSB first and with/without byte swap.

1. **MSB First Transfer (When the byte swap is disabled.)**
   Data (Byte1 [T15 to T08] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register.
   Bit values in the shift register are shifted and transmitted in the order of T15 → T14 → … → T00 as transmit data.
2. **MSB First Transfer (When the byte swap is enabled.)**
   Byte values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte1 [T07 to T00] to Byte0 [T15 to T08].
   Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → … → T00 → T15 → T14→ … → T08 as transmit data.
3. **LSB-first transfer (When the byte swap is disabled.)**
   Bit values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte1 [T08 to T15].
   Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … → T15 as transmit data.
4. **LSB-first transfer (When the byte swap is enabled.)**
   Bit values of each byte of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte1 [T08 to T15] to Byte0 [T00 to T07].
   Bit values in the shift register are shifted and transmitted in the order of T08 → T09 → … → T15 → T00 → T01 → … → T07 as transmit data.

Figure 36.26 Byte swap with MSB/LSB transfer (16 bit)

Note: When using the byte swap, set 16 bits or 32 bits to the data length (SPCMD.SPB[4:0] setting).
If setting the other length, the behavior is not guaranteed.

Note: When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0) .
If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.

Note: Set SPDCR.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

36.4.4.5 Byte swap reception

When byte swap is enabled, the data in the shift register, swapped in 8-bit units, is copied to the receive buffer.

(1) 32-bit data length

Figure 36.27 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 32-bit data length, using a combination of MSB/LSB first and with/without byte swap.

1. **MSB First Transfer (When the byte swap is disabled.)**
   The first received data (R31) is stored in bit 0 of the shift register, and received data is shifted in the order of R31 → R30 → … → R00.
   When necessary RSPCK cycles are input and data is stored from Byte3 [R31 to R24] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.
2. **MSB First Transfer (When the byte swap is enabled.)**
   The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → … → R00 → R15 → R14 → … → R08 → R23 → R22 → … → R16 → R31 → R30 → … → R24.
   When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte3 [R31 to R24], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].
3. **LSB-first transfer (When the byte swap is disabled.)**
   The first received data (R00) is stored in bit 0 of the shift register, and received data is shifted in the order of R00 → R01 → … → R31.

When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte3 [R24 to R31], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

4. **LSB-first transfer (When the byte swap is enabled.)**
   The first received data (R24) is stored in bit 0 of the shift register, and received data is shifted in the order of R24 → R25 → … → R31 → R16 → R17 → … → R23 → R08 → R09 → … → R15 → R00 → R01 → … → R07.
   When necessary RSPCK cycles are input and data is stored from Byte3 [R24 to R31] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

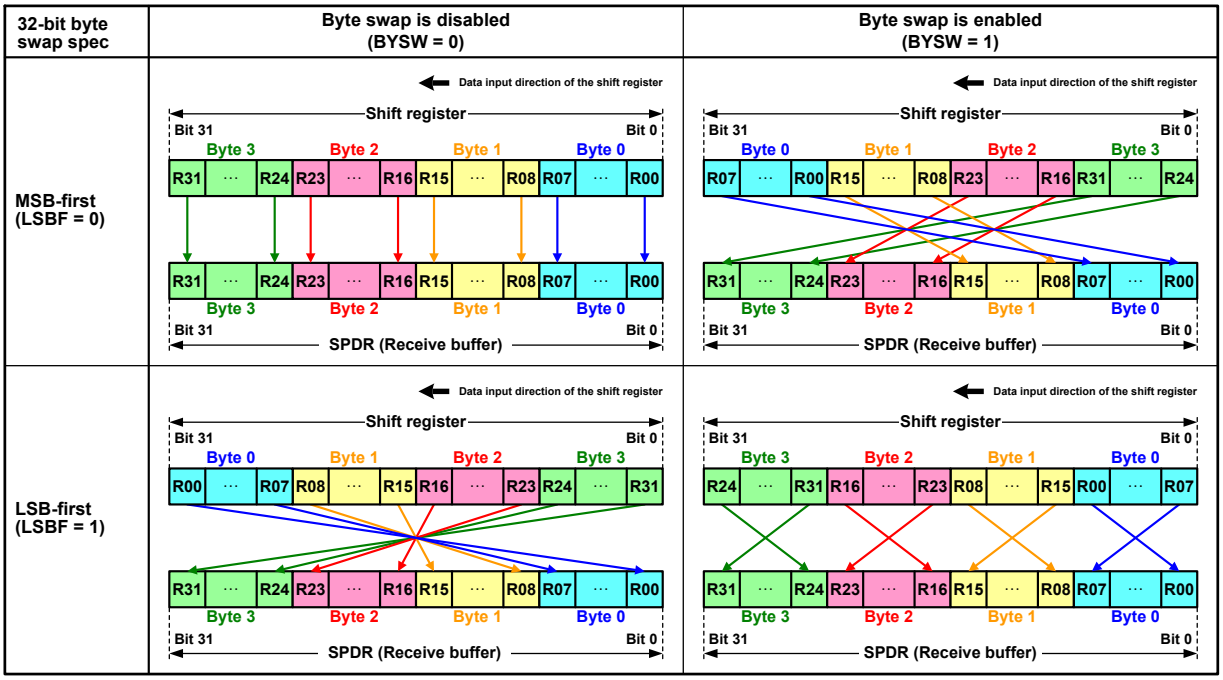


Figure 36.27 Byte swap with MSB/LSB reception (32 bit)

(2) 16-bit data length

Figure 36.28 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 16-bit data length, using a combination of MSB/LSB first and with/without byte swap.

1. **MSB First Transfer (When the byte swap is disabled.)**
   The first received data (R15) is stored in bit 0 of the shift register, and received data is shifted in the order of R15 → R14 → … → R00.
   When necessary RSPCK cycles are input and data is stored from Byte1 [R15 to R08] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.

2. **MSB First Transfer (When the byte swap is enabled.)**
   The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → … → R00 → R15 → R14 → … → R08.
   When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte1 [R15 to R08], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte1 [R15 to R08] to Byte0 [R07 to R00].

3. **LSB-first transfer (When the byte swap is disabled.)**
   The first received data (R00) is stored in bit 15 of the shift register, and received data is shifted in the order of R00 → R01 → … → R15.
   When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte1 [R08 to R15], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte1 [R15 to R08] to Byte0 [R07 to R00].

4. **LSB-first transfer (When the byte swap is enabled.)**
   The first received data (R08) is stored in bit 0 of the shift register, and received data is shifted in the order of R08 → R09 → … → R15 → R00 → R01 → … → R07.
   When necessary RSPCK cycles are input and data is stored from Byte1 [R08 to R15] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte1 [R15 to R08] to Byte0 [R07 to R00].

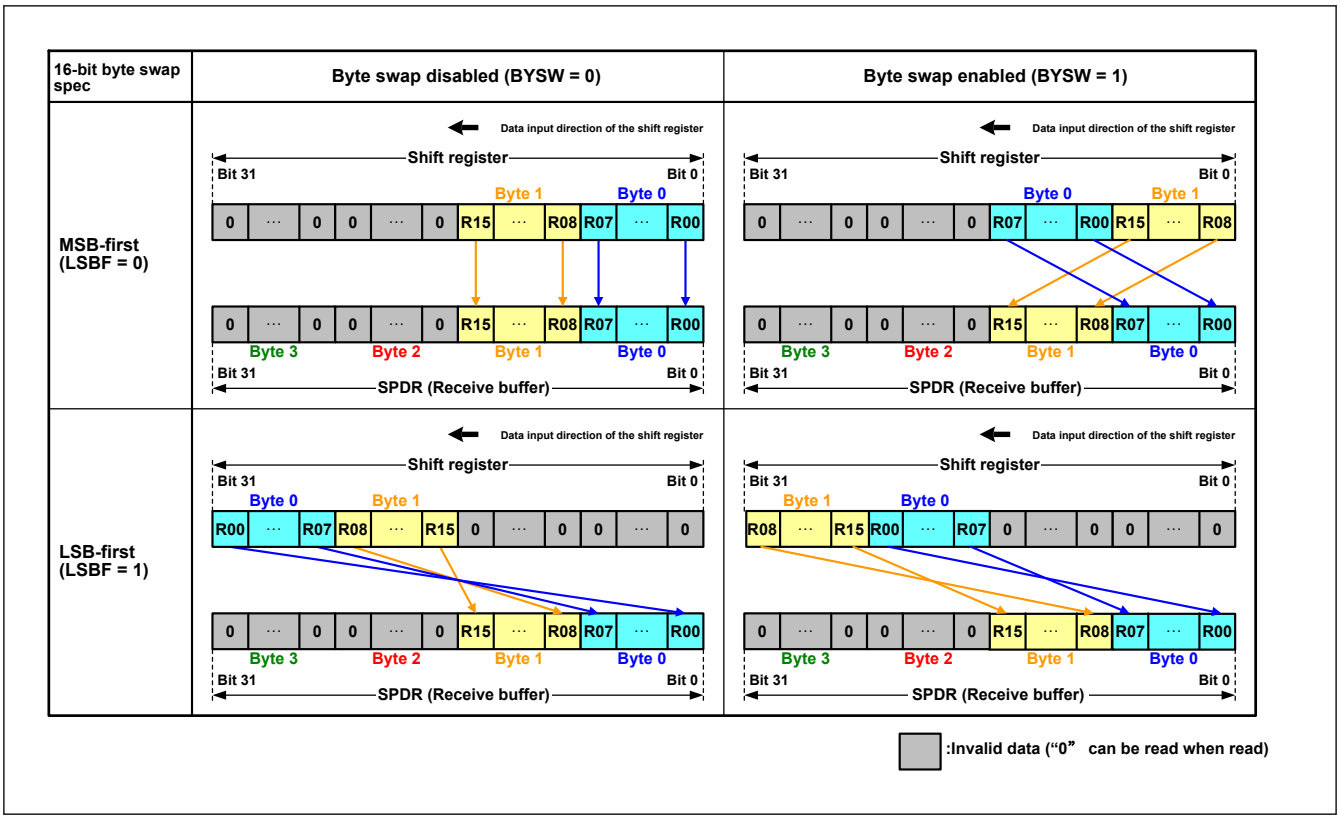


Figure 36.28 Byte swap with MSB/LSB reception (16 bit)

Note: When using the byte swap, set 16 bits or 32 bits to the data length (SPCMD.SPB[4:0] setting) .
If setting the other length, the behavior is not guaranteed.

Note: When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0).
If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.

Note: Set SPDCR2.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

36.4.5 Transfer Format (Frame Format)

36.4.5.1 CPHA = 0

Figure 36.29 shows an example of transfer format of 8-bit data serial transfer when SPCMD.CPHA = 0. However, when the SPI is in slave mode (SPCR.MSTR = 0) with the CPHA bit set to 0, clock synchronous operation (SPCR.SPMS = 1) is not guaranteed. In Figure 36.29, RSPCK (CPOL = 0) is the RSPCK signal waveform when SPCMD.CPOL = 0, and RSPCK (SPCMD.CPOL = 1) is the RSPCK signal waveform when SPCMD.CPOL = 1. The sampling timing shows the time when the SPI latches serial transfer data in the shift register. Input and output directions of each signal depend on SPI settings. (See section 36.4.2. SPI Pin Control.)

When SPCMD.CPHA = 0, output of valid data to the MOSI signal and valid data drive to the MISO signal start at the SSL signal assertion timing. The first RSPCK signal change timing after the SSL signal is asserted is the first transfer data latch timing. After this timing, data is sampled in each RSPCK cycle. The MOSI signal and the MISO signal change always after the half RSPCK cycles of the transfer data latch timing. The set CPOL bit value does not affect the RSPCK signal operation timing but affects only the signal polarity.

Period t1 is the period (RSPCK delay) from SSL signal assertion to RSPCK oscillation. Period t2 is the period (SSL negation delay) from RSPCK oscillation stop to SSL signal negation. Period t3 is the period (next-access delay) to inhibit SSL signal assertion for the next transfer after serial transfer is completed. Periods t1, t2, and t3 are controlled by the master device in the SPI system. For t1, t2, and t3 when this LSI (SPI) is in master mode, see section 36.4.13.1. Master Mode Operation.

[In the Motorola-SPI case]

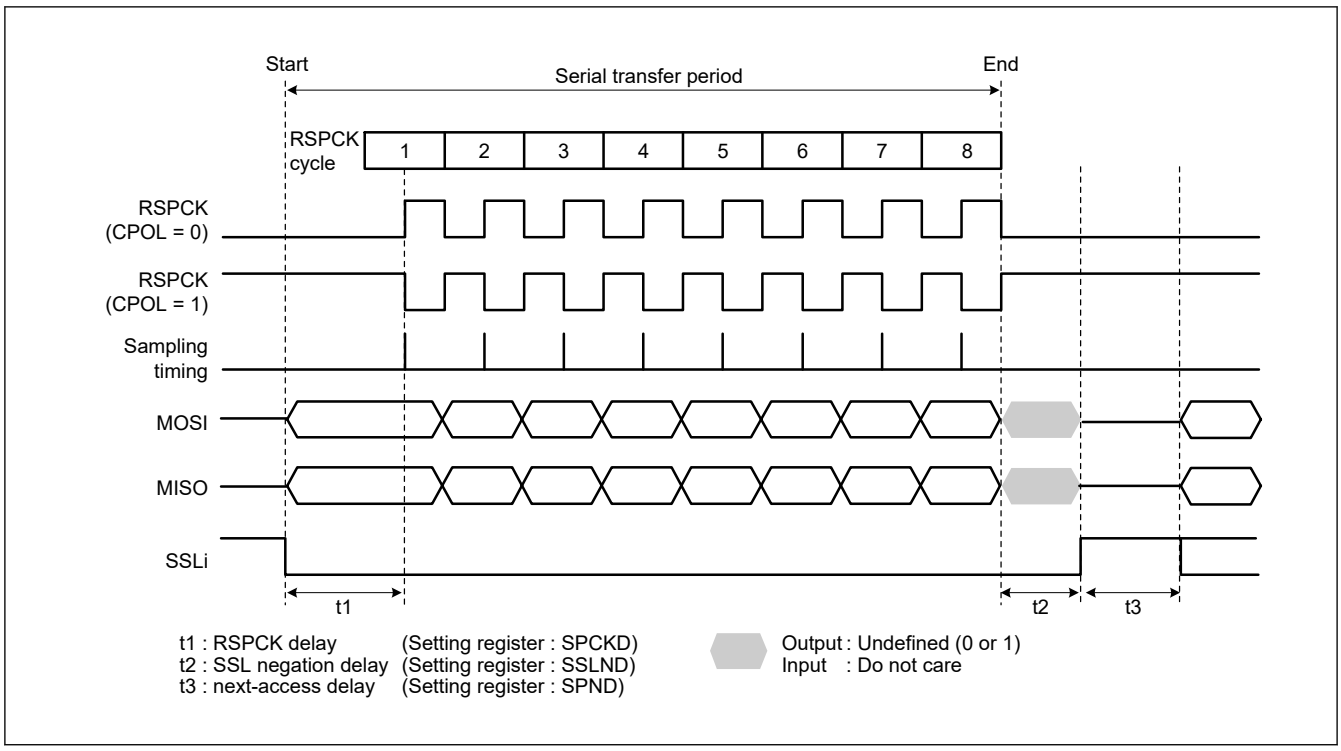


Figure 36.29 SPI transfer format (CPHA = 0, SPFRF = 0)

36.4.5.2 CPHA = 1

Figure 36.30 shows an example of transfer format of 8-bit data serial transfer when SPCMD.CPHA = 1. However, when SPCR.SPMS = 1, communication is performed by using only RSPCK, MOSI, and MISO signals without using the SSL signal. In Figure 36.30, RSPCK (CPOL = 0) is the RSPCK signal waveform when SPCMD.CPOL = 0, and RSPCK (SPCMD.CPOL = 0) is the RSPCK signal waveform when SPCMD.CPOL = 0. The sampling timing shows the time when the SPI latches serial transfer data in the shift register. Input and output directions of each signal depend on SPI mode (master or slave mode). (See section 36.4.2. SPI Pin Control.)

When SPCMD.CPHA = 1, invalid data drive to the MISO signal starts at the SSL signal assertion timing. Output of valid data to the MOSI and MISO signals starts at the first RSPCK signal change timing after the SSL signal is asserted.

After this timing, data is updated in each RSPCK cycle. Transfer data is always latched after 1/2 RSPCK cycle of this timing. The set CPOL bit value does not affect the RSPCK signal operation timing but affects only the signal polarity.

Periods t1, t2, and t3 are the same as when CPHA = 0. For t1, t2, and t3 when this LSI (SPI) is in master mode, see section 36.4.13.1. Master Mode Operation.

[In the Motorola-SPI case]

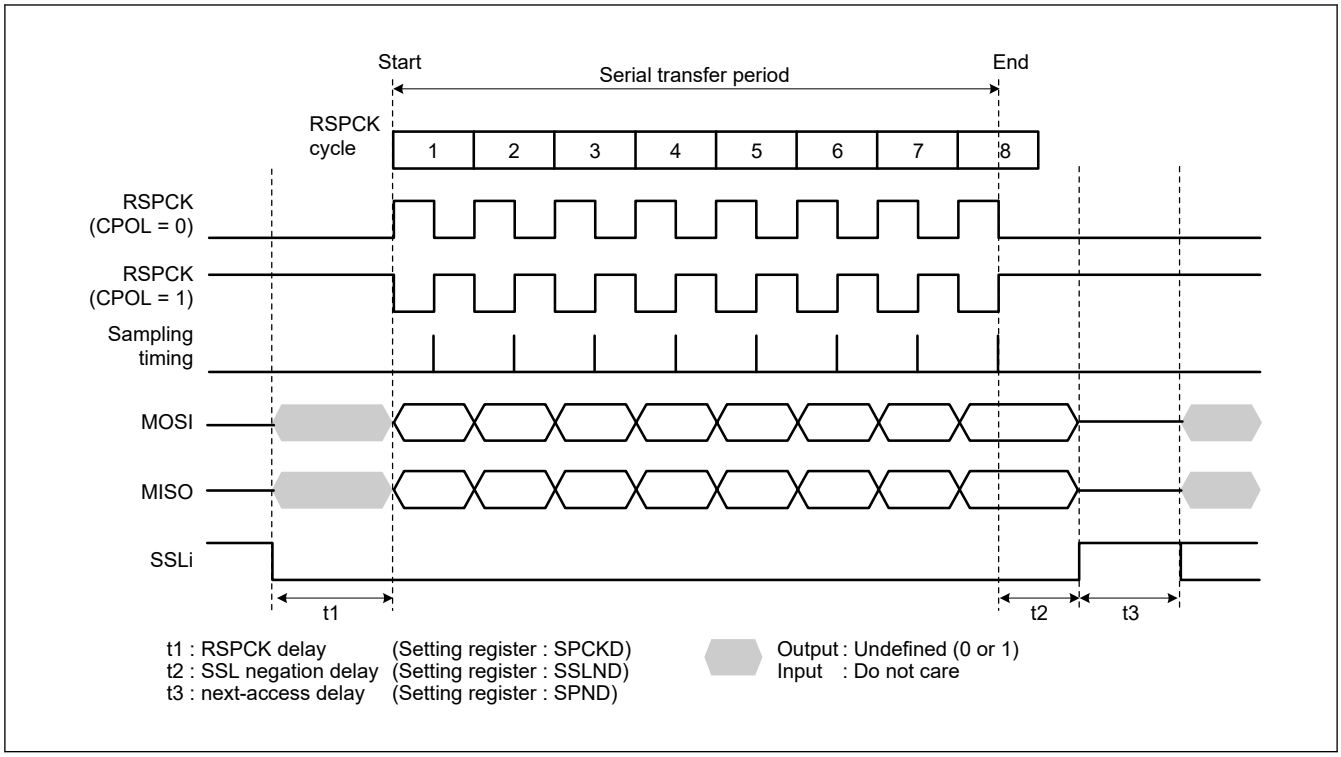


Figure 36.30 SPI transfer format (CPHA = 1, SPFRF = 0)

36.4.6 Communications Operating Mode

Transmit-Receive serial communication, transmit-only operation, and Receive-only operation are selected by setting the Communication Mode Select bits (TXMD[1:0]) of the SPCR register.

SPDR access described in Figure 36.31, Figure 36.32, and Figure 36.33 shows an access to the SPDR register. W shows a write cycle.

36.4.6.1 Transmit-Receive Serial Communication (TXMD[1:0] = 00b)

Figure 36.31 shows an example of operation when the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 00b. In the example in Figure 36.31, the SPI performs 8-bit data serial transfer with the settings of RTRG = FIFO stage - 1, SPDCR2.TTRG = 0, SPCMD.CPHA = 1, and SPCMD.CPOL = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

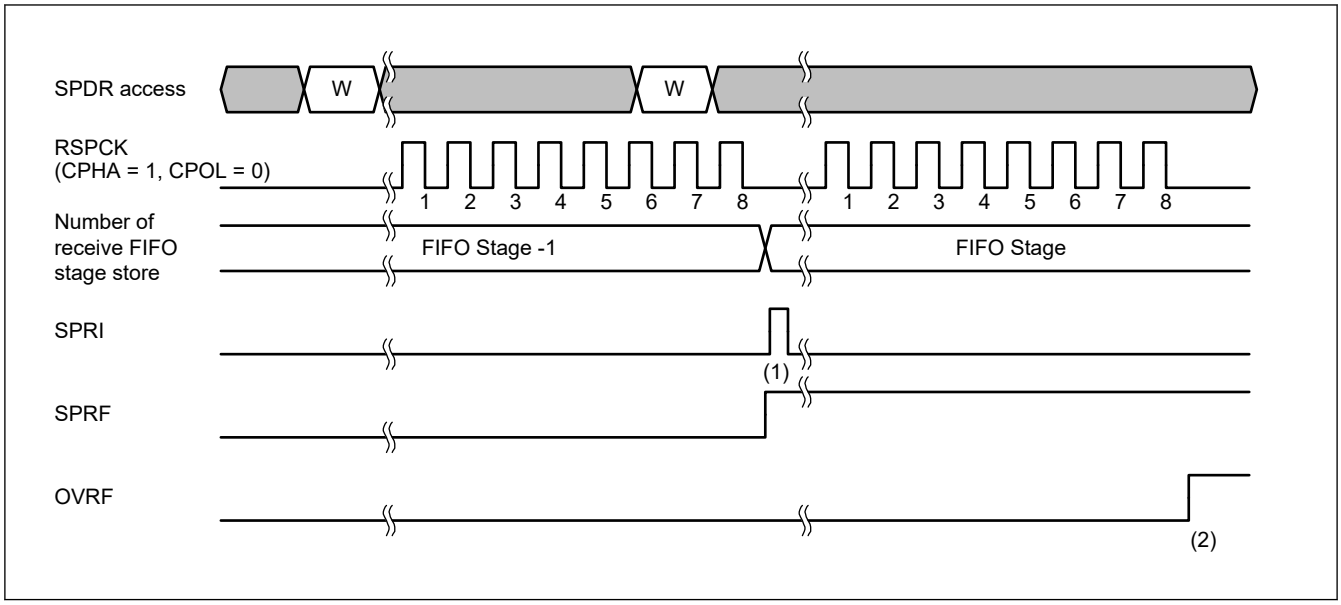


Figure 36.31 Operation example of TXMD[1:0] = 00b

The following describes operation of flags at timings (1) and (2) in the figure above.

(1) When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request SPRI (setting the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the OVRF flag in the SPSR register to 1 and discards the received data in the shift register.

In Transmit-Receive serial communication (TXMD[1:0] = 00b), transmit data is transmitted and receive data is received. Therefore, the SPRF flag and the OVRF flag are set to 1 at timings (1) and (2) respectively.

36.4.6.2 Transmit-Only Serial Communication (TXMD[1:0] = 01b)

Figure 36.32 shows an example of operation when the communication mode select bits (TXMD[1:0]) in the SPCR register are set to 01b. In the example in Figure 36.32, the SPI performs 8-bit data serial transfer with the settings of SPDCR2.TTRG[1:0] = 00b, SPDCR2.RTRG[1:0] = 00b, SPCMD.CPHA = 1, and SPCMD.CPOL = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

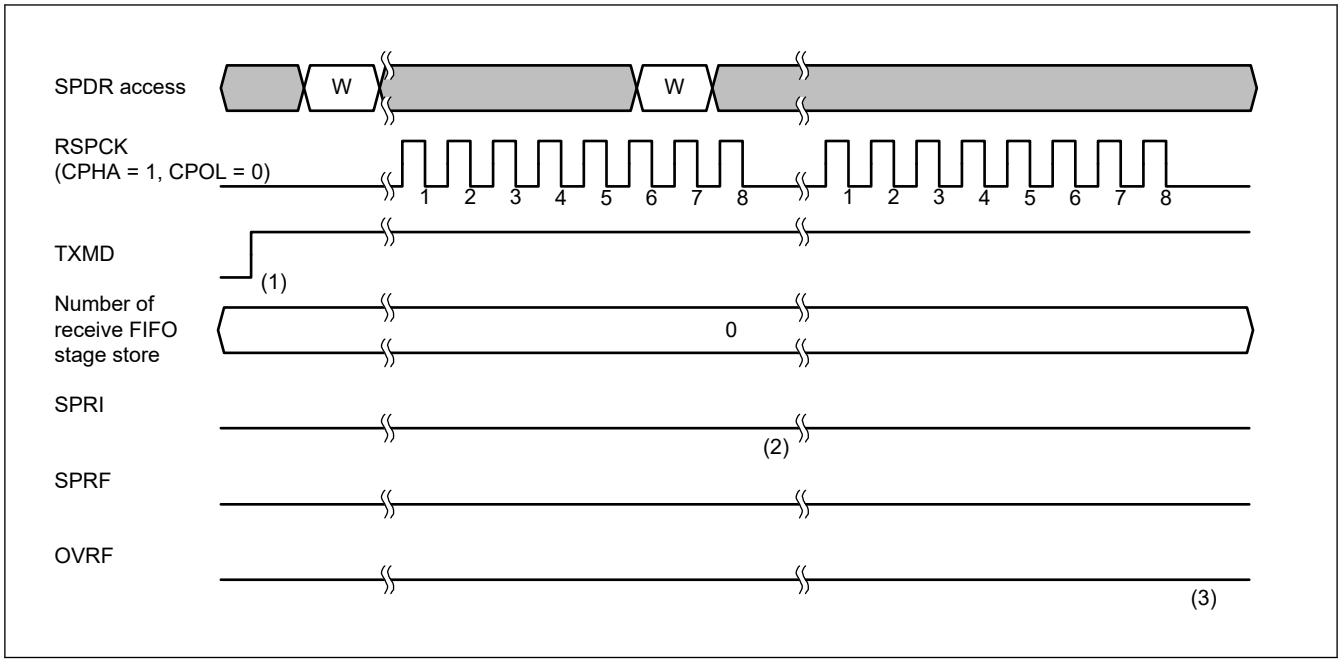


Figure 36.32 Operation example of TXMD[1:0] = 01b

The following describes operation of flags at timings (1) to (3) in the figure above.

(1) Before setting TXMD[1:0] to 01b (transmit-only serial communication), confirm that the receive buffer is empty and that the SPRF and OVRF flags in the SPSR register are 0.

(2) In transmit-only serial communication (TXMD[1:0] = 01b), when serial transfer ends without receiving data in the receiving FIFO of SPDR, the SPRF flag remains 0 and the shift register data is not copied to the receive buffer.

(3) The OVRF flag remains 0 even after serial transfer ends because previously received data is not remaining in the SPDR's receive buffer, and the shift register data is not copied to the receive buffer.

In transmit-only serial communication (TXMD[1:0] = 01b), transmit data is transmitted but receive data is not received.

Therefore, the SPRF flag and the OVRF flag remain 0 at each timing of (1) to (3).

36.4.6.3 Receive-Only Serial Communication (TXMD[1:0] = 10b)

Figure 36.33 shows an example of operation when the communication mode select bit (TXMD[1]) in the SPI control register (SPCR) is set to 1. In the example in Figure 36.33, the SPI performs 8-bit data serial transfer with the settings of SPDCR2.TTRG[1:0] = 0, SPDCR2.RTRG[1:0] = FIFO stage - 1, SPCMD.CPHA = 1, and CPOL in SPCMD = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

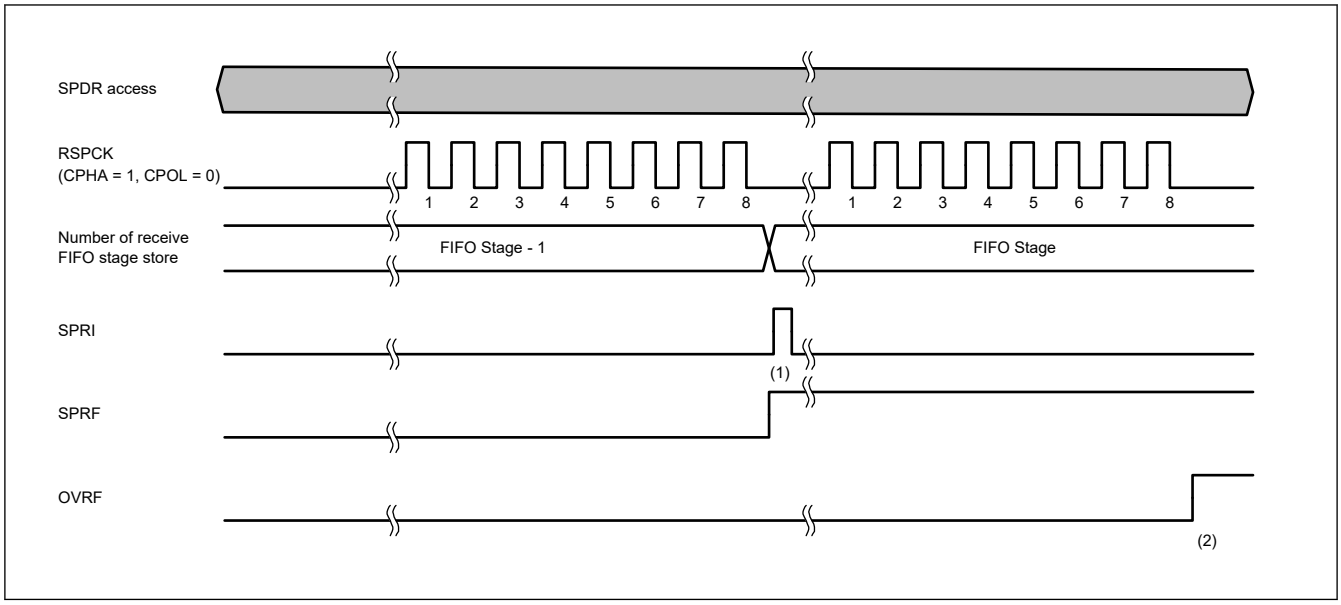


Figure 36.33 Operation example of TXMD[1:0] = 10b

The following describes operation of flags at timings (1) and (2) in the figure above.

(1) When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request SPRI (setting the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the OVRF flag in the SPSR.SPI register to 1 and discards the received data in the shift register.

36.4.7 Transmission Buffer Empty/Receive Buffer Full Interrupt

Figure 36.34 show an example of operation of SPI transmit buffer empty interrupt (SPTI) and SPI receive buffer full interrupt (SPRI) respectively. "SPDR access" described in Figure 36.34 and shows an access to the SPDR register. W shows a write cycle and R shows a read cycle. In each example of operation, the SPI performs 8-bit data serial transfer with the settings of SPCR.TXMD[1:0] = 00b, SPDCR2.TTRG = 0, SPDCR2.RTRG = 0, SPCMD.CPHA = 0 (Figure 36.34), and SPCMD.CPOL = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

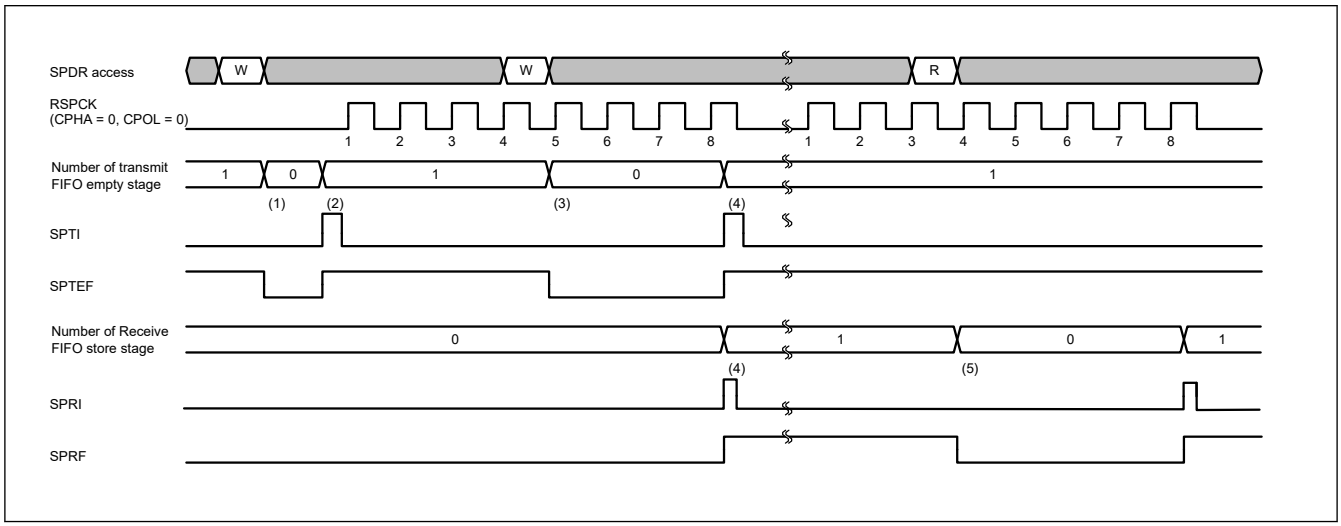


Figure 36.34 Example of SPTI interrupt and SPRI interrupt operations

The following describes operation at timings (1) to (5) in the figure above.

(1) When transmit data is written to SPDR before the data for the next transfer is not set to the FIFO of the SPDR, the SPI writes data to the transmission buffer. When transmit data is written to SPDR in one processing routine using DMAC, the SPSR.SPTEF flag is cleared to 0 at the last access.

(2) When the shift register is empty, the SPI copies transmit buffer data to the shift register. At this time, if transmit FIFO empty stage number > TTRG value, then the SPI generates a transmit buffer empty interrupt request (SPTI) and sets the SPTEF flag to 1. How to start serial transfer depends on SPI mode. (See section 36.4.13. SPI Operation and section 36.4.14. Clock Synchronous Operation.)

(3) When transmit data is written to SPDR upon a transmit buffer empty interrupt request (SPTI) or in the transmit buffer empty determination processing due to the SPTEF flag, the SPI writes data to the transmit buffer. When the transmit data is written to SPDR in one processing routine using DMAC, the SPTEF flag is cleared to 0 at the last access. The SPI does not copy transmit buffer data to the shift register because the shift register contains serial transfer data.

(4) When serial transfer ends (data sampling clock edge in the final bit is detected) while the receive buffer of SPDR > FIFO stage number, the SPI copies the received data in the shift register to the receive buffer. At this time, a receive buffer full interrupt request is generated and the SPRF flag is set to 1. When serial transfer ends, the shift register becomes empty. If the next transfer data is set in the transmit FIFO before serial transfer ends, the SPI set the SPTEF flag to 1 and copies transmit buffer data to the shift register. Even if received data is not copied from the shift register to the receive buffer while an overrun error is present, the SPI determines that the shift register is empty at the end of serial transfer, enabling data transfer from the transmit buffer to the shift register.

(5) When SPDR is read upon a receive buffer full interrupt request (SPRI) or in the receive buffer full determination processing due to the SPRF flag, received data can be read. If the received data is read from SPDR in one processing routine using DMAC, the SPRF flag is cleared to 0 at the last access.

When transmit data is written to the SPDR register while no empty stages in the transmit FIFO, the SPI does not update transmit buffer data. Write transmit data to the SPDR register upon a transmit buffer empty interrupt request or check the empty or not the transmission buffer by the SPTEF flag.

While SPCR.SPTIE = 1, when the SPI is disabled (SPCR.SPE = 0), an SPI transmit buffer empty interrupt is generated. However, SPI transmit buffer empty interrupt is inhibited by disabling SPI transmit buffer empty interrupt (SPCR.SPTIE = 0) at the same time as SPE bit setting.

When serial transfer ends while data is stored in the receive FIFO for the number of FIFO stages, the SPI detects an overrun error without copying data from the shift register to the receive buffer (as described in section 36.4.10. Error Detection). To prevent overrun of received data, read received data before the next serial transfer ends.

Transmit buffer state and receive buffer state can be monitored by an SPI transmit buffer empty interrupt and an SPI receive buffer full interrupt or a corresponding interrupt flag or by the SPTEF and SPRF flags.

36.4.8 Idle Interrupt

The idle interrupt during master mode operation is when the SPCP[2:0] of the SPSSR register becomes 000b (start of sequence control), the IDLNF flag in the SPSR register is set to 1 and an idle interrupt request is made during master mode operation. An interrupt request is also made by clearing the SPCR.SPE bit to 0.

(1) In the Motorola-SPI case

Figure 36.35 shows an example of idle interrupt operation during normal operation.

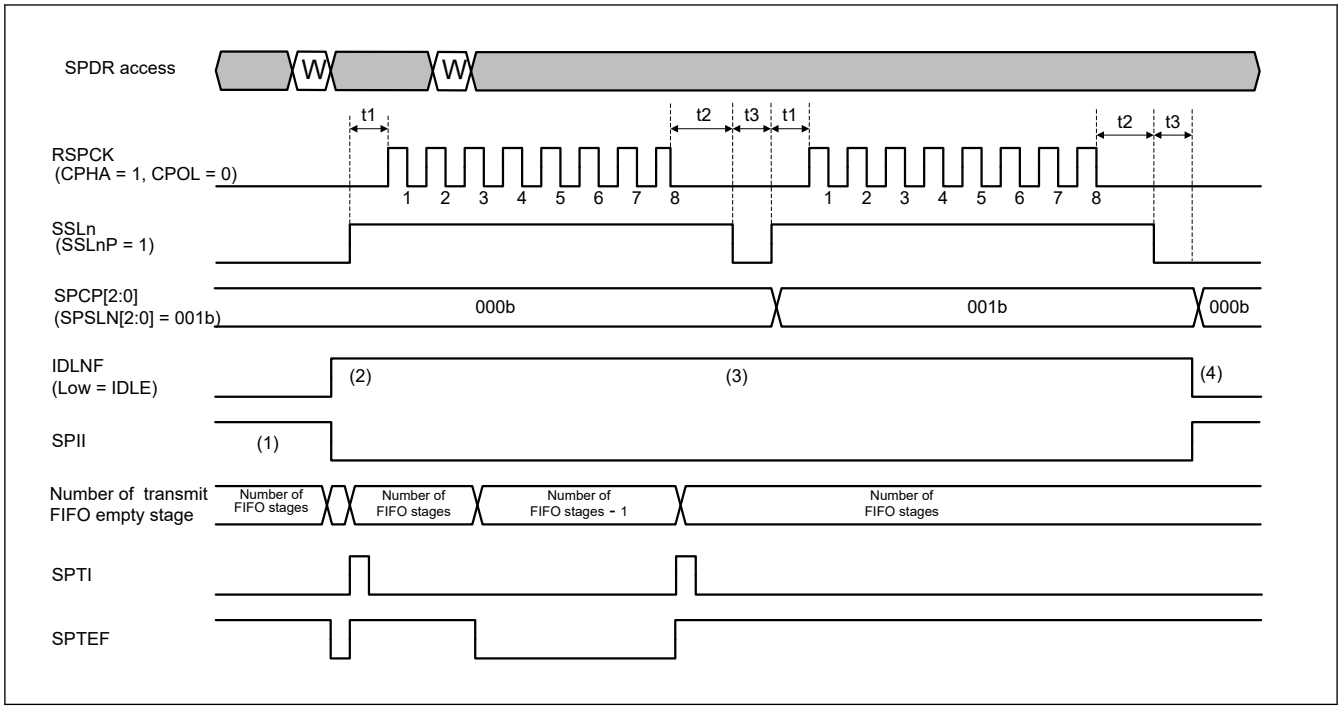


Figure 36.35 Example of idle interrupt operation (master mode / Motorola-SPI)

(1) At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data makes sets the IDLNF flag to 1 (BUSY). When the SPIIE bit in the SPI control register (SPCR) is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.

(2) After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.

(3) The SPCP[2:0] bits change the command to the next command the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.

(4) The IDLNF flag is cleared to 0 (IDLE) the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPIIE bit is 1 currently, an SPII interrupt is output.

(2) In the TI-SSP case

Figure 36.36 shows an example of idle interrupt operation during normal operation.

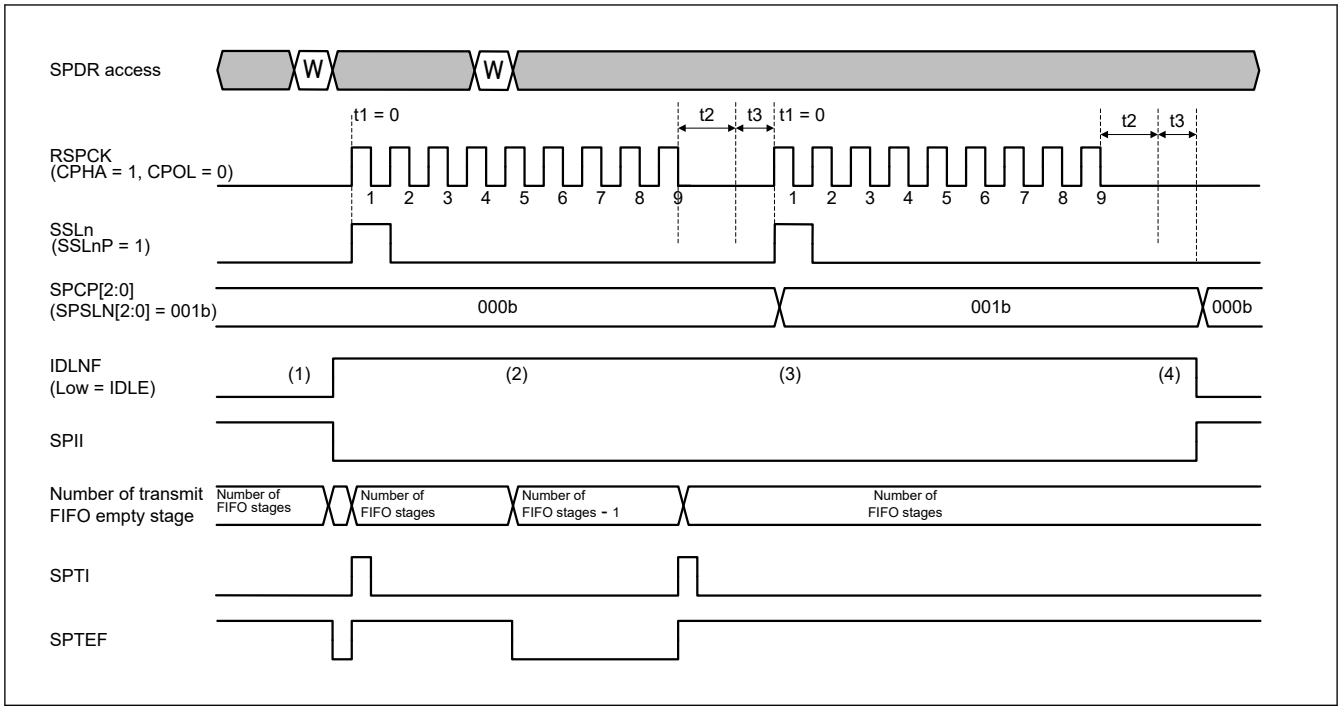


Figure 36.36 Example of idle interrupt operation (master mode / TI-SSP)

(1) At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data makes sets the IDLNF flag to 1 (BUSY). When the SPCR.SPIIE bit is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.

(2) After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.

(3) The SPCP[2:0] bits change the command to the next command the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.

(4) The IDLNF flag is cleared to 0 (IDLE) the end of t3 cycle because the next command is 000b and the next transmit data is not present. When SPCR.SPIIE = 1 currently, an SPII interrupt is output.

36.4.9 Communication End Interrupt

36.4.9.1 Transmit-Receive/Transmit-only in Master Mode

Refer to the description of the CENDF bit in section 36.3.19. SPTFSR : SPI Transfer FIFO Status Register and section 36.3.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Master Mode.

(1) In the Motorola-SPI case

Figure 36.37 shows an example of communication end interrupt operation during transmit-receive/transmit master mode.

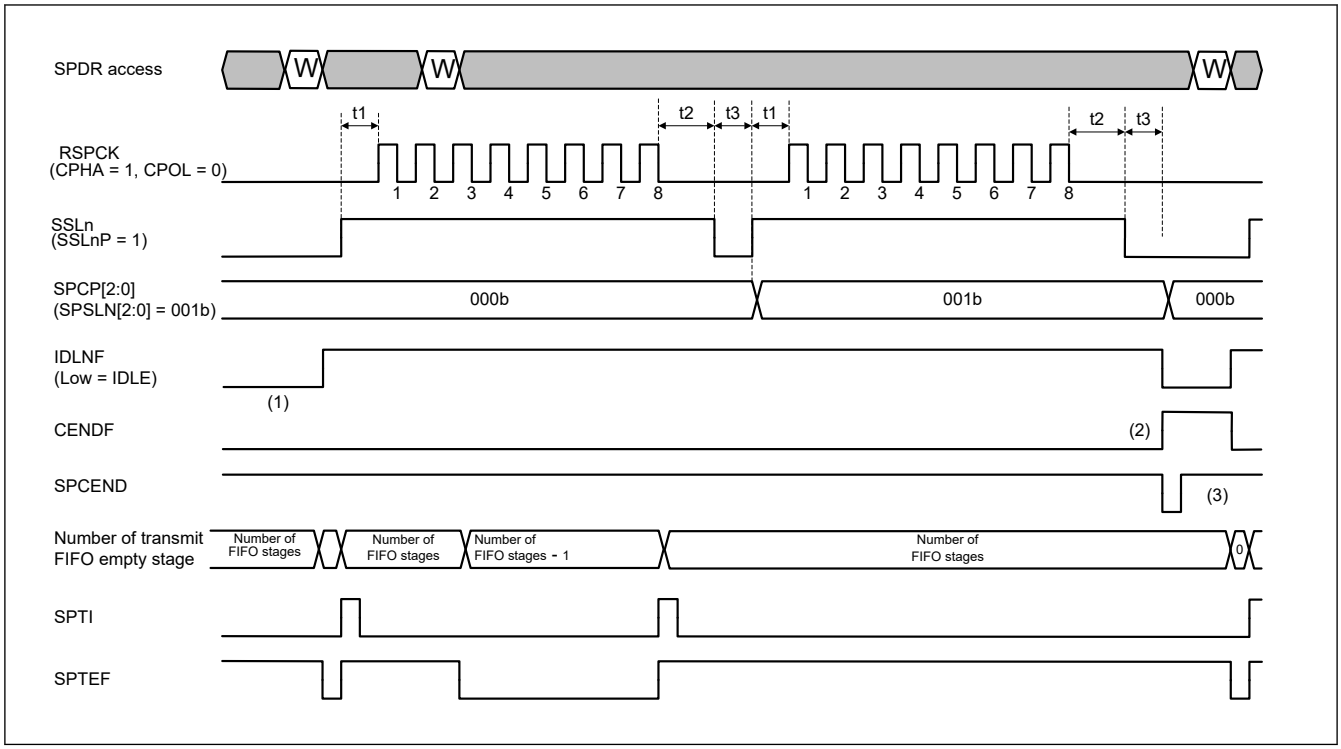


Figure 36.37 Example of communication end interrupt operation (transmit-receive/transmit-only master mode / Motorola-SPI)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication starts, and these are kept during communication.

(2) The CENDF flag is 1 (communication end) at the end of t3 cycle, because the next command is 000b and there is no next transmit data. The SPCEND interrupt is then output with 1 PCLKM cycle width if the CENDIE bit is 1.

(3) The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX), or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

(2) In the TI-SSP case

Figure 36.38 shows an example of communication end interrupt operation during transmit-receive/transmit-only master mode.

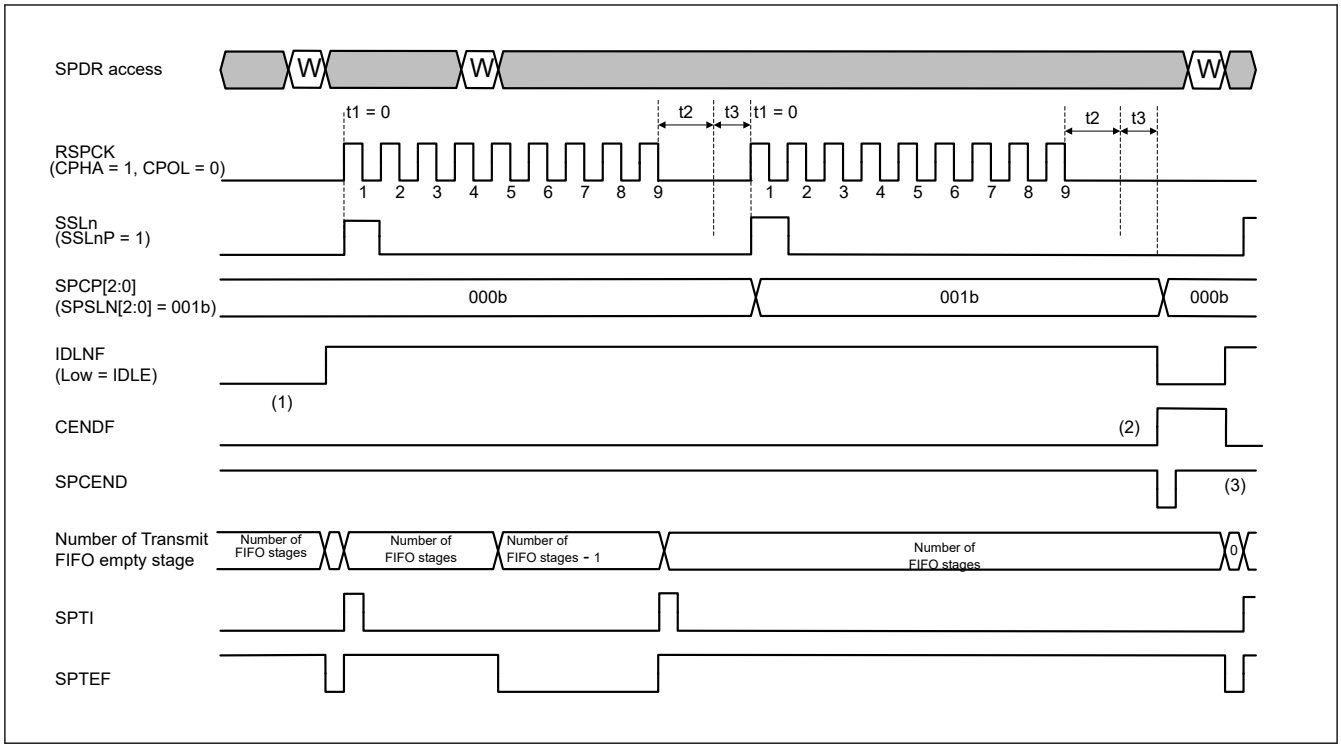


Figure 36.38 Example of communication end interrupt operation (transmit-receive/transmit-only master mode / TI-SSP)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication starts, and these are kept during communication.

(2) The CENDF flag is 1 (communication end) at the end of t3 cycle, because the next command is 000b and there is no next transmit data. The SPCEND interrupt is then output with 1 PCLKM cycle width if the CENDIE bit is 1.

(3) The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX), or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

36.4.9.2 Receive-only in Master Mode

Refer to the description of the CENDF bit in section 36.3.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Receive-only in Master Mode.

Figure 36.39 shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] = 0x00.

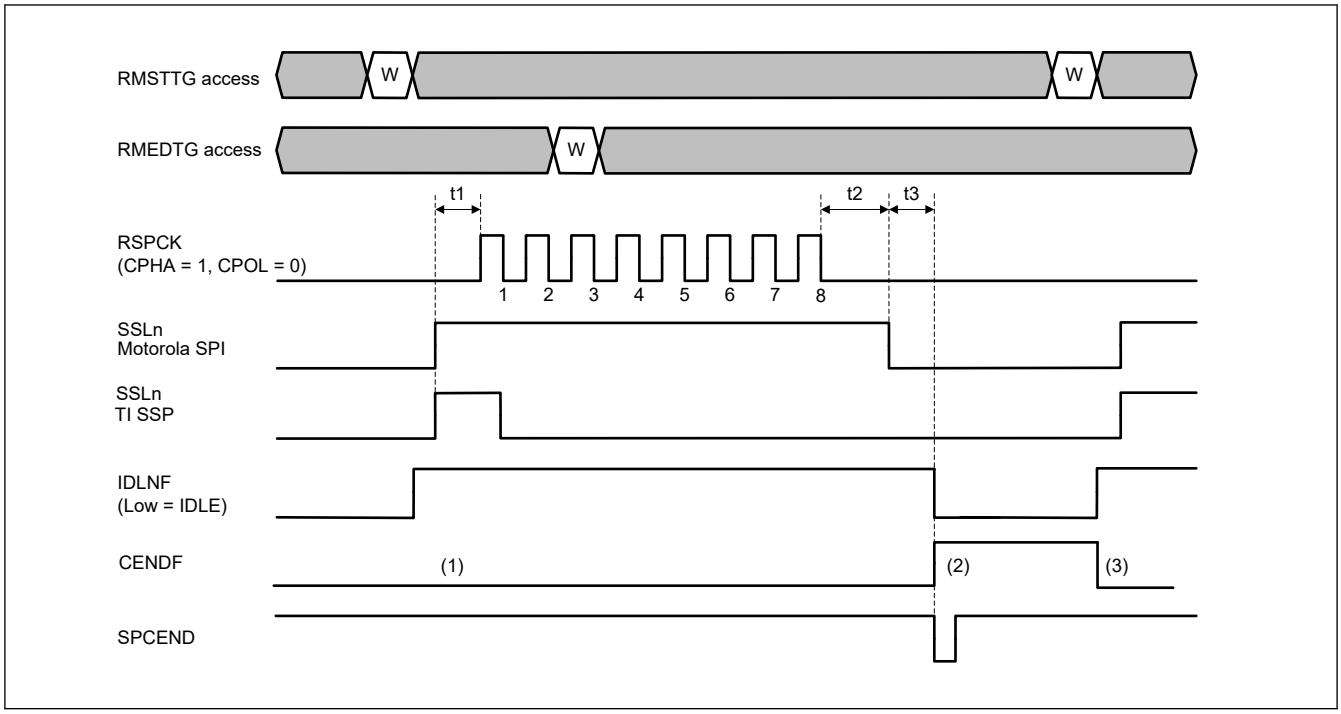


Figure 36.39 Example of communication end interrupt operation (receive-only master mode / Motorola-SPI) at RMFM[4:0] = 0x00

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication starts, and these are kept during communication.

(2) The CENDF flag is 1 (communication end) at the end of t3 cycle, by writing 1 to RMEDTG during the communication frame. The SPCEND interrupt is then output with 1 PCLKM cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared when writing 1 to RMSTTG, or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

Figure 36.40 shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] ≠ 0x00.

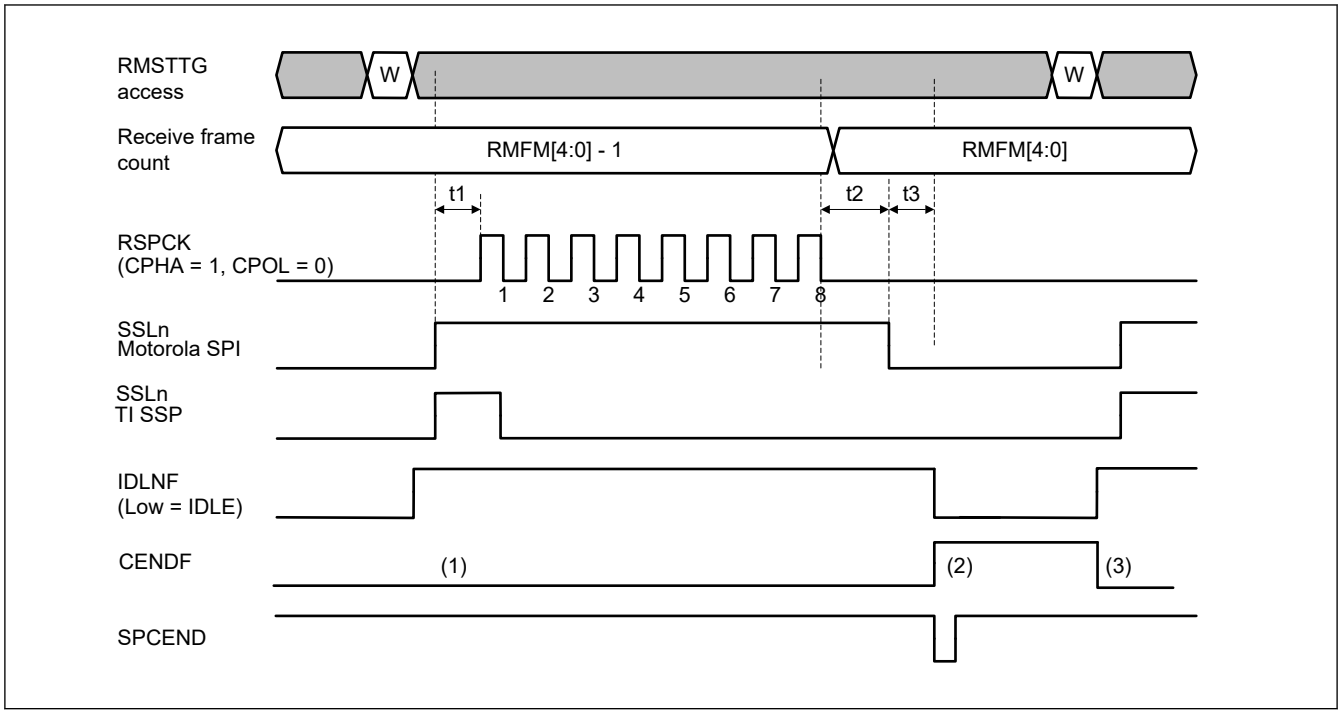


Figure 36.40 Example of communication end interrupt operation (receive-only master mode / Motorola-SPI) at RMFM[4:0] ≠ 0x00

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication starts, and these are kept during communication.

(2) The CENDF flag is 1 (communication end) at the end of t3 cycle, after receiving the number of frames set by RMFM[4:0]. The SPCEND interrupt is then output with 1 PCLKM cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared when writing 1 to RMSTTG, or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

In slave mode operation, the output timing of the communication end interrupt is dependent on the value of the SPCR.SPMS bit (SPI mode select bit). The clear timing of the communication end interrupt is dependent on the communication mode (transmit-receive or transmit-only or receive-only).

36.4.9.3 Transmit-Receive/Transmit-only in Slave Mode on SPI Operation (4-wire)

See the description for the CENDF bit in section 36.3.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Slave Mode (4-wire).

(1) In the Motorola-SPI case

Figure 36.41 shows an example of communication end interrupt operation during transmit-receive/transmit-only slave mode on SPI operation.

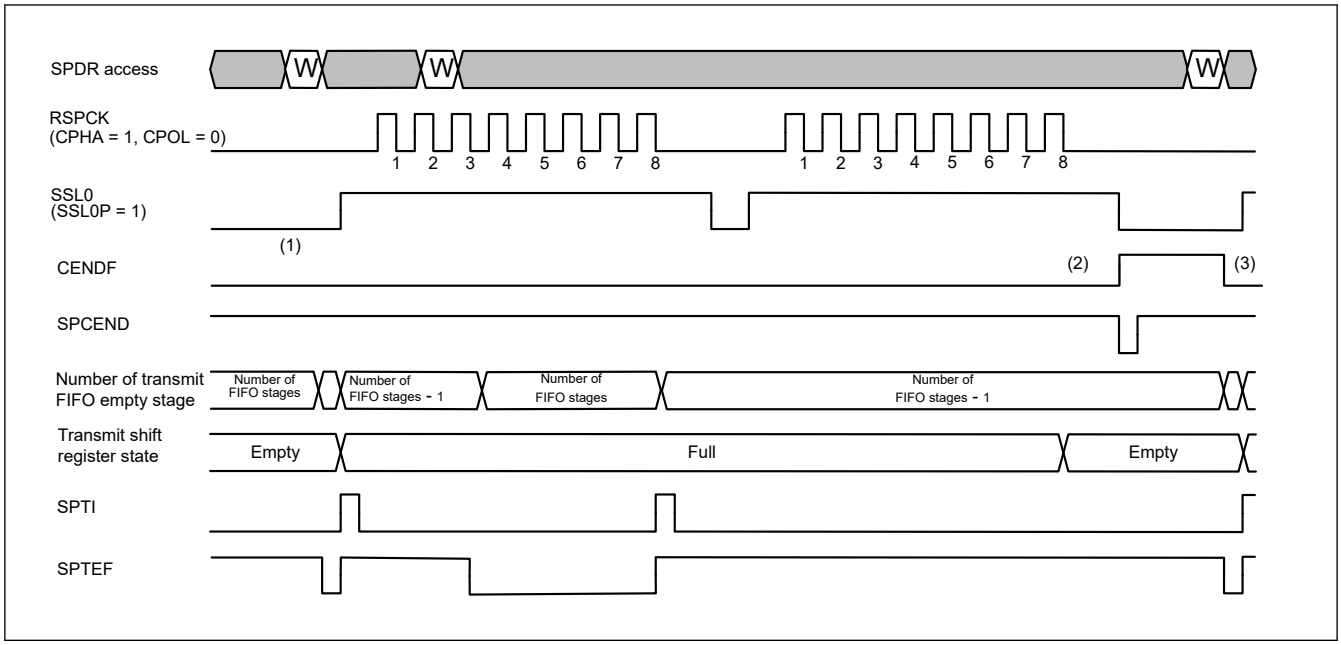


Figure 36.41 Example of communication end interrupt operation (transmit-receive/transmit-only slave mode on SPI operation / Motorola-SPI)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication starts, and these are kept during communication.

(2) The CENDF flag is 1 (communication end) at the timing of SSL0 negate, when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. The SPCEND interrupt is then output with 1 PCLKM cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared when the next transmition data is written to the transmit buffer (SPTX), or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

(2) In the TI-SSP case

Figure 36.42 shows an example of communication end interrupt operation during transmit-receive/transmit-only slave mode on SPI operation.

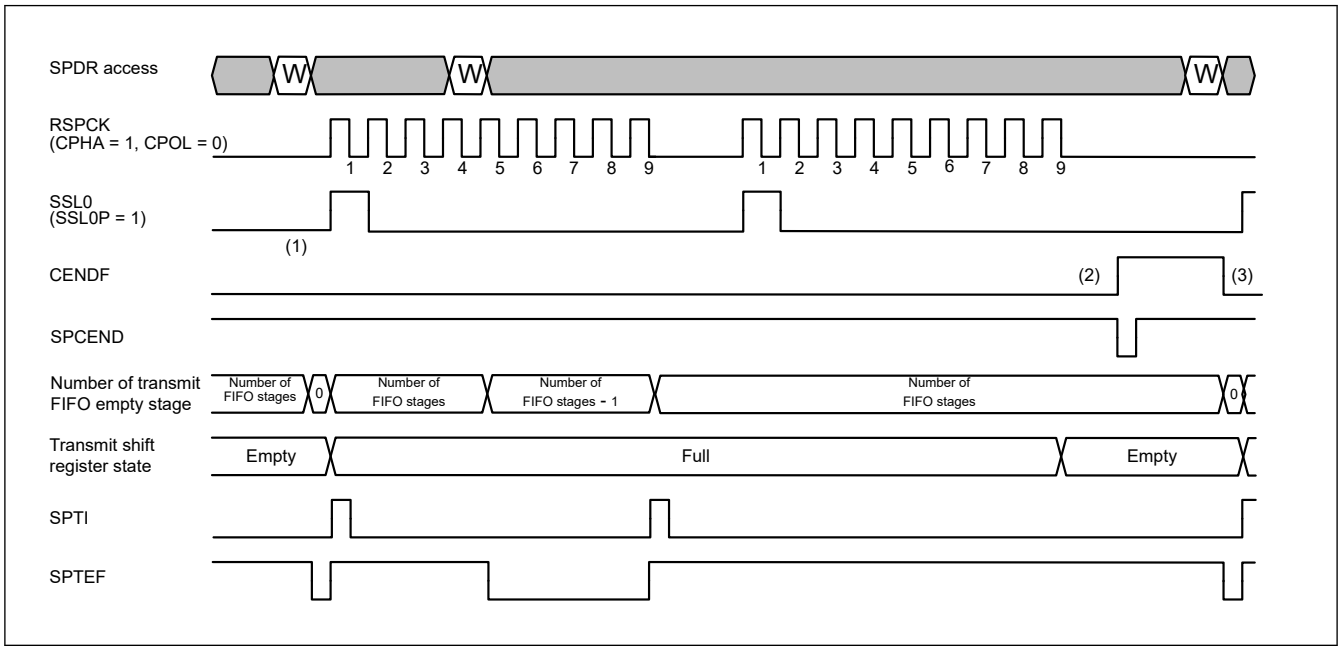


Figure 36.42 Example of communication end interrupt operation (transmit-receive/transmit-only slave mode on SPI operation / TI-SSP)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication start, and these have kept during communication.

(2) The CENDF flag is 1 (communication end) at the RSPCK last data bit sampling, when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. The SPCEND interrupt is then outputs with PCLKM 1 cycle width if the CENDIE bit is 1.

(3) The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX), or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

36.4.9.4 Receive Only in Slave Mode on SPI Operation (4-wire)

Refer to the description of the CENDF bit in section 36.3.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Receive-only in Slave Mode (4-wire).

(1) In the Motorola-SPI case

Figure 36.43 shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).

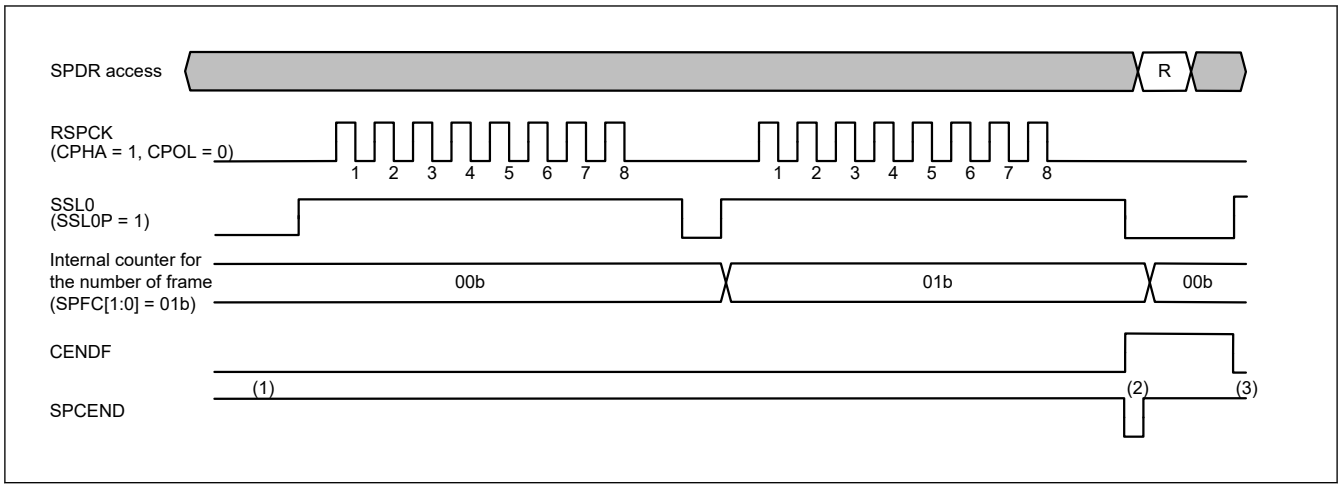


Figure 36.43 Example of communication end interrupt operation (receive-only slave mode on SPI operation / Motorola-SPI)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication start, and these have kept during communication.

(2) After store the frames for the SPDCR.SPFC register set value in the receive buffer, the CENDF flag becomes 1 (communication completed) at the timing of SSL0 negation. The SPCEND interrupt is then outputs with PCLKM 1 cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared at the SSL0 assert when the next transmission start, or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

(2) In the TI-SSP case

Figure 36.44 shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).

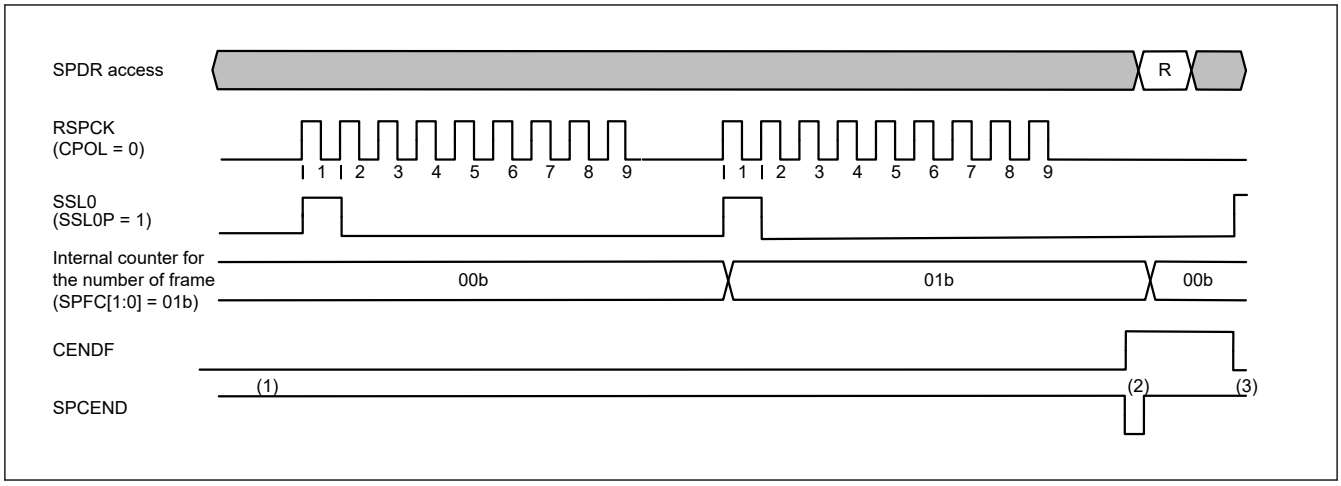


Figure 36.44 Example of communication end interrupt operation (receive-only slave mode on SPI operation / TI-SSP)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication start, and these have kept during communication.

(2) The CENDF flag is 1 (communication end) at the RSPCK last data bit sampling, when the last frame transmission ends. The SPCEND interrupt is then outputs with PCLKM 1 cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared at the SSL0 assert when the next transmission start, or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

36.4.9.5 Transmit-Receive/Transmit-only in Slave Mode on Clock Synchronous Operation (3-wire)

Refer to the description of the CENDF bit in section 36.3.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Receive-only in Slave Mode (4-wire).

Figure 36.45 shows an example of communication end interrupt operation during transmit-receive/transmit slave mode on clock synchronous operation (3-wire).

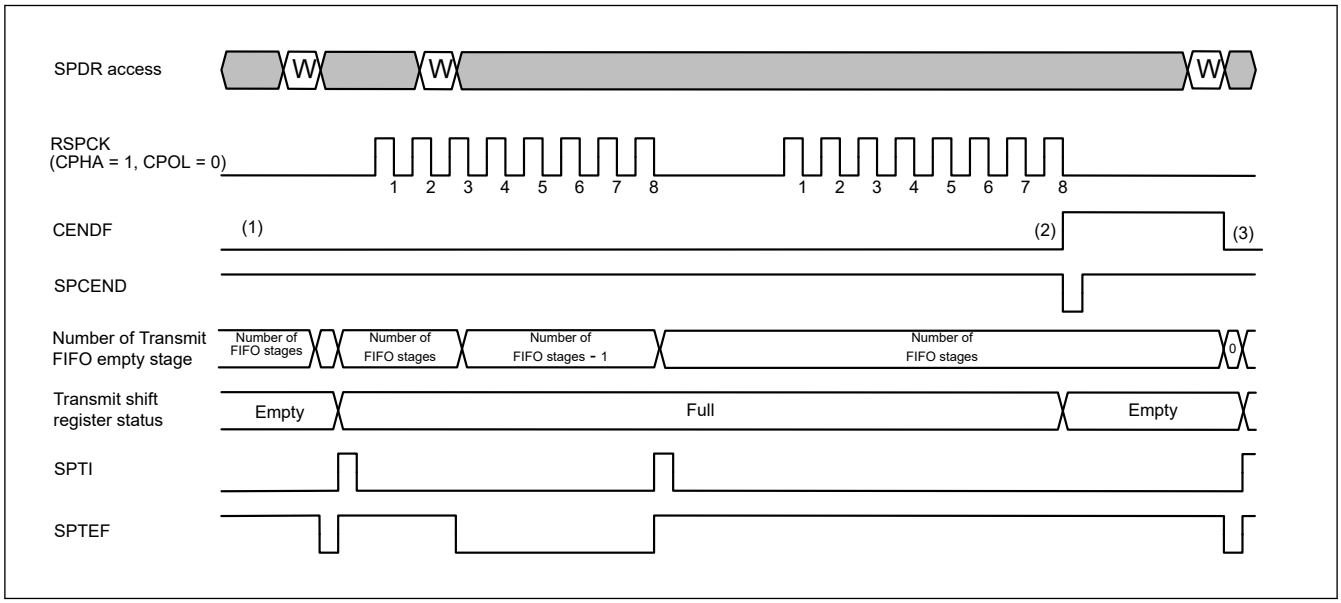


Figure 36.45 Example of communication end interrupt operation (receive-only slave mode on SPI operation / Motorola-SPI)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication start, and these have kept during communication.

(2) When the next transfer data is not set in the transmit FIFO and the transmit shift register is empty, the SPCEND interrupt is then outputs with PCLKM 1 cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX), or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

36.4.9.6 Receive only in Slave Mode on Clock Synchronous Operation (3-wire)

Refer to the description of the CENDF bit in section 36.3.19. SPTFSR : SPI Transfer FIFO Status Register and section 36.3.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Receive -only in Slave Mode on Clock Synchronous (3-wire).

Figure 36.46 shows an example of communication end interrupt operation during receive only slave mode on clock synchronous operation.

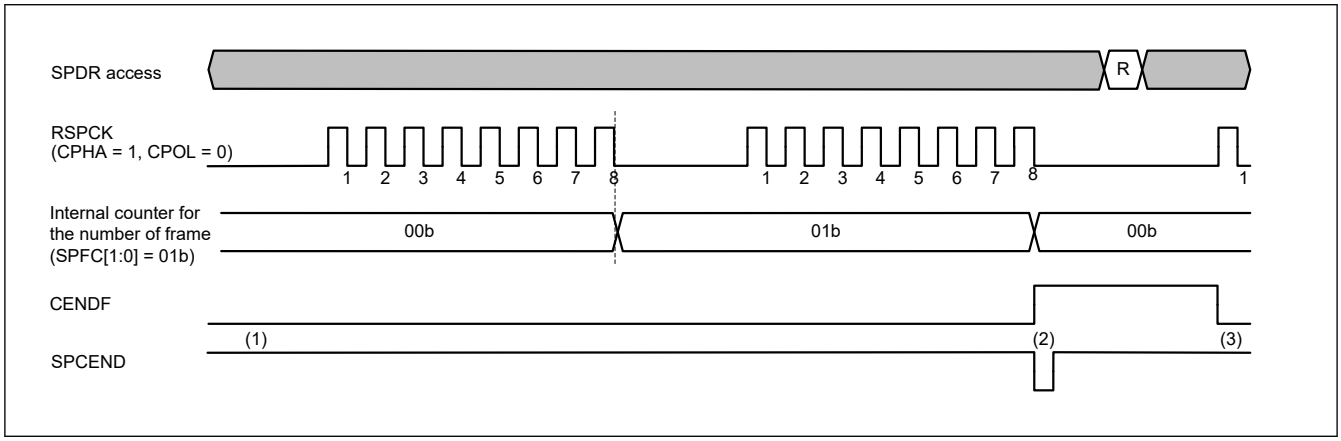


Figure 36.46 Example of communication end interrupt operation (receive-only slave mode on clock synchronous operation)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication start, and these have kept during communication.

(2) The CENDF flag is set to 1 (communication completed) at the timing of the last data bit sampling of RSPCK in the last frame communication when the last frame of the SPDCR.SPFC bit set value is received. The SPCEND interrupt is then outputs with PCLKM 1 cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared at the first edge of RSPCK for the next transmission, or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

36.4.9.7 Common Operation

In this Section, the operation common to each mode communication in section 36.4.9.1. Transmit-Receive/Transmit-only in Master Mode to section 36.4.9.6. Receive only in Slave Mode on Clock Synchronous Operation (3-wire) is explained. When the enable of SPI communication end interrupt (CENDIE) is 0, at the time of communication completion, a flag of communication end (CENDF) is set, but no interrupt is output. However, if the enable of communication end interrupt (CENDIE) is set to 1 before clearing the flag of communication end (CENDF) while the enable of SPI function (SPE) is 1, the communication end interrupt is output.

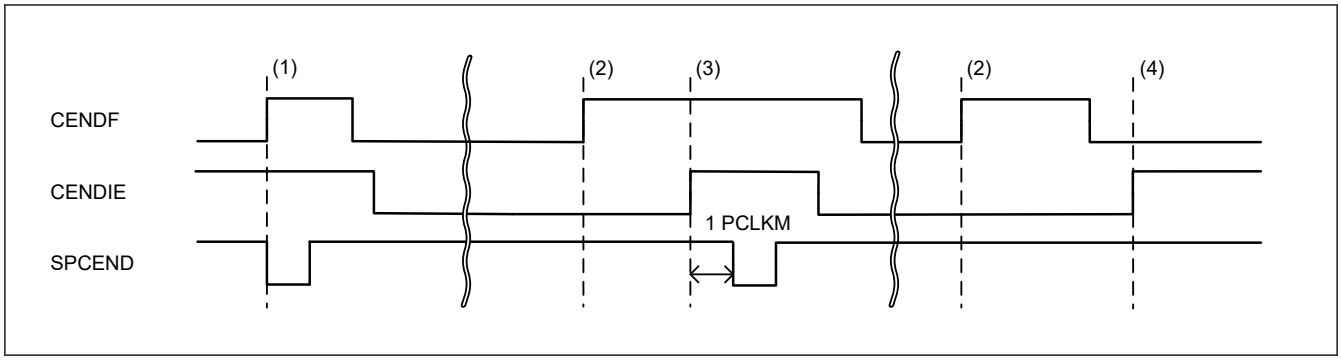


Figure 36.47 Example of communication end interrupt operation (enable control)

(1) When the enable of SPI communication end interrupt (CENDIE) is 1, at the time of communication completion, the following two are the same timing

a flag of communication end (CENDF)

the communication end interrupt

(2) When the enable of SPI communication end interrupt (CENDIE) is 0, at the time of communication completion, the following occurs, but no interrupt.

a flag of communication end (CENDF)

(3) After (2), if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) and the flag of communication end (CENDF) are 1, the communication end interrupt is output after 1 PCLKM.

(4) After (2), even if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) or the flag of communication end (CENDF) is 0, the communication end interrupt is not output.

36.4.10 Error Detection

In normal serial transfer of SPI, data written to the transmit buffer in the SPDR register is transmitted in serial and received serial data can be read from the SPDR's receive buffer. Depending on the transmit buffer or receive buffer status when SPDR is accessed and the SPI status at the start or end of serial transfer, abnormal transfer is performed.

When some abnormal transfer operation takes place, the SPI detects it as an underrun error, overrun error, parity error, or mode fault error. Table 36.12 shows the relationship between abnormal transfer operation and the SPI's error detection function.

Table 36.12 Abnormal transfer occurrence conditions and error detection function of SPI (1 of 2)

	Transfer occurrence condition	SPI operation	Error detection
1	SPDR is written when while no empty stages in the transmit FIFO.	• Transmit buffer data is retained. • No write data is present.	None
2	SPDR is read while no data stored in receive FIFO.	Received data and previously received serial data are read.	None
3	Serial transfer starts in transmit slave mode or transmit-only slave mode, before transmit data output is ready.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Underrun error
4	Serial transfer ends when data is stored in the receive FIFO for the number of FIFO stages.	• Receive FIFO data is retained. • No serial receive data is present.	Overrun error
5	Incorrect parity bit has been received with the parity function enabled in following mode. • Transmit / receive-only master mode • Transmit / receive slave mode • Receive-only slave mode	The parity error flag is asserted.	Parity error
6	The SSL0 input signal is asserted when the serial transfer is idle state in multi-master mode.	• Driving of the RSPCK, MOSI, SSL1 to SSL3 output signals is stopped. • SPI function is disabled.	Mode fault error

Table 36.12 Abnormal transfer occurrence conditions and error detection function of SPI (2 of 2)

	Transfer occurrence condition	SPI operation	Error detection
7	The SSL0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCK, MOSI, SSL1 to SSL3 output signals is stopped. • SPI function is disabled.	Mode fault error
8	[In the Motorola-SPI case] The SSL0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Mode fault error
9	[In the TI-SSP case] The SSL0 input signal is asserted during serial transfer in slave mode.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Mode fault error

The SPI does not detect any error for the operation shown in No.1 in Table 36.12. To prevent missing of data during SPDR writing, write data to SPDR upon a transmit buffer empty interrupt request or while the SPSR.SPTEF flag is 1.

The SPI does not detect any error also for the operation shown in No.2 in Table 36.12. To prevent reading of unnecessary data, read data from SPDR upon a receive buffer full interrupt request or while the SPSR.SPRF flag is 1.

The underrun error shown in No.3 in Table 36.12 is described in section 36.4.10.4. Underrun Error.

The overrun error shown in No.4 in Table 36.12 is described in section 36.4.10.1. Overrun Error.

The parity error shown in No.5 in Table 36.12 is described in section 36.4.10.2. Parity Error.

The mode fault error shown in No.6 to No.9 in Table 36.12 is described in section 36.4.10.3. Mode Fault Error.

For transmission and reception interrupts, see section 36.4.7. Transmission Buffer Empty/Receive Buffer Full Interrupt.

36.4.10.1 Overrun Error

When serial transfer ends while the receive buffer in the SPDR register is full, the SPI detects an overrun error and sets the OVRF flag in SPSR to 1. While the OVRF flag is 1, the SPI does not copy shift register data to the receive buffer. Therefore, data before an error occurs is retained in the receive buffer. To clear the OVRF flag in SPSR to 0, issue a system reset or 1 is written to the SPSRC.OVRFC bit.

Figure 36.48 shows operation of the SPRF flag and the OVRF flag in SPSR. "SPSR access" and "SPDR access" described in Figure 36.48 show an access to SPSR and SPDR respectively. "W" shows a write cycle and "R" shows a read cycle. In the example in Figure 36.48, the SPI performs 8-bit data serial transfer with the settings of SPCMD.CPHA = 1 and SPCMD.CPOL = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

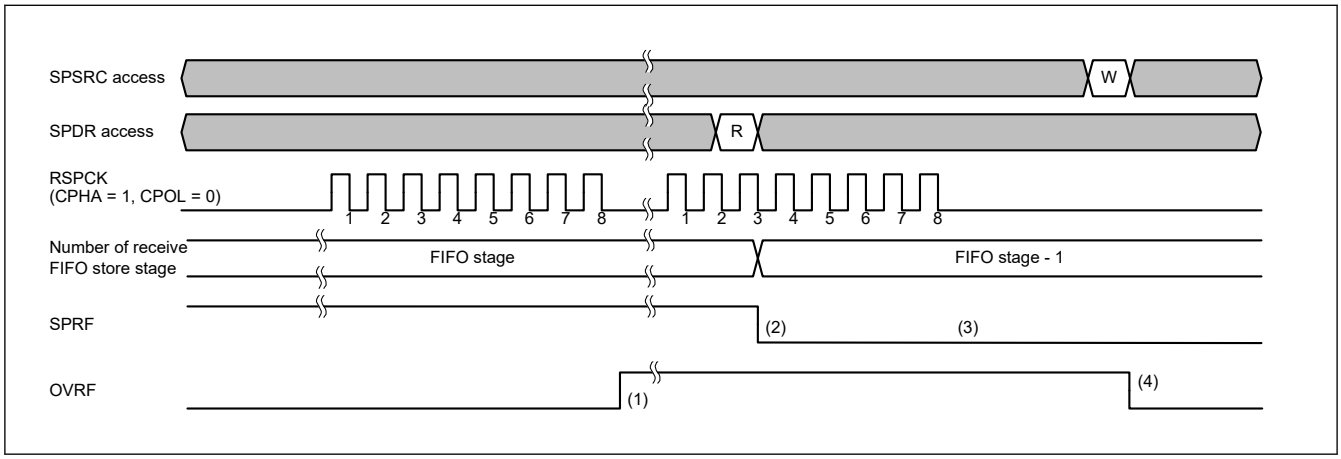


Figure 36.48 Example of operation of SPRF and OVRF flags

The following describes operation of flags at timings (1) to (4) in the figure above.

(1) When serial transfer ends while data is stored for the number of FIFO stages, the SPI detects an overrun error and sets the OVRF flag to 1. The SPI does not copy shift register data to the receive buffer. The SPI does not detect a

parity error even when SPPE = 1. In master mode, the SPI copies the value of pointer to the SPCMD register to the SPSSR.SPECM[2:0] bits.

(2) The SPI can read receive buffer data by reading SPDR. At this time, the SPRF flag is cleared to 0 at the last access when the received data is read from SPDR in one processing routine using DMAC.

(3) When serial transfer ends while the OVRF flag is 1 (overrun error), the SPI does not copy shift register data to the receive buffer (SPRF flag remains 0) and no SPI receive buffer full interrupt is generated. The SPI does not detect a parity error even when SPPE = 1. In master mode, the SPI does not update the SPSSR.SPECM[2:0] bits. When the SPI does not copy received data from the shift register to receive buffer while an overrun error is present, the SPI also determines that the shift register is empty at the end of serial transfer, enabling data transfer from the transmit buffer to the shift register.

(4) When 1 is written to the SPSRC.OVRFC bit.

Whether an overrun error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To perform serial transfer without using an SPI error interrupt, read SPDR and then immediately read SPSR to quickly detect an overrun error. When the SPI is used in master mode, the pointer value to SPCMD when an error is present can be checked by reading the SPSSR.SPECM bits. When an overrun error occurs and the OVRF flag is set to 1, successful reception is disabled until the OVRF flag is cleared to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 36.49 and Figure 36.50 show the clock stop waveform when a serial transfer continues while the reception buffer is full in master mode.

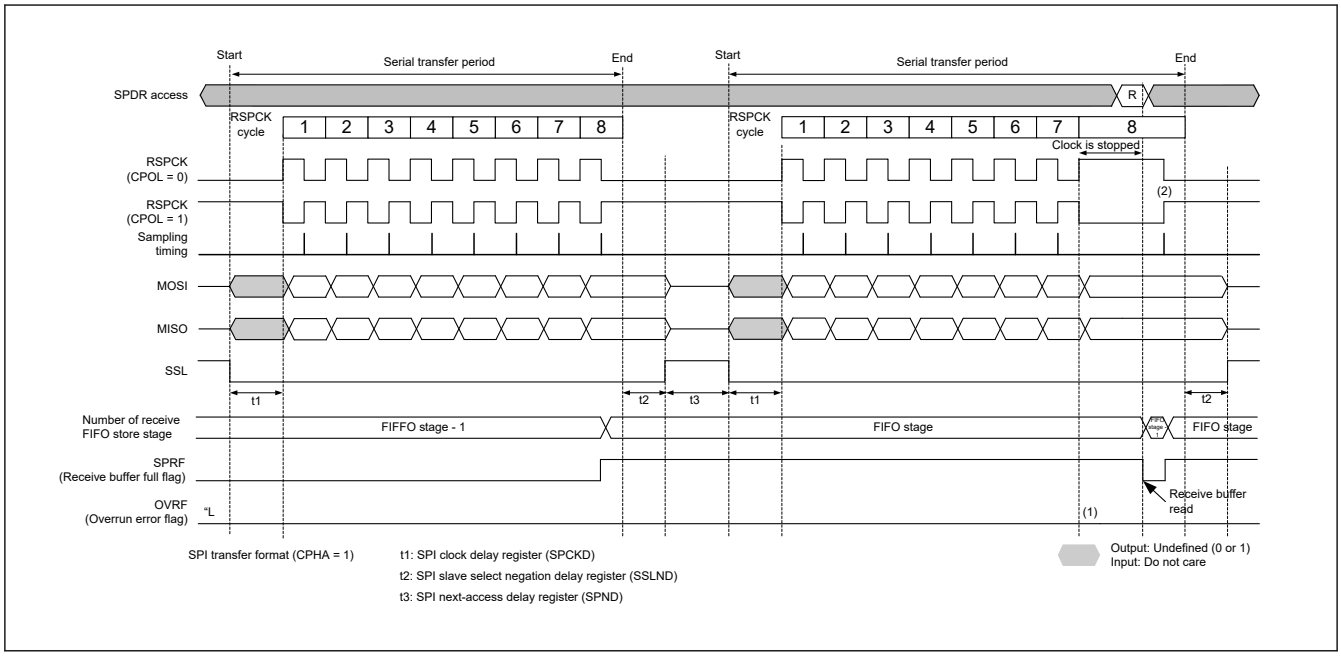


Figure 36.49 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 1)

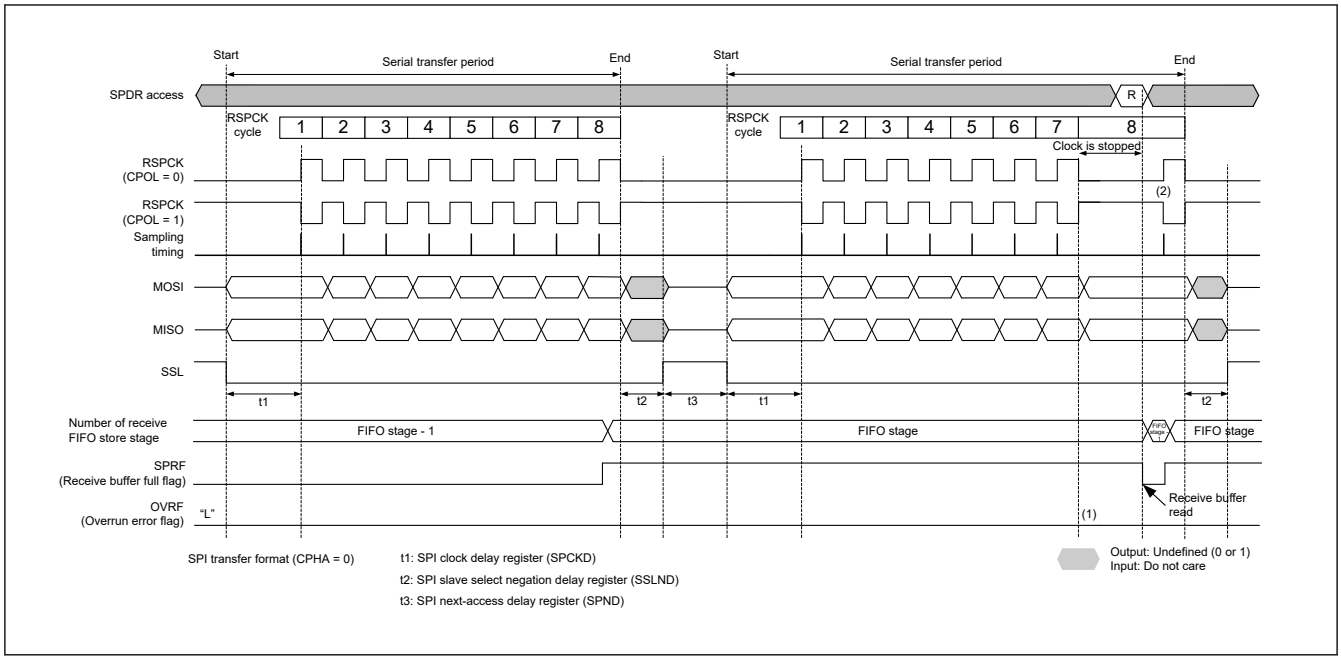


Figure 36.50 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 0)

The following describes operation of flags at timings (1) and (2) in the figure above.

(1) While data is stored in the receive FIFO for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.

(2) Receive buffer data can be read by reading SPDR during clock stop. After the receive buffer data has been read, the RSPCK clock restarts.

Overrun error does not occur when RSPCK automatic stop function is enabled for transfer with no delay of between frames during burst transfer in master mode. The Figure 36.51 and the Figure 36.52 show the clock stop waveform, when there is no delay between frames at burst transfer and the serial transfer continues in the data is stored in the receive FIFO for the number of FIFO stages.

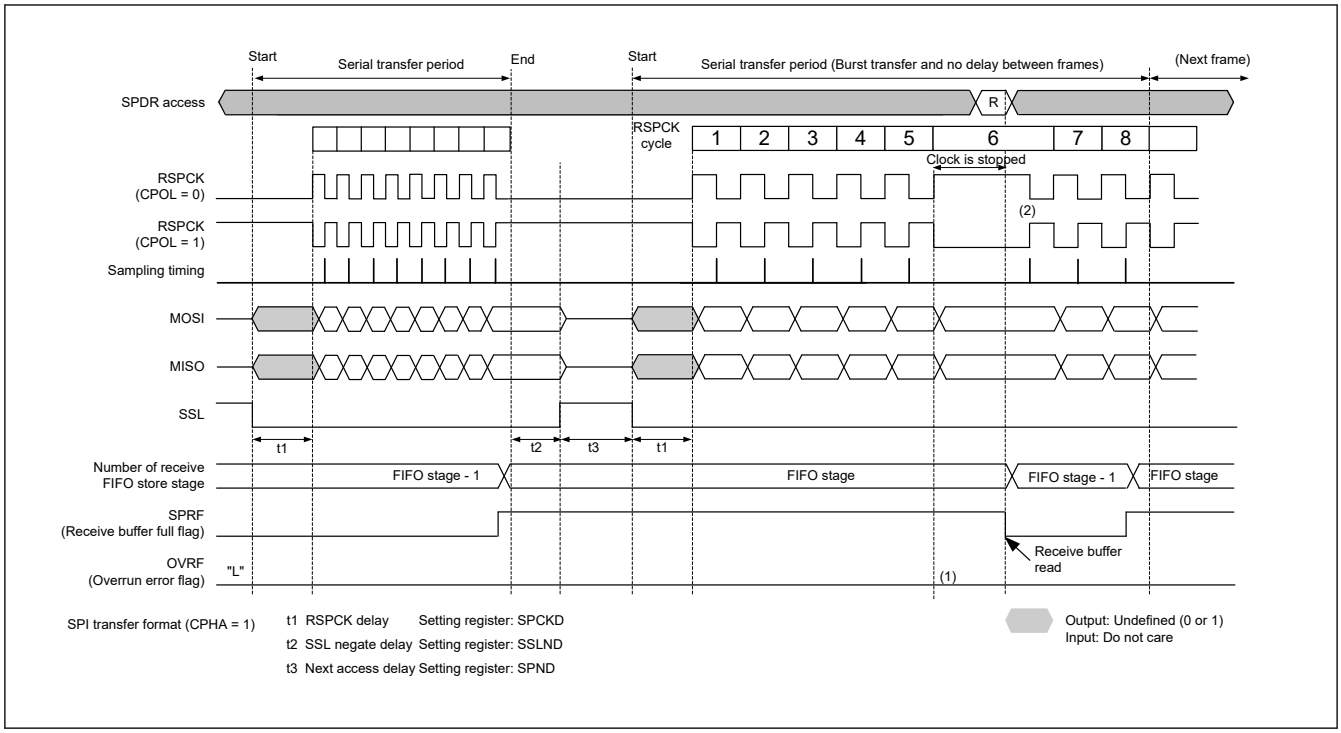


Figure 36.51 Clock stop waveform when serial transfer continues in the receive buffer full with data is stored for the number of FIFO stages in master mode (at burst transfer and no delay between frames CPHA = 1)

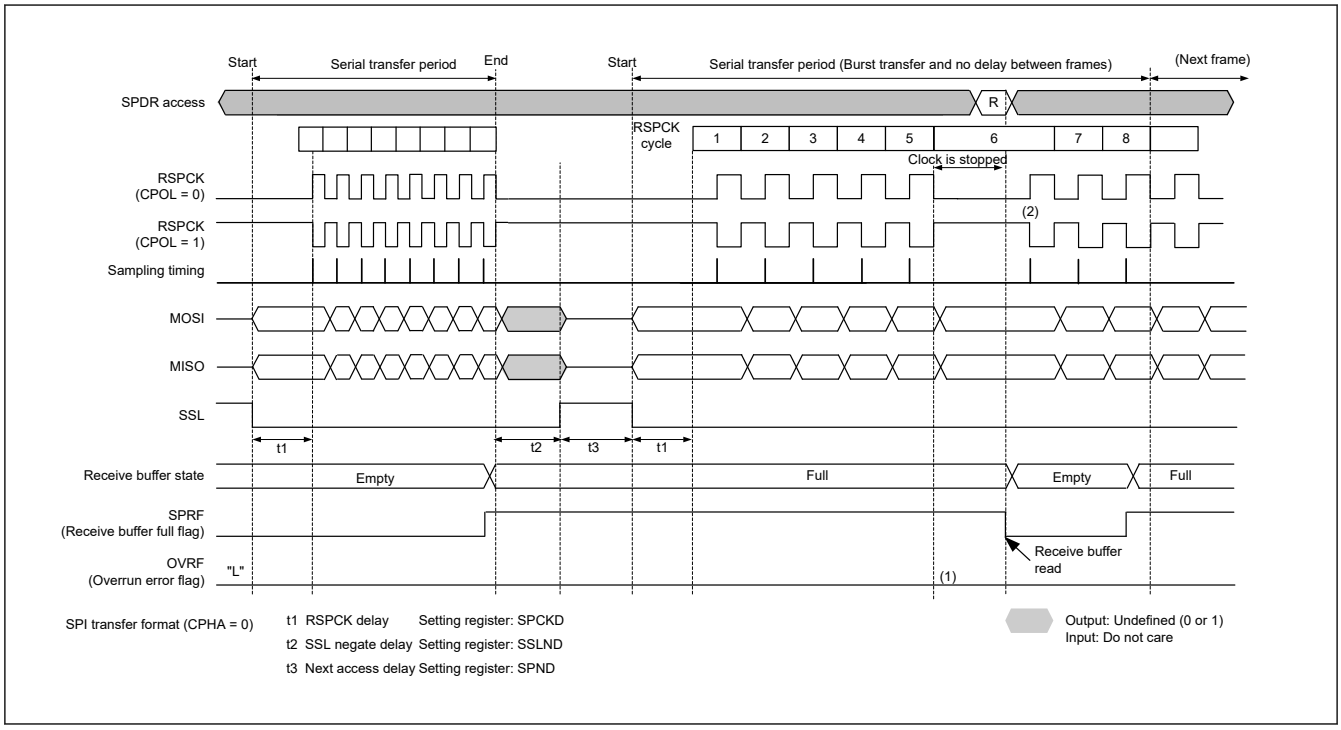


Figure 36.52 Clock stop waveform when serial transfer continues i with data is stored for the number of FIFO stages in master mode (at burst transfer and no delay between frames CPHA = 0)

The following describes operation of flags at timings (1) and (2) in the figure above.

(1) While the data is stored for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.

(2) Receive buffer data can be read by reading SPDR during clock stop. After the receive buffer data has been read, the RSPCK clock restarts.

36.4.10.2 Parity Error

After transfer in transmit-receive or receive-only master mode, transmit-receive slave mode or receive only slave mode while the SPCR.SPPE = 1, the SPI checks occurrence of a parity error. When the SPI detects a parity error in received data, the PERF flag in the SPSR register is set to 1. While the OVRF flag is 1, the SPI does not copy shift register data to the receive buffer. Therefore, parity error in received data is not detected. To clear the PERF flag in SPSR to 0, issue a system reset or write 1 to the SPSRC.PERFC bit.

Figure 36.53 shows operation of the OVRF and PERF flags in SPSR. "SPSR access" described in Figure 36.53 shows an access to SPSR. "W" shows a write cycle and "R" shows a read cycle. In the example in Figure 36.53, the SPI performs transmit-receive serial communication while SPCR.SPPE = 1. The SPI performs 8-bit data serial transfer with the settings of SPCMD.CPHA = 1 and SPCMD.CPOL = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

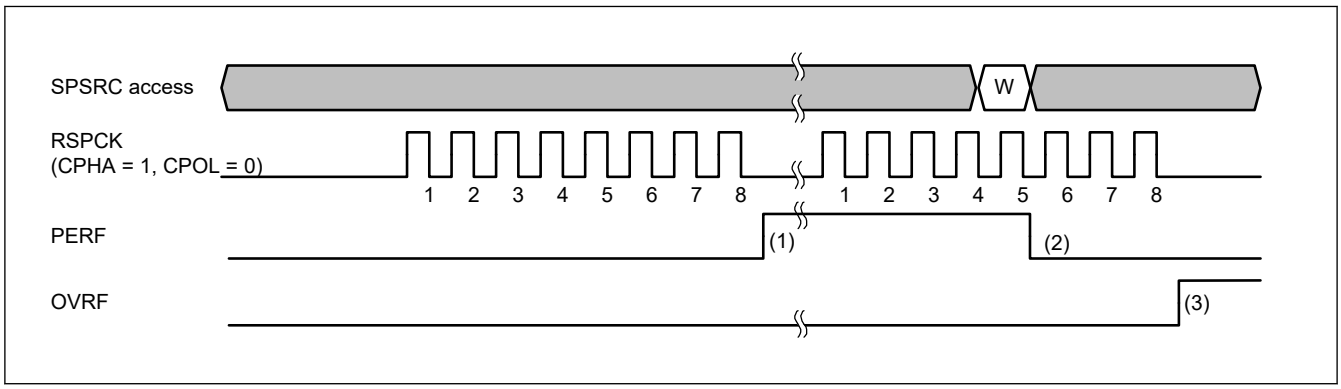


Figure 36.53 Example of operation of OVRF and PERF flags

The following describes operation of flags at timings (1) to (3) in the figure above.

(1) When the SPI does not detect an overrun error and terminates the serial transfer, the SPI copies shift register data to the receive buffer. When the SPI checks the received data and detects a parity error at this time, the PERF flag is set to 1. In master mode, the SPI copies the value of pointer to the SPCMD register to the SPSSR.SPCM [2:0] bits.

(2) When 1 is written to the SPSRC.PERFC bit.

(3) When the SPI detects an overrun error and terminates the serial transfer, the SPI does not copy shift register data to the receive buffer and does not detect a parity error.

Whether a parity error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To use an SPI error interrupt, set the SPCR.SPEIE bit to 1. To perform serial transfer without using an SPI error interrupt, read SPDR to quickly detect a parity error. When the SPI is used in master mode, the pointer value to SPCMD when an error is present can be checked by reading the SPSSR.SPECM[2:0] bits.

36.4.10.3 Mode Fault Error

When the SPCR.MSTR = 1, SPCR.SPMS = 0, and SPCR.MODFEN = 1, the SPI operates in multi-master mode. When an active level is input to the SSL0 input signal of the SPI in multi-master mode, the SPI detects a mode fault error regardless of serial transfer status and sets the MODF flag in the SPSR register to 1. When a mode fault error is detected, the SPI copies the value of pointer to the SPCMD register to the SPSSR.SPECM[2:0] bits. The SSL0 signal active level depends on the SSL0P bit in the SSLP register.

While SPCR.MSTR = 0, the SPI operates in slave mode. When SPCR.SPMS = 0 and SPCR.MODFEN = 1 in slave mode, if the SSL0 input signal is negated during the serial transfer period (from valid data drive start to final valid data latch), the SPI detects a mode fault error while any of the following 2 conditions is met.

(1) In the Motorola-SPI case

When the SSL0 input signal is negated while serial data transfer.

(2) In the TI-SSP case

When the SSL0 input signal is asserted while serial data transfer. However, during burst transfer, no error is detected even if the SSL0 input signal is asserted during the last bit of frame.

When the SPI detects a mode fault error, it stops driving output signals and clears the SPCR.SPE bit.

When the SPE bit is cleared, the SPI function is disabled (as described in section 36.4.12. SPI Initialization). In a multi-master configuration, the mastership can be released by stopping driving output signals and disabling the SPI function by using a mode fault error.

Whether a mode fault error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To detect a mode fault error without using an SPI error interrupt, poll SPSR. When the SPI is used in master mode, the pointer value to SPCMD when an error is present can be checked by reading the SPSSR.SPECM[2:0] bits.

While the MODF flag = 1, the SPI ignores writing 1 to the SPE bit. To enable the SPI function after a mode fault error is detected, clear the MODF flag to 0 without fail.

36.4.10.4 Underrun Error

While the SPI is operating in slave mode (SPCR.MSTR = 0) and the communication mode select SPCR.TXMD[1:0] bits are set to 00b or 01b, if serial transfer is started before transmit data output is ready with the SPCR.SPE bit set to 1 (SPI function enabled), the SPI detects an underrun error and sets the MODF and UDRF flags in the SPSR register to 1.

When the SPI detects an underrun error, it stops driving output signals and clears the SPE bit to 0. When the SPE bit is cleared, the SPI function is disabled (as described in section 36.4.12. SPI Initialization)

Whether an underrun error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR.

To detect an underrun error without using an SPI error interrupt, poll SPSR.

While the MODF flag = 1, the SPI ignores writing 1 to the SPE bit. To enable the SPI function after an underrun error is detected, clear the MODF flag to 0 without fail.

36.4.11 Received Data Ready Detection

When SPCR.TXMD[1:0] = 00b, 01b, or 11b, and SPDRCR.SPDRC[7:0] ≠ 0x00, after receiving data in the receive FIFO during communication (SPE = 1), SPDR.SPDRF flag is set to 1 when the received data is not stored even after the number of received FIFOs ≤ the threshold value and the value set in SPDRCR.SPDRC[7:0] has elapsed.

When the receive data ready is detected, the interrupt can be selected as SPRI or SPEI with the SPCR.SPDRES bit.

Figure 36.54 shows an example of reception data ready detection operation.

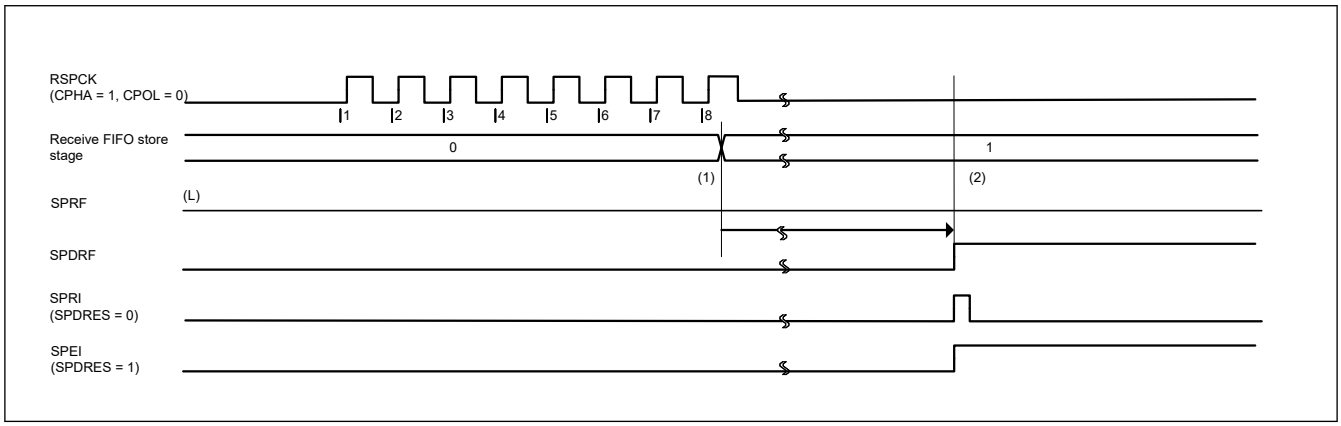


Figure 36.54 Received data ready

The following describes operation at the timings indicated by (1) and (2) in the figure.

(1) Store the received data in the receive FIFO. SPRF is 0, because receive FIFO store stage ≤ RTRG.

(2) Set SPDRF and assert SPRI or SPEI because there is no writing to the receive FIFO for the amount of SPDRC[7:0] set from (1).

36.4.12 SPI Initialization

When 0 is written to the SPCR.SPE bit or when the SPI clears the SPE bit to 0 after it detects a mode fault error or an underrun error, the SPI disables the SPI function and initializes a part of module functions. Furthermore, the SPI initializes

all module functions when a system reset occurs. The following describes initialization by clearing the SPCR.SPE bit and by using a system reset.

36.4.12.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is cleared to 0, the SPI performs the following initialization:

- Suspends the ongoing serial transfer.
- Stops driving output signals (Hi-Z) in slave mode.
- Resets the internal SPI state.
- SPSR.SPTEF flag = 1.

In initialization by clearing the SPE bit, control bits in SPI are not initialized. For this reason, re-setting the SPE bit to 1 activates the SPI with the pre-initialization transfer mode.

Values of the CENDF flag, the SPRF flag, OVRF flag, MODF flag, PERF flag, and UDRF flag in SPSR are not initialized. The SPSSR register value is not initialized either. For this reason, it is possible to read receive buffer data and check the communication completion status and the error status during SPI transfer also after the SPI is initialized.

SPSR.SPTEF flag is initialized to 1. Therefore, if the SPCR.SPTIE bit is set to 1 after the SPI is initialized, an SPI transmit buffer empty interrupt is generated. When the CPU initializes the SPI, write 0 to the SPE bit and the SPTIE bit to disable an SPI transmit buffer empty interrupt.

36.4.12.2 System Reset

In initialization by using a system reset, all SPI control bits, status bits, and data register are initialized in addition to the initialization described in section 36.4.12.1. Initialization by Clearing the SPE Bit to completely initialize the SPI.

36.4.13 SPI Operation

36.4.13.1 Master Mode Operation

Single master mode operation and multi-master mode operation are different from each other only in mode fault error detection (see section 36.4.10. Error Detection). The SPI in single master mode (SPI) detects no mode fault error.

The SPI in multi-master mode detects a mode fault error. The following describes operations common to single master mode and multi-master mode.

(1) Starting a Serial Transfer

When data is written to the SPDR register while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn, n = 0 to 3) data in SPDR.

While the shift register is empty, the SPI copies transmit buffer data to the shift register to start serial transfer. After the SPI copies transmit data to the shift register, it changes the shift register status to full. Upon completion of serial transfer, the SPI changes the shift register status to empty. The shift register status cannot be monitored.

For details about the SPI transfer format, see section 36.4.5. Transfer Format (Frame Format). The SSL output signal polarity depends on the set SSLP register value.

(2) Terminating a Serial Transfer

[Except Receive-only in Master Mode]

After the SPI detects the RSPCK edge corresponding to the final sampling timing regardless of the SPCMD.CPHA bit value, the SPI terminates serial transfer. When the number of data stored in the receive FIFO < the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPDR register after serial transfer.

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in master mode depends on the set value of the SPCMD.SPB[4:0] bits. The SSL output signal polarity depends on the set SSLP register value. For details about the SPI transfer format, see section 36.4.5. Transfer Format (Frame Format).

[Receive-only in Master Mode]

When any of the following 2 conditions is met, then SPI terminating the serial transfer.

- After the SPI detects the RSPCK edge corresponding to the final sampling timing regardless of the SPCMD.CPHA bit value, the SPI terminates serial transfer.
- When writing SPCRRM.RMEDTG = 1 during the serial transfer period, the SPI terminates the serial transfer.

When the number of data stored in the receive FIFO < the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPDR register after serial transfer.

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in master mode depends on the set value of the SPCMD.SPB[4:0] bits. The SSL output signal polarity depends on the set SSLP register value. For details about the SPI transfer format, see section 36.4.5. Transfer Format (Frame Format).

(3) Sequence Control

The transfer format in master mode is determined as follows.

The transfer format in master mode is determined by the SPSCR register, SPCMDm (m = 0 to 7) register, SPBR register, SPCKD register, SSLND register, and SPND register.

The SPSCR register is used to determine the sequence configuration for serial transfer to be performed by the SPI in master mode. The SSL output signal value, MSB first/LSB first, data length, a part of bit rate settings, RSPCK polarity and phase, SPCKD enable/disable, SSLND enable/disable, and SPND enable/disable are set in SPCMD0 to SPCMD7.

A part of bit rate settings is set in SPBR, the SPI clock delay value is set in SPCKD, the SSL negation delay value is set in SSLND, and the access delay value is set in SPND.

The SPI configures the sequence that structures a part of or whole of SPCMD0 to SPCMD7 according to the sequence length specified in SPSCR. The SPI has a pointer to SPCMD that configures the sequence. This pointer value can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPE bit in the SPCR register is set to 1 to enable the SPI function, the SPI sets the pointer to commands in SPCMD0 and applies the SPCMD0 setting to the transfer format at the start of serial transfer. The SPI increments the pointer the next-access delay period ends in each data transfer. When the serial transfer corresponding to the final command that configures the sequence is completed, the SPI sets the pointer in SPCMD0. Thus sequence is repeatedly executed.

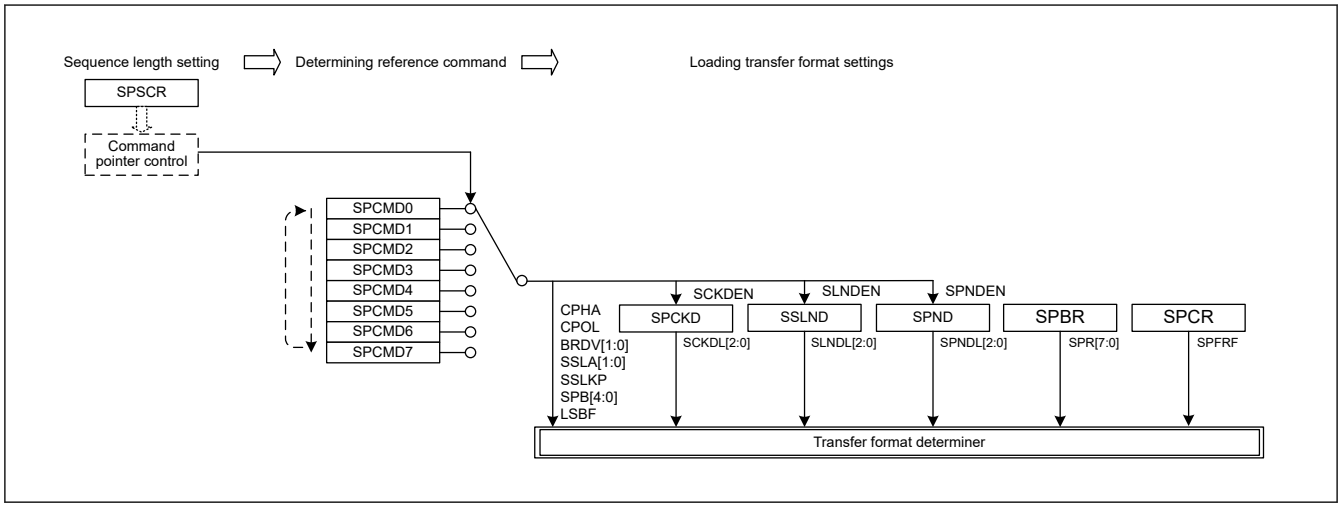


Figure 36.55 Determining serial transfer method in master mode

In this section, a frame consists of data (SPDR) and configuration (SPCMDm).

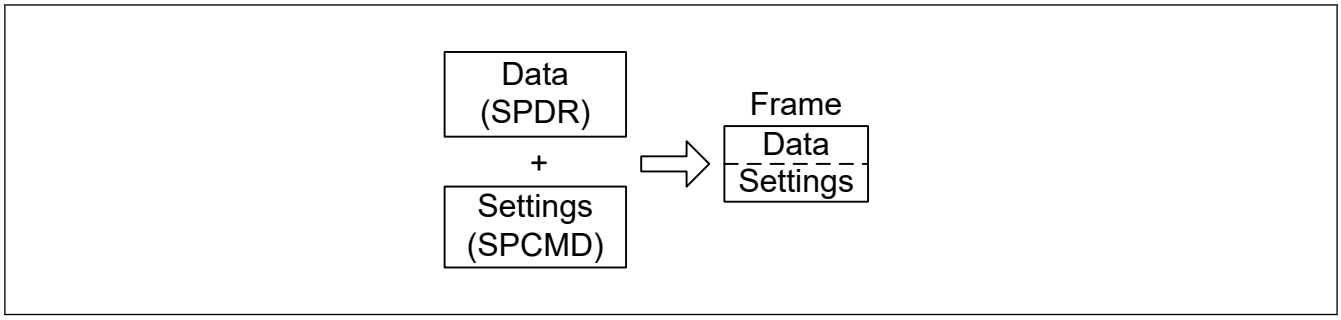


Figure 36.56 Conceptual diagram of frame

Figure 36.57 shows the correspondence between the SPI command register and transmit buffer (SPTXn, n = 0 to 3)/receive buffer (SPRXn, n = 0 to 3) in the sequence operation.

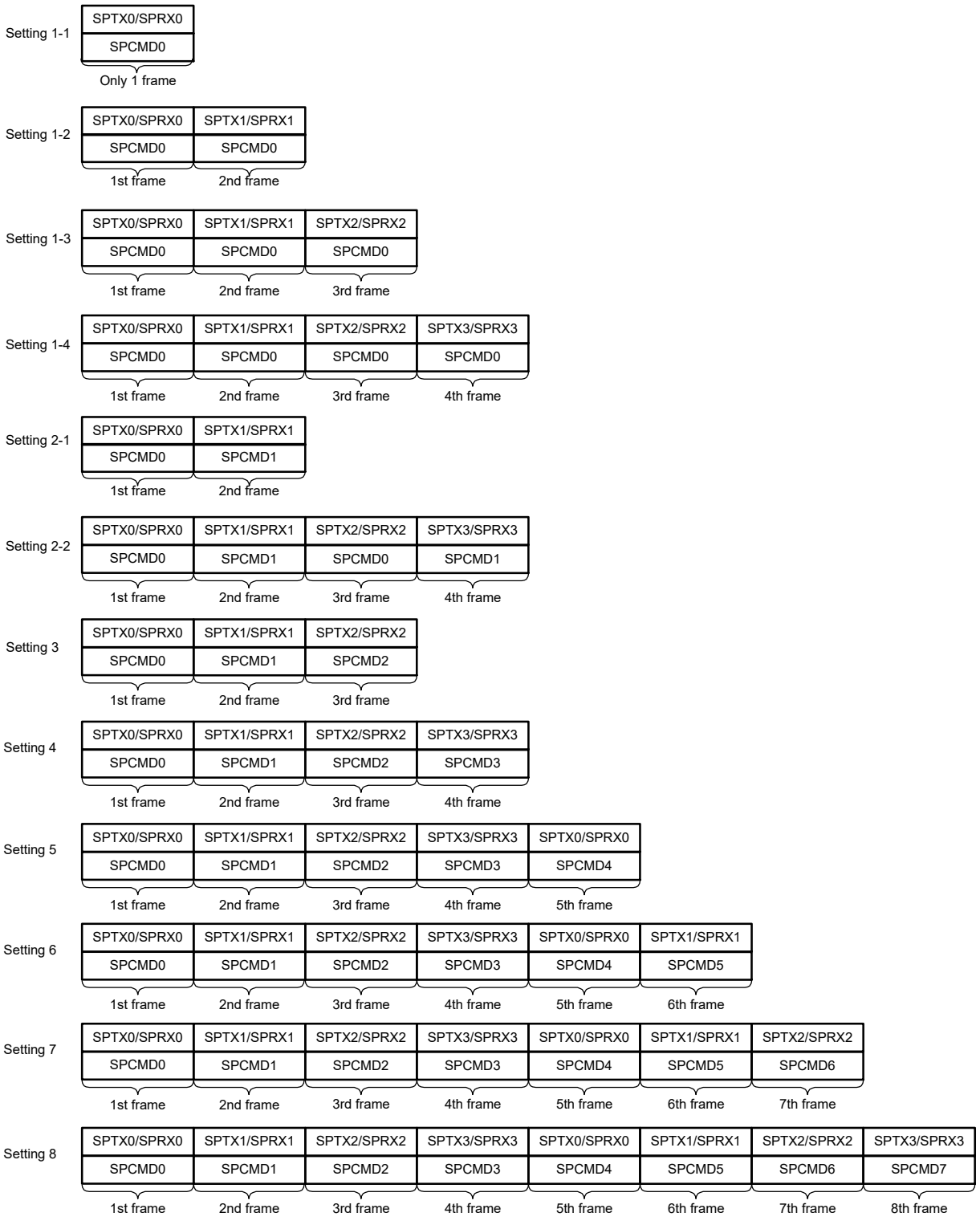


Figure 36.57 Correspondence between SPI command register and transmit/receive buffers in sequence operation

(4) Burst Transfer

This section describes burst transfer during transmit-receive / transmit-only operation.

[In the Motorola-SPI case]

When the SPCMD.SSLKP bit, which the SPI references in the current serial transfer, is 1, the SPI retains the SSL signal level during serial transfer until the SSL signal assertion of the next serial transfer starts. When the SSL signal level in the next serial transfer is the same as the SSL signal level in the current serial transfer, the SPI can continuously perform serial transfer while holding the SSL signal assertion status (burst transfer).

(a) SPCR.BFDS = 0

Figure 36.58 shows an example of SSL signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (8) shown in Figure 36.58. The SSL output signal polarity depends on the set the SSLP register.

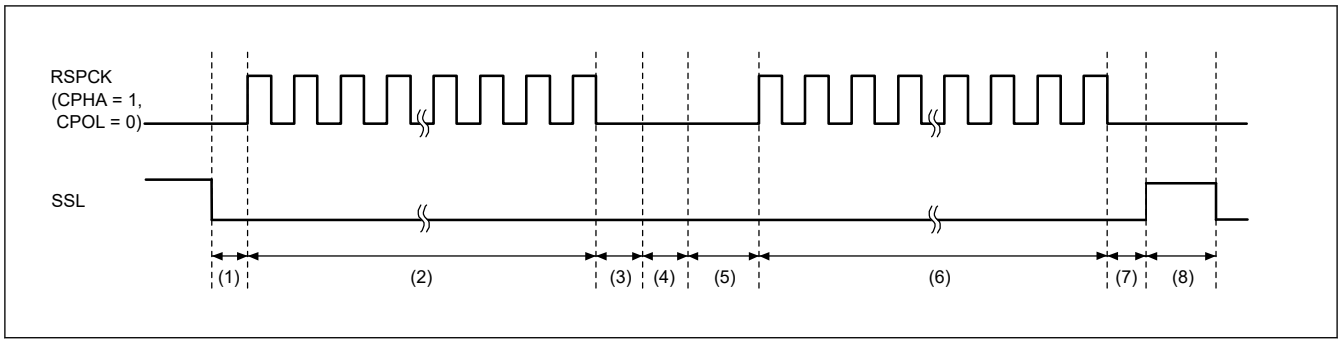


Figure 36.58 Example of burst transfer operation using SSLKP bit (BFDS = 0, SPFRF = 0)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0.

(2) Perform serial transfer according to SPCMD0.

(3) Insert an SSL negation delay.

(4) The SSL signal value in SPCMD0 is retained because SPCMD0.SSLKP = 1. This period additionally continues for 5 PCLKSPIn cycles (at minimum) that is the same as the next-access delay time of SPCMD0. When the shift register is empty after the minimum period has passed, this period continues until transmit data for the next transfer is stored in the shift register.

(5) Assert the SSL signal and insert an RSPCK delay according to SPCMD1.

(6) Perform serial transfer according to SPCMD1.

(7) Insert SSL negate delay.

(8) The SSL signal is negated because SPCMD1.SSLKP = 0. Furthermore, the next-access delay is inserted according to SPCMD1.

If the SSL signal output setting in SPCMD with the SSLKP bit set to 1 differs from the SSL signal output setting in SPCMD to be used for the next transfer, the SPI changes the SSL signal state when the SSL signal corresponding to the next-transfer command is asserted (5). Note that, if an SSL signal change like this takes place, slaves that drive the MISO signal may conflict with each other, which may cause collision of signal level.

The SPI in master mode references the SSL signal operation in the module when SSLKP is not used. Even when SPCMD.CPHA = 0, the SPI can accurately start serial transfer by using the next-transfer SSL signal assertion detected internally. For this reason, burst transfer in master mode is enabled regardless of the set CPHA bit value. (See section 36.4.13. SPI Operation.)

(b) SPCR.BFDS = 1

Figure 36.59 shows an example of SSL signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 36.59. The SSL output signal polarity depends on the set SSSLP register value.

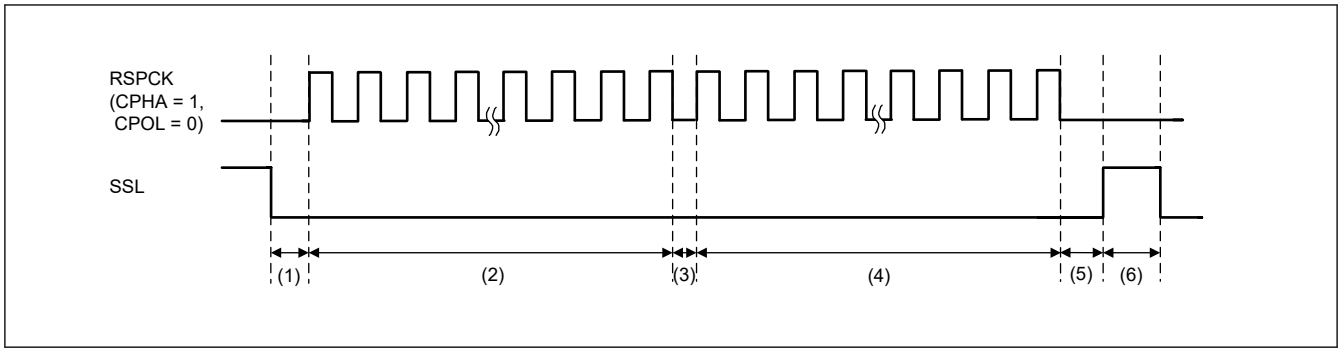


Figure 36.59 Example of burst transfer operation using SSLKP bit (BFDS = 1, SPFRF = 0)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst transmisshon.

(2) Perform serial transfer according to SPCMD0. Wait last clock until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.

(3) The value of SSL signal according to SPCMD0 was hold, because SPCMD0.SSLKP = 1. RSPCK negate period between frames is 0.5RSPCK, if the shift register is not empty.

(4) Perform serial transfer according to SPCMD1.

(5) Insert SSL negate delay for the last frame.

(6) The SSL signal is negated because SPCMD1.SSLKP = 0. Furthermore, the next-access delay is inserted according to SPCMD1.

[In the TI-SSP case]

SPI asserts the SSL signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSL signal for one cycle at the start of the next serial transfer (burst transfer).

(a) SPCMD.SSLKP = 1 and SPCR.BFDS = 1

SPCMD0 to SPCMD1 are shown in Figure 36.60. The following shows an example of SSL signal operation and serial data MISO/MOSI when burst transfer is realized using the settings. The SSL output signal polarity depends on the set SPI slave select polarity register (SSLP) value.

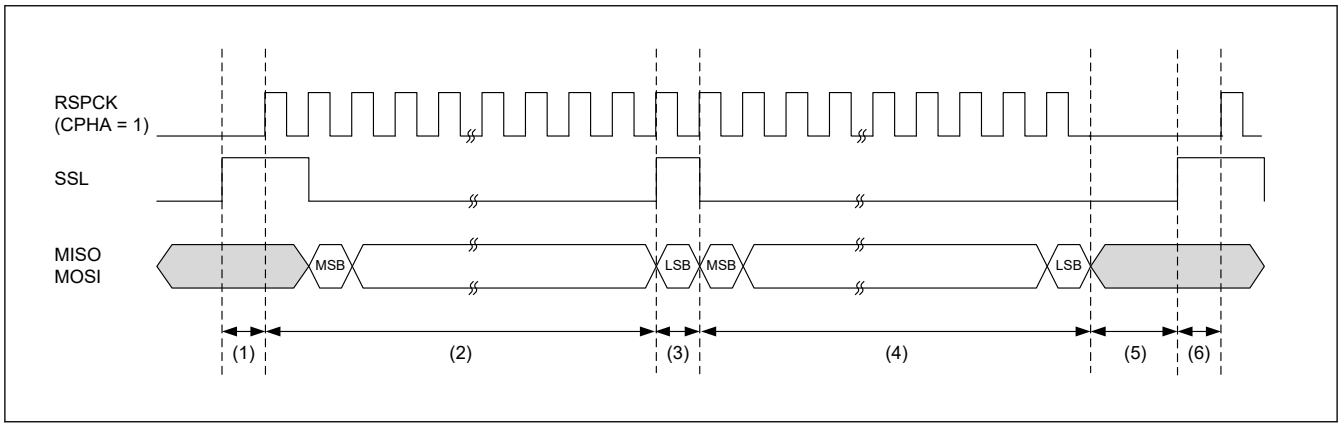


Figure 36.60 Example of burst transfer operation (SPFRF = 1)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst transmission.

(2) Perform serial transfer according to SPCMD0.

(3) Final data transfer and SSL assertion are performed simultaneously. If the shift register is empty during the RSPCK negation period between frames, wait for the output of the last clock until the transmission data for the next transfer is stored in the shift register.

(4) Perform serial transfer according to SPCMD1.

(5) Insert SSL negate delay for the last frame.

(6) The SSL signal is negated because the SPCMD1.SSLKP = 0.

If the SSL signal output setting in SPCMD with the SSLKP bit set to 1 differs from the SSL signal output setting in SPCMD to be used for the next transfer, the SPI changes the SSL signal state when the SSL signal corresponding to the next-transfer command is asserted (5). Note that, if an SSL signal change like this takes place, slaves that drive the MISO signal may conflict with each other, which may cause collision of signal level.

This section describes burst transfer during receive-only operation.

[In the Motorola-SPI case]

When the SPCMD.SSLKP bit, which the SPI references in the current serial transfer, is 1, the SPI retains the SSL signal level during serial transfer until the SSL signal assertion of the next serial transfer starts. When the SSL signal level in the next serial transfer is the same as the SSL signal level in the current serial transfer, the SPI can continuously perform serial transfer while holding the SSL signal assertion status (burst transfer).

(a) SPCR.BFDS = 0

Figure 36.61 shows an example of SSL signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (8) shown in Figure 36.61. The SSL output signal polarity depends on the set SSLP register value.

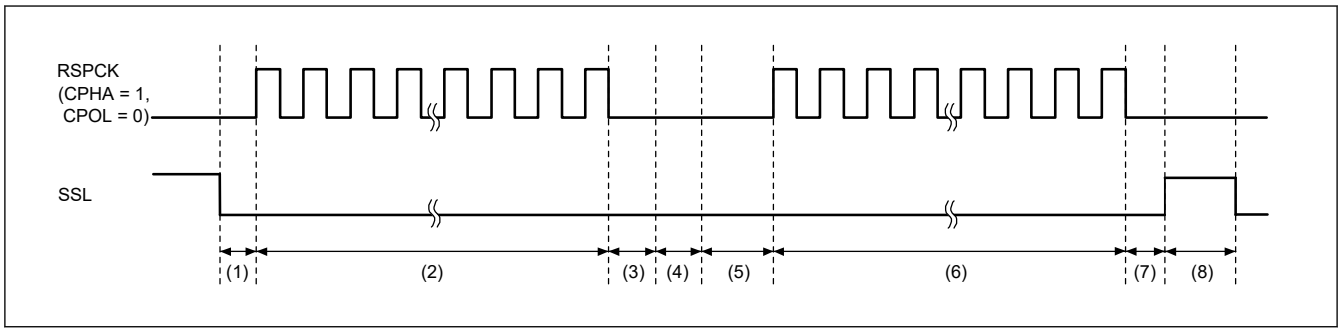


Figure 36.61 Example of burst transfer operation using SSLKP bit (BFDS = 0, SPFRF = 0)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0.

(2) Perform serial transfer according to SPCMD0.

(3) Insert an SSL negation delay.

(4) The SSL signal value in SPCMD0 is retained because SPCMD0.SSLKP = 1. This period additionally continues for 5 PCLKSPIn cycles (at minimum) that is the same as the next-access delay time of SPCMD0.

(5) Assert the SSL signal and insert an RSPCK delay according to SPCMD1.

(6) Perform serial transfer according to SPCMD1.

(7) Insert SSL negate delay.

(8) The SSL signal is negated because SPCMD1.SSLKP = 0. Furthermore, the next-access delay is inserted according to SPCMD1

If the SSL signal output setting and the SSL signal output setting between SPCMDs used for burst transfer are different, SPI switches the SSL signal state when the SSL signal corresponding to the next transfer command is asserted (5). Note that, if an SSL signal change like this takes place, slaves that drive the MISO signal may conflict with each other, which may cause collision of signal level.

The SPI in master mode references the SSL signal operation in the module when SSLKP is not used. Even when SPCMD.CPHA = 0, the SPI can accurately start serial transfer by using the next-transfer SSL signal assertion detected internally. For this reason, burst transfer in master mode is enabled regardless of the set CPHA bit value. (See section 36.4.13. SPI Operation.)

(b) SPCR.BFDS = 1

Figure 36.62 shows an example of SSL signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 36.62. The SSL output signal polarity depends on the set SSLP register value.

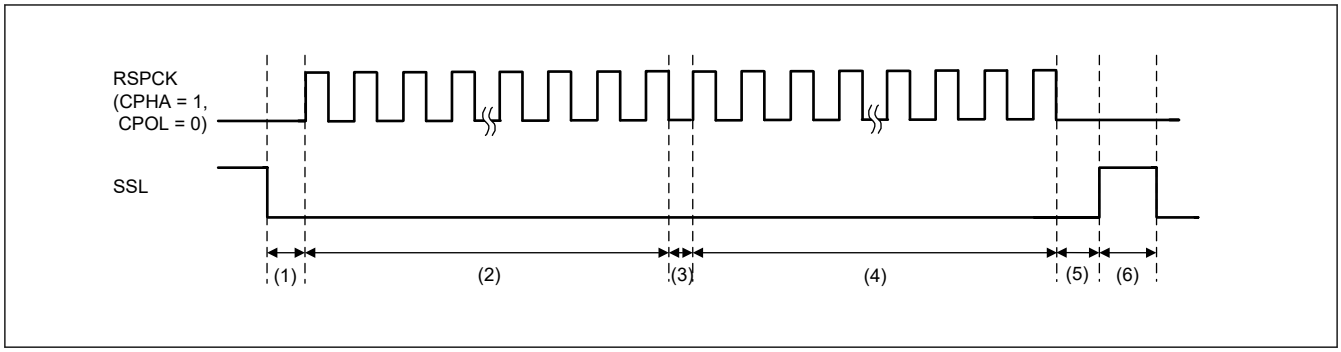


Figure 36.62 Example of burst transfer operation using SSLKP bit (BFDS = 1, SPFRF = 0)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst.

(2) Perform serial transfer according to SPCMD0.

(3) Since it is not the last frame, the SSL signal value at SPCMD0 is retained. RSPCK negation between frames is 0.5 RSPCK for the next frame.

(4) Perform serial transfer according to SPCMD1.

(5) Insert SSL negate delay for the last time.

(6) The SSL signal is negated because SPCMD1.SSLKP = 0. Furthermore, the next-access delay is inserted according to SPCMD1.

Note: "Last frame": Frame set by RMFM[4:0] bits when SPCRRM.RMFM[4:0] ≠ 0x00
Or, a frame in which SPCRRM.RMEDTG = 1 has been accepted.

[In the TI-SSP case]

SPI asserts the SSL signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSL signal for one cycle at the start of the next serial transfer (burst transfer).

(a) SPCMD.SSLKP = 1 and SPCR.BFDS = 1

SPCMD0 to SPCMD1 are shown in Figure 36.63. The following shows an example of SSL signal operation and serial data MISO/MOSI when burst transfer is realized using the settings. The SSL output signal polarity depends on the set SSLP register value.

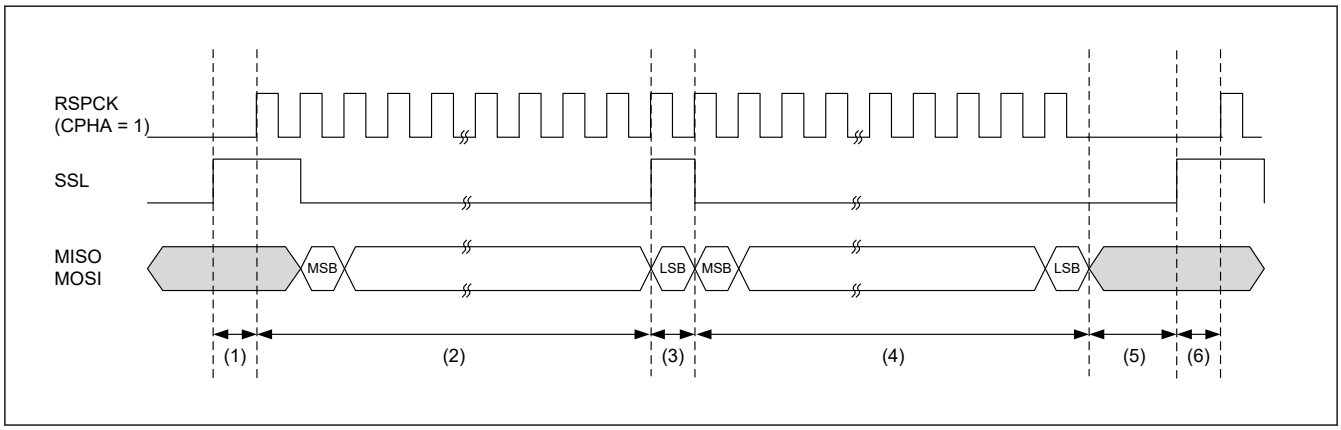


Figure 36.63 Example of burst transfer operation (SPFRF = 0)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst.

(2) Perform serial transfer according to SPCMD0.

(3) Final data transfer and SSL assertion are performed simultaneously.

(4) Perform serial transfer according to SPCMD1.

(5) Insert Output disable delay for the last frame.

(6) The SSL signal is negated because SPCMD1.SSLKP = 0.

Note: "Last frame": Frame set by RMFM[4:0] bits when SPCRRM.RMFM[4:0] ≠ 0x00
Or, a frame in which SPCRRM.RMEDTG = 1 has been accepted.

If the SSL signal output setting between the SPCMDs used for burst transfer differs from the SSL signal output setting, SPI switches the SSL signal state when the SSL signal corresponding to the next transfer command is asserted (5). Note that, if an SSL signal change like this takes place, slaves that drive the MISO signal may conflict with each other, which may cause collision of signal level.

(5) RSPCK Delay (t1)

The RSPCK delay value of the SPI in master mode depends on the settings of the SPCMD.SCKDEN and SPCKD.SCKDL[2:0] bits. The SPI determines SPCMD to be referenced in serial transfer by the pointer control, and then determines the RSPCK delay value for serial transfer by using the selected SPCMD.SCKDEN bit and SPCKD.SCKDL[2:0] bits as shown in Table 36.13. For the definition of RSPCK delay, see section 36.4.5. Transfer Format (Frame Format).

RSPCK delay insert to only the first frame of burst transmission, when transmit without "Between Burst Transfer Frames Delay" (SPCMD.SSLKP = 1, SPCR.BFDS = 1).

Table 36.13 Relationship between SCKDEN/SPCKD and RSPCK delay value

SPCMDm.SCKDEN bit	SPCKD.SCKDL[2:0] bits	RSPCK delay value	
		Motorola-SPI	TI-SSP
0	000b to 111b	1 RSPCK	0 RSPCK
1	000b	1 RSPCK	1 RSPCK
	001b	2 RSPCK	2 RSPCK
	010b	3 RSPCK	3 RSPCK
	011b	4 RSPCK	4 RSPCK
	100b	5 RSPCK	5 RSPCK
	101b	6 RSPCK	6 RSPCK
	110b	7 RSPCK	7 RSPCK
	111b	8 RSPCK	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value of the SPI in master mode depends on the settings of the SPCMD.SLNDEN bit and SSLND.SLNDL[2:0]. The SPI determines SPCMD to be referenced in serial transfer by the pointer control, and then determines the SSL negation delay value for serial transfer by using the selected SPCMD.SLNDEN and SSLND.SLNDL[2:0] as shown in Table 36.14. For the definition of SSL negation delay, see section 36.4.5. Transfer Format (Frame Format).

SSL negate delay insert to only the last frame of burst transmission, when transmit without "Between Burst Transfer Frames Delay" (SPCMD.SSLKP = 1, SPCR.BFDS = 1).

Table 36.14 Relationship between SSLND and SSL negation delay value (1 of 2)

SLNDEN	SSLND.SLNDL[2:0] bits	SSL negation delay value
0	000b to 111b	1 RSPCK cycle

Table 36.14 Relationship between SSLND and SSL negation delay value (2 of 2)

SLNDEN	SSLND.SLNDL[2:0] bits	SSL negation delay value
1	000b	1 RSPCK cycle
	001b	2 RSPCK cycle
	010b	3 RSPCK cycle
	011b	4 RSPCK cycle
	100b	5 RSPCK cycle
	101b	6 RSPCK cycle
	110b	7 RSPCK cycle
	111b	8 RSPCK cycle

(7) Next-Access Delay (t3)

The next-access delay value of the SPI in master mode depends on the settings of the SPCMD.SPNDEN bit and SPND.SPNDL[2:0] bits. The SPI determines SPCMD to be referenced in serial transfer by the pointer control, and then determines the RSPCK delay value for serial transfer by using the selected SPCMD.SPNDEN bit and SPND.SPNDL[2:0] bits as shown in Table 36.15. For the definition of next-access delay, see section 36.4.5. Transfer Format (Frame Format).

Next-Access delay insert to only the last frame of burst transmission, when transmit without "Between Burst Transfer Frames Delay" (SPCMD.SSLKP = 1, SPCR.BFDS = 1).

Table 36.15 Relationship between SPNDEN bit, SPND bits, and next-access delay value

SPNDEN bit	SPND.SPNDL[2:0] bits	Next-access delay value
0	000b to 111b	1 RSPCK + 5 PCLKSPIn cycle
1	000b	1 RSPCK + 5 PCLKSPIn cycle
	001b	2 RSPCK + 5 PCLKSPIn cycle
	010b	3 RSPCK + 5 PCLKSPIn cycle
	011b	4 RSPCK + 5 PCLKSPIn cycle
	100b	5 RSPCK + 5 PCLKSPIn cycle
	101b	6 RSPCK + 5 PCLKSPIn cycle
	110b	7 RSPCK + 5 PCLKSPIn cycle
	111b	8 RSPCK + 5 PCLKSPIn cycle

(8) Initialization Flowchart

Figure 36.64 shows an example of initialization flow when using the SPI in master-mode SPI operation. For how to set the interrupt controller, DMAC, and input/output ports, see descriptions of each block.

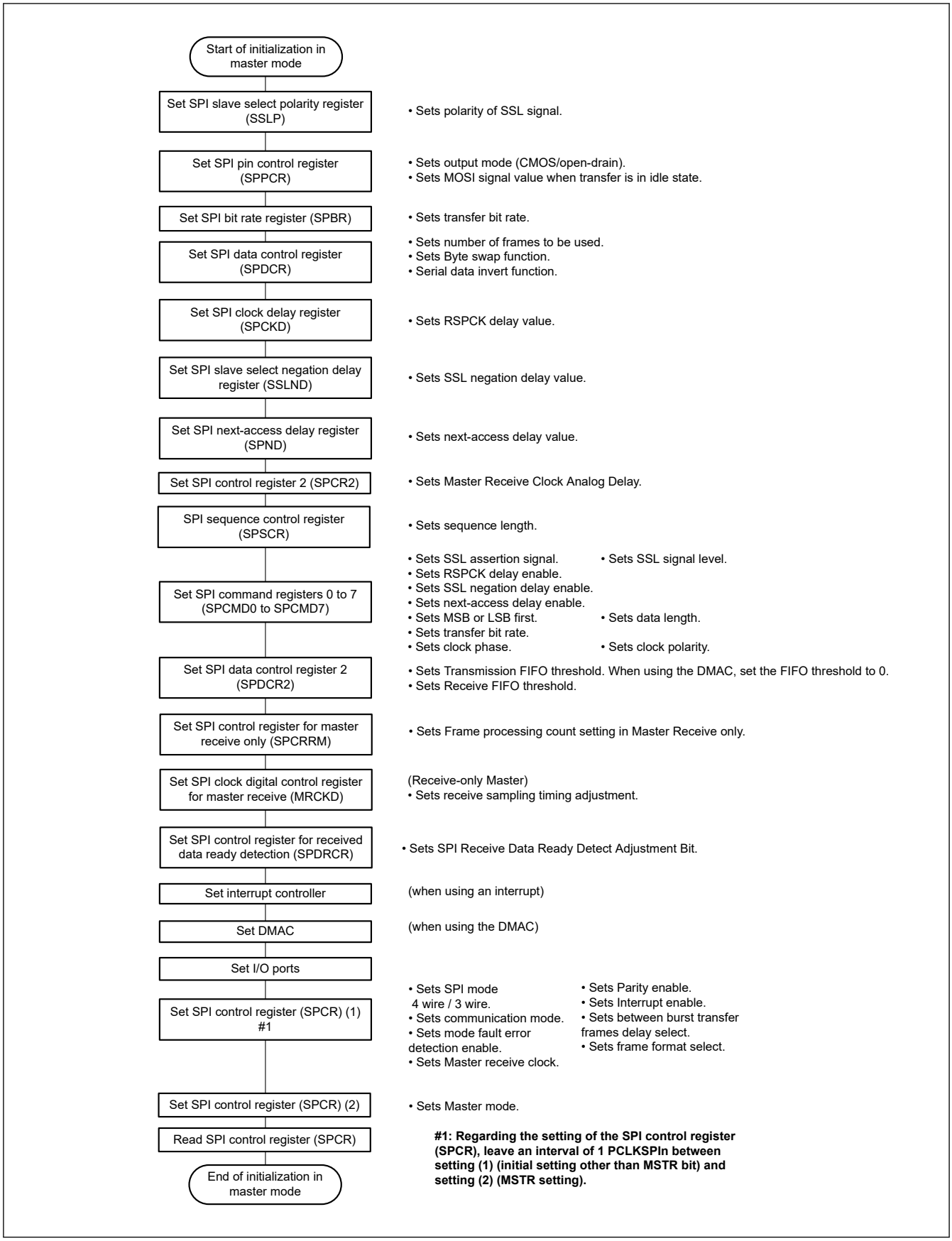


Figure 36.64 Example of initialization flowchart in master mode (SPI operation)

(9) Software Processing Flow

Figure 36.65 to Figure 36.68 show examples of the flow of software processing.

(a) Transmit Processing Flow

Enabling an SPII or SPCEND interrupt after final transmit data is written makes it possible to notify the CPU that transmission of all data has been completed.

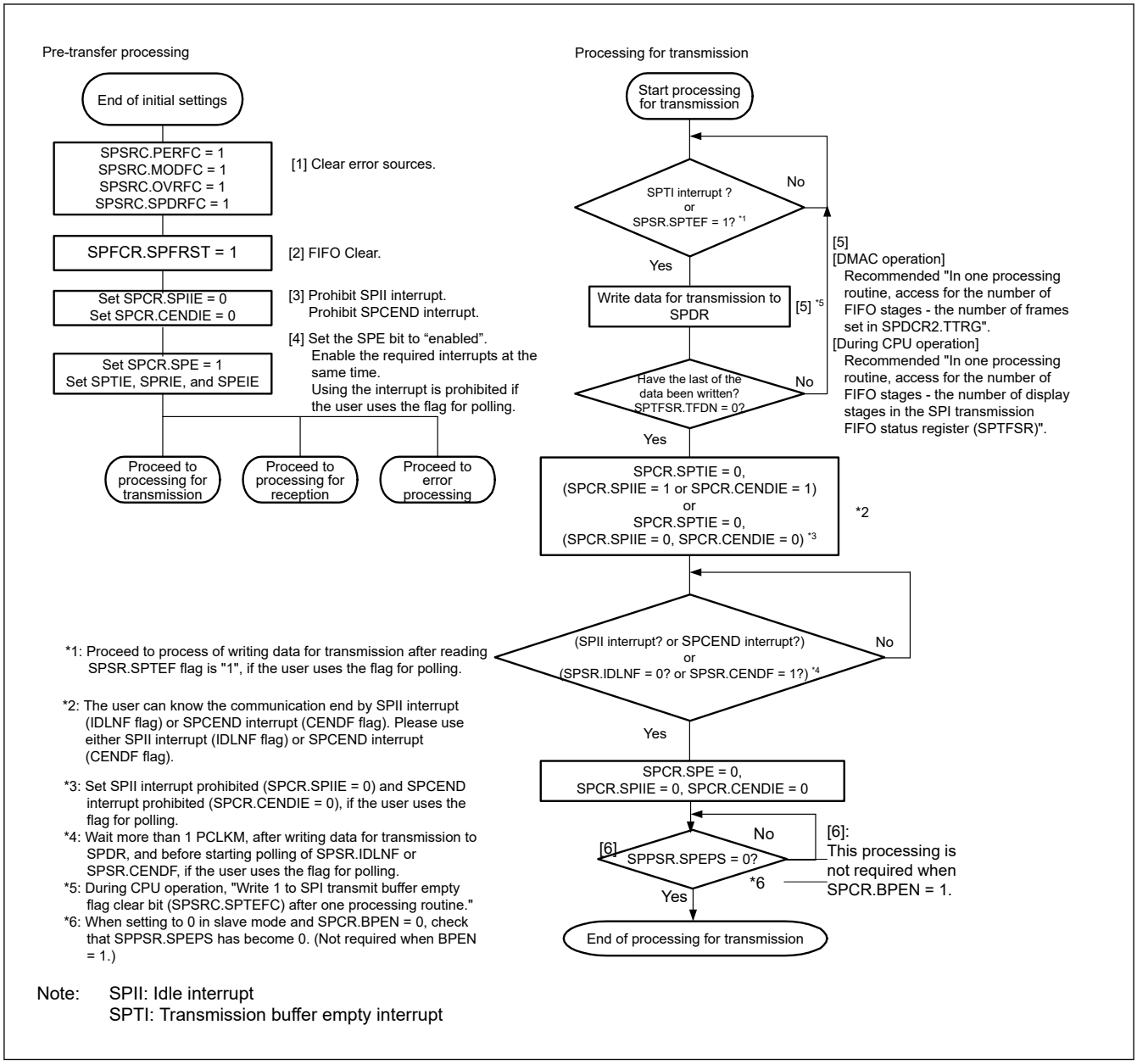


Figure 36.65 Software processing flowchart in master mode (transmission)

(b) Reception Processing Flow

The SPI always requires transmission because it does not have reception-only operation.

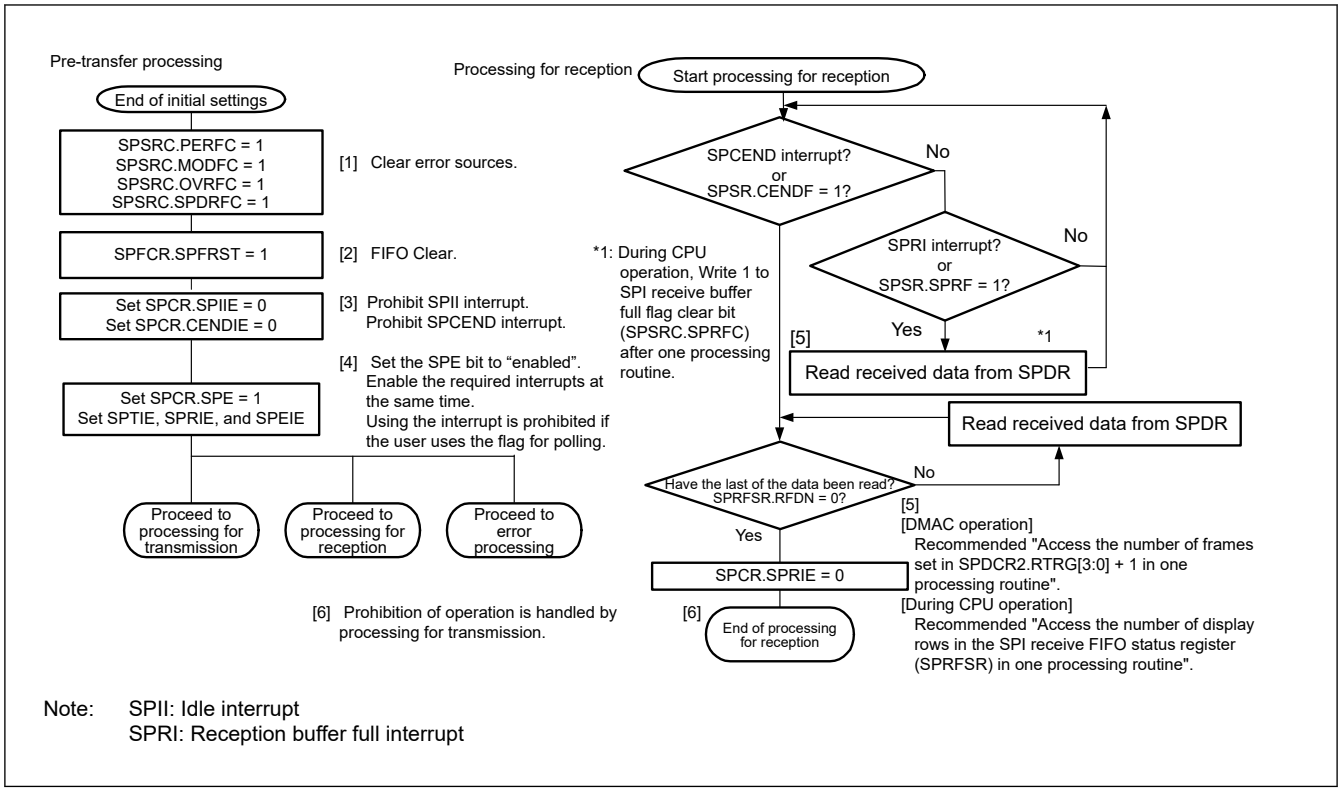


Figure 36.66 Software processing flowchart in master mode (reception)

(c) Master Reception-only processing flow

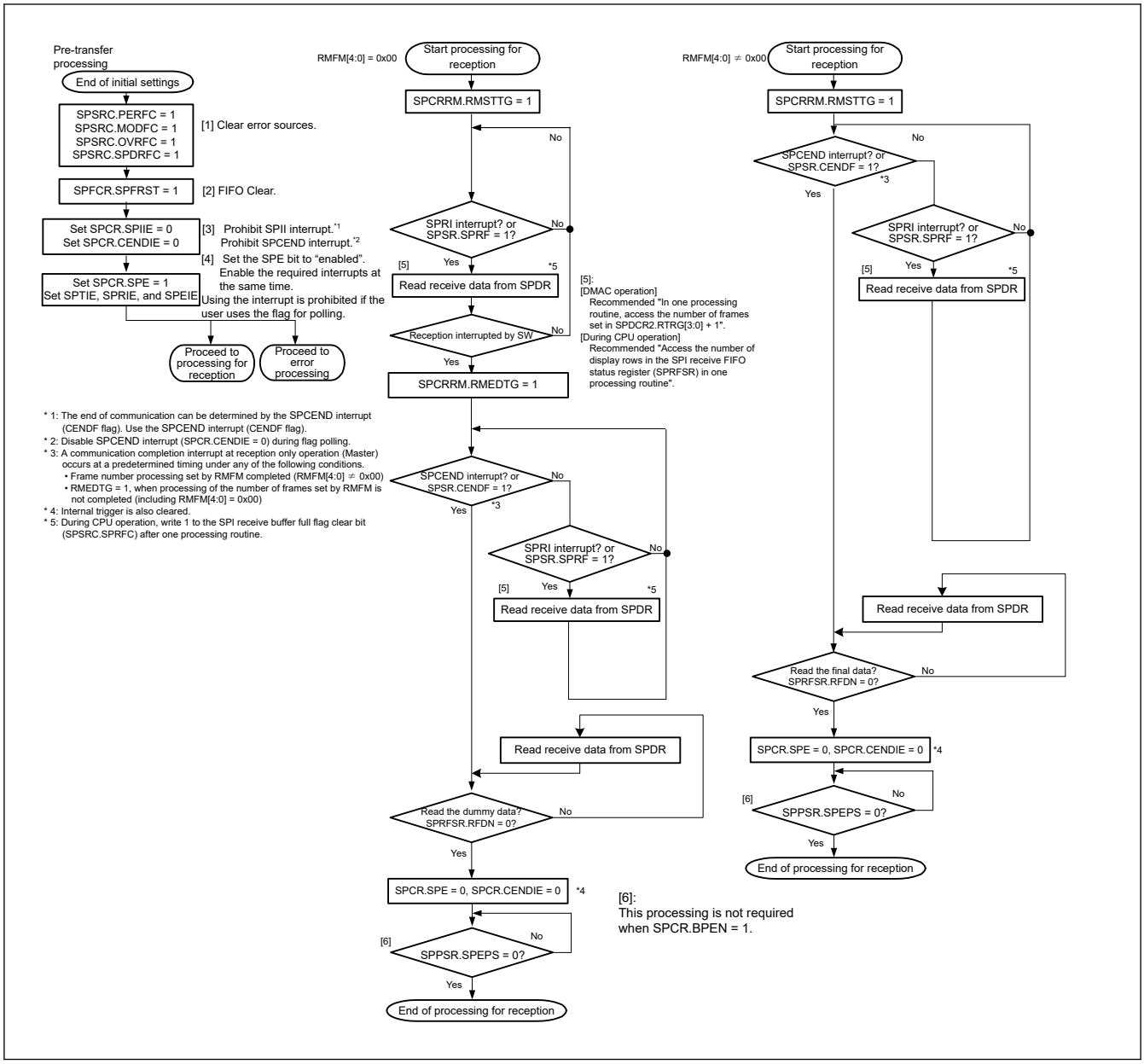


Figure 36.67 Software processing flowchart in master mode (reception-only)

(d) Error processing flow

When a mode fault error occurs, the SPE bit is automatically cleared to stop transmission and reception operations.

However, other error sources do not clear the SPE bit or stop transmission and reception operations. For this reason, if an error occurs due to another source different from the source of the error occurred first, the SPECM[2:0] value is updated. Therefore, we recommend that you clear the SPE bit to stop the ongoing operation.

When interrupts are used and an error has occurred, clear an Interrupt flag in the error processing because an SPTI interrupt request or an SPRI interrupt request may be remaining in an Interrupt flag. Furthermore, when an SPRI interrupt request is remaining, read the receive buffer to initialize the SPI's internal sequencer.

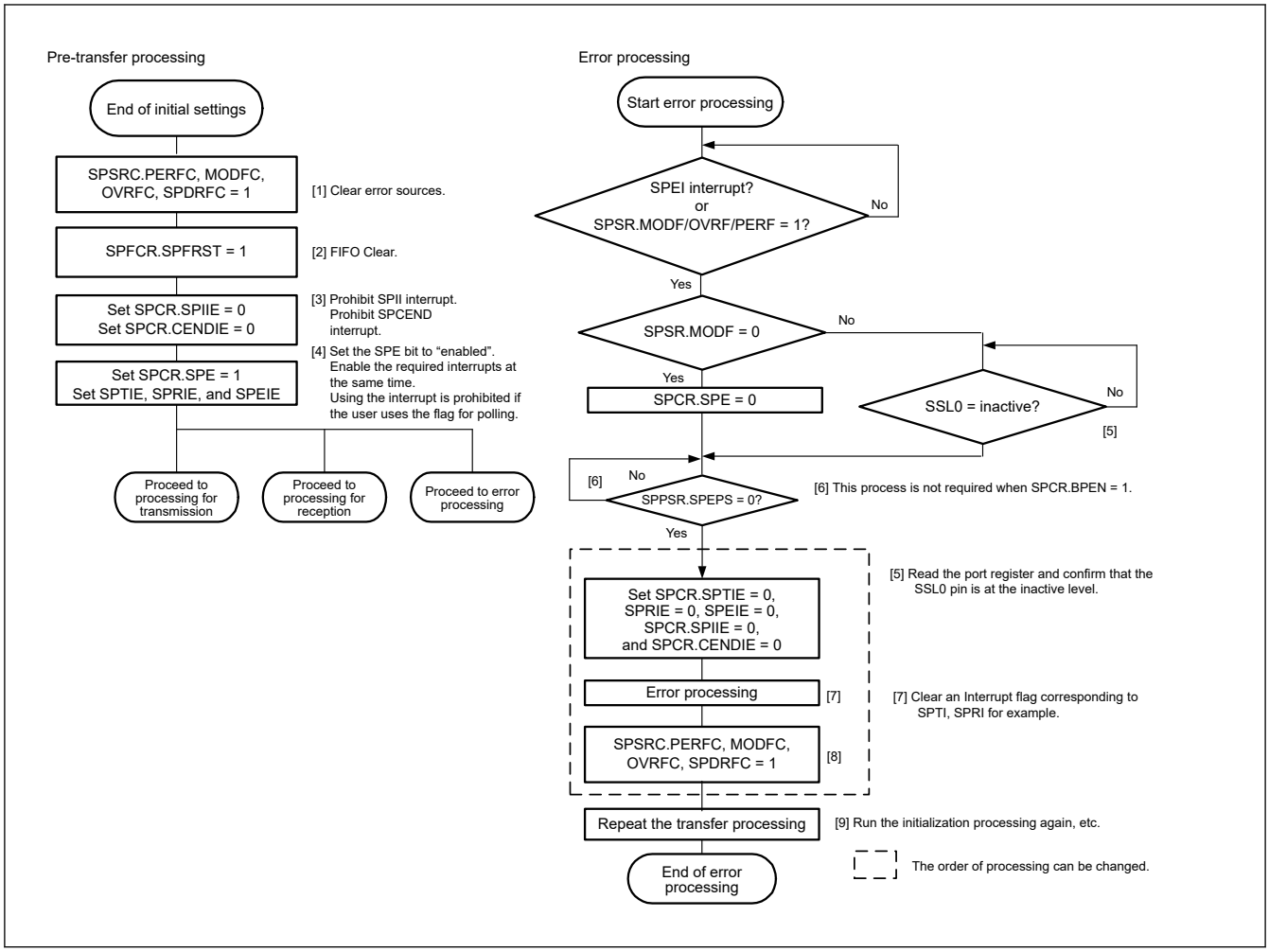


Figure 36.68 Flowchart for master mode (error processing)

36.4.13.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPI detects SSL0 input signal assertion while SPCMD0.CPHA = 0, valid data drive to the MISO output signal must be started. For this reason, the SSL0 input signal assertion becomes a trigger to start serial transfer.

While SPCMD0.CPHA = 1, when the SPI detects the first RSPCK edge with the SSL0 input signal asserted, valid data drive to the MISO output signal must be started. For this reason, the first RSPCK edge in the SSL0 signal assertion state becomes a trigger to start serial transfer when SPCMD0.CPHA = 1.

The SPI starts driving the MISO output signal at the SSL0 signal assertion timing independently of the CPHA bit setting. Data that the SPI outputs is enabled or disabled depending on the set CPHA bit value.

For details about the SPI transfer format, see section 36.4.5. Transfer Format (Frame Format). The SSL0 input signal polarity depends on the set SSL0P bit value in the SSLP register.

(2) Terminating a Serial Transfer

Upon detecting the RSPCK edge corresponding to the final sampling timing regardless of the value of SPCMD0.CPHA bit, the SPI terminates serial transfer. When the number of data stored in the receive FIFO is smaller than the number of FIFO stages, the SPI copies received data from the shift register to the receive buffer in the SPDR register after serial transfer. The SPI also changes the shift register status to empty after serial transfer regardless of the receive buffer status. When the SPI detects SSL0 input signal negation during a period from serial transfer start to the end of serial transfer, a mode fault error occurs. (See section 36.4.10. Error Detection.)

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in slave mode depends on the set value of the SPCMD0.SPB[4:0] bits. The SSL0 input signal polarity depends on the set value of SSLP.SSL0P bit. For details about the SPI transfer format, see section 36.4.5. Transfer Format (Frame Format).

(3) Notes on Single-Slave Operations

[In the Motorola-SPI case]

Upon detecting the SSL0 input signal assertion edge while SPCMD0.CPHA = 0, the SPI starts serial transfer. When using the SPI in single slave mode in a configuration as shown in the example in Figure 36.9, the SPI cannot start serial transfer correctly while CPHA is set to 0 because the SSL0 input signal is always active. To correctly perform transmission and reception of the SPI in slave mode in a configuration where the SSL0 input signal is fixed to active state, set the CPHA bit to 1. If the CPHA bit needs to be set to 0, do not fix the SSL0 input signal.

[In the TI-SSP case]

When TI SSP is set, when SPI is used as a single slave in the configuration shown in the example of Figure 36.9, the SSL0 input signal is always fixed to the inactive state, SPI cannot start serial transfer correctly.

When using a single slave, use the configuration shown in the example in Figure 36.8.

(4) Burst Transfer

[In the Motorola-SPI case]

When SPCMD0.CPHA = 1, continuous serial transfer (burst transfer) is possible while maintaining the SSL0 input signal assertion state. When SPCMD0.CPHA = 1, the period (from the first RSPCK edge in the SSL0 input signal active state until the sampling timing for receiving the final bit) corresponds to the serial transfer period. Even if the SSL0 input signal is always at the active level, start of an access can be detected, which allows burst transfer.

When SPCMD0.CPHA = 0, the second and subsequent serial burst transfers cannot be performed correctly for the same reason as (3).

[In the TI-SSP case]

In serial transfer, data transfer starts after the SSL input signal is asserted for RSPCK 1 cycle. Since frame transfer starts from the SSL input signal, SSL must be asserted between frames.

(5) Initialization Flowchart

Figure 36.69 shows an example of initialization flow when using the SPI in slave-mode SPI operation. For how to set the interrupt controller, DMAC, and input/output ports, see descriptions of each block.

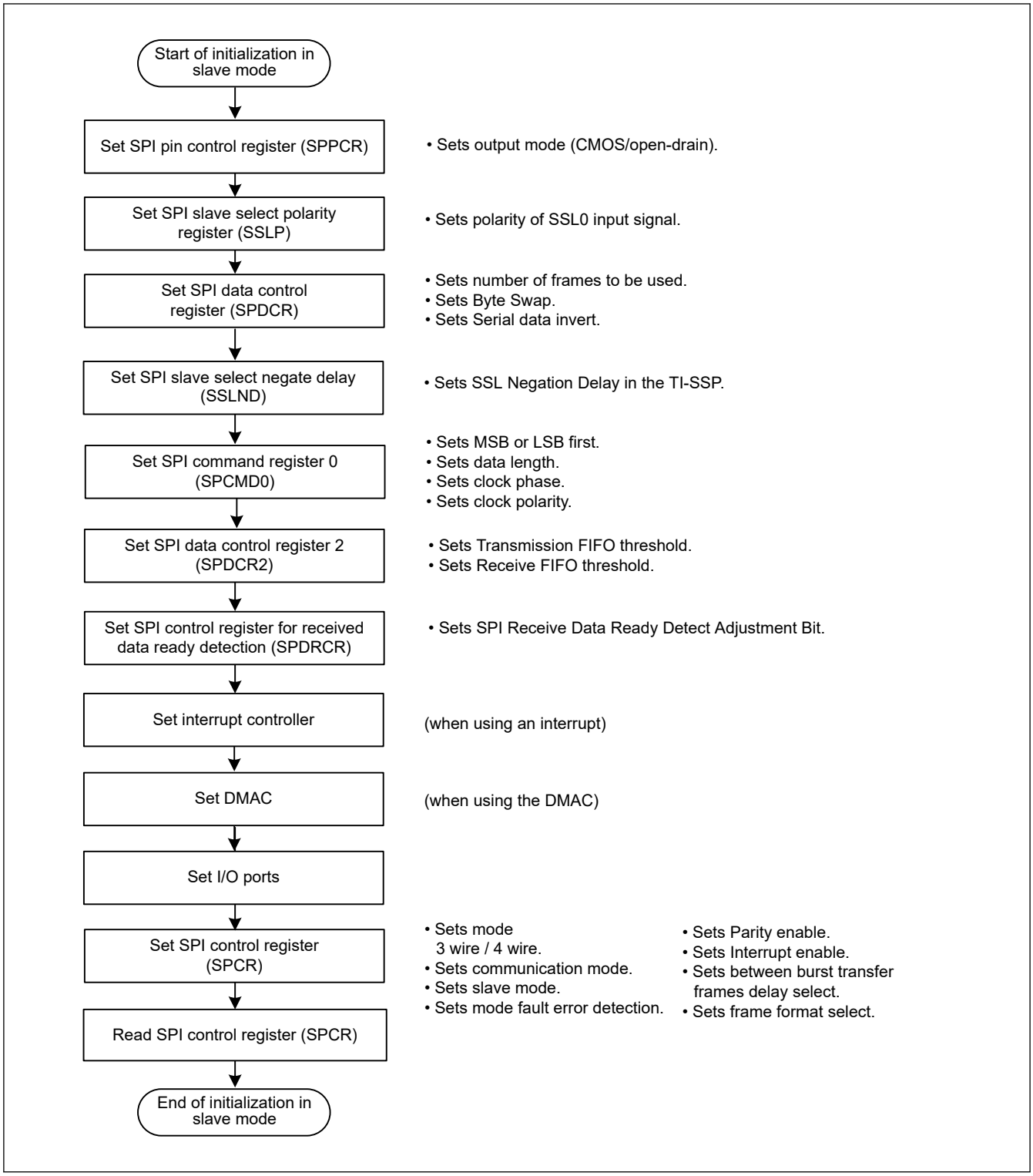


Figure 36.69 Example of initialization flow in slave mode

(6) Software Processing Flow

Figure 36.70 to Figure 36.73 show examples of the flow of software processing.

(a) Transmit Processing Flow

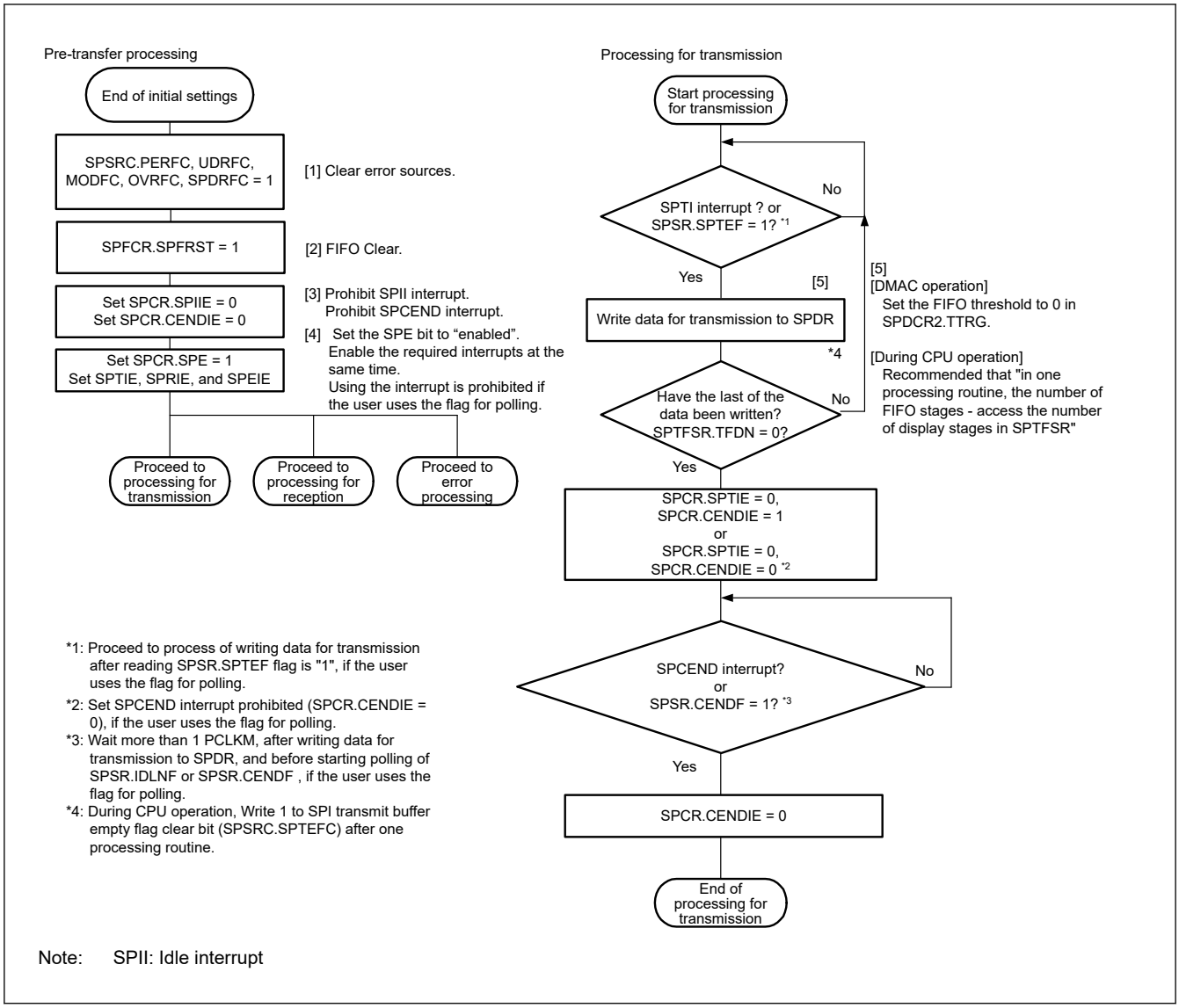


Figure 36.70 Software processing flowchart in slave mode (transmission)

(b) Reception Processing Flow

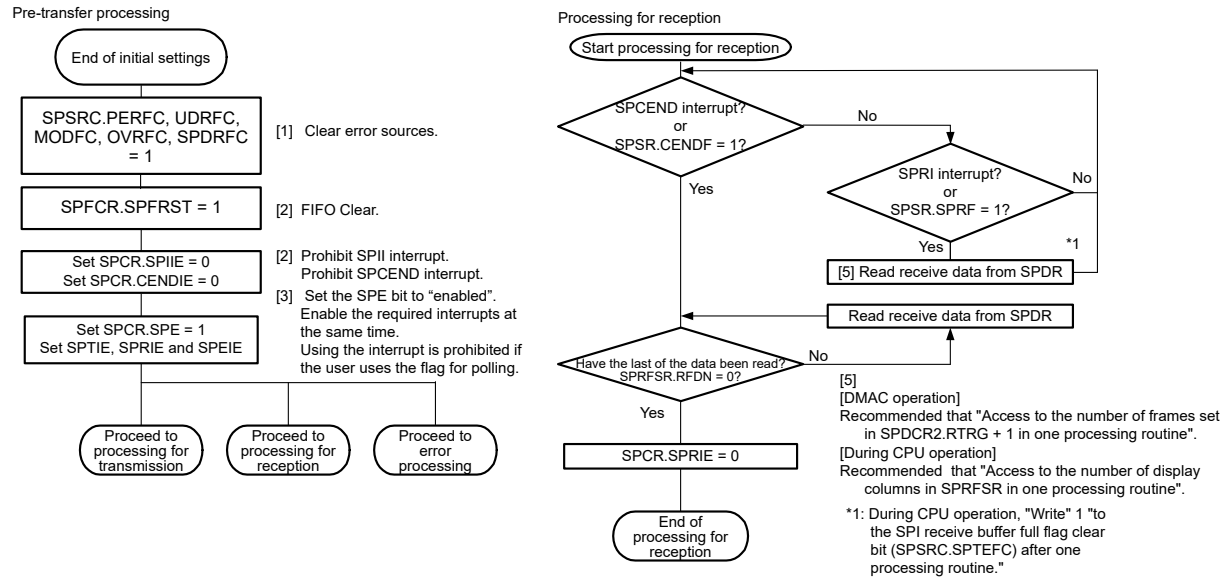


Figure 36.71 Flowchart in slave mode (reception)

(c) Master Reception-only processing Flow

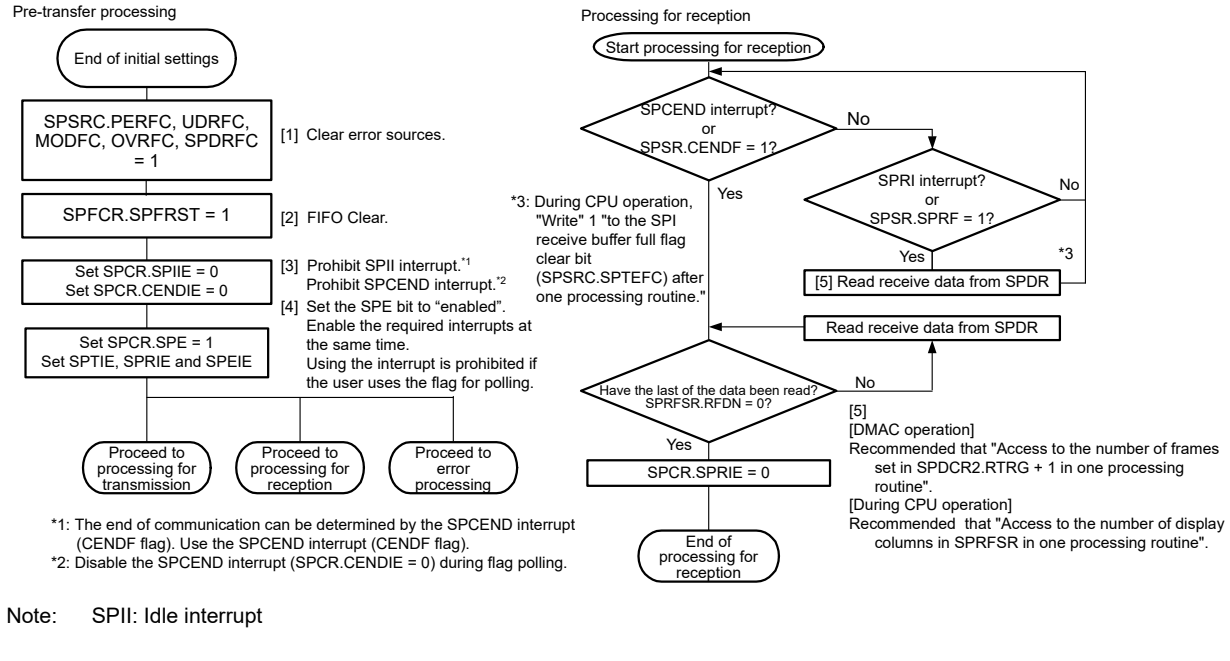
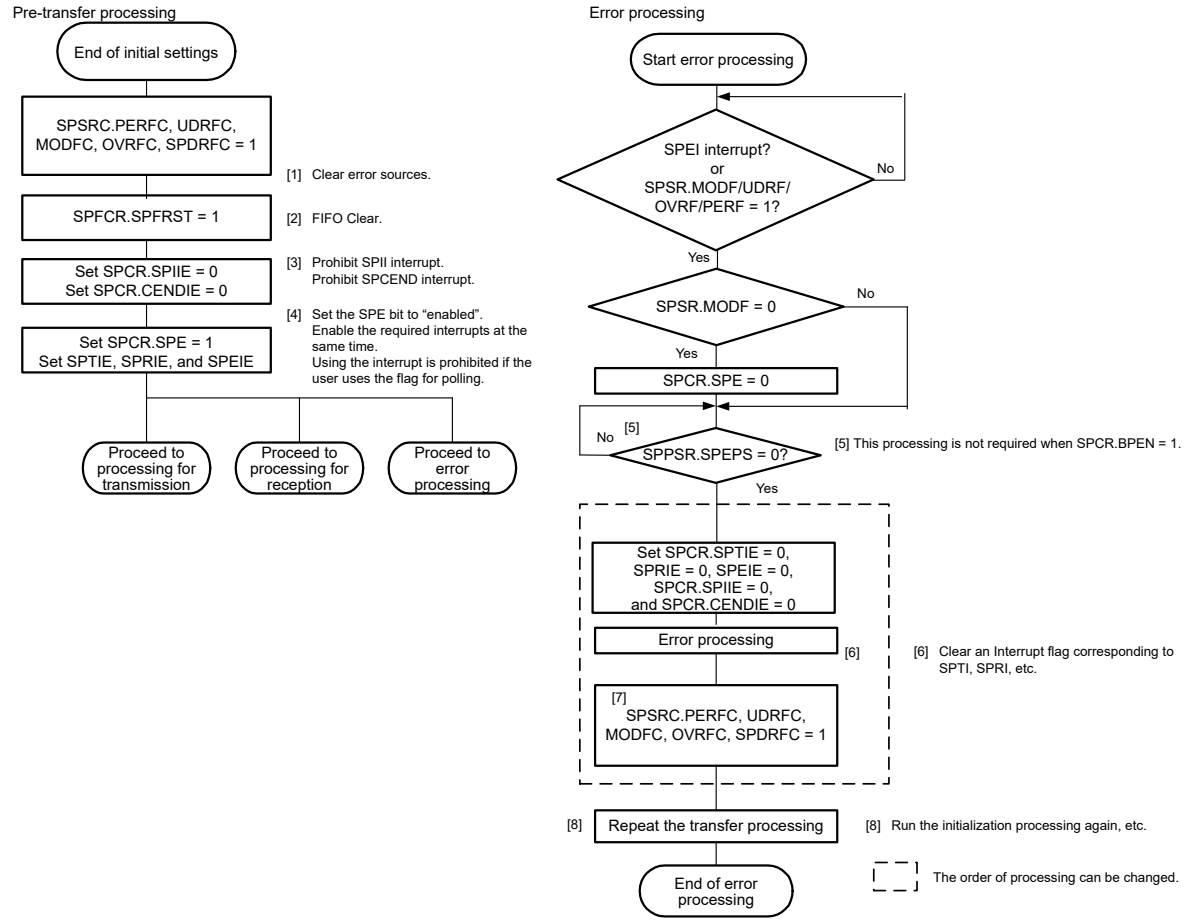


Figure 36.72 Software processing flowchart in slave mode (reception-only)

(d) Error processing flow

In slave mode operation, the MODF flag can be cleared regardless of the SSL0 pin status even when a mode fault error is present.

When interrupts are used and an error has occurred, clear an Interrupt flag in the error processing because an SPTI interrupt request or an SPRI interrupt request may be remaining in an Interrupt flag. Furthermore, when an SPRI interrupt request is remaining, read the receive buffer to initialize the SPI's internal sequencer.



Note: SPEI: SPI error interrupt
SPII: Idle interrupt
SPRI: Reception buffer full interrupt
SPTI: Transmission buffer empty interrupt

Figure 36.73 Software processing flowchart in slave mode (error processing)

36.4.14 Clock Synchronous Operation

When SPCR.SPMS bit = 1, the SPI operates in synchronization with clock. Clock synchronous operation enables communication by using three pins RSPCK, MOSI, and MISO without using SSL pins. SSL pins can be used as I/O ports.

Clock synchronous operation does not use SSL pins for communication, but the module operates internally in the similar manner to SPI operation. In master mode operation and slave mode operation, communication is possible according to the flow similar to SPI operation, but no mode fault error is detected because SSL pins are not used.

In clock synchronous operation, if the SPCMD.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0), subsequent operation is not guaranteed.

36.4.14.1 Master Mode Operation

(1) Starting a Serial Transfer

When data is written to the SPDR register while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn, n = 0 to 3) data in SPDR. While the shift register is empty, the SPI copies transmit buffer data to the shift register to start serial transfer. After the SPI copies transmit data to the shift register, it changes the shift register status to full. Upon completion of serial transfer, the SPI changes the shift register status to empty. The shift register status cannot be monitored.

For details about the SPI transfer format, see section 36.4.5. Transfer Format (Frame Format). In clock synchronous operation, however, the SSL0 output signal is not used for communication.

(2) Terminating a Serial Transfer

Upon detecting the RSPCK edge corresponding to the final sampling timing, the SPI terminates serial transfer. When the number of data stored in the receive FIFO < the number of FIFO stages, the SPI copies received data from the shift register to the receive buffer in the SPDR register after serial transfer.

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in master mode depends on the set value of the SPCMD.SPB[4:0] bits. For details about the SPI transfer format, see section 36.4.5. Transfer Format (Frame Format). In clock synchronous operation, however, the SSL0 output signal is not used for communication.

(3) Sequence Control

The transfer format in master mode is determined by the SPSCR, SPCMDm (m = 0 to 7), SPBR, SPCKD, SSLND, and SPND registers. These settings are valid though SSL signals are not output in clock synchronous operation.

The SPSCR register is used to determine the sequence configuration for serial transfer to be performed by the SPI in master mode. MSB first/LSB first, data length, a part of bit rate settings, RSPCK polarity and phase, SPCKD enable/disable, SSLND enable/disable, and SPND enable/disable are set in SPCMD0 to SPCMD7. A part of bit rate settings is set in SPBR, the SPI clock delay value is set in SPCKD, the SSL negation delay value is set in SSLND, and the access delay value is set in SPND.

The SPI configures the sequence that structures a part of or whole of SPCMD0 to SPCMD7 according to the sequence length specified in SPSCR. The SPI has a pointer to SPCMD that configures the sequence. This pointer value can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 to enable the SPI function, the SPI sets the pointer to commands in SPCMD0 and applies the SPCMD0 setting to the transfer format at the start of serial transfer. The SPI increments the pointer when the next-access delay period ends in each data transfer. When the serial transfer corresponding to the final command that configures the sequence is completed, the SPI sets the pointer in SPCMD0. Thus sequence is repeatedly executed.

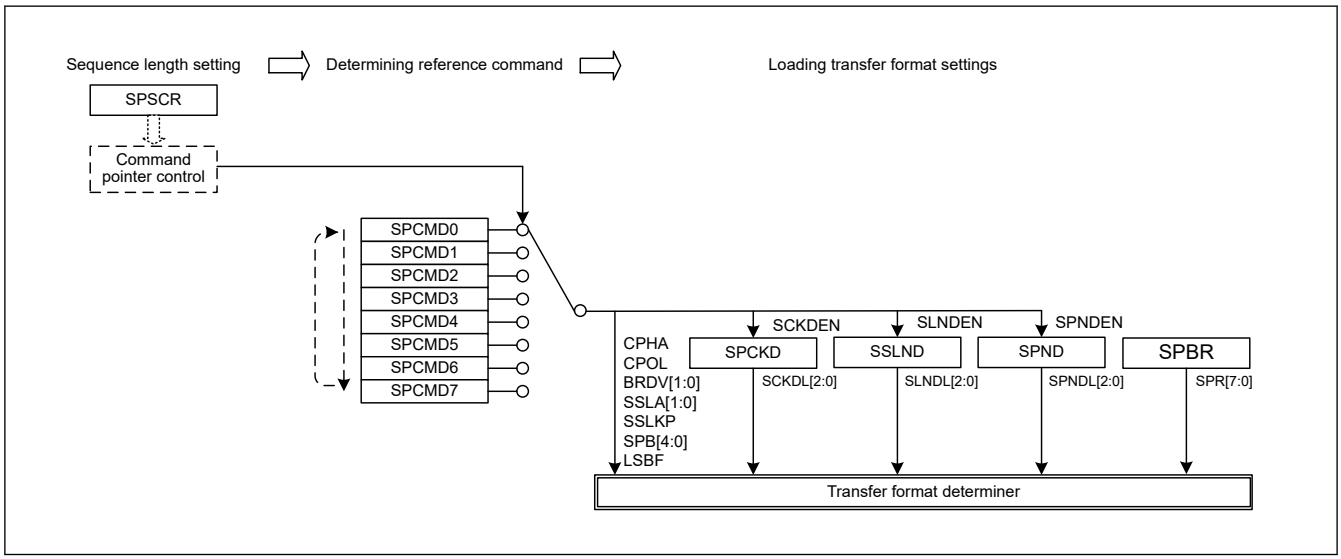


Figure 36.74 Determining serial transfer method in master mode

In this section, a frame consists of data (SPDR) and configuration (SPCMDm (m = 0 to 7)).

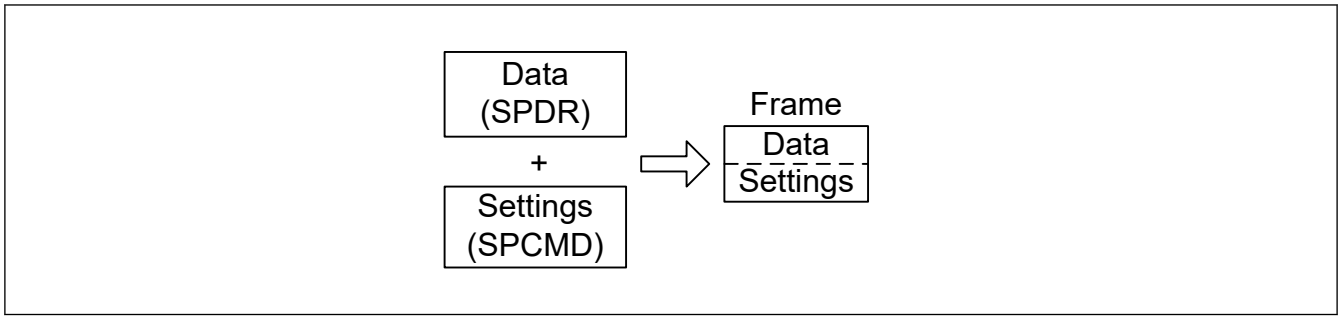


Figure 36.75 Conceptual diagram of frame

Figure 36.76 shows the correspondence between the command in the sequence operation performed and the transmit buffer/ receive buffer.

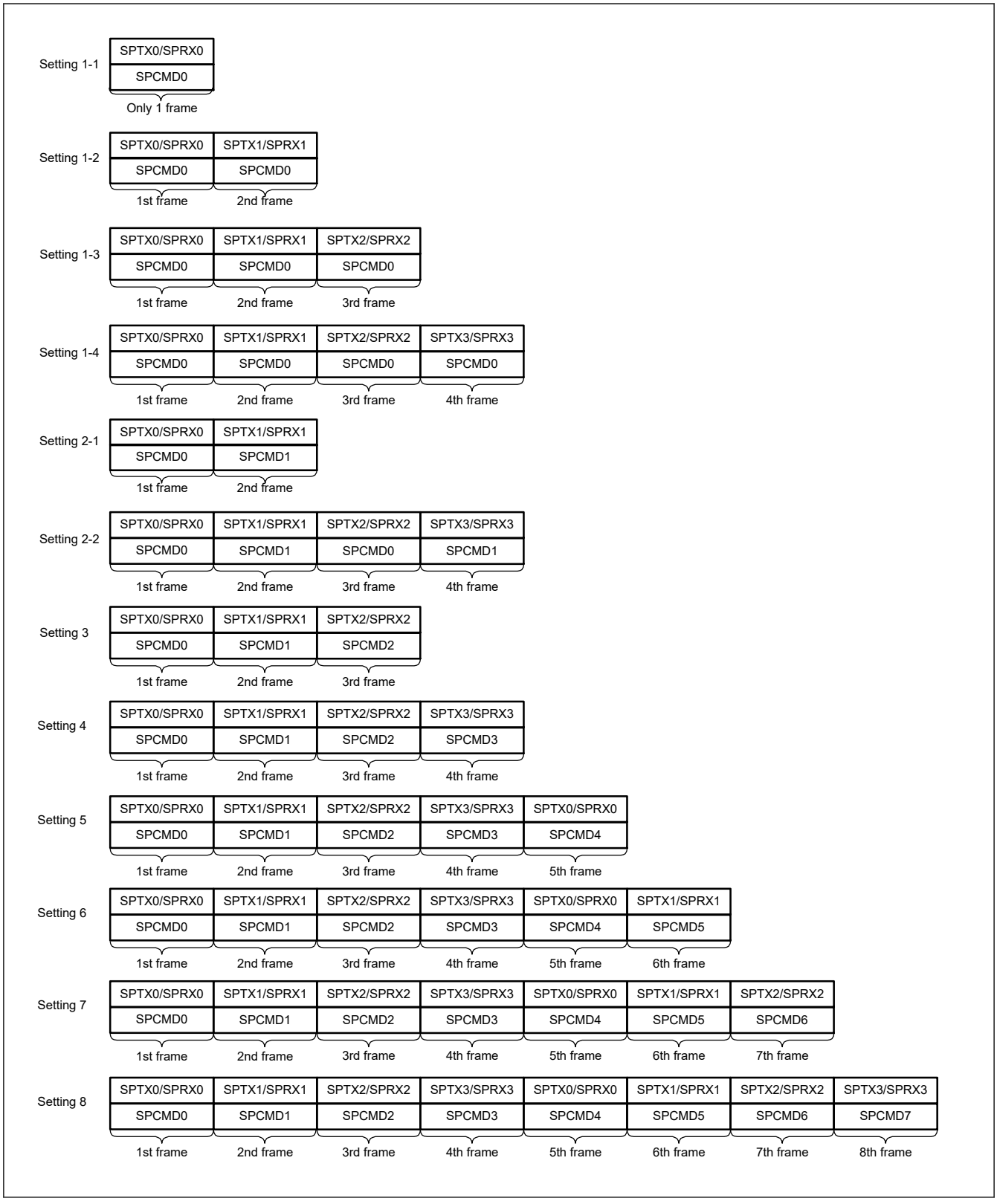


Figure 36.76 Correspondence between SPI command register and transmit/receive buffers in sequence operation

(4) Initialization Flow

Figure 36.77 shows an example of initialization flow when using the SPI in master-mode clock synchronous operation. For how to set the interrupt controller, DMAC, and input/output ports, see descriptions of each block.

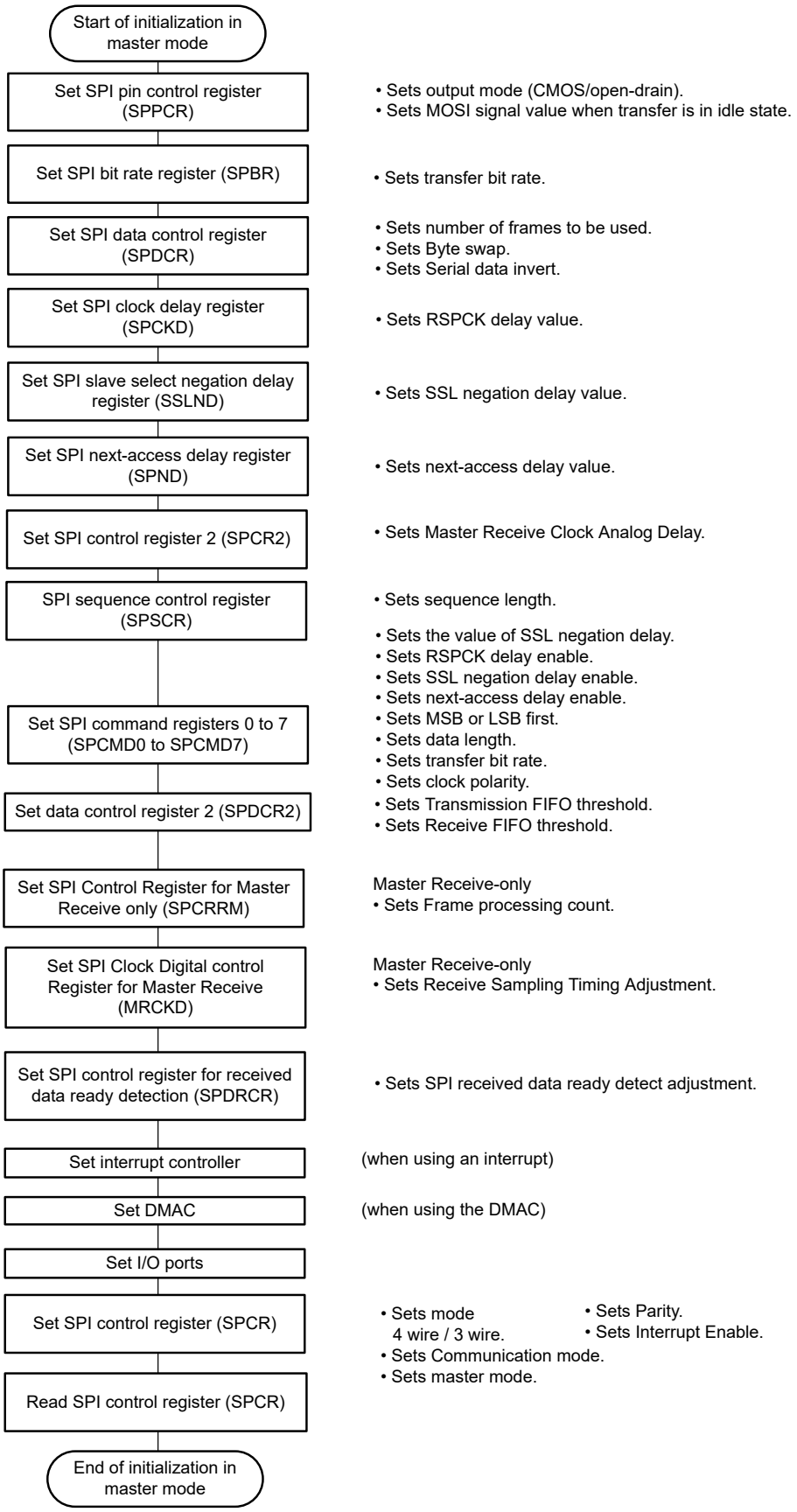


Figure 36.77 Example of initialization flowchart in master mode (clock synchronous operation)

(5) Flow of Software Processing

The software processing flow is omitted here because it is similar to the flow for SPI operation. However, no mode fault error occurs.

36.4.14.2 Slave Mode Operation

(1) Starting a Serial Transfer

When SPCR.SPMS = 1, the first RSPCK edge becomes a trigger to start serial transfer of the SPI.

While SPCR.SPMS = 1, the SPI always drives the MISO output signal.

For details about the SPI transfer format, see section 36.4.5. Transfer Format (Frame Format). In clock synchronous operation, however, the SSL0 input signal is not used.

(2) Terminating a Serial Transfer

Upon detecting the RSPCK edge corresponding to the final sampling timing, the SPI terminates serial transfer. When the number of data stored in the receive FIFO < the number of FIFO stages, the SPI copies received data from the shift register to the receive buffer in the SPDR register after serial transfer. The SPI also changes the shift register status to empty after serial transfer regardless of the receive buffer status. The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in slave mode depends on the set value of the SPCMD0.SPB[4:0] bits. For details about the SPI transfer format, see section 36.4.5. Transfer Format (Frame Format).

(3) Initialization Flowchart

Figure 36.78 shows an example of initialization flow when using the SPI in slave-mode clock synchronous operation.

For how to set the interrupt controller, DMAC, and input/output ports, see descriptions of each block.

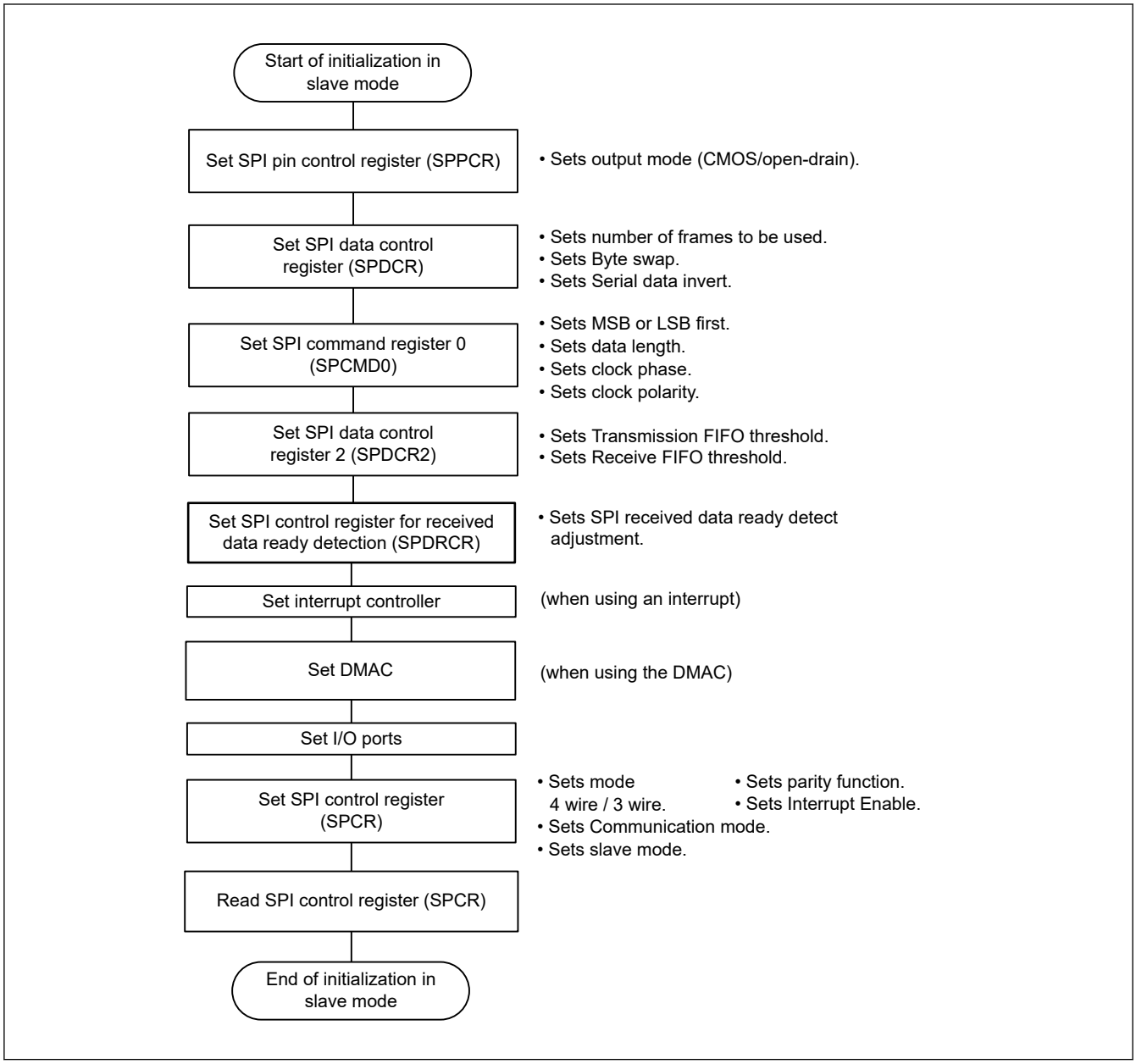


Figure 36.78 Example of initialization flow in slave mode

(4) Flow of Software Processing

The software processing flow is omitted here because it is similar to the flow for SPI operation. However, no mode fault error occurs.

36.4.15 Loopback Mode

When 1 is written to the SPPCR.SPLP bit or SPPCR.SPLP2 bit, the SPI disconnects the path between MISO pin and shift register (when SPCR.MSTR = 1) or between MOSI pin and shift register (when SPCR.MSTR = 0), and then connects the shift register input path to the output path. When SPCR.MSTR = 1, the SPI does not disconnect the path between MOSI pin and shift register. When SPCR.MSTR = 0, the SPI does not disconnect the path between MISO pin and shift register.

When serial transfer is performed in loopback mode, transmit data or inverted transmit data of the SPI becomes received data of the SPI.

Figure 36.79 shows the relationship between settings of the SPLP2 and SPLP bits and received data.

Table 36.16 SPLP2 and SPLP bit settings and received data

SPPCR.SPLP2 bit	SPPCR.SPLP bit	Received data
0	0	Input data from the MOSI pin or MISO pin
0	1	Inversed transmit data
1	0	Transmit data
1	1	Transmit data

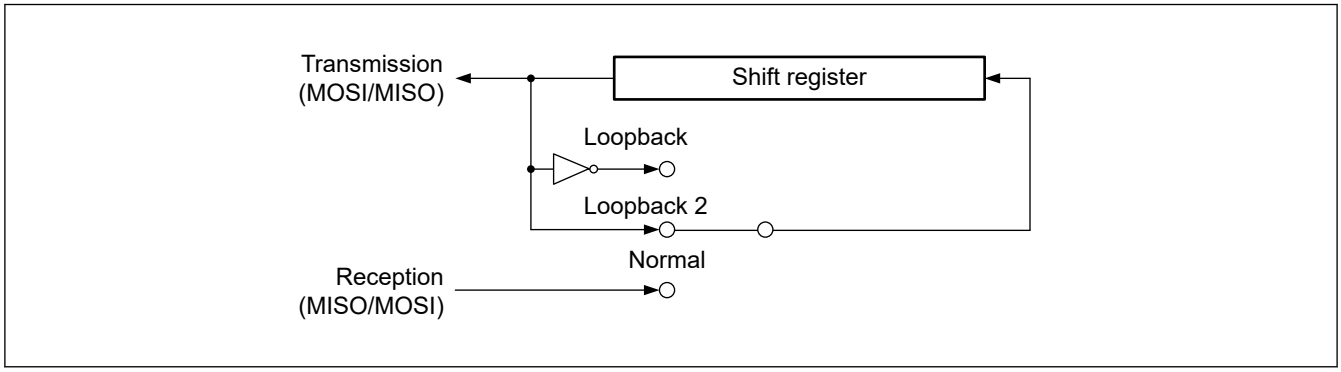


Figure 36.79 Configuration of shift register I/O paths in loopback mode

36.4.16 Self-Diagnosis of Parity Function

The parity circuit consists of a section to add a parity bit to transmit data and a section to detect an error in receive data.

To detect a failure of these sections of the parity circuit, self-diagnosis of the parity circuit is performed according to the flow in Figure 36.80.

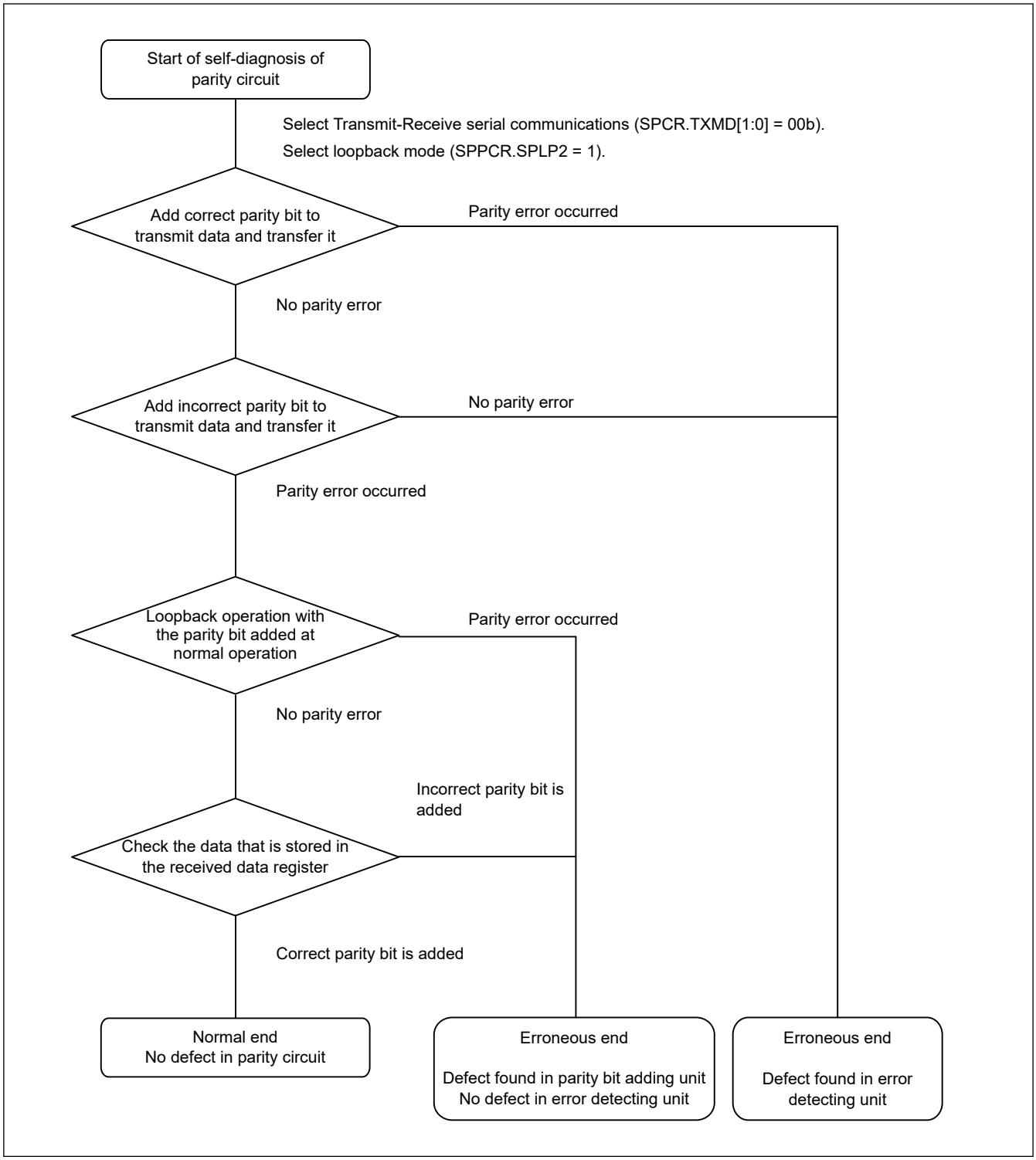


Figure 36.80 Parity circuit self-diagnosis flow

36.4.17 Interrupt Requests

The SPI has the following interrupt sources: receive buffer full, transmit buffer empty, mode fault, underrun, overrun, parity error, SPI idle, communication end, and received data ready. Furthermore, the DMAC can be activated by a receive buffer full interrupt request and a transmit buffer empty interrupt request to start data transfer.

Mode fault, underrun, overrun, parity error, and received data ready (SPCR.SPDRES = 1) interrupt requests are allocated to vector addresses in SPEI. Therefore, identify these interrupt sources by reading respective flags. Table 36.17 lists SPI's interrupt sources. When each interrupt condition in Table 36.17 is met, an interrupt is generated. Clear the receive buffer full and transmit buffer empty interrupt sources by performing data transfer.

When starting transmission or reception by using the DMAC, configure the DMAC to enable it, and then configure the SPI.

Even when the transmit buffer empty interrupt or receive buffer full interrupt condition is met while an Interrupt flag is 1, an interrupt request is not output to ICU but is internally retained. (Only one request per source can be internally retained.) When an Interrupt flag is cleared to 0, the retained interrupt request is output to ICU and then automatically cleared. An internally retained interrupt request can also be cleared by setting the corresponding interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) to 0.

Table 36.17 Interrupt sources of SPI

Interrupt source	Symbol	Interrupt condition	DMAC activation
Reception buffer full	SPRI	SPRIE = 1 and (receive buffer full (SPRF = 1) | (SPDRES = 0 and Receive data ready (SPDRF = 1)))	Activated
Transmission buffer empty	SPTI	SPTIE = 1 and transmit buffer empty (SPTEF = 1)	Activated
SPI errors (mode fault, underrun, overrun, and parity error, received data ready)	SPEI	SPEIE = 1 and (MODF = 1 | UDRF = 1 | OVRF = 1 | PERF = 1 | (SPDRES = 1 and SPDRF = 1))	—
SPI idle	SPII	SPIIE = 1 and IDLNF = 0	—
Communication end*1	SPCEND	CENDIE = 1 and CENDF = 1	—

Note 1. See section 36.4.9.7. Common Operation for details on the communication completion interrupt output conditions.

36.5 Usage Notes

36.5.1 Notes on Communication Start

When an Interrupt flag is 1 at the beginning of communication, perform the following procedure to clear an interrupt request before enabling operation (setting the SPCR.SPE bit to 1).

1. Confirm that communication is not going (SPCR.SPE = 0).
2. Set the corresponding interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) to 0.
3. Read the corresponding interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) and confirm that it is 0.
4. Set the Interrupt flag to 0.

36.5.2 Notes on the Module-Stop Function

To stop this module using the power consumption reducing function, be sure to set the SPE bit in SPCR to 0 to terminate communication and then use the power consumption reducing function.

36.5.3 Notes on SPRF and SPTEF Flags

When using the SPRF and SPTEF flags in SPSR by polling them, set the SPRIE and SPTIE bits in SPCR to 0.

36.5.4 Notes on Receive Sampling Function

When using the receive sampling function (sampling receive data with MRIOCLK or MRCLK), set the SSLND register considering the delay of the receive data sampling clock (MRIOCLK or MRCLK) from RSPCK. Please see section 36.3.3. SSLND : SPI Slave Select Negation Delay Register.

36.5.5 Notes on Burst Transfer in Master Mode

During burst transfer, changing the following settings between SPCMD with the SSLKP bit set to 1 and SPCMD used in the next transfer is prohibited.

1. SSL Signal Assertion setting (SSLA[1:0] bits)
2. RSPCK output setting (CPHA, CPOL, BRDV[1:0] bits)

36.5.6 Notes on TI-SSP in Slave Mode

In the slave TI SSP mode, the delay between frames must observe the following intervals.

Secure the following intervals on the master side.

[Master side: interval from RSPCK edge of last bit to RSPCK edge of next SSL] > [Slave side: Output enable delay time = PCLKSPIn × (1 to 2) + SLNDL[2:0] setting value]

36.5.7 Notes at SPE = 1

(1) When SPCR.SPE = 1, if the following register bits are rewritten, subsequent operations are not guaranteed.

Table 36.18 Communication end event generating conditions (receive only slave mode) (1 of 2)

Register name	Bit name
SPCKD	SCKDL[2:0]
SSLND	SLNDL[2:0]
SPND	SPNDL[2:0]
MRCKD	ARST[2:0]
SPCR	BPEN
	MSTR
	TXMD[1:0]
	SPFRF
	SPMS
	MODFEN
	BFDS
	SCKASE
	PTE
	SPOE
	SPPE
	SPSCKSEL
SPCRRM	RMFM[4:0]
SPDRCR	SPDRC[7:0]
SPPCR	MOIFE
	MOIFV
	SPOM
	SPLP2
	SPLP
SPCR2	SPSCKDL[2:0]
SSLP	SSL3P
	SSL2P
	SSL1P
	SSL0P
SPBR	SPR[7:0]
SPSCR	SPSLN[2:0]

Table 36.18 Communication end event generating conditions (receive only slave mode) (2 of 2)

Register name	Bit name
SPCMD0*1	SSLA[1:0]
	SPB[4:0]
	SCKDEN
	SLNDEN
	SPNDEN
	LSBF
	SSLKP
	BRDV[1:0]
	CPOL
	CPHA
SPDCR	SPFC[1:0]
	SINV
	SPRDTD
	SLSEL[1:0]
	BYSW
SPDCR2	TTRG[1:0]
	RTRG[1:0]
SPFCR	SPFRST

Note 1. Rewriting prohibited in slave mode. In master mode, this is possible only when there is no next transfer data in the transmit FIFO.

(2) After rewriting from SPE = 1 to 0 when SPCR.BPEN = 0, or when SPSR.MODF is set to 1, check that the SPPSR.SPEPS bit is set to 0, then do the next operation.

37. Expanded Serial Peripheral Interface (xSPI)

37.1 Overview

This LSI has two units of Expanded Serial Peripheral Interface (xSPI). The xSPI protocol specifies the interface for Memory Devices, which provides high data throughput, low signal count, and limited backward compatibility with legacy SPI devices. The electrical interface can deliver up to 266 MB per second raw data throughput.

Table 37.1 lists the xSPI specifications, Figure 37.1 shows a block diagram, and Table 37.2 lists the I/O pins.

Table 37.1 xSPI Specifications

Item	Description
Number of channels	2 units
Protocol	Compliant for the xSPI protocol
Data transmission and reception	Issue the transaction for up to 2 Slave as Master
Transfer speed	Support the transfer at xSPI266
Mode	• Support Protocol modes below – 1/4/8pin with SDR/DDR (1S-1S-1S, 4S-4D-4D, 8D-8D-8D) – 2/4pin with SDR (1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S) • Configurable address length • Configurable initial access latency cycle • Support XiP mode
xSPI function	• Support Write Data Mask • Support In-band Reset • Memory-mapping – Support up to 256 MB address space (Sum of CS0 and CS1 is up to 256 MB.) – Prefetch function for burst-read with low latency – Outstanding buffer for burst-write with high throughput • Manual command – Configurable up to 4 commands – Status Register Polling function • Input Strobe port timing shift
Interrupt source	2 interrupts
Module-stop function	Module-stop state can be set to reduce power consumption

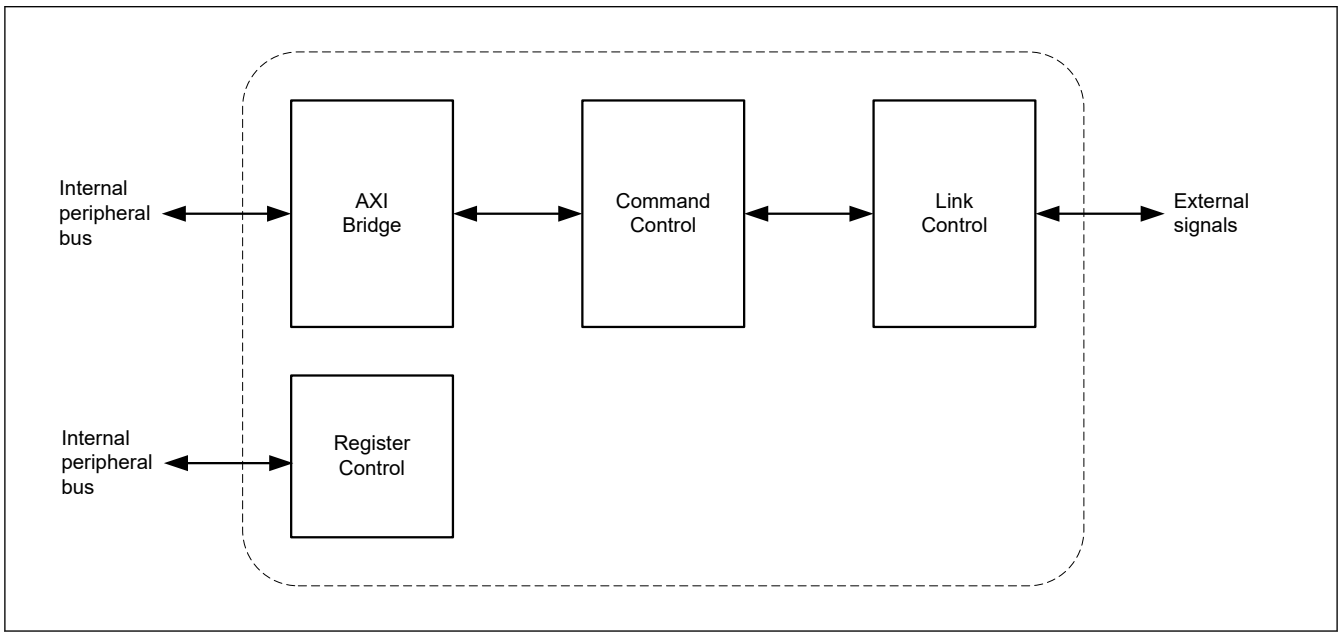


Figure 37.1 Block diagram

Table 37.2 xSPI I/O pins

Channel	Pin name	I/O	Function
xSPIm (m = 0, 1)	XSPIm_CKP	Output	Clock Positive
	XSPIm_CKN*1	Output	Clock Negative
	XSPIm_CS0#	Output	Chip Select for slave0
	XSPIm_CS1#	Output	Chip Select for slave1
	XSPIm_DS	I/O	Read Data Strobe / Write Data Mask
	XSPIm_IO0	I/O	Data 0 input/output
	XSPIm_IO1	I/O	Data 1 input/output
	XSPIm_IO2	I/O	Data 2 input/output
	XSPIm_IO3	I/O	Data 3 input/output
	XSPIm_IO4	I/O	Data 4 input/output
	XSPIm_IO5	I/O	Data 5 input/output
	XSPIm_IO6	I/O	Data 6 input/output
	XSPIm_IO7	I/O	Data 7 input/output
	XSPIm_RESET0#*1	Output	Master reset status for slave0
	XSPIm_RESET1#*1	Output	Master reset status for slave1
	XSPIm_RSTO0#*1	Input	Slave reset status for slave0
	XSPIm_RSTO1#*1	Input	Slave reset status for slave1
	XSPIm_INT0#*1	Input	Interrupt for slave0
	XSPIm_INT1#*1	Input	Interrupt for slave1
	XSPIm_ECS0#*1	Input	Error Correction Status for slave0
	XSPIm_ECS1#*1	Input	Error Correction Status for slave1
	XSPIm_WP0#*1	Output	Write Protect for slave0
	XSPIm_WP1#*1	Output	Write Protect for slave1

Note 1. These pins are available in unit 0 only.

Table 37.3 Address map

Channel	Internal address	Space
xSPI0	0x4000_0000 to 0x4FFF_FFFF	CS0 + CS1
xSPI1	0x5000_0000 to 0x5FFF_FFFF	CS0 + CS1

37.2 Register Map

Table 37.4 xSPI register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x801C_0000 + 0x1000 × m (m = 0, 1)	WRAPCFG	xSPI Wrapper Configuration Register	—
0x801C_0004 + 0x1000 × m (m = 0, 1)	COMCFG	xSPI Common Configuration Register	—
0x801C_0008 + 0x1000 × m (m = 0, 1)	BMCFG	xSPI Bridge Map Configuration Register	—
0x801C_0010 + 0x1000 × m + 0x010 × n (m = 0, 1)	CMCFG0CSn	xSPI Command Map Configuration Register 0 CSn (n = 0, 1)	—
0x801C_0014 + 0x1000 × m + 0x010 × n (m = 0, 1)	CMCFG1CSn	xSPI Command Map Configuration Register 1 CSn (n = 0, 1)	—
0x801C_0018 + 0x1000 × m + 0x010 × n (m = 0, 1)	CMCFG2CSn	xSPI Command Map Configuration Register 2 CSn (n = 0, 1)	—

Table 37.4 xSPI register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x801C_0050 + 0x1000 × m + 0x004 × n (m = 0, 1)	LIOCFGCSn	xSPI Link I/O Configuration Register CSn (n = 0, 1)	—
0x801C_0060 + 0x1000 × m (m = 0, 1)	BMCTL0	xSPI Bridge Map Control Register 0	—
0x801C_0064 + 0x1000 × m (m = 0, 1)	BMCTL1	xSPI Bridge Map Control Register 1	—
0x801C_0068 + 0x1000 × m (m = 0, 1)	CMCTL	xSPI Command Map Control Register	—
0x801C_0070 + 0x1000 × m (m = 0, 1)	CDCTL0	xSPI Command Manual Control Register 0	—
0x801C_0074 + 0x1000 × m (m = 0, 1)	CDCTL1	xSPI Command Manual Control Register 1	—
0x801C_0078 + 0x1000 × m (m = 0, 1)	CDCTL2	xSPI Command Manual Control Register 2	—
0x801C_0080 + 0x1000 × m + 0x010 × n (m = 0, 1)	CDTBUFn	xSPI Command Manual Type Buf n (n = 0 to 3)	—
0x801C_0084 + 0x1000 × m + 0x010 × n (m = 0, 1)	CDABUFn	xSPI Command Manual Address Buf n (n = 0 to 3)	—
0x801C_0088 + 0x1000 × m + 0x010 × n (m = 0, 1)	CDD0BUFn	xSPI Command Manual Data 0 Buf n (n = 0 to 3)	—
0x801C_008C + 0x1000 × m + 0x010 × n (m = 0, 1)	CDD1BUFn	xSPI Command Manual Data 1 Buf n (n = 0 to 3)	—
0x801C_0100 + 0x1000 × m (m = 0, 1)	LPCTL0	xSPI Link Pattern Control Register 0	—
0x801C_0104 + 0x1000 × m (m = 0, 1)	LPCTL1	xSPI Link Pattern Control Register 1	—
0x801C_0108 + 0x1000 × m (m = 0, 1)	LIOCTL	xSPI Link I/O Control Register	—
0x801C_0130 + 0x1000 × m + 0x020 × n (m = 0, 1)	CCCTL0CSn	xSPI Command Calibration Control Register 0 CSn (n = 0, 1)	—
0x801C_0134 + 0x1000 × m + 0x020 × n (m = 0, 1)	CCCTL1CSn	xSPI Command Calibration Control Register 1 CSn (n = 0, 1)	—
0x801C_0138 + 0x1000 × m + 0x020 × n (m = 0, 1)	CCCTL2CSn	xSPI Command Calibration Control Register 2 CSn (n = 0, 1)	—
0x801C_013C + 0x1000 × m + 0x020 × n (m = 0, 1)	CCCTL3CSn	xSPI Command Calibration Control Register 3 CSn (n = 0, 1)	—
0x801C_0140 + 0x1000 × m + 0x020 × n (m = 0, 1)	CCCTL4CSn	xSPI Command Calibration Control Register 4 CSn (n = 0, 1)	—
0x801C_0144 + 0x1000 × m + 0x020 × n (m = 0, 1)	CCCTL5CSn	xSPI Command Calibration Control Register 5 CSn (n = 0, 1)	—
0x801C_0148 + 0x1000 × m + 0x020 × n (m = 0, 1)	CCCTL6CSn	xSPI Command Calibration Control Register 6 CSn (n = 0, 1)	—
0x801C_014C + 0x1000 × m + 0x020 × n (m = 0, 1)	CCCTL7CSn	xSPI Command Calibration Control Register 7 CSn (n = 0, 1)	—
0x801C_0180 + 0x1000 × m (m = 0, 1)	VERSTT	xSPI Version Register	—
0x801C_0184 + 0x1000 × m (m = 0, 1)	COMSTT	xSPI Common Status Register	—
0x801C_0188 + 0x1000 × m + 0x004 × n (m = 0, 1)	CASTTCSn	xSPI Calibration Status Register CSn (n = 0, 1)	—
0x801C_0190 + 0x1000 × m (m = 0, 1)	INTS	xSPI Interrupt Status Register	—
0x801C_0194 + 0x1000 × m (m = 0, 1)	INTC	xSPI Interrupt Clear Register	—
0x801C_0198 + 0x1000 × m (m = 0, 1)	INTE	xSPI Interrupt Enable Register	—
0x8029_3000 + 0x100 × m (m = 0, 1)	IOVOLCTL	xSPI IO Voltage Control Register	PRCRN.PRC3
0x8029_3004 + 0x100 × m (m = 0, 1)	CS0ENDAD	xSPI CS0 End Address Register	PRCRN.PRC3
0x8029_3008 + 0x100 × m (m = 0, 1)	CS1STRAD	xSPI CS1 Start Address Register	PRCRN.PRC3
0x8029_300C + 0x100 × m (m = 0, 1)	CS1ENDAD	xSPI CS1 End Address Register	PRCRN.PRC3

Table 37.5 xSPI related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	MRCTLA.MRCTLA04	MSTPCRA.MSTPCRA04	SLVACCCTL8.xSPI0_SL
1	MRCTLA.MRCTLA05	MSTPCRA.MSTPCRA05	SLVACCCTL8.xSPI1_SL

Note: Access to memory area is protected by TZC-400. Its slave access control register is SLVACCCTL8.TZCXSPI_SL.

37.3 Register Descriptions

37.3.1 xSPI Configuration Registers

These registers configure xSPI Master function. These registers should be configured in the initialization phase before issuing xSPI transaction.

37.3.1.1 WRAPCFG : xSPI Wrapper Configuration Register

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DSSFTCS1[4:0]					—	—	—	—	—	—	—	—	—	—	—	DSSFTCS0[4:0]					—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DSSFTCS0[4:0]	DS shift for slave0 These bits configure the number of delay cell for XSPIm_DS pin. It is used to adjust the DS sampling timing. 1 cell delay is around 125 ps. When enabled automatic calibration, it can be updated automatically. 0x00: No shift 0x01: Add a delay of 1 cell ⋮ 0x1E: Add a delay of 30 cells 0x1F: Add a delay of 31 cells	R/W
23:13	—	These bits are read as 0. The write value should be 0.	R/W
28:24	DSSFTCS1[4:0]	DS shift for slave1 The function is same as one of slave0.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to configure xSPI Master function.

37.3.1.2 COMCFG : xSPI Common Configuration Register

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OENEGEX	OEASTEX
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	OEASTEX	Output Enable Asserting extension This bit extends 1 cycle output enable of Data and DS during output enable asserting. When set to 1, CS asserting should be extended (LIOCFGCSn.CSASTEX = 1 (n = 0, 1)). This bit shall not be used in case of no latency cycle. Because xSPI output data could be conflicted with xSPI input data. 0: No extend 1 cycle Output enable 1: Extend 1 cycle Output enable	R/W
17	OENEGEX	Output Enable Negating extension This bit extends 1 cycle output enable of Data and DS during output enable negating. This bit should not be used in case of no latency cycle. Because SPI output data could be conflicted with SPI input data. 0: No extend 1 cycle Output enable 1: Extend 1 cycle Output enable	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to configure xSPI Master function.

37.3.1.3 BMCFG : xSPI Bridge Map Configuration Register

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CMBTIM[7:0]								—	—	—	—	—	—	—	PREEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MWRSIZE[7:0]								MWRCOMB	—	—	—	—	—	—	WRMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WRMD	AXI Write Response mode This bit selects the timing of system bus write response in memory-mapping mode. When set to 1, it returns the response after transmitting a frame on xSPI bus. When enabled this mode, Memory Write Combination mode must be disabled. 0: Return response after storing to Internal Write Buffer 1: Return response after issuing write transaction to xSPI bus	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	MWRCOMB	Memory Write Combination mode This bit selects to combine the xSPI data in write access of memory-mapping mode. When set to 0, xSPI data size depends on AXI Burst type. When set to 1, it depends on Memory Write Size bit. When set to 1, any write transaction could be held in this xSPI master temporarily. 0: Disable combination mode 1: Enable combination mode	R/W
15:8	MWRSIZE[7:0]	Memory Write Size These bits select the size to combine incremental address in memory-mapping mode. It transmits an xSPI frame with the data combined up to the configured size while the address is incremental. When detected non-incremental address or a read transaction before reaching to the target size, it transmits the pending data into xSPI bus. 0x00: Combine incremental address up to 4 bytes 0x01: Combine incremental address up to 8 bytes ⋮ 0x0E: Combine incremental address up to 60 bytes 0x0F: Combine incremental address up to 64 bytes Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
16	PREEN	Prefetch enable This bit enables prefetch function for read transaction in memory-mapping mode. It could reduce the latency for read transaction with incremental address. 0: Disable prefetch function 1: Enable prefetch function	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
31:24	CMBTIM[7:0]	Combination timer This field specifies expiration period of combination timer. 0x00 means disabling the combination timer. When the timer is expired, the data in the combination buffer is pushed to memory device.	R/W

This register has functions to configure xSPI Master function.

37.3.1.4 CMCFG0CSn : xSPI Command Map Configuration Register 0 CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x010 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADDRPCD[7:0]								ADDRPEN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	ARYAMD	WPBSTMD	ADDSIZE[1:0]		FFMT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	FFMT[1:0]	Frame format These bits configure xSPI frame format in memory-mapping mode. Please see Table 37.11 for detail. 0 0: Normal format: Command 1 byte, Address ADDSIZE, Data to AXI bus transaction. 0 1: 8D-8D-8D profile 1.0 format: Command 2 bytes, Address ADDSIZE, Data to AXI bus transaction 1 0: 8D-8D-8D profile 2.0 Command Modifier format: Command & Modifier 6 bytes, Data to AXI bus transaction 1 1: 8D-8D-8D profile 2.0 Commands with Extended Command Modifier format: Command & Modifier 6 bytes, Data to AXI bus transaction	R/W
3:2	ADDSIZE[1:0]	Address size These bits configure the number of address byte in memory-mapping mode. In case of 8D-8D-8D profile 2.0, it should be configured to 4 bytes. 0 0: 1 byte (256-byte address space) 0 1: 2 bytes (64 KB address space) 1 0: 3 bytes (16 MB address space) 1 1: 4 bytes (4 GB address space)	R/W
4	WPBSTMD	Wrapping burst mode When this field is set to 1, the wrapping boundary between system bus access and xSPI memory shall be matched. 0: Separate xSPI transfer at the wrapping address boundary 1: Not separate xSPI transfer at the wrapping address boundary	R/W
5	ARYAMD	Array address mode When this field is set to 1, address for memory is mapped as {A[25:10], A[9:4], 6{RSV}, A[3:0]} where A[25:0] is normal address, and RSV is reserved value (0b). This field is effective only when FFMT = 1xb. 0: Normal address mode 1: Array address mode	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
23:16	ADDRPEN[7:0]	Address Replace Enable These bits select the bits to replace for MSByte of AXI bus address in memory-mapping mode. 0: No replacement (xSPI frame address field is same as AXI bus address) 1: Replacement	R/W
31:24	ADDRPCD[7:0]	Address Replace Code These bits configure the code to replace the MSByte of AXI bus address in memory-mapping mode. It replaces the corresponding bits when Address Replace Enable bit is set to 1.	R/W

This register has functions to configure xSPI Master function.

37.3.1.5 CMCFG1CSn : xSPI Command Map Configuration Register 1 CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x014 + 0x010 × n

Bit position:	31											20			16	15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RDLATE[4:0]				RDCMD[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RDCMD[15:0]	Read command These bits configure the command field of read transaction in memory-mapping mode. Normal format and 8D-8D-8D profile 2.0 format use only upper 1 byte. 8D-8D-8D profile 1.0 format uses 2 bytes.	R/W
20:16	RDLATE[4:0]	Read latency cycle These bits configure the latency cycle of read transaction in memory-mapping mode. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to configure xSPI Master function.

37.3.1.6 CMCFG2CSn : xSPI Command Map Configuration Register 2 CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x018 + 0x010 × n

Bit position:	31											20			16	15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	WRLATE[4:0]				WRCMD[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WRCMD[15:0]	Write command These bits configure the command field of write transaction in memory-mapping mode. Normal format and 8D-8D-8D profile 2.0 format use only upper 1 byte. 8D-8D-8D profile 1.0 format uses 2 bytes.	R/W

Bit	Symbol	Function	R/W
20:16	WRLATE[4:0]	Write latency cycle These bits configure the latency cycle of write transaction in memory-mapping mode. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to configure xSPI Master function.

37.3.1.7 LIOCFGCSn : xSPI Link I/O Configuration Register CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x050 + 0x004 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DDRSMPEX[3:0]				SDRSMPSFT[3:0]				SDRSMPMD	SDRDRV	CSNEGEX	CSASTEX	CSMIN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WRMSKMD	LATEMD	PRTMD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PRTMD[9:0]	Protocol mode These bits configure the protocol mode and the pin to sample data inputs. In case of using not SPI clock but Data strobe for sampling in SDR mode, it is required to set PRTMD[9] to 1. 0x000: 1S-1S-1S 0x3B2: 4S-4D-4D 0x3FF: 8D-8D-8D 0x048: 1S-2S-2S 0x049: 2S-2S-2S 0x090: 1S-4S-4S 0x092: 4S-4S-4S Others: Setting prohibited	R/W
10	LATEMD	Latency mode This bit selects the behavior of initial access latency phase for both direct-manual mode and memory-mapping mode. When set to 0, the latency cycle is equal to each configured cycle from transmitting address field. When set to 1, the latency cycle is incremented from the last byte-pair of Address field and is extended 2 times of each configured cycle depending on data strobe port. It is used only for profile 2.0 frame format of 8D-8D-8D protocol mode with 6 bytes command/address field. Please refer to xSPI protocol for detail.	R/W

Value: Function	Frame format	Usage
0: Configurable latency	profile 2.0	The configurable latency cycle should be set as minus 1.
	Others	Latency cycle increments after address field.
1: Variable latency	profile 2.0	Latency cycle increments from address [23:16]. And it should not be set to 1.
	Others	Not support

Bit	Symbol	Function	R/W
11	WRMSKMD	Write mask mode This bit selects to use Data strobe port as write data mask. It could be useful for write access of odd byte. It is used only for 8D-8D-8D protocol mode. 0: Write mask disable 1: Write mask enable	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
19:16	CSMIN[3:0]	CS minimum idle term This bit configures the minimum cycle between xSPI frames. 0x0: 1 cycle 0x1: 2 cycles ⋮ 0xE: 15 cycles 0xF: 16 cycles	R/W
20	CSASTEX	CS asserting extension This bit extends 1 cycle chip select pins when asserting. 0: No extension 1: Extend 1 cycle	R/W
21	CSNEGEX	CS negating extension This bit extends 1 cycle chip select pins when negating. 0: No extension 1: Extend 1 cycle	R/W
22	SDRDRV	SDR driving timing This bit configures the timing of data output in SDR. This bit should not be used in case of no latency cycle. Because SPI output data could be conflicted with SPI input data. 0: Drive at 1/2 cycle before CK rising-edge 1: Drive at CK rising-edge	R/W
23	SDRSMPMD	SDR Sampling mode This bit selects the edge of sampling in SDR. In DDR, regardless of this setting, it samples data input ports with both edges of DS. When set to 1, it samples at rising-edge before falling-edge. 0: Samples data input at falling-edge 1: Samples data input at rising-edge	R/W
27:24	SDRSMPSFT[3:0]	SDR Sampling window shift These bits shift the timing of CK sampling in SDR. In case of using DS in SDR, there is no influence on the behavior. In case of DDR or using DS in SDR, it should be set to 0. 0x0: Sample without delay 0x1: Sample at 1 cycle delay ⋮ 0x6: Sample at 6 cycle delay 0x7: Sample at 7 cycle delay Others: Setting prohibited	R/W
31:28	DDRSMPEX[3:0]	DDR sampling window extend These bits configure the cycle of extending the sampling window in DDR. In DDR, the input data is sampled during the expected cycle soon after latency cycle. The input data out of range is ignored. It can be configured depending on DS propagation delay. 0x0: Expand no cycle 0x1: Expand 1 cycle ⋮ 0x6: Expand 6 cycles 0x7: Expand 7 cycles Others: Setting prohibited	R/W

This register has functions to configure xSPI Master function.

37.3.2 xSPI Control Registers

These registers control xSPI Master function.

37.3.2.1 BMCTL0 : xSPI Bridge Map Control Register 0

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CS1ACC[1:0]		CS0ACC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	CS0ACC[1:0]	AXI bus channel to slave0 memory area access enable These bits enable the access from AXI bus channel to CS0 memory. 0 0: Read/Write disable 0 1: Read enable, Write disable 1 0: Read disable, Write enable 1 1: Read/Write enable	R/W
3:2	CS1ACC[1:0]	AXI bus channel to slave1 memory area access enable These bits enable the access from AXI bus channel to CS1 memory. 0 0: Read/Write disable 0 1: Read enable, Write disable 1 0: Read disable, Write enable 1 1: Read/Write enable	R/W
7:4	—	These bits are read as 1. The write value should be 1.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function.

37.3.2.2 BMCTL1 : xSPI Bridge Map Control Register 1

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PBUF CLR	—	MWRP USH	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	The write value should be 0.	W
8	MWRPUSH	Memory Write Data Push This bit requests to push the pending data in combination mode. 0: No command 1: Push request	W
9	—	The write value should be 0.	W

Bit	Symbol	Function	R/W
10	PBUFCLR	Prefetch Buffer clear This bit requests to clear the prefetch buffer when enabled the prefetch function. It should not be set during memory access (COMSTT.MEMACC = 1). 0: No command 1: Clear request	W
31:11	—	The write value should be 0.	W

This register has functions to control xSPI Master function.

37.3.2.3 CMCTL : xSPI Command Map Control Register

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x068

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	XIPEN	XIPEXCODE[7:0]								XIPENCODE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	XIPENCODE[7:0]	XiP mode enter code These bits configure the code to enter XiP mode in memory-mapping mode.	R/W
15:8	XIPEXCODE[7:0]	XiP mode exit code These bits configure the code to exit XiP mode in memory-mapping mode.	R/W
16	XIPEN	XiP mode enable This bit enables XiP mode in memory-mapping mode. When set to 1, XiP enter code is inserted in the latency field, and the command field in next transaction is omitted. When set to 0, XiP exit code is inserted in the latency field. And it is set to 0 automatically when transmitting XiP disable pattern. It should not be used for 8D-8D-8D protocol mode profile 2.0 frame format. 0: Disable XiP mode 1: Enable XiP mode	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function.

37.3.2.4 CDCTL0 : xSPI Command Manual Control Register 0

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PERREP[3:0]				—	—	—	PERITV[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TRNUM[1:0]		CSSEL	—	PERMD	TRREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRREQ	Transaction request This bit requests to issue the transaction of manual-command. When set to 1, it starts the transaction. It is cleared to 0 when the transaction completed. The transaction is canceled by clearing to 0 while the transaction is ongoing. 0: No transaction 1: Request transaction	R/W
1	PERMD	Periodic mode This bit enables the periodic transaction mode. When set to 1, it repeats a transaction periodically and compares the read value with the expected value. It alternates the status polling operation for external memory. 0: Direct manual-command mode 1: Periodic manual-command mode	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue manual-command. 0: CS0 1: CS1	R/W
5:4	TRNUM[1:0]	Transaction number These bits configure the number of transactions in normal manual-command mode. In periodic manual-command, regardless of this setting, it issues the transaction using only command buffer 0. 0 0: Issue 1 command (using command buffer 0) 0 1: Issue 2 commands (using command buffer 0-1) 1 0: Issue 3 commands (using command buffer 0-2) 1 1: Issue 4 commands (using command buffer 0-3)	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
20:16	PERITV[4:0]	Periodic transaction interval These bits configure the interval of transaction in periodic manual-command mode. Too short interval compared with CPU bus cycle could result in no store into command buffer0. The interval should be longer than 4 times of CPU bus cycle. 0x00: 2 (= 2^1) cycles 0x01: 4 (= 2^2) cycles ⋮ 0x1E: 2,147,483,648 (= 2^{31}) cycles 0x1F: 4,294,967,296 (= 2^{32}) cycles	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
27:24	PERREP[3:0]	Periodic transaction repeat These bits configure the number of transaction repetitions in periodic manual-command mode. 0x0: 1 (= 2^0) time 0x1: 2 (= 2^1) times ⋮ 0xE: 16384 (= 2^{14}) times 0xF: 32768 (= 2^{15}) times	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function.

37.3.2.5 CDCTL1 : xSPI Command Manual Control Register 1

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x074

Bit position:	31																												0
Bit field:	PEREXP[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

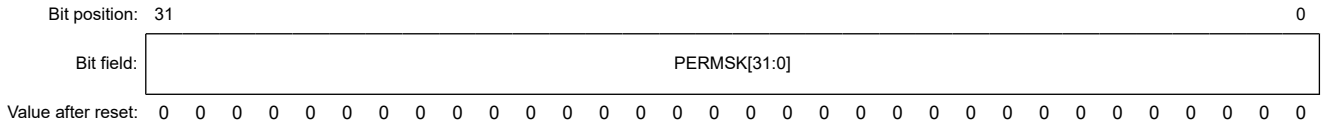
Bit	Symbol	Function	R/W
31:0	PEREXP[31:0]	Periodic transaction expected value These bits configure the expected value to compare with the read value in periodic manual-command mode. For example, in case of comparing 1 byte, the lower byte should be configured.	R/W

This register has functions to control xSPI Master function.

37.3.2.6 CDCTL2 : xSPI Command Manual Control Register 2

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x078



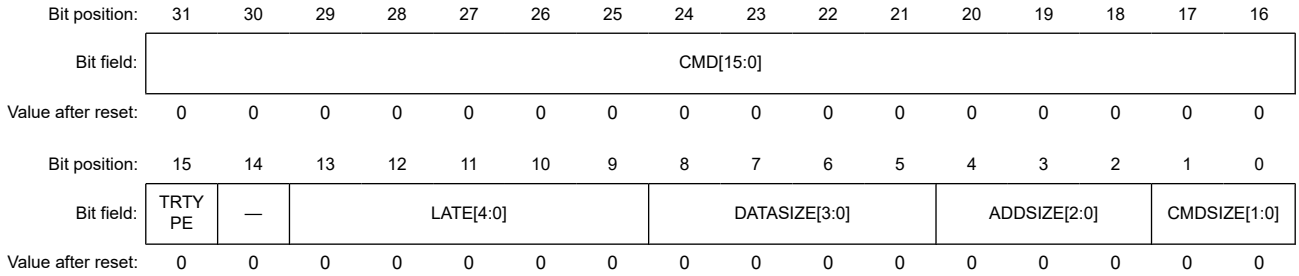
Bit	Symbol	Function	R/W
31:0	PERMSK[31:0]	Periodic transaction masked value These bits configure the masked value for the expected value in periodic manual-command mode. When set 1 to any bit, the corresponding bit configured as expected value (CDCTL1.PEREXP[31:0]) is ignored. In 8D-8D-8D, the data bytes are transferred only in byte pairs on SPI bus. It means the dummy read data could be stored. It should be masked for unused bits. For example, in case of read lower 1 byte, it should be configured to 0xFFFFFF00.	R/W

This register has functions to control xSPI Master function.

37.3.2.7 CDTBUFn : xSPI Command Manual Type Buf n (n = 0 to 3)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x080 + 0x010 × n



Bit	Symbol	Function	R/W
1:0	CMDSIZE[1:0]	Command Size These bits configure the size of command field. In case of 8D-8D-8D, it should be fixed to 10b. It should not be configured both command size and address size to zero. 0 0: 0 bytes (No command phase) 0 1: 1 byte 1 0: 2 bytes Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
4:2	ADDSIZE[2:0]	Address size These bits configure the size of address field. 0 0 0: 0 bytes (No address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Others: Setting prohibited	R/W
8:5	DATASIZE[3:0]	Write/Read Data Size These bits configure the size of data field. In 8D-8D-8D, the data bytes are transferred only in byte pairs on SPI bus. For example, even if configuring 1 byte for read, 2 bytes data is received. The last byte should be ignored. The 0 bytes must not configured for read transaction. 0x0: 0 bytes (No data phase) 0x1: 1 byte ⋮ 0x7: 7 bytes 0x8: 8 bytes Others: Setting prohibited	R/W
13:9	LATE[4:0]	Latency cycle These bits configure the latency cycle in manual-command mode. 0x0: No latency 0x1: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	TRTYPE	Transaction Type This bit selects the type of transaction. 0: Read transaction (Readout data from slave device) 1: Not read transaction	R/W
31:16	CMD[15:0]	Command (1-2 bytes) These bits configure the command field in manual-command mode. The number of bytes configured in Command Size bit is transferred. 1S-1S-1S, 4S-4D-4D: CMD[15:8] is command field, CMD[7:0] is not used. 8D-8D-8D profile 1.0: CMD[15:8] is command field, CMD[7:0] is extension field. 8D-8D-8D profile 2.0: CMD[15:0] is upper 2 bytes of command & modifier field. (bit 47-32 in xSPI protocol)	R/W

This register has functions to control xSPI Master function.

37.3.2.8 CDABUFn : xSPI Command Manual Address Buf n (n = 0 to 3)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x084 + 0x10 × n

Bit position:	31																												0
Bit field:	ADD[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ADD[31:0]	Address These bits configure the address field in manual-command mode. 1S-1S-1S, 4S-4D-4D, 8D-8D-8D profile 1.0: It is address field. 8D-8D-8D profile 2.0: It is lower 4 bytes of command & modifier field. (bit 31-0 in xSPI protocol)	R/W

This register has functions to control xSPI Master function.

37.3.2.9 CDD0BUFn : xSPI Command Manual Data 0 Buf n (n = 0 to 3)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x088 + 0x10 × n

Bit position:	31																												0
Bit field:	DATA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Write/Read Data These bits configure the data field in manual-command mode. In case of write transaction, the write data should be configured. In case of read transaction, the read data is stored.	R/W

This register has functions to control xSPI Master function.

37.3.2.10 CDD1BUFn : xSPI Command Manual Data 1 Buf n (n = 0 to 3)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x08C + 0x10 × n

Bit position:	31																												0
Bit field:	DATA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Write/Read Data These bits configure the data field in manual-command mode. In case of write transaction, the write data should be configured. In case of read transaction, the read data is stored.	R/W

This register has functions to control xSPI Master function.

37.3.2.11 LPCTL0 : xSPI Link Pattern Control Register 0

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	XD2VAL	—	—	XD2LEN[4:0]					XD1VAL	—	—	XD1LEN[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	XDPIN[1:0]		CSSEL	—	—	PATREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PATREQ	Pattern request This bit requests to issue the pattern. When set to 1, it starts the pattern. It is cleared to 0 when the pattern completed. 0: No request XiP Disable pattern 1: Request XiP Disable pattern	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue a pattern. 0: slave0 (CS0) 1: slave1 (CS1)	R/W
5:4	XDPIN[1:0]	XiP Disable pattern pin These bits select the data output pins to transmit XiP Disable pattern. 0 0: 1 pin 0 1: 2 pins 1 0: 4 pins 1 1: 8 pins	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
20:16	XD1LEN[4:0]	XiP Disable pattern 1st phase length These bits select the length of 1st phase in XiP disable pattern. The pattern with zero-length both 1st phase and 2nd phase should not be configured. 0x0: 0 cycles 0x1: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	XD1VAL	XiP Disable pattern 1st phase value This bit selects the value of 1st phase in XiP disable pattern. 0: Low drive 1: High drive	R/W
28:24	XD2LEN[4:0]	XiP Disable pattern 2nd phase length These bits select the length of 2nd phase in XiP disable pattern. 0x00: 0 cycles 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
30:29	—	These bits are read as 0. The write value should be 0.	R/W
31	XD2VAL	XiP Disable pattern 2nd phase value This bit selects the value of 2nd phase in XiP disable pattern. 0: Low drive 1: High drive	R/W

This register has functions to control xSPI Master function.

37.3.2.12 LPCTL1 : xSPI Link Pattern Control Register 1

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	RSTSU[2:0]			—	RSTWID[2:0]			—	—	RSTREP[1:0]		CSSEL	—	PATREQ[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PATREQ[1:0]	Pattern request These bits request to issue the pattern. When set to 01b or 10b, it starts the pattern. It is cleared to 00b when the pattern completed. 0 0: No request 0 1: Request Reset pattern 1 0: Request CS only pattern 1 1: Setting prohibited	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue a pattern. 0: slave0 (CS0) 1: slave1 (CS1)	R/W
5:4	RSTREP[1:0]	Reset pattern repeat These bits select the repeating time to toggle CS from LOW to HIGH. 0 0: 4 times (Specified on Reset Signaling Protocol) 0 1: 5 times 1 0: 6 times 1 1: 7 times	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RSTWID[2:0]	Reset pattern width These bits configure the width of cycle in reset pattern and CS only pattern. It toggles CS with the configured cycle. 0 0 0: 2 (= 2^1) cycles 0 0 1: 4 (= 2^2) cycles ⋮ 1 1 0: 128 (= 2^7) cycles 1 1 1: 256 (= 2^8) cycles	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	RSTSU[2:0]	Reset pattern data output setup time These bits configure the number of setup cycle for data output based on the edge of CS in reset pattern. It needs enough setup time because xSPI slave samples any data at the rising edge of CS. This cycle of setup time should be less than the cycle of reset pattern width (RSTWID[2:0]). 0 0 0: 1 cycle 0 0 1: 2 cycles ⋮ 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function.

37.3.2.13 LIOCTL : xSPI Link I/O Control Register

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSTCS1	RSTCS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WPCS1	WPCS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	WPCS0*1	WP drive for slave0 This bit controls the value of Write Protect (WP0#) port. It can be useful only for xSPI slave with write protect port. 0: Drive Low level 1: Drive High level	R/W
1	WPCS1*1	WP drive for slave1 This function is same as WPCS0. 0: Drive Low level 1: Drive High level	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	RSTCS0*1	Reset drive for slave0 This bit controls the value of xSPI reset (RESET0#) port. It can be useful only for xSPI slave with reset port. 0: Drive Low level 1: Drive High level	R/W
17	RSTCS1*1	Reset drive for slave1 This function is same as RSTCS0. 0: Drive Low level 1: Drive High level	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. For unit 1, these bits are reserved. The write value should be 0.

This register has functions to control xSPI Master function.

37.3.2.14 CCCTL0CSn : xSPI Command Calibration Control Register 0 CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x130 + 0x020 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CASFTEND[4:0]					—	—	—	CASFTSTA[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CAITV[4:0]					—	—	—	—	—	—	CANOWR	CAEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CAEN	Automatic Calibration Enable This bit enables the automatic calibration. When set to 1, it transmits the calibration sequence periodically and adjusts the value of phase shift. When set to 0 during the calibration sequence, it stops after completed ongoing calibration sequence, and then this bit is cleared. 0: Disable automatic calibration 1: Enable automatic calibration	R/W
1	CANOWR	Calibration no write mode This bit selects to omit write command in calibration sequence. It can be used for any slave device with fixed calibration pattern data. 0: Calibration sequence with write command 1: Calibration sequence without write command	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
12:8	CAITV[4:0]	Calibration interval These bits configure the interval between calibration patterns. 0x00: 2 (= 2^1) cycle wait 0x01: 4 (= 2^2) cycle wait ⋮ 0x1E: 2,147,483,648 (= 2^{31}) cycle wait 0x1F: 4,294,967,296 (= 2^{32}) cycle wait	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	CASFTSTA[4:0]	Calibration DS shift start value These bits configure the start value of DS shift.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	CASFTEND[4:0]	Calibration DS shift end value These bits configure the end value of DS shift. It should be equal or more than the start value (CASFTSTA).	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function.

37.3.2.15 CCCTL1CSn : xSPI Command Calibration Control Register 1 CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x134 + 0x020 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CARDLATE[4:0]					—	—	—	CAWRLATE[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CADATASIZE[3:0]				CAADDSIZE[2:0]			CACMDSIZE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CACMDSIZE[1:0]	Command Size These bits configure the size of command field. In case of 8D-8D-8D, it should be fixed to 10b. It should not be configured both command size and address size to zero. 0 0: 0 bytes (No command phase) 0 1: 1 byte 1 0: 2 bytes 1 1: Setting prohibited	R/W

Bit	Symbol	Function	R/W
4:2	CAADDSIZE[2:0]	Address size These bits configure the size of address field. 0 0 0: 0 bytes (No address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Others: Setting prohibited	R/W
8:5	CADATASIZE[3:0]	Write/Read Data Size These bits configure the size of data field. In 8D-8D-8D, it should be configured with even byte. 0x0: 1 byte 0x1: 2 bytes ⋮ 0xE: 15 bytes 0xF: 16 bytes	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
20:16	CAWRLATE[4:0]	Write Latency cycle These bits configure the latency cycle in calibration frame. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	CARDLATE[4:0]	Read Latency cycle These bits configure the latency cycle in calibration frame. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.3.2.16 CCCTL2CSn : xSPI Command Calibration Control Register 2 CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x138 + 0x020 × n

Bit position:	31															16	15														0
Bit field:	CARDCMD[15:0]																CAWRCMD[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CAWRCMD[15:0]	Calibration pattern write command These bits configure the calibration pattern write command.	R/W
31:16	CARDCMD[15:0]	Calibration pattern read command These bits configure the calibration pattern read command.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.3.2.17 CCCTL3CSn : xSPI Command Calibration Control Register 3 CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x13C + 0x020 × n

Bit position:	31																												0
Bit field:	CAADD[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CAADD[31:0]	Calibration pattern address These bits configure the calibration pattern address.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.3.2.18 CCCTL4CSn : xSPI Command Calibration Control Register 4 CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x140 + 0x020 × n

Bit position:	31																												0
Bit field:	CADATA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.3.2.19 CCCTL5CSn : xSPI Command Calibration Control Register 5 CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x144 + 0x020 × n

Bit position:	31																												0
Bit field:	CADATA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

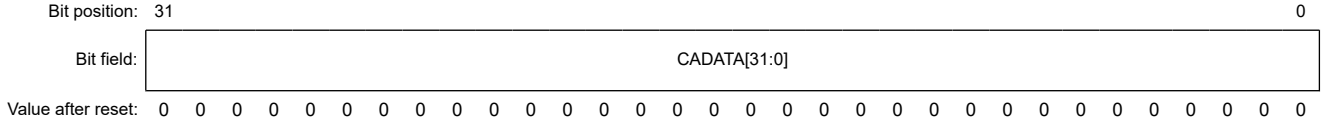
Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.3.2.20 CCCTL6CSn : xSPI Command Calibration Control Register 6 CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x148 + 0x020 × n



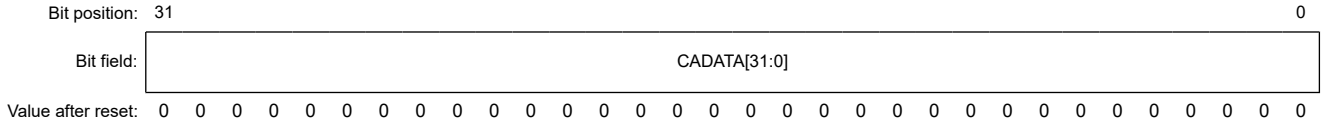
Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

37.3.2.21 CCCTL7CSn : xSPI Command Calibration Control Register 7 CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x14C + 0x020 × n



Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

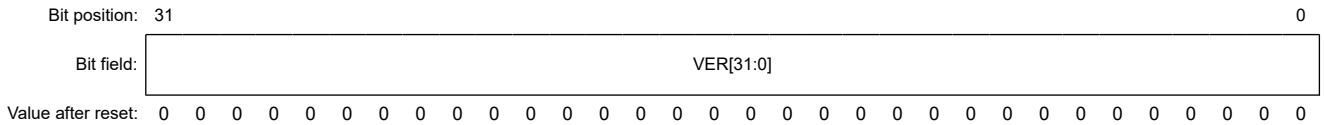
37.3.3 xSPI Status Registers

These registers monitor the status of xSPI Master.

37.3.3.1 VERSTT : xSPI Version Register

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x180



Bit	Symbol	Function	R/W
31:0	VER[31:0]	Version These bits indicate this IP version.	R

This register indicates the status of xSPI Master.

37.3.3.2 COMSTT : xSPI Common Status Register

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x184

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	RSTOCS1	INTCS1	ECSCS1	—	RSTOCS0	INTCS0	ECSCS0
Value after reset:	0	0	0	0	0	0	0	0	0	*1	*1	*1	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	WRBUFNE	—	PBUFNE	—	—	—	MEMACC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MEMACC	Memory access ongoing 0: AXI bus is not accessing to memory. 1: AXI bus is accessing to memory.	R
3:1	—	These bits are read as 0.	R
4	PBUFNE	Prefetch Buffer Not Empty 0: Empty 1: Not empty	R
5	—	This bit is read as 0.	R
6	WRBUFNE	Write Buffer Not Empty 0: Empty 1: Not empty	R
15:7	—	These bits are read as 0.	R
16	ECSCS0*2	ECS monitor for slave0 This bit indicates the value of ECS port. 0: Low level 1: High level	R
17	INTCS0*2	INT monitor for slave0 This bit indicates the value of INT port. 0: Low level 1: High level	R
18	RSTOCS0*2	RSTO monitor for slave0 This bit indicates the value of RSTO port. 0: Low level 1: High level	R
19	—	This bit is read as 0.	R
20	ECSCS1*2	ECS monitor for slave1 This function is same as ECSCS0. 0: Low level 1: High level	R
21	INTCS1*2	INT monitor for slave1 This function is same as INTCS0. 0: Low level 1: High level	R
22	RSTOCS1*2	RSTO monitor for slave1 This function is same as RSTOCS0. 0: Low level 1: High level	R
31:23	—	These bits are read as 0.	R

Note 1. Value after reset is 1 for unit 0 and 0 for unit 1.
Note 2. These bits are reserved for unit 1.

This register indicates the status of xSPI Master.

37.3.3.3 CASTTCSn : xSPI Calibration Status Register CSn (n = 0, 1)

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x188 + 0x004 × n

Bit position:	31																												0
Bit field:	CASUC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CASUC[31:0]	Calibration Success These bits indicate the calibration success for each DS shift value. It is updated when completed each calibration sequence. CASUC[x] indicates calibration success in DS shift value = x.	R

This register indicates the status of xSPI Master.

37.3.4 xSPI Interrupt Registers

These registers control the interrupt function of xSPI Master.

37.3.4.1 INTS : xSPI Interrupt Status Register

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x190

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1	CASUCCS0	CAFAILCS1	CAFAILCS0	—	—	—	—	—	—	—	BUSERR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1	INTCS0	—	—	ECSCS1	ECSCS0	—	—	DSTOCS1	DSTOCS0	PERTO	INICMP	PATCMP	CMDCMP
Value after reset:	0	0	*1	*1	0	0	*1	*1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMP	Command Completed This bit is set to 1 when completed the requested manual-command. In direct manual-command, it means all transactions completed. In periodic manual-command, it means the read data matches with the expected data. 0: No detection 1: Detection	R
1	PATCMP	Pattern Completed This bit is set to 1 when completed the requested pattern. 0: No detection 1: Detection	R
2	INICMP	Initial Sequence Completed This bit is set to 1 when completed the initial sequence. 0: No detection 1: Detection	R
3	PERTO	Periodic transaction timeout This bit is set to 1 when the read value does not match with the expected value in periodic manual-command mode. 0: No detection 1: Detection	R

Bit	Symbol	Function	R/W
4	DSTOCS0	DS timeout for slave0 This bit is set to 1 when lost DS in read transaction with using DS. It means not receiving the data during expected read phase. In this case, xSPI master stops the read transaction and the following transaction. 0: No detection 1: Detection	R
5	DSTOCS1	DS timeout for slave1 This function is same as DSTOCS0. 0: No detection 1: Detection	R
7:6	—	These bits are read as 0.	R
8	ECSCS0*2	ECC error detection for slave0 This bit is set to 1 when detected the falling edge on ECS0# port. It can be useful only for xSPI slave with ECC detection function. 0: No detection 1: Detection	R
9	ECSCS1*2	ECC error detection for slave1 This function is same as ECSCS0. 0: No detection 1: Detection	R
11:10	—	These bits are read as 0.	R
12	INTCS0*2	Interrupt detection for slave0 This bit is set to 1 when detected the falling edge on INT0# port. It can be useful only for xSPI slave with interrupt function. 0: No detection 1: Detection	R
13	INTCS1*2	Interrupt detection for slave1 This function is same as INTCS0. 0: No detection 1: Detection	R
19:14	—	These bits are read as 0.	R
20	BUSERR	AXI bus error This bit is set to 1 when an error response occurs on AXI bus.	R
27:21	—	These bits are read as 0.	R
28	CAFAILCS0	Calibration failed for slave0 This bit is set to 1 when failed calibration. 0: No detection 1: Detection	R
29	CAFAILCS1	Calibration failed for slave1 This function is same as CAFAILCS0. 0: No detection 1: Detection	R
30	CASUCCS0	Calibration success for slave0 This bit is set to 1 when success calibration. 0: No detection 1: Detection	R
31	CASUCCS1	Calibration success for slave1 This function is same as CASUCCS0. 0: No detection 1: Detection	R

Note 1. Value after reset is 0 for unit 0 and 1 for unit 1.
Note 2. These bits are reserved for unit 1.

This register indicates the status of interrupt. The bits in this register are cleared to 0 when writing 1 on the corresponding bit of INTC register.

37.3.4.2 INTC : xSPI Interrupt Clear Register

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x194

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1C	CASUCCS0C	CAFAILCS1C	CAFAILCS0C	—	—	—	—	—	—	—	BUSERRC	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1C	INTCS0C	—	—	ECSCS1C	ECSCS0C	—	—	DSTOCS1C	DSTOCS0C	PERTOC	INICMPC	PATCMPC	CMDCMPC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMPC	Command Completed interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
1	PATCMPC	Pattern Completed interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
2	INICMPC	Initial Sequence Completed interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
3	PERTOC	Periodic transaction timeout interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
4	DSTOCS0C	DS timeout for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
5	DSTOCS1C	DS timeout for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
7:6	—	The write value should be 0.	W
8	ECSCS0C*1	ECC error detection for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
9	ECSCS1C*1	ECC error detection for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
11:10	—	The write value should be 0.	W
12	INTCS0C*1	Interrupt detection for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
13	INTCS1C*1	Interrupt detection for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
19:14	—	The write value should be 0.	W
20	BUSERRC	AXI bus error interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
27:21	—	The write value should be 0.	W

Bit	Symbol	Function	R/W
28	CAFAILCS0C	Calibration failed for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
29	CAFAILCS1C	Calibration failed for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
30	CASUCCS0C	Calibration success for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
31	CASUCCS1C	Calibration success for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W

Note 1. For unit 1, these bits are reserved. The write value should be 1.

This register clears the status of interrupt.

37.3.4.3 INTE : xSPI Interrupt Enable Register

Base address: XSPIm = 0x801C_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x198

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1E	CASUCCS0E	CAFAILCS1E	CAFAILCS0E	—	—	—	—	—	—	—	BUSERRE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1E	INTCS0E	—	—	ECSCS1E	ECSCS0E	—	—	DSTOCS1E	DSTOCS0E	PERTOE	INICMPE	PATCMPE	CMDCMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMPE	Command Completed interrupt enable 0: Disabled 1: Enabled	R/W
1	PATCMPE	Pattern Completed interrupt enable 0: Disabled 1: Enabled	R/W
2	INICMPE	Initial Sequence Completed interrupt enable 0: Disabled 1: Enabled	R/W
3	PERTOE	Periodic transaction timeout interrupt enable 0: Disabled 1: Enabled	R/W
4	DSTOCS0E	DS timeout for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
5	DSTOCS1E	DS timeout for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	ECSCS0E*1	ECC error detection for slave0 interrupt enable 0: Disabled 1: Enabled	R/W

Bit	Symbol	Function	R/W
9	ECSCS1E*1	ECC error detection for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	INTCS0E*1	Interrupt detection for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
13	INTCS1E*1	Interrupt detection for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
19:14	—	These bits are read as 0. The write value should be 0.	R/W
20	BUSERRE	AXI bus error interrupt enable 0: Disabled 1: Enabled	R/W
27:21	—	These bits are read as 0. The write value should be 0.	R/W
28	CAFAILCS0E	Calibration failed for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
29	CAFAILCS1E	Calibration failed for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
30	CASUCCS0E	Calibration success for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
31	CASUCCS1E	Calibration success for slave1 interrupt enable 0: Disabled 1: Enabled	R/W

Note 1. For unit 1, these bits are reserved. The write value should be 0.

This register enables the interrupt.

37.3.5 xSPI Voltage and Size Registers

37.3.5.1 IOVOLCTL : xSPI IO Voltage Control Register

Base address: XSPI_MISC = 0x8029_3000 + 0x100 × m (m = 0, 1)

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	XSPI_MDV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	XSPI_MDV	Operation voltage of xSPI IO domain 0: 1.8 V 1: 3.3 V	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to configure xSPI Master function.

37.3.5.2 CS0ENDAD : xSPI CS0 End Address Register

Base address: XSPI_MISC = 0x8029_3000 + 0x100 × m (m = 0, 1)

Offset address: 0x004

Bit position:	31																												0
Bit field:	CS0_END_ADD[31:0]																												
Value after reset:	0	1	0	*1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	CS0_END_ADD[31:0]	CS0 (slave 0) area end address Value after reset is different between units. xSPI unit 0: 0x400F_FFFF xSPI unit 1: 0x500F_FFFF	R/W

Note 1. 0 for unit 0 and 1 for unit 1

This register is used to specify CS0 area end address. Set the value according to size of the connected slave device.

Note that CS0 area start address is fixed to 0x4000_0000 for unit 0 and 0x5000_0000 for unit 1. They are not configurable.

37.3.5.3 CS1STRAD : xSPI CS1 Start Address Register

Base address: XSPI_MISC = 0x8029_3000 + 0x100 × m (m = 0, 1)

Offset address: 0x008

Bit position:	31																												0
Bit field:	CS1_STR_ADD[31:0]																												
Value after reset:	0	1	0	*1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CS1_STR_ADD[31:0]	CS1 (slave 1) area start address Value after reset is different between units. xSPI unit 0: 0x4800_0000 xSPI unit 1: 0x5800_0000	R/W

Note 1. 0 for unit 0 and 1 for unit 1

This register is used to specify CS1 area start address.

37.3.5.4 CS1ENDAD : xSPI CS1 End Address Register

Base address: XSPI_MISC = 0x8029_3000 + 0x100 × m (m = 0, 1)

Offset address: 0x00C

Bit position:	31																												0
Bit field:	CS1_END_ADD[31:0]																												
Value after reset:	0	1	0	*1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	CS1_END_ADD[31:0]	CS1 (slave 0) area end address Value after reset is different between units. xSPI unit 0: 0x480F_FFFF xSPI unit 1: 0x580F_FFFF	R/W

Note 1. 0 for unit 0 and 1 for unit 1

This register is used to specify CS1 area end address. Set the value according to size of the connected slave device.

37.4 Operation

xSPI Master interface has the functions to issue the transaction for external memory with xSPI Slave interface. It allows to write to registers in external memory or read from it.

This xSPI Master mainly has two modes to issue the transaction. One is a manual-command mode; Software configures all fields of xSPI frame and starts the transaction by software request. The other is a memory-mapping mode; it automatically converts AXI bus access for pre-configured memory area into xSPI transaction. It enables to access from AXI bus to external memory area outside of chip via xSPI bus.

This section describes the xSPI bus operation, the direct control of xSPI frame (manual-command), the control of memory access (memory-mapping), the error operation, and the flow to operation.

37.4.1 xSPI Bus

This section describes the xSPI bus operation.

37.4.1.1 Supported Protocol Mode

This xSPI Master supports various protocol modes. It is configured by Protocol mode bits (LIOCFGCSn.PRTMD[9:0]). Table 37.6 lists the supported protocol mode.

Table 37.6 Supported protocol mode

Protocol mode	Function	PRTMD[9:0]	Note
1S-1S-1S	Command, Address, and Data fields are transferred at SDR with using 1 pin. Read data is sampled with CK.	0x000	Specified by xSPI protocol
4S-4D-4D	Command field is transferred at SDR with using 4 pins. Address and Data fields are transferred at DDR with using 4 pins. Read data is sampled with DS.	0x3B2	Specified by xSPI protocol
8D-8D-8D	Command, Address, and Data fields are transferred at DDR with using 8 pins. Read data is sampled with DS.	0x3FF	Specified by xSPI protocol
1S-2S-2S	Command field is transferred at SDR with using 1 pin. Address and Data fields are transferred at SDR with using 2 pins. Read data is sampled with CK.	0x048	—
2S-2S-2S	Command, Address, and Data fields are transferred at SDR with using 2 pins. Read data is sampled with CK.	0x049	—
1S-4S-4S	Command field is transferred at SDR with using 1 pin. Address and Data fields are transferred at SDR with using 4 pins. Read data is sampled with CK.	0x090	—
4S-4S-4S	Command, Address, and Data fields are transferred at SDR with using 4 pins. Read data is sampled with CK.	0x092	—

Note: In case of XiP mode enable, XiP code is inserted in Latency field. It is valid only for memory-mapping mode.

Table 37.7 lists a description of the main internal signals.

Table 37.7 Internal signals (1 of 2)

Signal name	I/O	Function
clk_spi	I	xSPI control clock
spi_ck	O	xSPI Clock (Same as XSPIm_CKP pin)
spi_cs0	O	xSPI Chip Select for slave0 (Same as XSPIm_CS0# pin)
spi_cs1	O	xSPI Chip Select for slave1 (Same as XSPIm_CS1# pin)
spi_doe[7:0]	O	xSPI Data Output Enable
spi_do[7:0]	O	xSPI Data Output
spi_di[7:0]	I	xSPI Data Input
spi_dsoe	O	xSPI Data Strobe Output Enable
spi_dso	O	xSPI Data Strobe Output

Table 37.7 Internal signals (2 of 2)

Signal name	I/O	Function
spi_dsi	I	xSPI Data Strobe Input

The bytes of Command and Address fields are transferred in highest order to lowest order sequence. The sequential bytes of Data field are transferred in lowest address to highest address order. In case of using multiple pins, the least significant bit of each byte is placed on spi_do/di[0] with each higher order bit on the successively higher numbered spi_do/di signals.

Figure 37.2 shows timing-chart for 1S-1S-1S protocol mode. The spi_do[0] signal is used for output data and the spi_di[1] signal is used for input data.

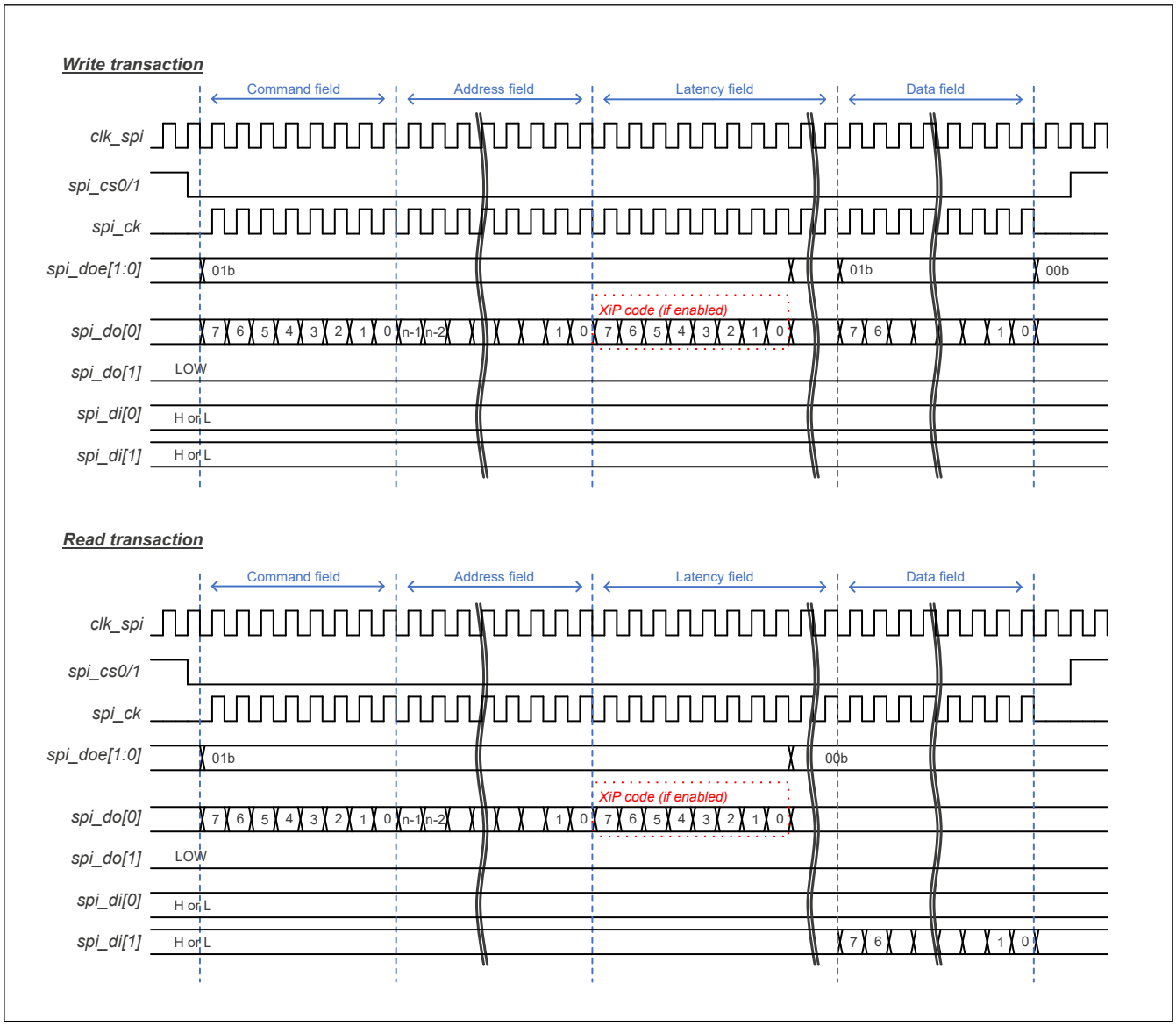


Figure 37.2 1S-1S-1S timing-chart

Figure 37.3 shows timing-chart for 1S-2S-2S protocol mode.

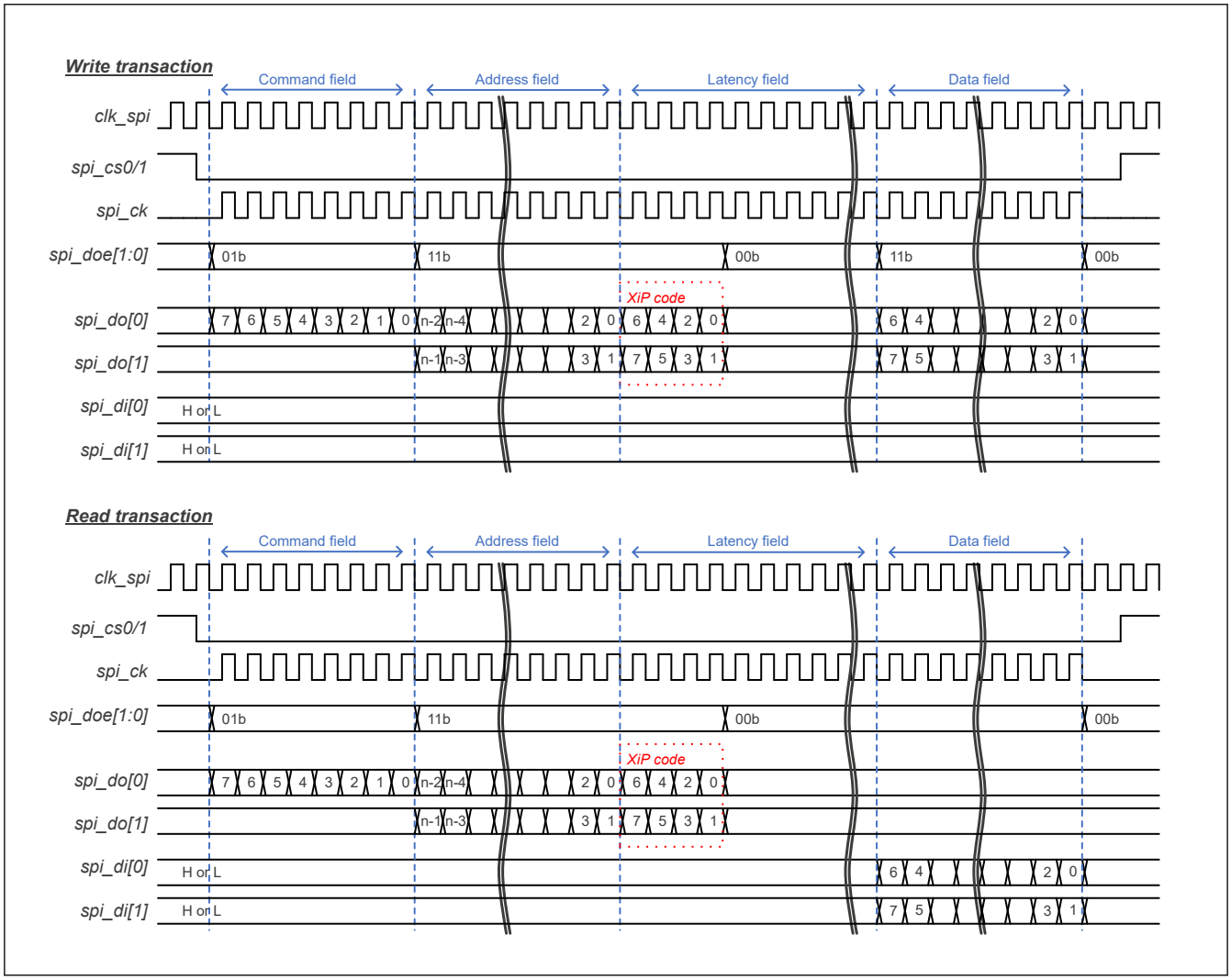


Figure 37.3 1S-2S-2S timing-chart

Figure 37.4 shows timing-chart for 4S-4D-4D protocol mode.

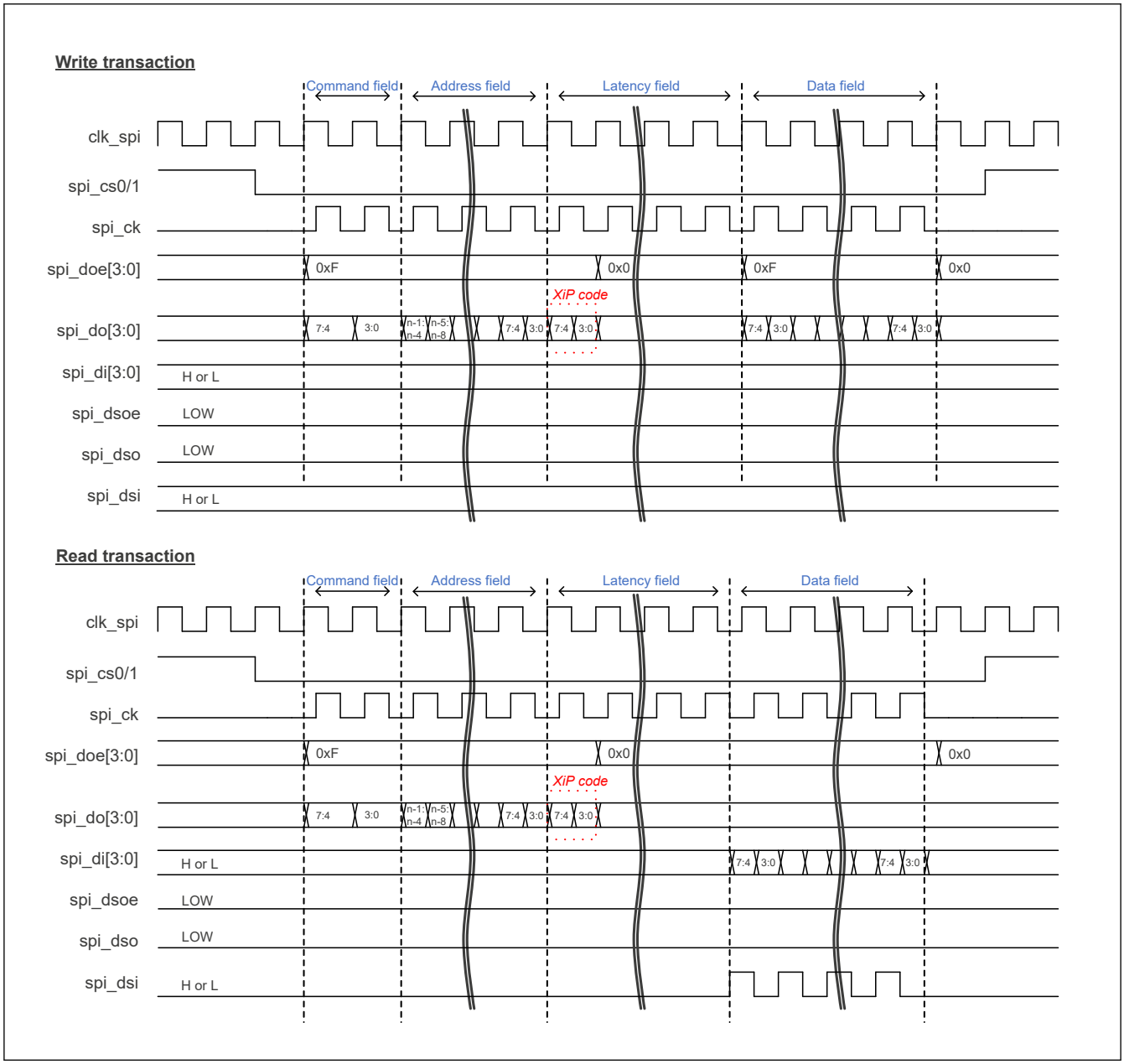


Figure 37.4 4S-4D-4D timing-chart

Figure 37.5 shows timing-chart for 8D-8D-8D profile 1.0 protocol mode. In 8D-8D-8D, the data is always transmitted with byte-pair. In case each field is odd byte, last one byte is padded with invalid data.

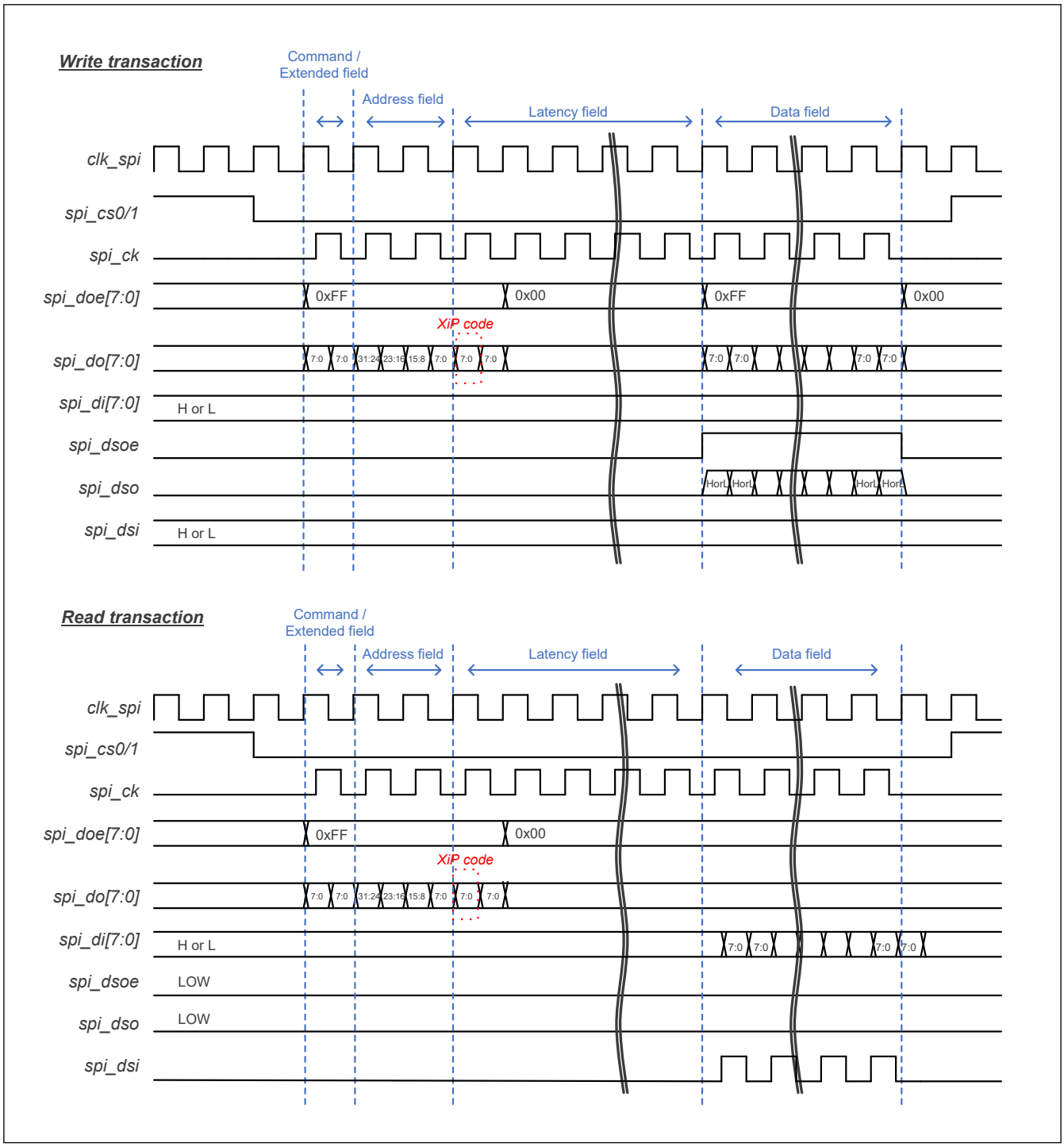


Figure 37.5 8D-8D-8D profile 1.0 timing-chart

Figure 37.6 shows timing-chart for 8D-8D-8D profile 2.0 protocol mode. In 8D-8D-8D, the data is always transmitted with byte-pair. In case each field is odd byte, last one byte is padded with invalid data.

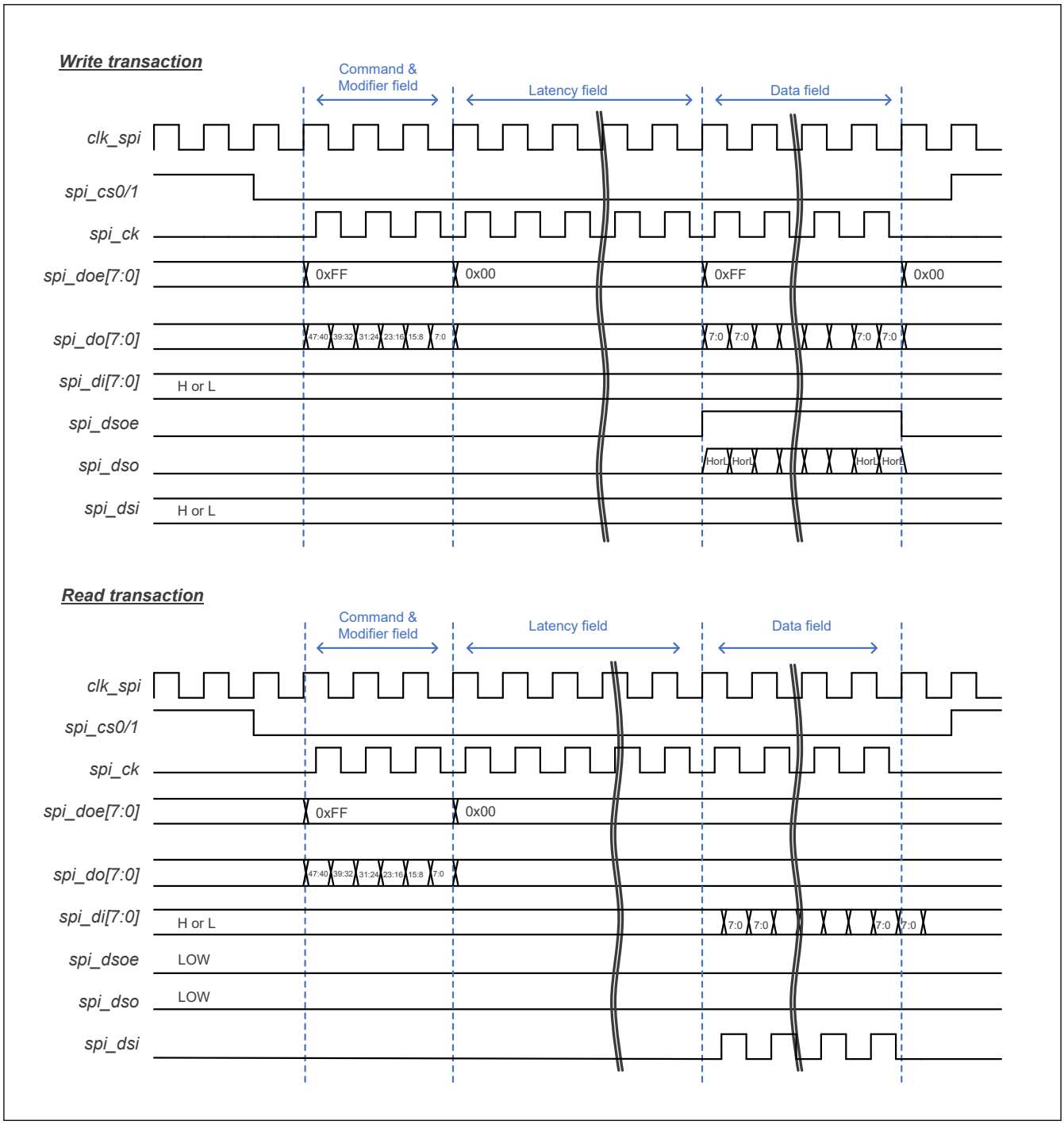


Figure 37.6 8D-8D-8D profile 2.0 timing-chart

37.4.1.2 xSPI Frame Interval

The interval between xSPI frames is configured with CS minimum idle term bits (LIOCFGCSn.CSMIN[3:0]). It depends on the specification of xSPI slave device.

37.4.1.3 xSPI Signals Timing Control

This xSPI Master supports both SDR and DDR. And it is possible to sample input data with Data-Strobe (DS) signal at SDR. For various modes and easy implementation, this xSPI Master could adjust the timing to drive/sample xSPI interface signals statically. Table 37.8 lists the summary of xSPI Interface signal timing control.

Table 37.8 Summary of xSPI signals timing control

Signal	Mode	Default operation	Timing Control (n = 0, 1)
CS (spi_cs) drive	—	Asserting 1 cycle before the rising-edge of first spi_ck	1 cycle extension for asserting with LIOCFGCSn.CSASTEX bit
		Negating 1.5 cycle after the falling-edge of last spi_ck	1 cycle extension for negating with LIOCFGCSn.CSNEGEX bit
CK (spi_ck) drive	SDR without DS SDR with DS	Reference point	—
	DDR with DS	Reference point	—
DO/DOE (spi_do/doe) drive	SDR without DS SDR with DS	Falling edge of clk_spi	0 or 0.5 cycle shift with LIOCFGCSn.SDRDRV bit
	DDR with DS	Both edges of clk_spi	—
DI (spi_di) sample	SDR without DS	Falling edge of spi_ck on expected data size	0 to 7 cycle shift (1 cycle unit) with LIOCFGCSn.SDRSMPSFT[3:0] bits 0 or 0.5 cycle shift with LIOCFGCSn.SDRSMPMD bit
	SDR with DS	Falling edge of spi_dsi signal on expected data size	Sample at rising edge with LIOCFGCSn.SDRSMPMD bit 0 to 1 cycle phase shift with WRAPCFG.DSSFTCSn[4:0] bits 0 to 7 cycle extension with LIOCFGCSn.DDRSMPEX[3:0] bits
	DDR with DS	Both edges of spi_dsi signal on expected data size	0 to 1 cycle phase shift with WRAPCFG.DSSFTCSn[4:0] bits 0 to 7 cycle extension with LIOCFGCSn.DDRSMPEX[3:0] bits

Note: In DDR on xSPI protocol, CK or DS should be aligned for center of data. It means to shift the phase by 0.25 cycle (90 degrees). This xSPI master supports to adjust this phase depending on the usage conditions.

Figure 37.7 shows the default operation and timing control for SDR without DS.

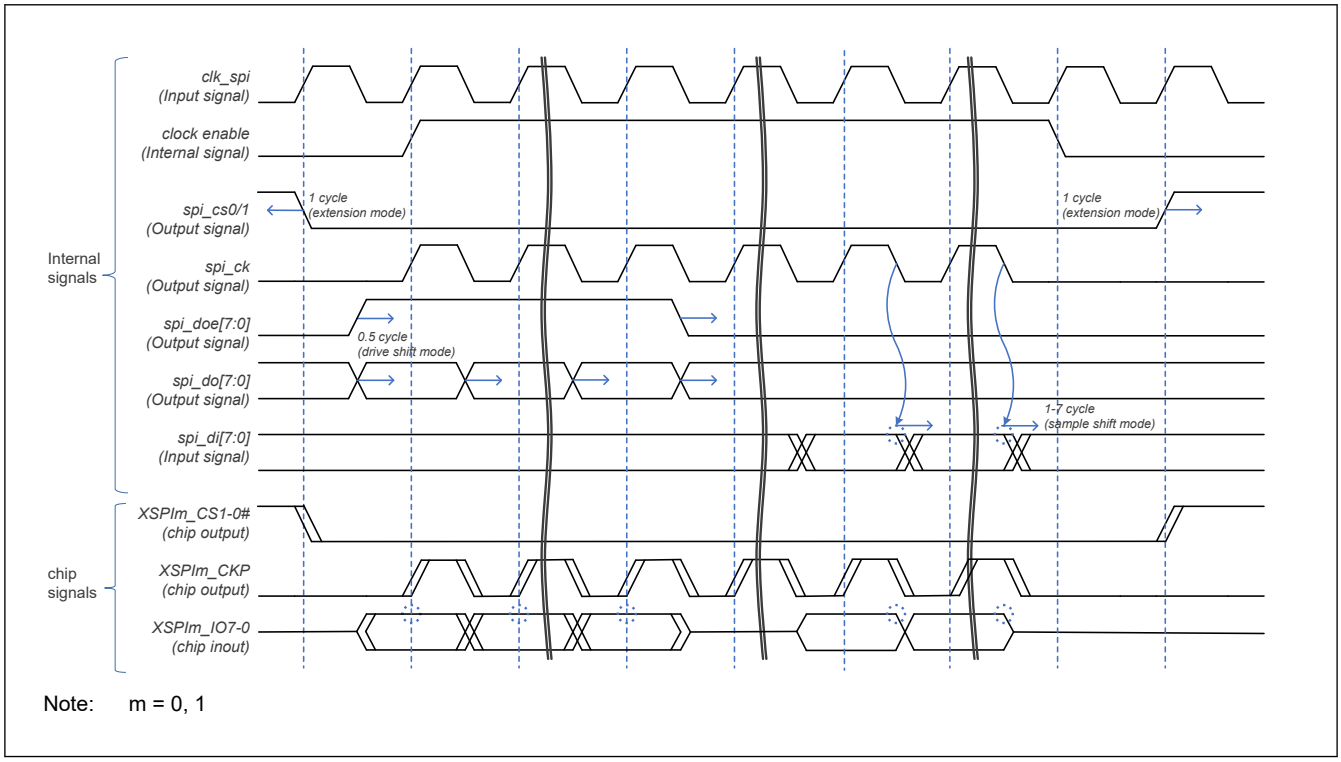


Figure 37.7 Timing control for SDR without DS

Figure 37.8 shows the default operation and timing control for SDR with DS.

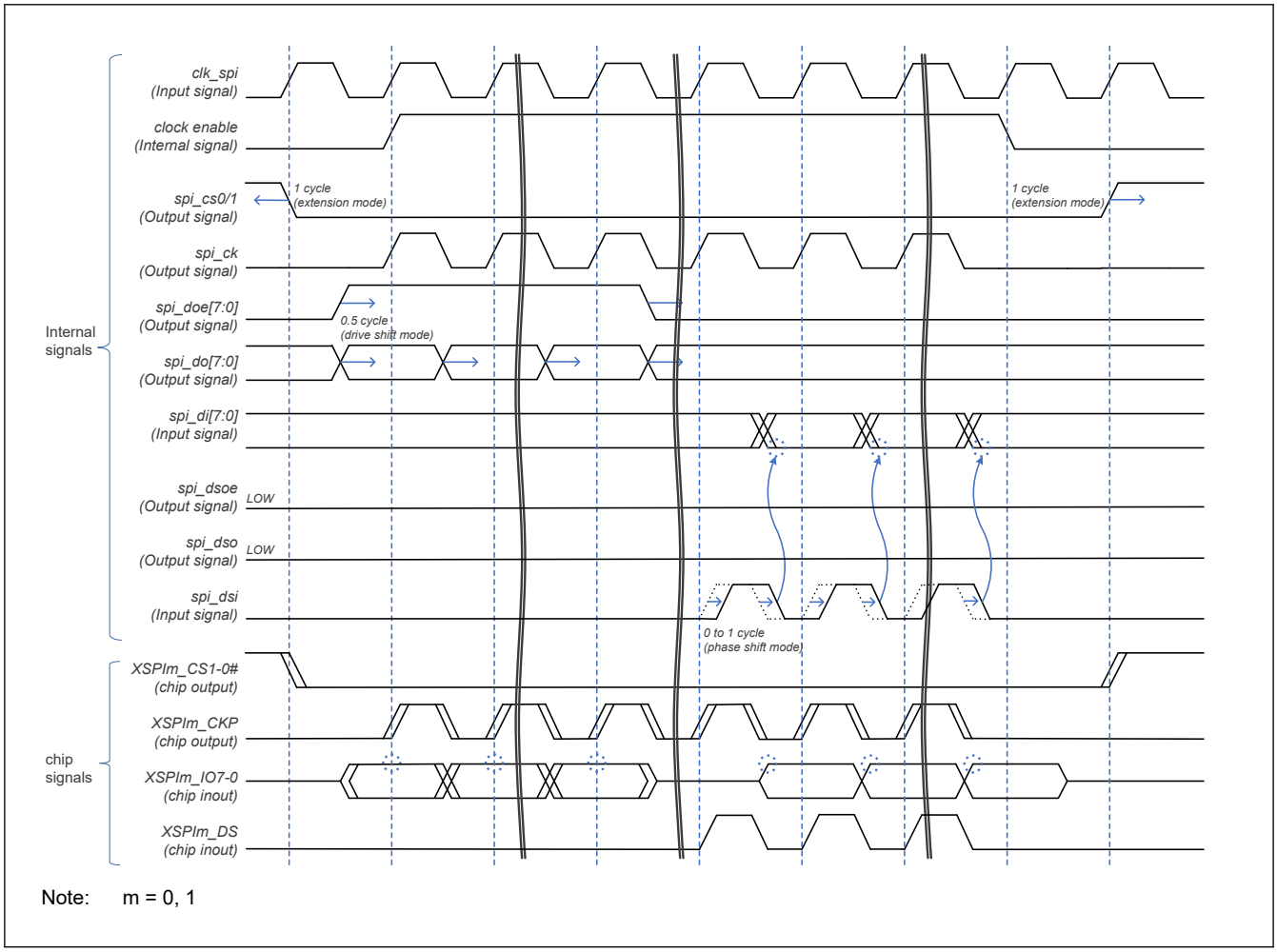


Figure 37.8 Timing control for SDR with DS

Figure 37.9 shows the default operation and timing control for DDR with DS.

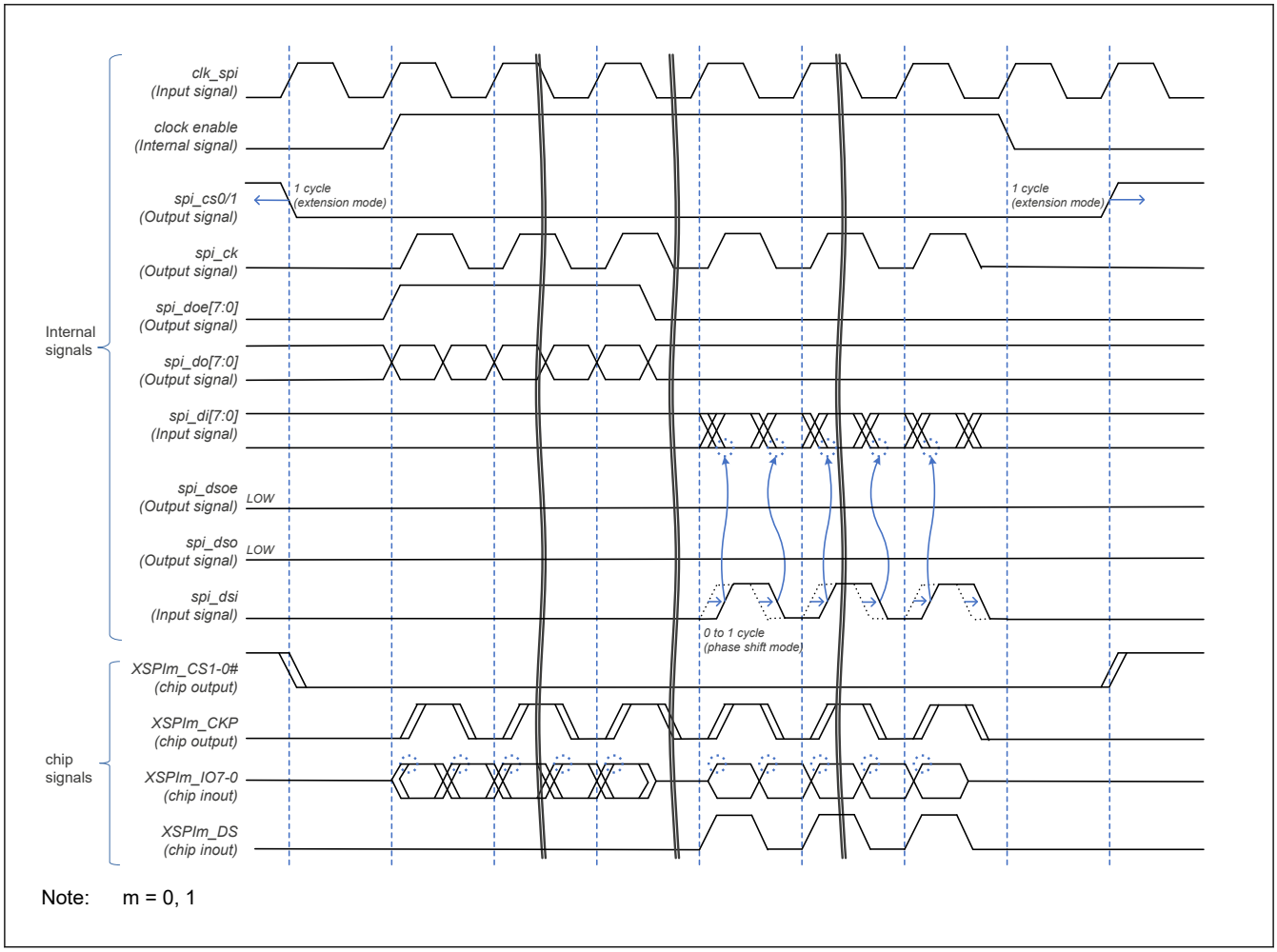


Figure 37.9 Timing control for DDR with DS

37.4.2 Manual-command

This section describes the manual-command mode. The manual-command has two functional modes: direct mode and periodic mode.

37.4.2.1 Direct Mode

This mode sequentially can issue up to four xSPI transactions configured and requested by Software. A series of transaction can be issued by a transaction request (CDCTL0.TRREQ = 1 with PERMD = 0). The number of transactions (CDCTL0.TRNUM[1:0]) can be configured up to 4. It can be used to change the mode or read the status of xSPI slave device.

Table 37.9 lists the configured register bits for direct manual-command. The operating flow is illustrated in Figure 37.23.

Table 37.9 Manual-command configuration for direct mode (1 of 2)

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
1st Transaction	CDTBUF0. TRTYPE	CDTBUF0. CMD[15:0]	CDTBUF0. CMDSIZE[1:0]	CDABUF0. ADD[31:0]	CDTBUF0. ADDSIZE[2:0]	CDDxBUF0. DATA[31:0]	CDTBUF0. DATASIZE[3:0]	CDTBUF0. LATE[4:0]
2nd Transaction	CDTBUF1. TRTYPE	CDTBUF1. CMD[15:0]	CDTBUF1. CMDSIZE[1:0]	CDABUF1. ADD[31:0]	CDTBUF1. ADDSIZE[2:0]	CDDxBUF1. DATA[31:0]	CDTBUF1. DATASIZE[3:0]	CDTBUF1. LATE[4:0]

Table 37.9 Manual-command configuration for direct mode (2 of 2)

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
3rd Transaction	CDTBUF2.TRTYPE	CDTBUF2.CMD[15:0]	CDTBUF2.CMDSIZE[1:0]	CDABUF2.ADD[31:0]	CDTBUF2.ADDSIZE[2:0]	CDDxBUF2.DATA[31:0]	CDTBUF2.DATASIZE[3:0]	CDTBUF2.LATE[4:0]
4th Transaction	CDTBUF3.TRTYPE	CDTBUF3.CMD[15:0]	CDTBUF3.CMDSIZE[1:0]	CDABUF3.ADD[31:0]	CDTBUF3.ADDSIZE[2:0]	CDDxBUF3.DATA[31:0]	CDTBUF3.DATASIZE[3:0]	CDTBUF3.LATE[4:0]

37.4.2.2 Periodic Mode

This mode periodically issues an xSPI read transaction configured and requested by Software. And it can compare the read value up to 4 bytes with expected value. The transaction is issued by a transaction request (CDCTL0.TRREQ = 1 with PERMD = 1). It can be used to alternate the status polling operation of xSPI slave device.

The periodic term is configured in Periodic transaction interval bits (CDCTL0.PERITV[4:0]). The expected value is configured in Periodic transaction expected and masked value bits (CDCTL1.PEREXP[31:0] and CDCTL2.PERMSK[31:0]). Table 37.10 lists the configured register bits for periodic manual-command. The operating flow is illustrated in Figure 37.24.

Table 37.10 Manual-command configuration for periodic mode

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
Read Transaction	CDTBUF0.TRTYPE	CDTBUF0.CMD[15:0]	CDTBUF0.CMDSIZE[1:0]	CDABUF0.ADD[31:0]	CDTBUF0.ADDSIZE[2:0]	CDDxBUF0.DATA[31:0]	CDTBUF0.DATASIZE[3:0]	CDTBUF0.LATE[4:0]

Note: In this mode, it is possible to repeat multiple commands by configuring multiple buffers. In the case, only the last command shall be Read transaction and the others shall be Write transaction. The read data of last command is compared with expected value.

37.4.3 Memory-mapping

This section describes the memory-mapping mode. This mode automatically converts AXI access for pre-configured memory area into xSPI transaction.

37.4.3.1 Configuration

In this operation, the payload of address and data field are delivered from AXI signals. The information of command field and size are delivered from the configured register bits. Table 37.11 lists the register bits configured for memory-mapping.

Table 37.11 Memory-mapping configuration for memory area access (n = 0, 1) (1 of 2)

AXI Transaction	Format Change mode	Command	Command size	Address size	Latency cycle
Write for slave n memory area	Normal	CMCFG2CSn.WRCMD[15:8]	1 byte	CMCFG0CSn.ADDSIZE[1:0]	CMCFG2CSn.WRLATE[4:0]
	8D-8D-8D profile 1.0	CMCFG2CSn.WRCMD[15:0]	2 bytes		
	8D-8D-8D profile 2.0 Command Modifier	CMCFG2CSn.WRCMD[15:8]	1 byte	5 bytes	
	8D-8D-8D profile 2.0 Extended Command Modifier	CMCFG2CSn.WRCMD[15:13]	3 bits	45 bits	

Table 37.11 Memory-mapping configuration for memory area access (n = 0, 1) (2 of 2)

AXI Transaction	Format Change mode	Command	Command size	Address size	Latency cycle
Read for slave n memory area	Normal	CMCFG1CSn. RDCMD[15:8]	1 byte	CMCFG0CSn. ADDSIZE[1:0]	CMCFG1CSn. RDLATE[4:0]
	8D-8D-8D profile 1.0	CMCFG1CSn. RDCMD[15:0]	2 bytes		
	8D-8D-8D profile 2.0 Command Modifier	CMCFG1CSn. RDCMD[15:8]	1 byte	5 bytes	
	8D-8D-8D profile 2.0 Extended Command Modifier	CMCFG1CSn. RDCMD[15:13]	3 bits	45 bits	

Note: The MSByte of Address can be replaced with Address Replace Enable and Code bits (CMCFG0CSn.ADDRPEN[7:0] / ADDRPCD[7:0]).

37.4.3.2 Write Access Operation

In AXI write access for memory area, this xSPI Master stores all payload data in internal bridge buffer and then issues a write transaction into xSPI slave. Figure 37.10 shows the operation summary.

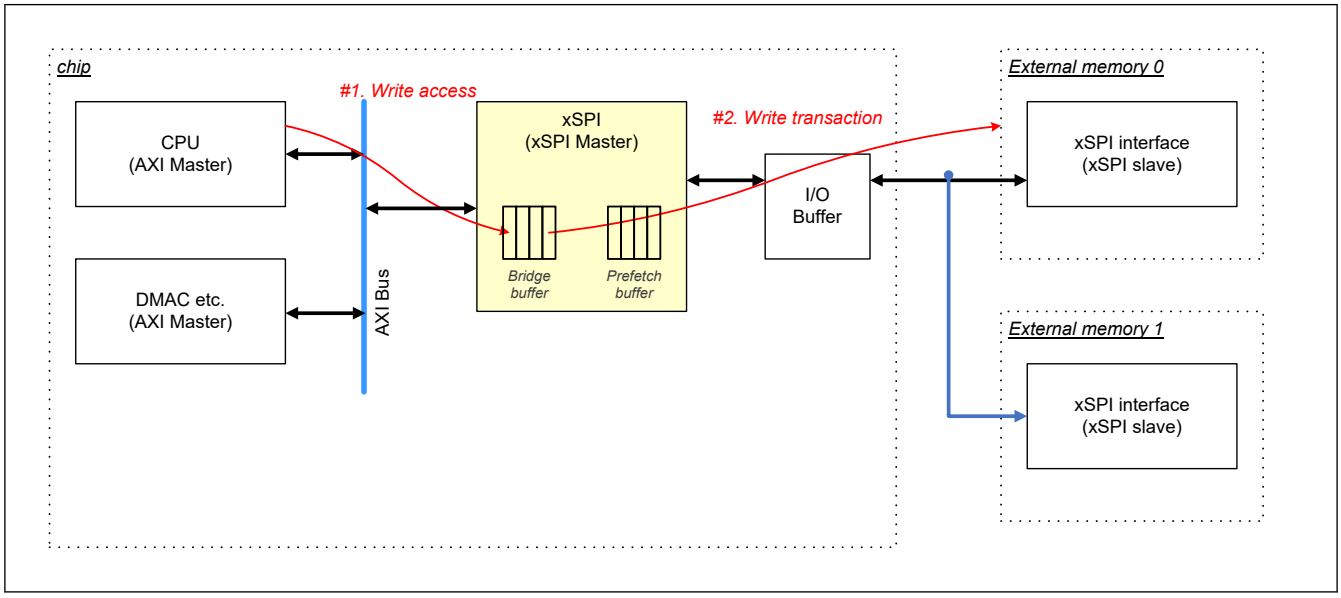


Figure 37.10 Write access for memory area

The operation of xSPI bus changes depending on AXI burst type. When the burst type is single type or incremental type, one AXI transaction triggers for one xSPI frame. When the burst type is wrap type and the CMCFG0CSx.WPBSTMD is 0, one AXI transaction triggers two xSPI frames. Figure 37.11 shows the relationship between AXI and xSPI frame.

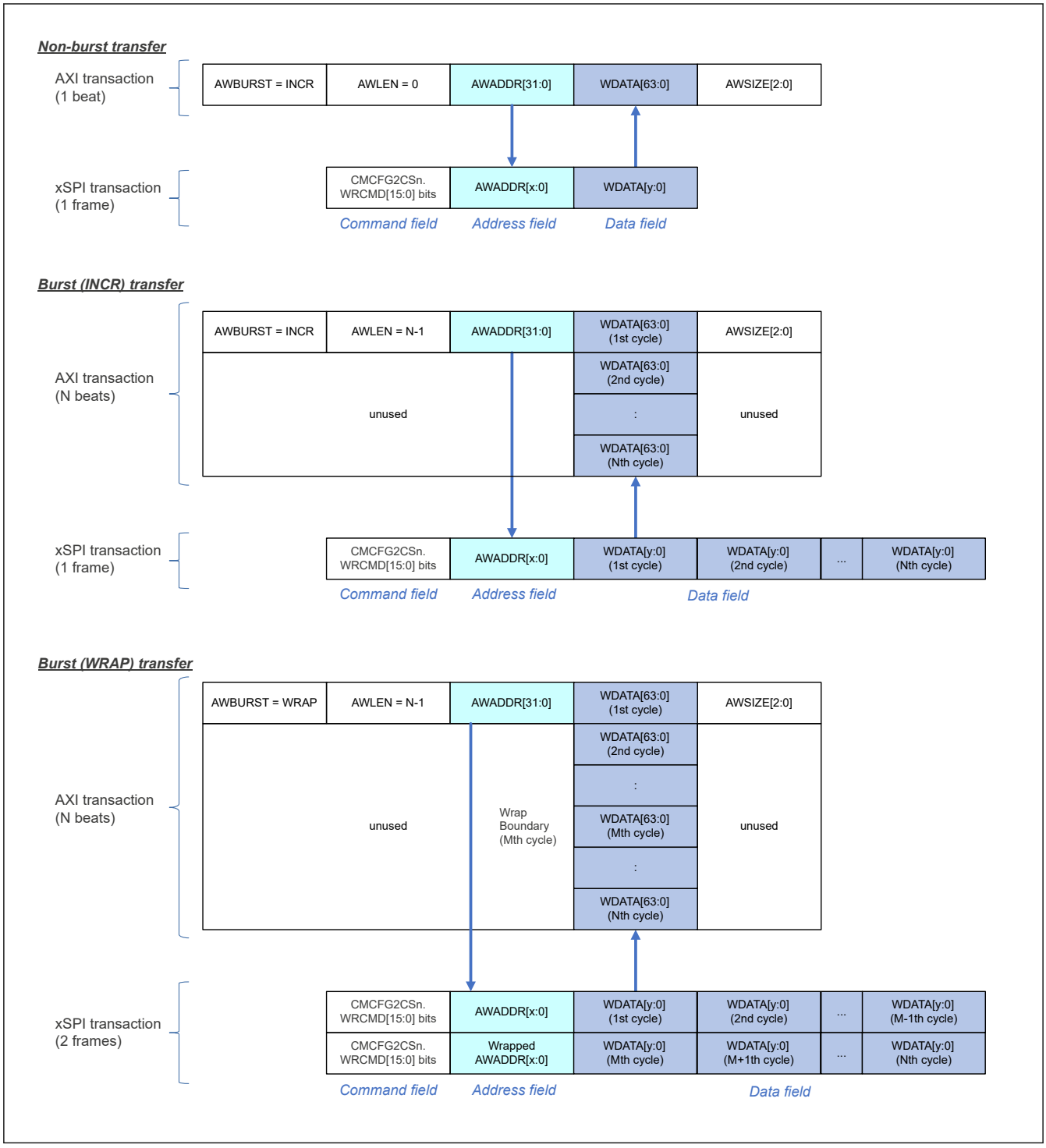


Figure 37.11 xSPI frame format in write access (Normal format)

37.4.3.3 Combination Function

In AXI write access for memory area, this xSPI master has the function to combinate the write data for high throughput on xSPI bus. When enabled this function (BMCFG.MWRCOMB = 1), this xSPI master transmits an xSPI frame with the selected size while the sequential address is incremental[*1]. When one of the below conditions is detected, even though not reaching to the target size (BMCFG.MWRSIZE), this xSPI master transmits the pending data into xSPI bus.

- Non-incremental address is detected.
- Different burst type is detected.
- Read transaction is detected.

- Access for different slave is detected.
- Memory Write Data Push bit (BMCTL1.MWRPUSHCHx) is set.

Note 1. The access which comply all below condition is considered as incremental address.
- Transaction type is INCR.
- The access's start address is continuous to the previous last write address.
  - The access's start position of WSTRB is treated as start address.
  - Previous access's last WSTRB is treated last write address.

This function could be useful for any slave device to request a chunk of data at a time. In the case, AXI Master should continue to provide the fixed data size with incremental address. Ex. there is any device to request to write in page unit.

Note that combination function should not be used in case of programming to Serial Flash since write enable command is basically need before data programming and it may cause the unexpected order of the transaction.

Figure 37.12 shows the operation when enable the combination function.

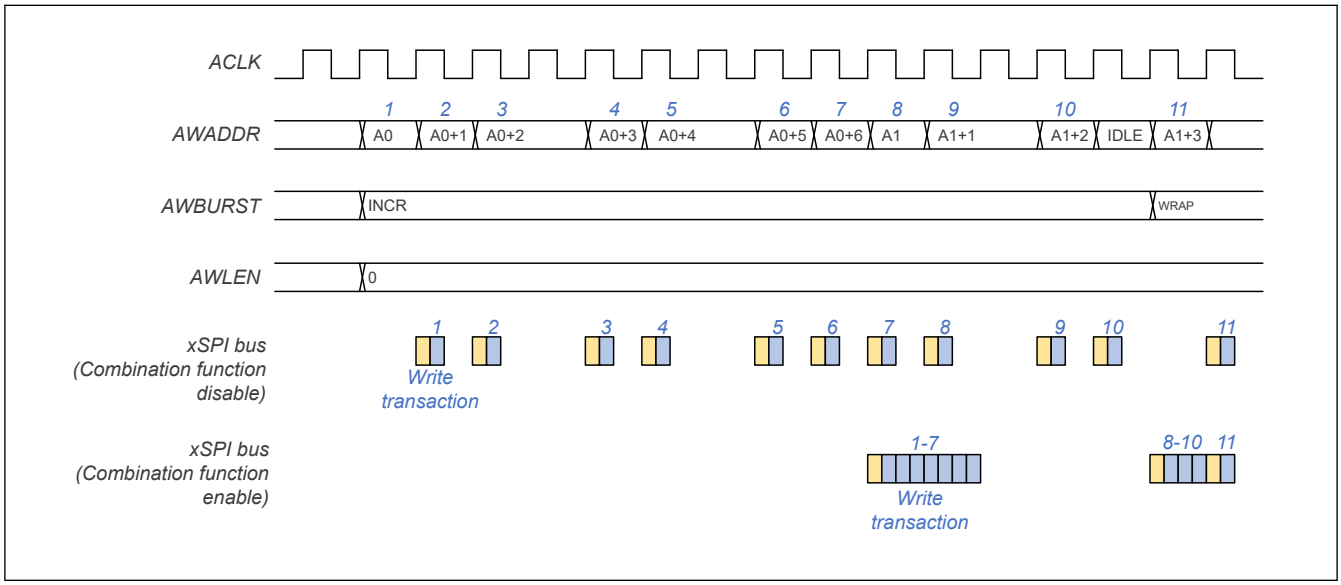


Figure 37.12 Combination function

Figure 37.13 shows data combined example with AXI access.

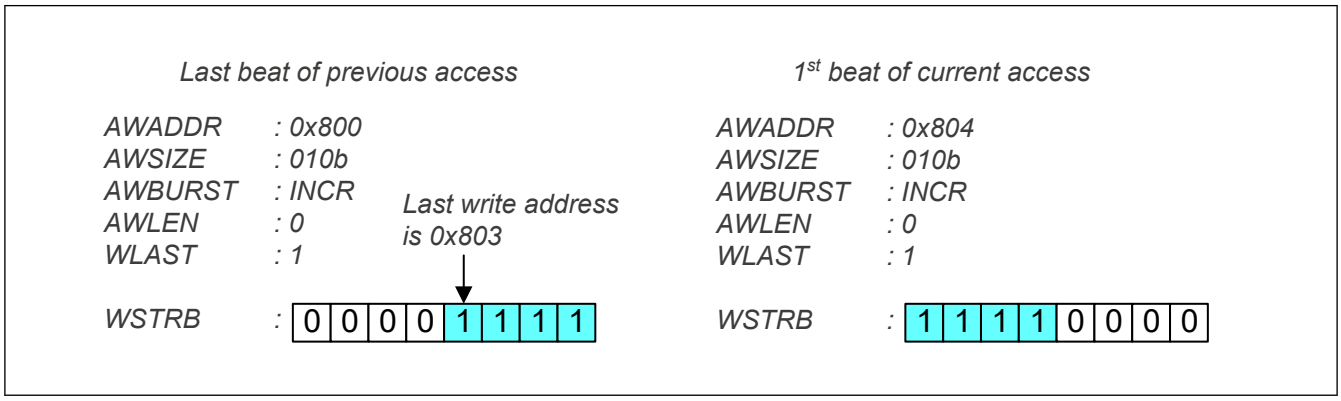


Figure 37.13 Data combined example with AXI access

37.4.3.4 Read Access Operation

In AXI read access for memory area, soon after detected the read access, this xSPI Master issues a read transaction into xSPI slave. Figure 37.14 shows the operation summary.

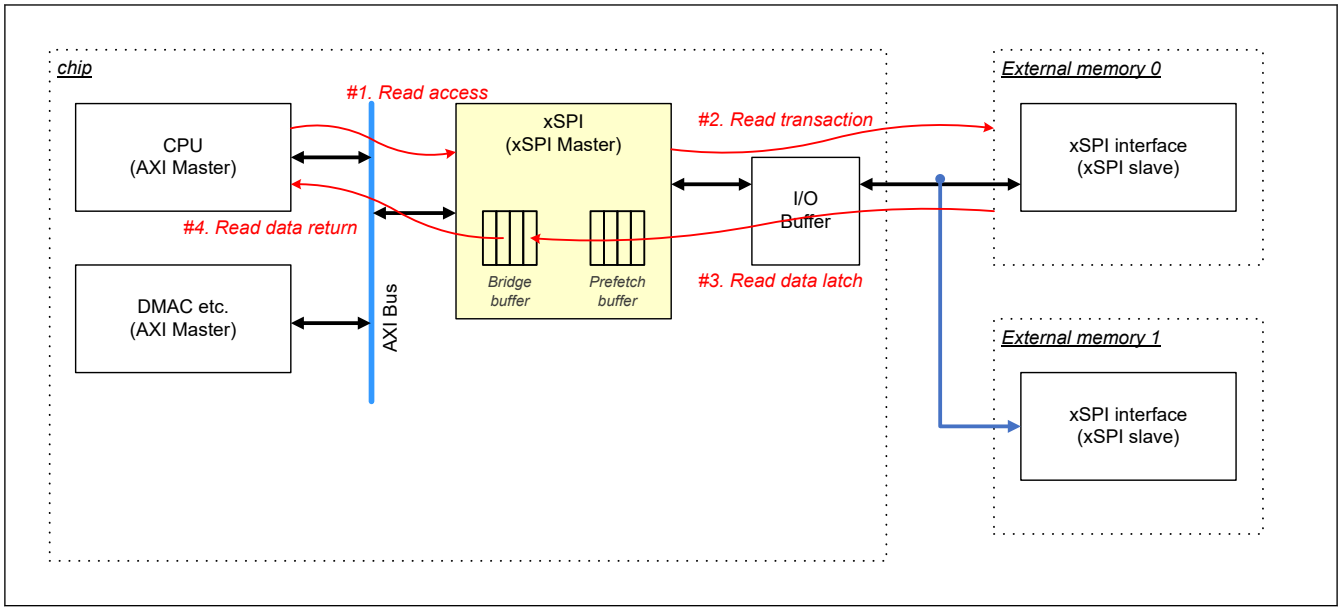


Figure 37.14 Read access for memory area

The operation of xSPI bus changes depending on burst type. When the type is single or increment type, one AXI read transaction triggers one xSPI frame. When the type is wrap type and the CMCFG0CSx.WPBSTMD is 0, one AXI read transactions triggers two xSPI frame. Figure 37.15 shows the relationship between AXI and xSPI frame.

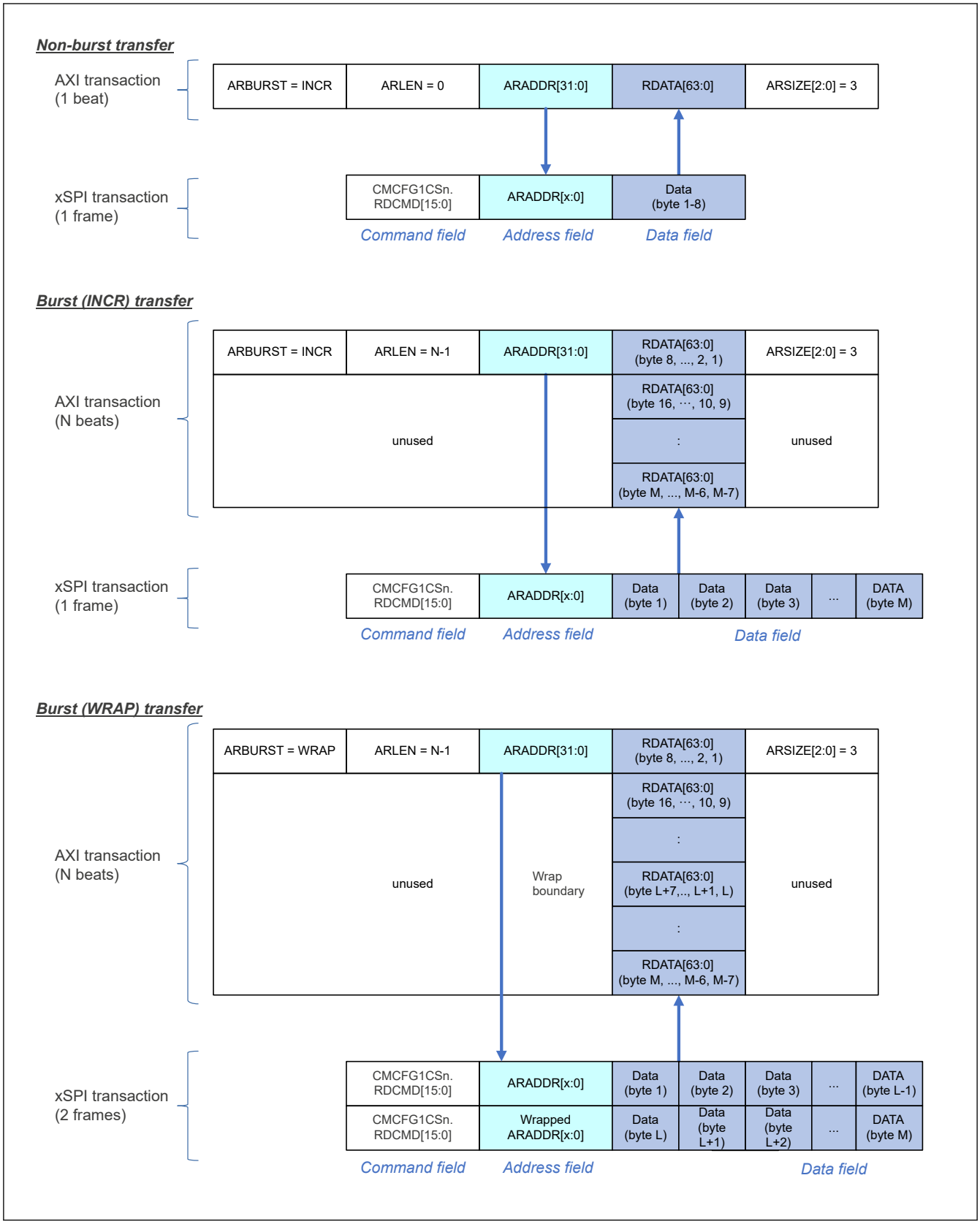


Figure 37.15 xSPI frame format in read access (Normal format)

37.4.3.5 Prefetch Function

In AXI read access for memory area, this xSPI Master has the function to prefetch the read data for lower latency. When enabled this function (BMCFG.PREEN = 1), this xSPI Master continues to read the incremental address and store the read

data from xSPI slave in internal prefetch buffer. And the following AXI read access searches for prefetch buffer. If found the target read data in prefetch buffer, this xSPI Master returns the data from prefetch buffer. If not found, this xSPI Master clears prefetch buffer and newly issues a read transaction into xSPI slave. This function is effective in application such as the consecutive read addresses are close. But it is not effective in application such as the consecutive read addresses are not incremental because xSPI read frame for prefetch uses xSPI bus. Figure 37.16 shows the operation summary.

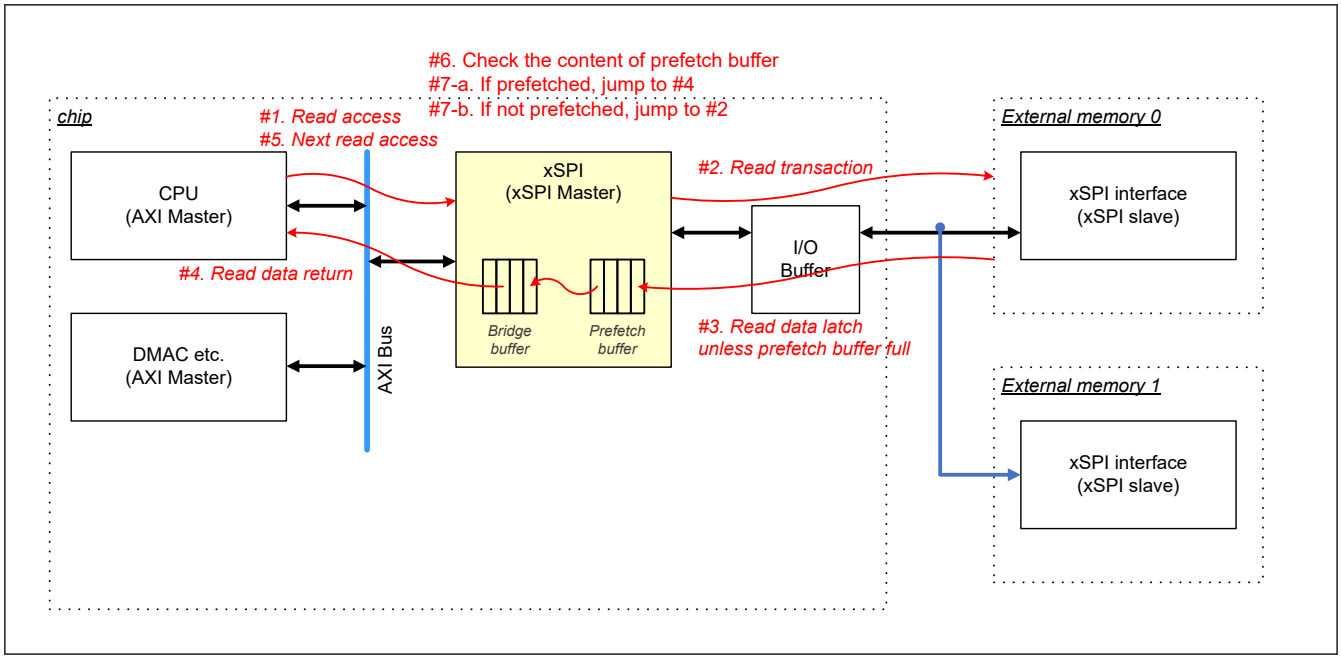


Figure 37.16 Read access for memory area with prefetch enabled

Note: When enabled this prefetch function, AXI Master could read from not a slave device but the internal prefetch buffer. When accessed to the same address from multiple AXI Masters, this xSPI Master does not guarantee to read the latest data. If AXI Master wish to read the latest data from a slave device, it should read after cleared the prefetch buffer (BMCTL1.PBUFCLR).

Note: Prefetch buffer is implemented as FIFO-based, and when read access is issued, the data before the access address is discarded from the buffer. And when next access is issued to the region which is discarded at previous access, this module issues the xSPI read access again to fill the prefetch buffer.

37.4.3.6 XiP Mode

Some slave devices have a mode (XiP mode) in which the command phase is not required for lower latency. While in this mode, the xSPI master skips sending the command and the slave device implicitly performs the command that was executed in the previous transaction. When enabled XiP mode bit (CMCTL.XIPEN = 1), this xSPI master inserts XiP enter code (CMCTL.XIPENCODE[7:0]) in latency field. When disabled XiP mode bit (CMCTL.XIPEN = 0), this xSPI master inserts XiP exit code (CMCTL.XIPEXCODE[7:0]) in latency field. This function is available only for memory-mapping mode.

And when transmitting XiP disable pattern, this xSPI master clears XiP mode bit and disables XiP mode.

Note: In case of no cycle inserting XiP code in latency field, this xSPI Master could not insert XiP code.

Note: XiP mode could be used only for unidirectional access to a slave. The write transaction and read transaction should be separated.

37.4.4 Pattern Control

This xSPI Master has the function to transmit any pattern which is not xSPI frame format. It starts to transmit by the pattern issue trigger bits (LPCTL0-1.PATREQ). It is possible to transmit 3 types of patterns.

37.4.4.1 XiP Disable Pattern

XiP Disable pattern transmits any pattern with the configured length and value (LPCTL0.XD1LEN[4:0] / XD1VAL / XD2LEN[4:0] / XD2VAL). It uses spi_ck, spi_doe/do signals. The number of output pin can be configured by XiP Disable

pattern pin bits (LPCTL0.XDPIN[1:0]). It may be used to disable XiP mode for legacy SPI. Figure 37.17 shows the timing-chart.

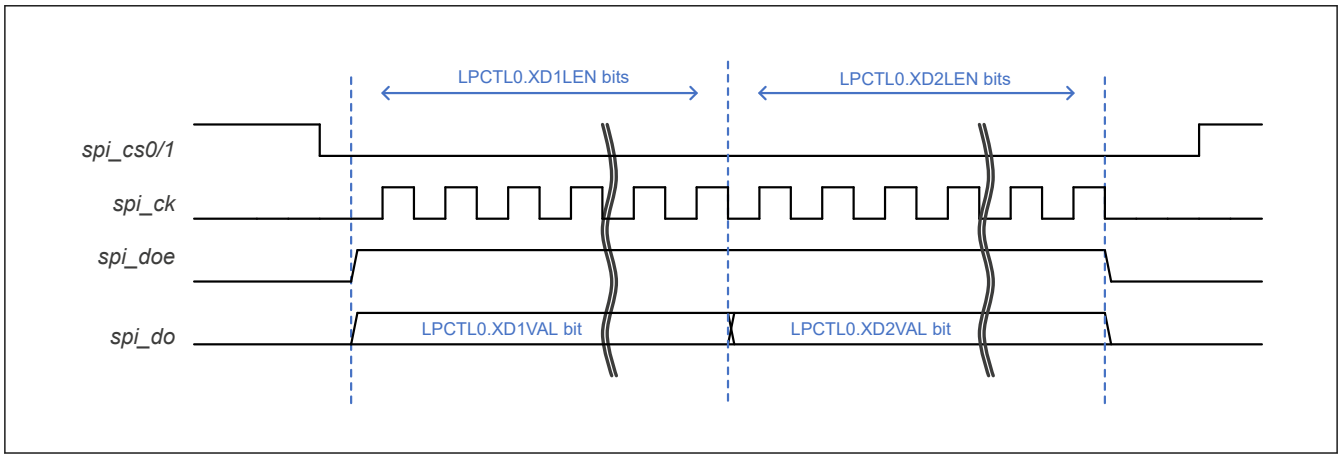


Figure 37.17 XiP Disable pattern

37.4.4.2 Reset Pattern

Reset pattern transmits the pattern specified in Serial Flash Reset Signaling Protocol. CS Low/High width is configured with Reset Pattern Length bits (LPCTL1.RSTWID[2:0]). xSPI slave will sample the data input at the rising edge of CS. Setup time for data output is configured with Reset pattern data output setup time bits (LPCTL1.RSTSU[2:0]). The setup time should be less than Reset pattern width always. Figure 37.18 shows the timing-chart.

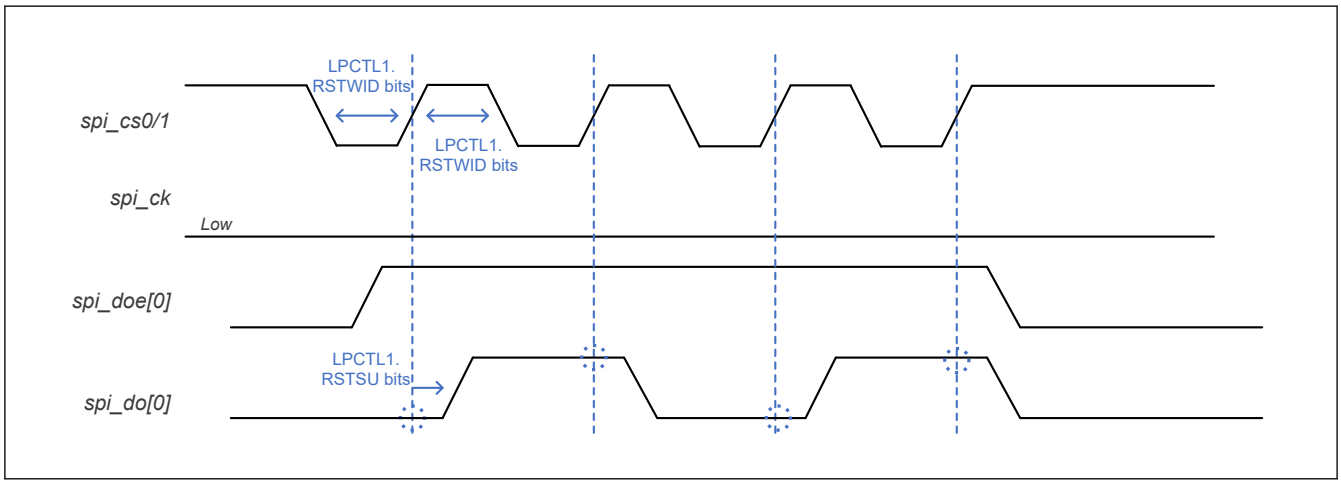


Figure 37.18 Reset pattern

Note: In the protocol, CS Low/High width is defined as minimum 500 ns and Setup time is defined as minimum 6 ns.

37.4.4.3 CS Only Pattern

CS Only pattern activates CS port with the configured length bits (LPCTL1.RSTWID[2:0]). It may be used to resume from Deep Power Down state. Figure 37.19 shows the timing-chart.

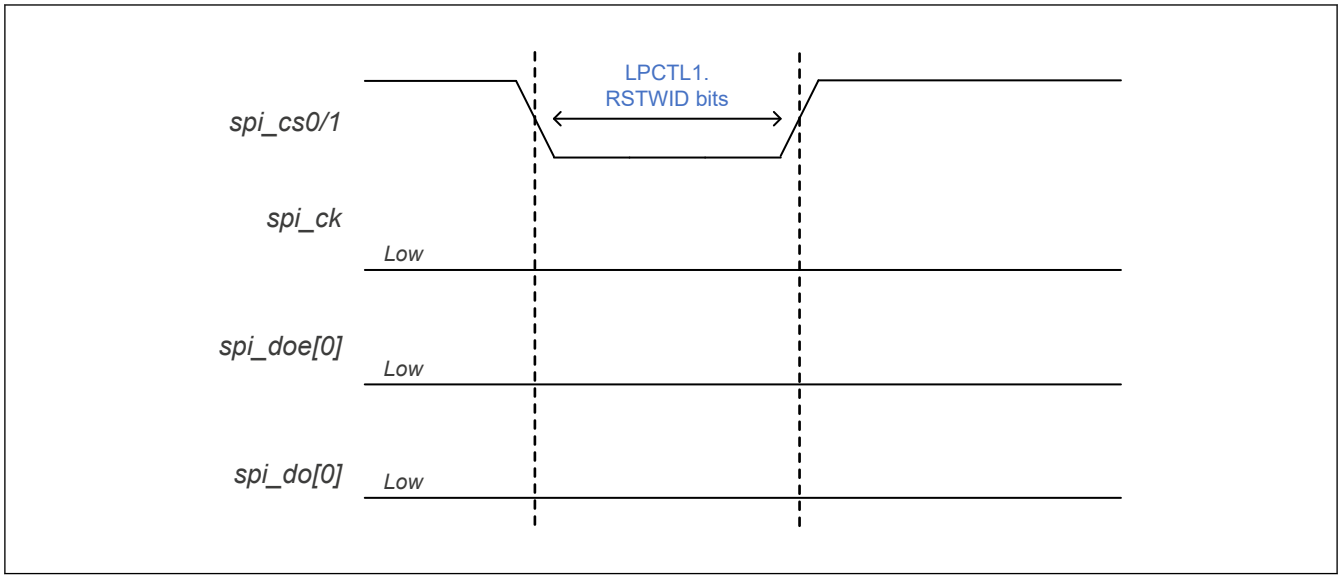


Figure 37.19 CS Only pattern

37.4.5 Integrity Checking

This xSPI Master can detect some errors. Table 37.12 lists the error list and the detail behavior.

When reset of xSPI master is needed due to fatal error, set module reset register (MRCTLA) to reset xSPI master. After confirming it is reset surely, release module reset.

Table 37.12 Error list

Error type	Event	Flag bit	Note (action)
Calibration failed	When the read data did not match the expected value during automatic calibration	INTS. CAFAILCS0-1	It results in writing unexpected data to xSPI slave.
AXI bus error	When an error response occurred on AXI slave interface for memory-mapping	INTS. BUSERR	This is a fatal error. This xSPI master shall be initialized.
ECC error detection	When detected the falling edge on ECSn# port It can be useful only for xSPI slave with ECC detection function.	INTS. ECSCS0-1	Only notify the error event of xSPI slave.
DS timeout	When DS does not toggle in read transaction with using DS	INTS. DSTOCS0-1	Both xSPI master and xSPI slave should be reset for fatal error.
Periodic transaction timeout	When the read value does not match with the expected value in periodic manual-command mode	INTS. PERTO	Depending on the status of function

37.4.6 Interrupts

This xSPI Master has an interrupt port.

It can monitor with Interrupt Status Register (INTS). In case of initialization phase, it can be programmable with Interrupt Enable register (INTE). Table 37.13 lists the related register bit.

Table 37.13 xSPI interrupt sources

Name	Interrupt Sources	CPU0 GIC Request	CPU1 GIC Request	DMAC Activation
XSPIm_INT	Interrupt	Possible	Possible	Not possible
XSPIm_INTERR	Error interrupt	Possible	Possible	Not possible

Note: m = 0, 1

Table 37.14 Interrupt register bit

Flag bit	Enable bit	Clear bit	Interrupt Sources (m = 0, 1)
CASUCCS1	CASUCCS1E	CASUCCS1C	XSPIm_INT
CASUCCS0	CASUCCS0E	CASUCCS0C	
CAFAILCS1	CAFAILCS1E	CAFAILCS1C	XSPIm_INTERR
CAFAILCS0	CAFAILCS0E	CAFAILCS0C	
BUSERR	BUSERRE	BUSERRC	XSPIm_INTERR
INTCS1	INTCS1E	INTCS1C	XSPIm_INTERR
INTCS0	INTCS0E	INTCS0C	
ECSCS1	ECSCS1E	ECSCS1C	XSPIm_INTERR
ECSCS0	ECSCS0E	ECSCS0C	
DSTOCS1	DSTOCS1E	DSTOCS1C	XSPIm_INTERR
DSTOCS0	DSTOCS0E	DSTOCS0C	
PERTO	PERTOE	PERTOC	XSPIm_INTERR
INICMP	INICMPE	INICMPC	XSPIm_INT
PATCMP	PATCMPE	PATCMPC	XSPIm_INT
CMDCMP	CMDCMPE	CMDCMPC	XSPIm_INT

37.4.7 Flows of Operations

37.4.7.1 Flow of Configuration

Figure 37.20 shows flow of configuration.

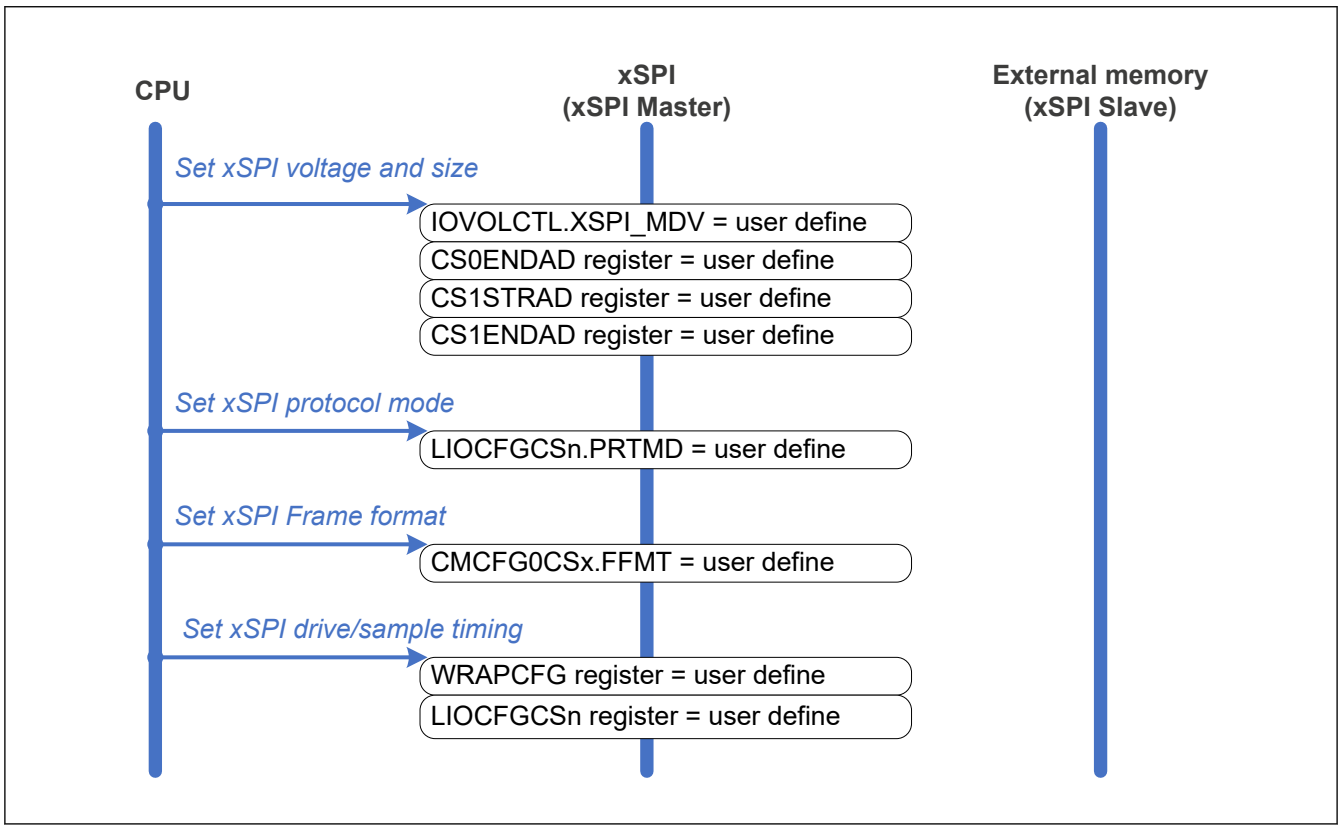


Figure 37.20 Flow of configuration

37.4.7.2 Flow of Communication Stop

Figure 37.21 shows flow of communication stop.

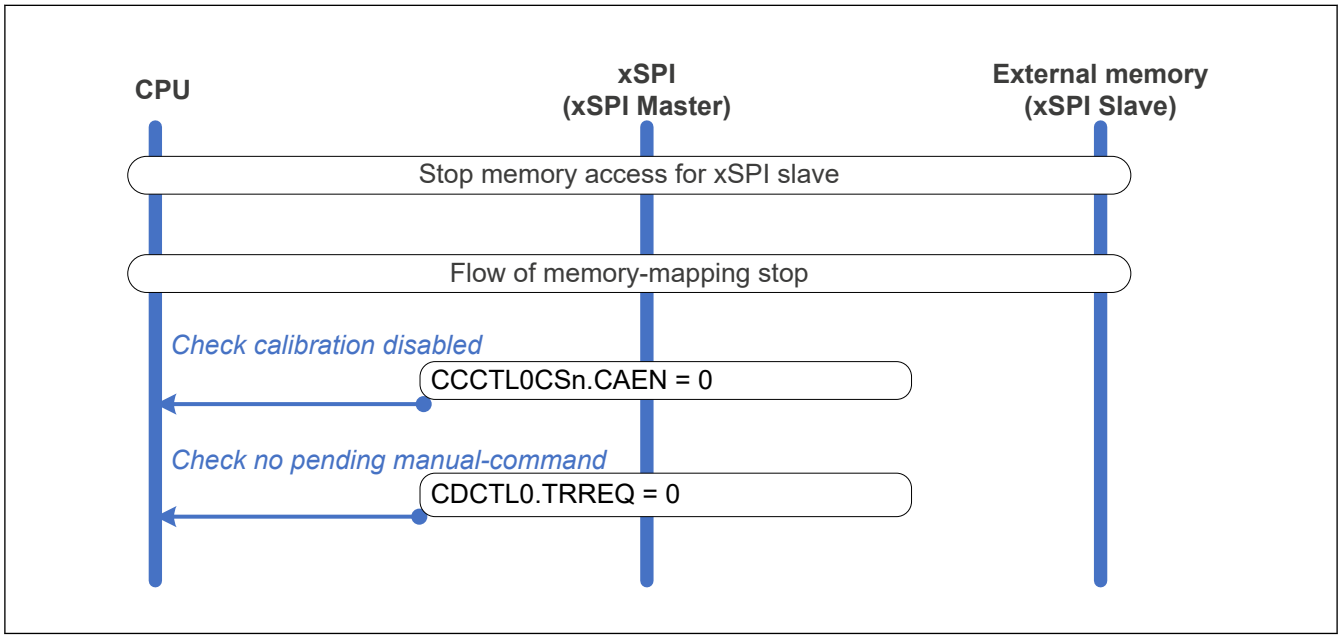


Figure 37.21 Flow of communication stop

Note: In case of re-config of any configuration register, all communication with xSPI slave shall be stopped completely. It means that the automatic calibration is disabled, and there is no pending manual-command and memory-mapping access.

37.4.7.3 Flow of Automatic Calibration

Figure 37.22 shows flow of automatic calibration.

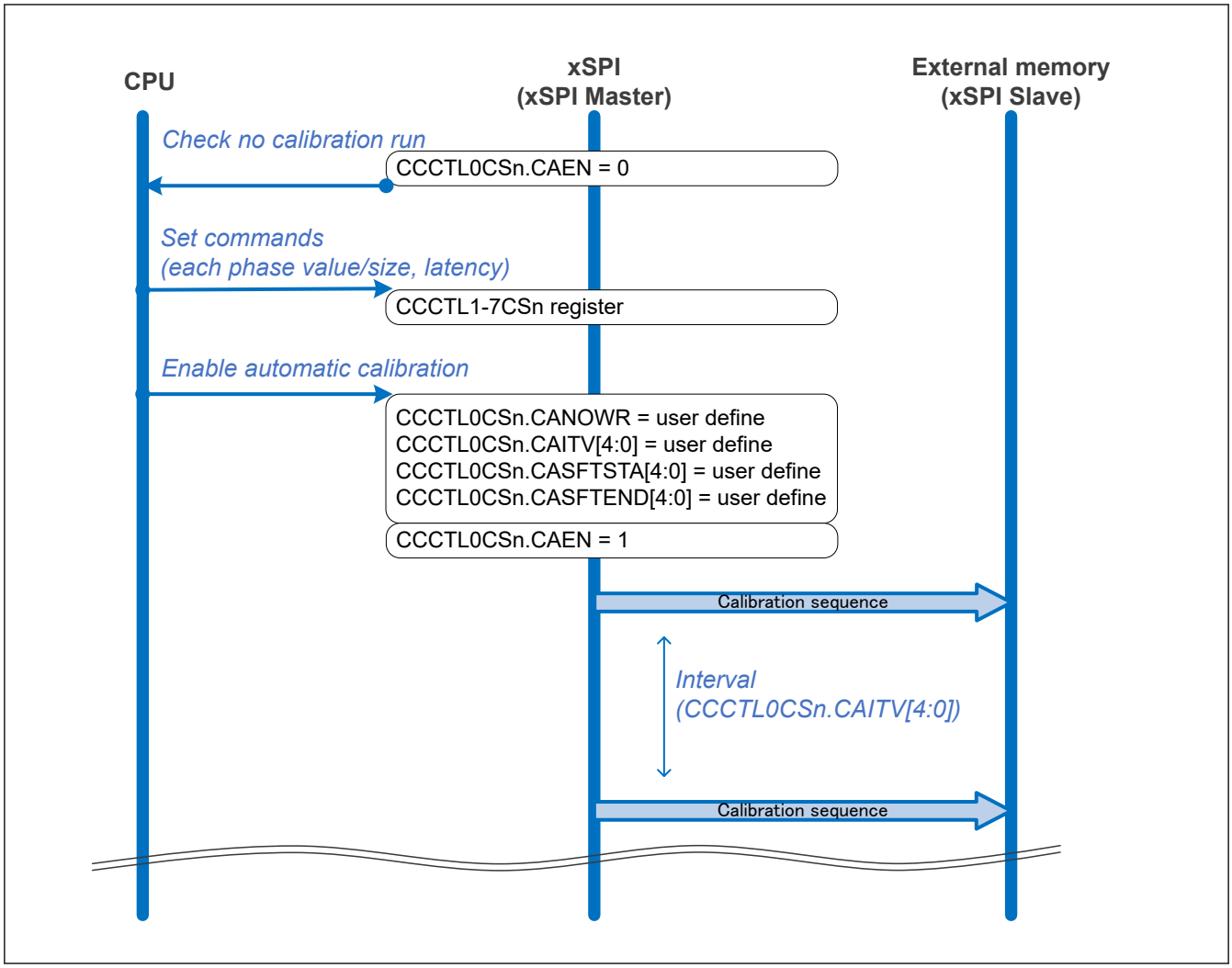


Figure 37.22 Flow of automatic calibration

37.4.7.4 Flow of Manual-command Procedure

Figure 37.23 shows manual-command procedure for direct mode.

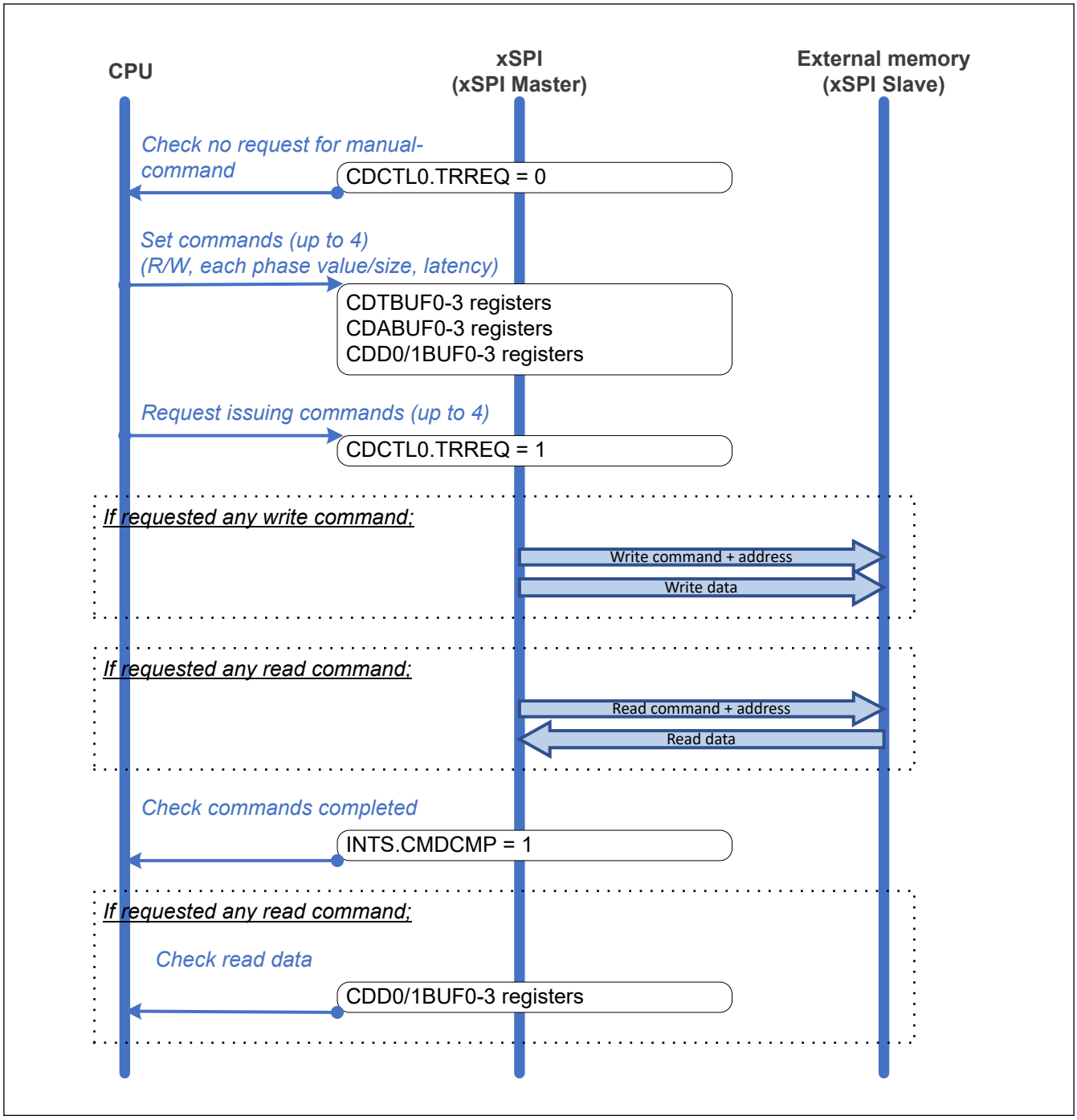


Figure 37.23 Flow of manual-command procedure for direct mode

Figure 37.24 shows manual-command procedure for periodic mode.

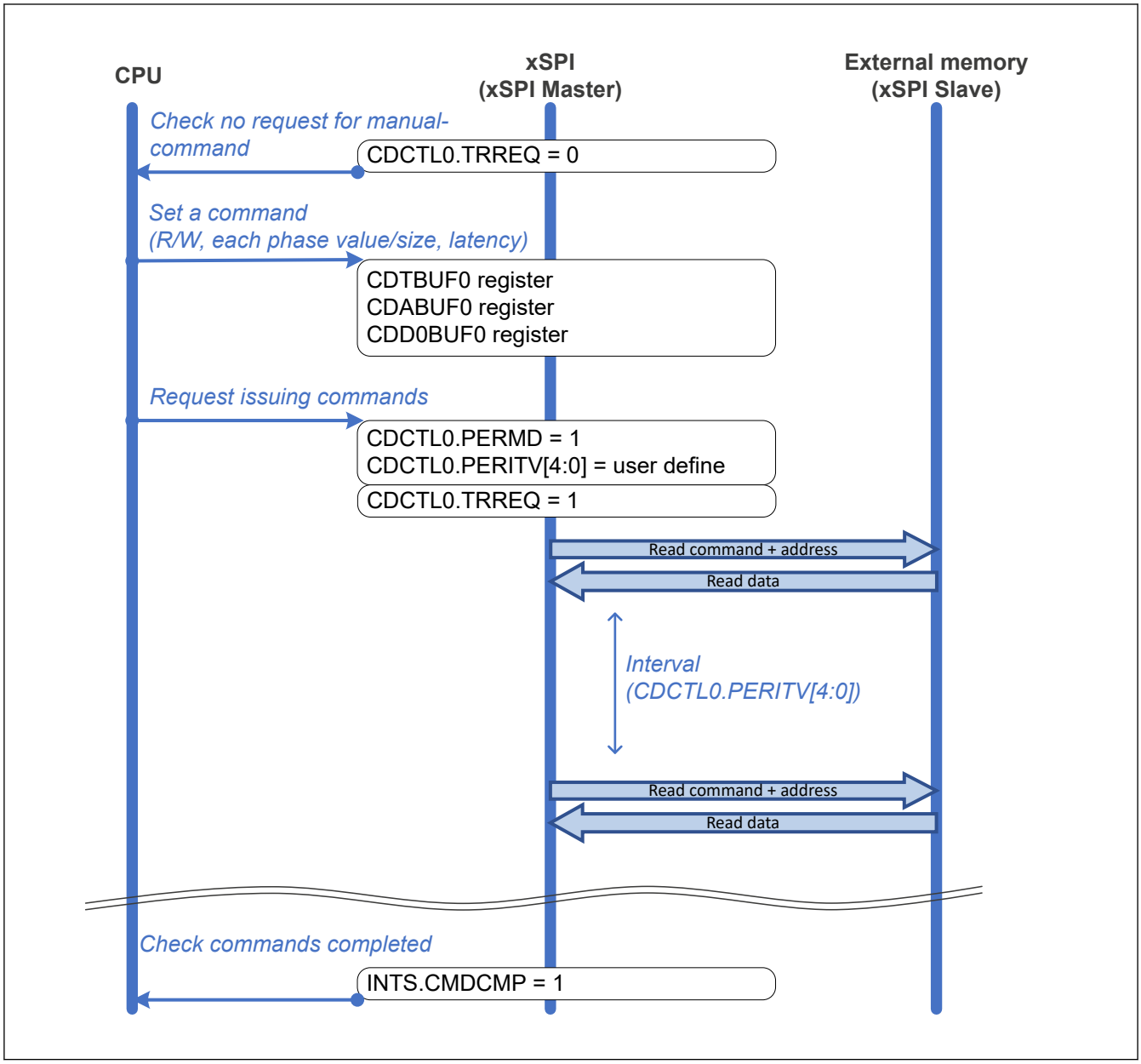


Figure 37.24 Flow of manual-command procedure for periodic mode

37.4.7.5 Flow of Memory-mapping

Figure 37.25 shows flow of memory-mapping.

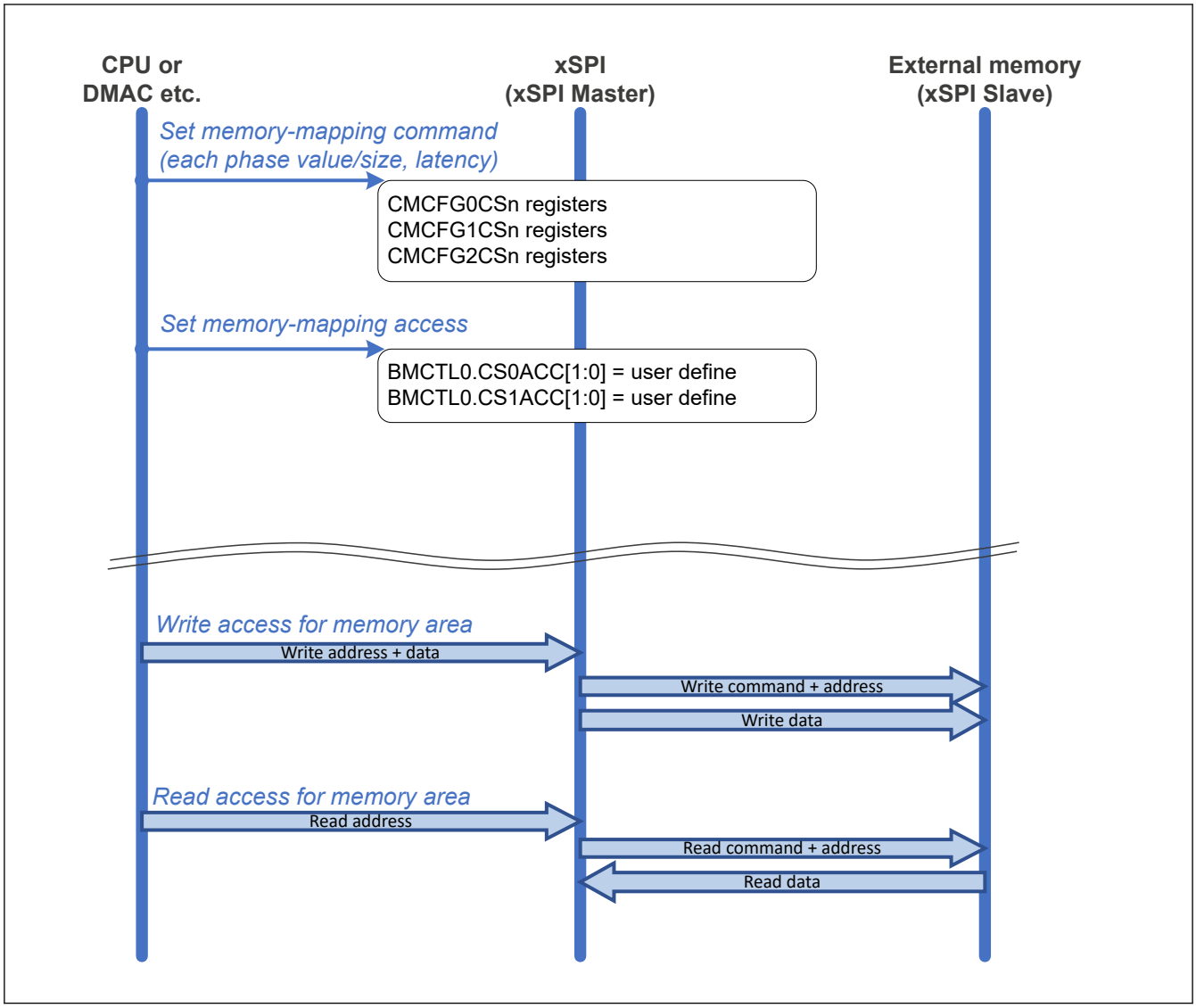


Figure 37.25 Flow of memory-mapping

37.4.7.6 Flow of Memory-mapping Stop

Figure 37.26 shows flow of memory-mapping stop.

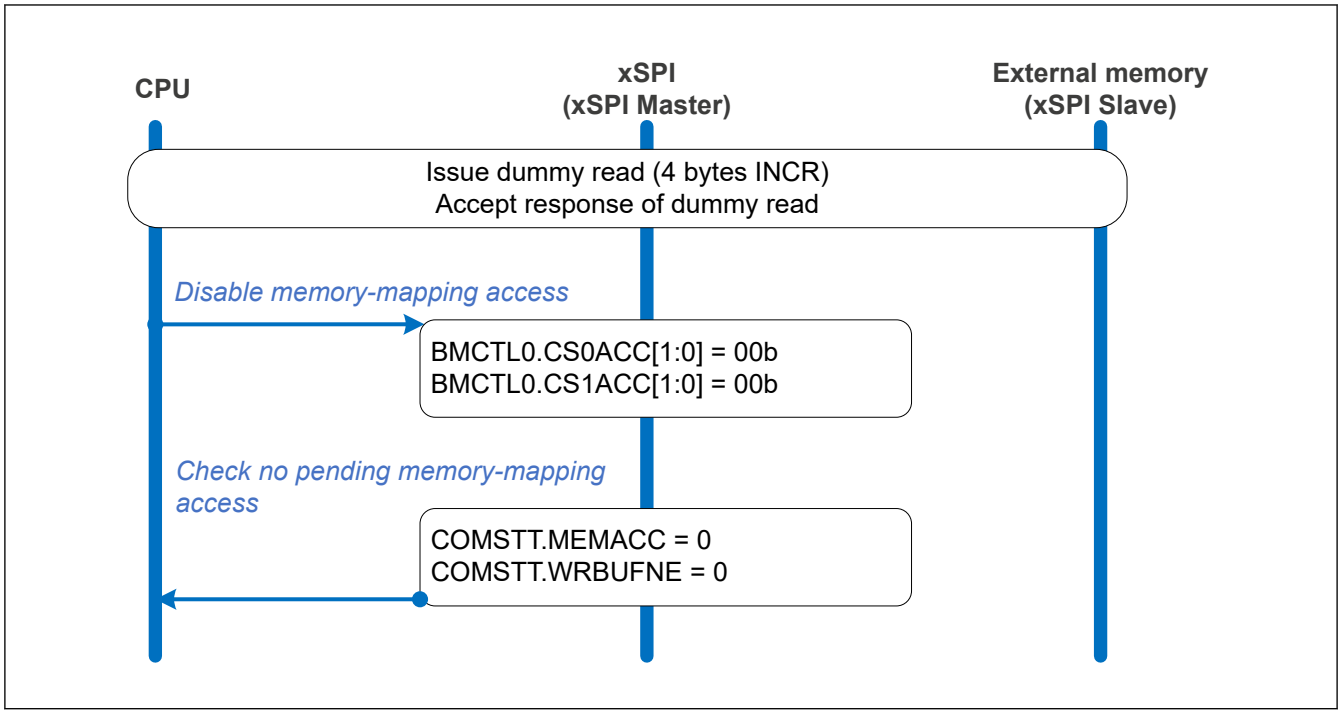


Figure 37.26 Flow of memory-mapping stop

37.4.7.7 Flow of Pattern Request

Figure 37.27 shows flow of pattern request. Before requesting any pattern, any ongoing commands should be completed or canceled.

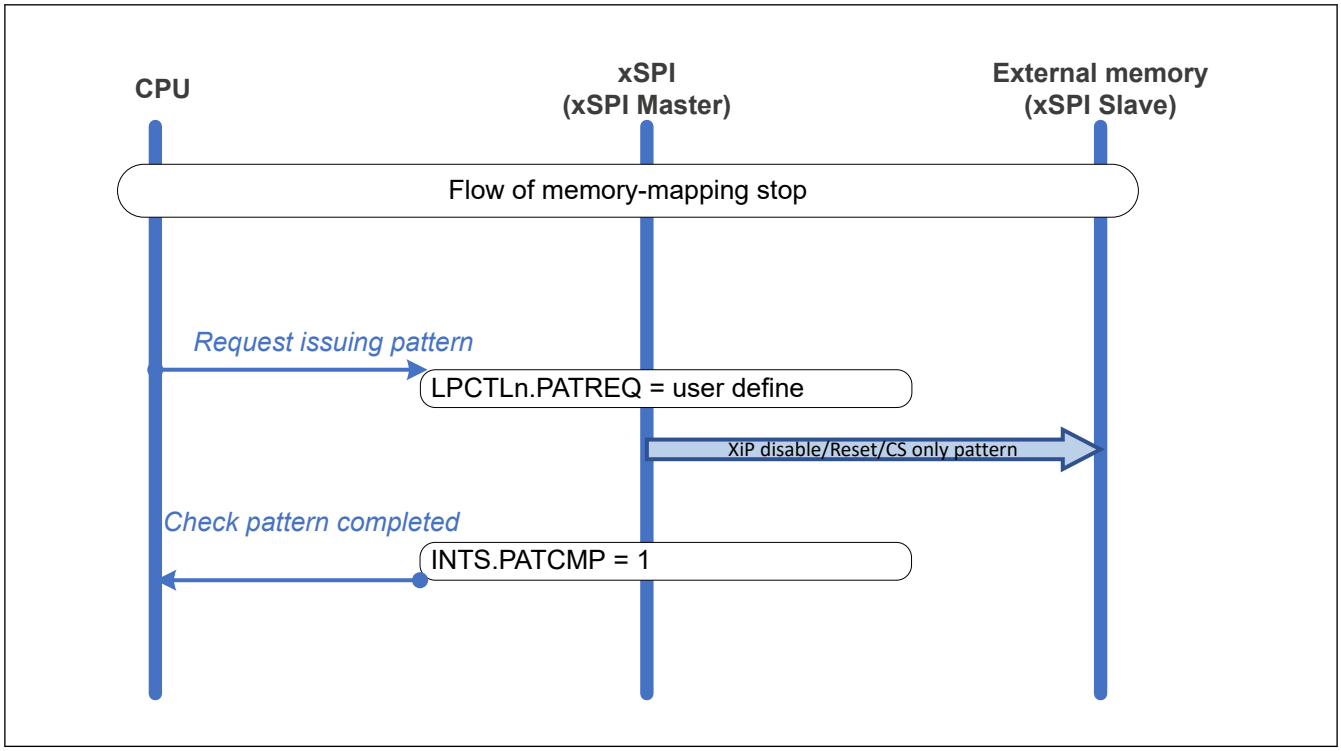


Figure 37.27 Flow of pattern request

37.4.7.8 Flow of XiP Mode

Figure 37.28 shows flow of XiP mode enable/disable.

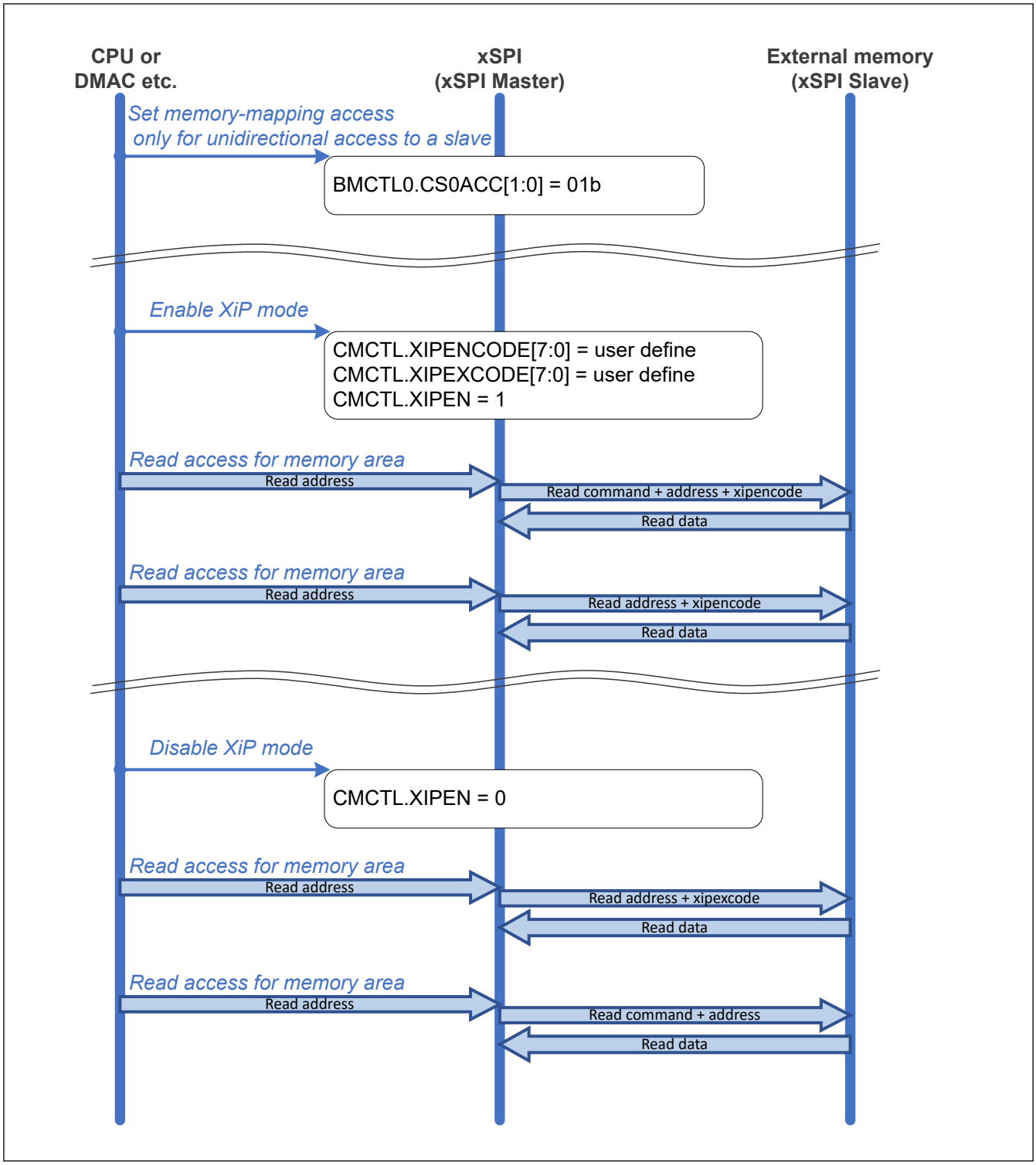


Figure 37.28 Flow of XiP mode enable/disable

37.4.8 Usage Notes

- If prefetch function is enabled and xSPI CS1 Start Address Register and CSn End Address Register (CS1STRAD and CSnENDAD) is not set correctly compared to size of connected slave device, xSPI master may access outside of the slave device address. It may cause DS timeout in case of 8D-8D-8D. Make sure to set the adequate start address and end address to CS1STRAD and CSnENDAD registers considering the size of the connected slave device.
- DDR access without DS is not supported.

38. CRC Operation Unit (CRC)

38.1 Overview

Cyclic redundancy check (CRC) operation units generate CRC codes. Table 38.1 lists the specifications of the CRC operation unit, and Figure 38.1 shows a block diagram of the CRC operation unit.

Table 38.1 CRC specifications

Parameter	Description	
Number of channels	2 channels	
Data size	8 bits	32 bits
Data for CRC calculation*1	CRC codes are generated for any desired data in 8n-bit units (where n is a natural number)	CRC codes are generated for any desired data in 32n-bit units (where n is a natural number)
CRC processor unit	8-bit parallel processing	32-bit parallel processing
CRC generating polynomial	One of three generating polynomials is selectable • 8-bit CRC $X^8 + X^2 + X + 1$ (CRC-8) • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ (CRC-16) $X^{16} + X^{12} + X^5 + 1$ (CRC-CCITT)	One of two generating polynomials is selectable • 32-bit CRC $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (CRC-32) $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ (CRC-32C)
CRC calculation switching	The order of the bits produced by CRC calculation can be switched for LSB first or MSB first communication	
Low power consumption	Module-stop state can be set.	

Note 1. The circuit does not have a function to divide data for calculation into CRC calculation units. Write data in 8-bit or 32-bit units.

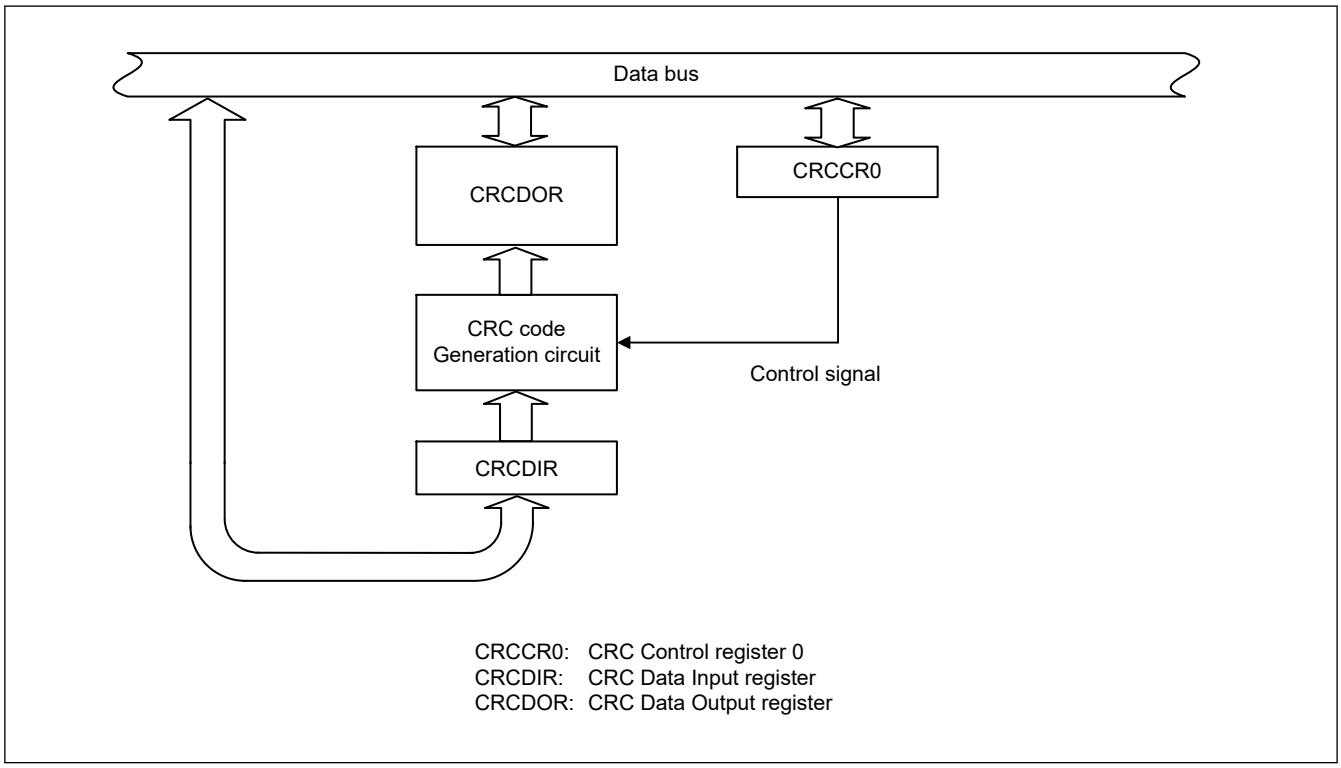


Figure 38.1 CRC block diagram

38.2 Register Map

Table 38.2 CRC register map

Address	Register symbol	Register name	Write protection
0x8000_4000 0x8100_4000	CRCCR0	CRC Control Register 0	—
0x8000_4004 0x8100_4004	CRCDIR/CRCDIR_BY	CRC Data Input Register	—
0x8000_4008 0x8100_4008	CRCDOR/CRCDOR_HA/CRCDOR_BY	CRC Data Output Register	—

Table 38.3 CRC related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	—	MSTPCRD.MSTPCRD09	SLVACCCTL1.CRC0_SL
1	—	MSTPCRG.MSTPCRG04	SLVACCCTL2.CRC1_SL

38.3 Register Descriptions

38.3.1 CRCCR0 : CRC Control Register 0

Base address: CRC0 = 0x8000_4000
CRC1 = 0x8100_4000

Offset address: 0x0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DORCLR	LMS	—	—	—	GPS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	GPS[2:0]	CRC Generating Polynomial Switching 0 0 0: No calculation is executed. 0 0 1: 8-bit CRC-8 ($X^8 + X^2 + X + 1$) 0 1 0: 16-bit CRC-16 ($X^{16} + X^{15} + X^2 + 1$) 0 1 1: 16-bit CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC-32 ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) 1 0 1: 32-bit CRC-32C ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) Others: No calculation is executed.	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	LMS	CRC Calculation Switching 0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
7	DORCLR	CRCDOR Register Clear 0: No effect 1: Clears the CRCDOR register	W

GPS[2:0] bits (CRC Generating Polynomial Switching)

Set these bits to select the CRC Generating Polynomial.

LMS bit (CRC Calculation Switching)

The setting this bit selects the order of the bits of generated CRC codes. The bit selects transmission of the lower-order byte of the CRC code first for LSB first communication, or the higher-order byte first for MSB first communication. For details on the transmission and reception of CRC codes, see section 38.4. Operation.

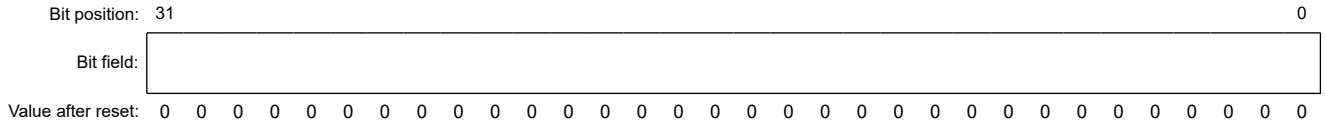
DORCLR bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0x00000000.

38.3.2 CRCDIR/CRCDIR_BY : CRC Data Input Register

Base address: CRC0 = 0x8000_4000
CRC1 = 0x8100_4000

Offset address: 0x4



Bit	Symbol	Function	R/W
31:0	n/a	CRC Data Input	R/W

CRCDIR or CRCDIR_BY is written with data for the CRC calculation.

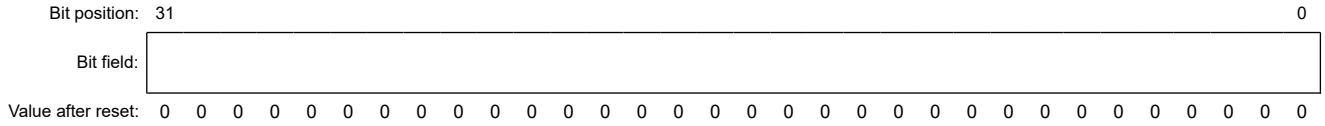
CRCDIR is the register symbol for R/W access in 32-bit units. When generating a 32-bit CRC code, the input data should be written to CRCDR.

CRCDIR_BY is the register symbol for R/W access in 8-bit units. When generating an 8- or 16-bit CRC code, the input data should be written to CRCDID_BY.

38.3.3 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register

Base address: CRC0 = 0x8000_4000
CRC1 = 0x8100_4000

Offset address: 0x8



Bit	Symbol	Function	R/W
31:0	n/a	CRC Data Output	R/W

When calculating with the initial value set to a value other than 0x00000000, rewrite the CRCDOR register.

The CRC code should be written to the CRCDIR register following the communication data. If the calculation result is 0x00000000, there is no error in the communication data.

If a 32-bit CRC is selected, the CRCDOR register should be accessed. If a 16-bit CRC is selected, the CRCDOR_HA register should be accessed. If an 8-bit CRC is selected, the CRCDOR_BY register should be accessed.

38.4 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first communication.

The following shows examples of generating the CRC code for input data (0xF0) using the 16-bit CRC-CCITT generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC Data Output Register (CRCDOR_HA) is cleared before the CRC calculation (when a 16-bit CRC is in use, bits [31:16] of CRCDOR are not updated).

When an 8-bit CRC is in use, the valid bits of the CRC code is obtained in the lower-order byte of CRCDOR. When a 32-bit CRC is in use, the valid bits of the CRC code is obtained in bits [31:0] of CRCDOR.

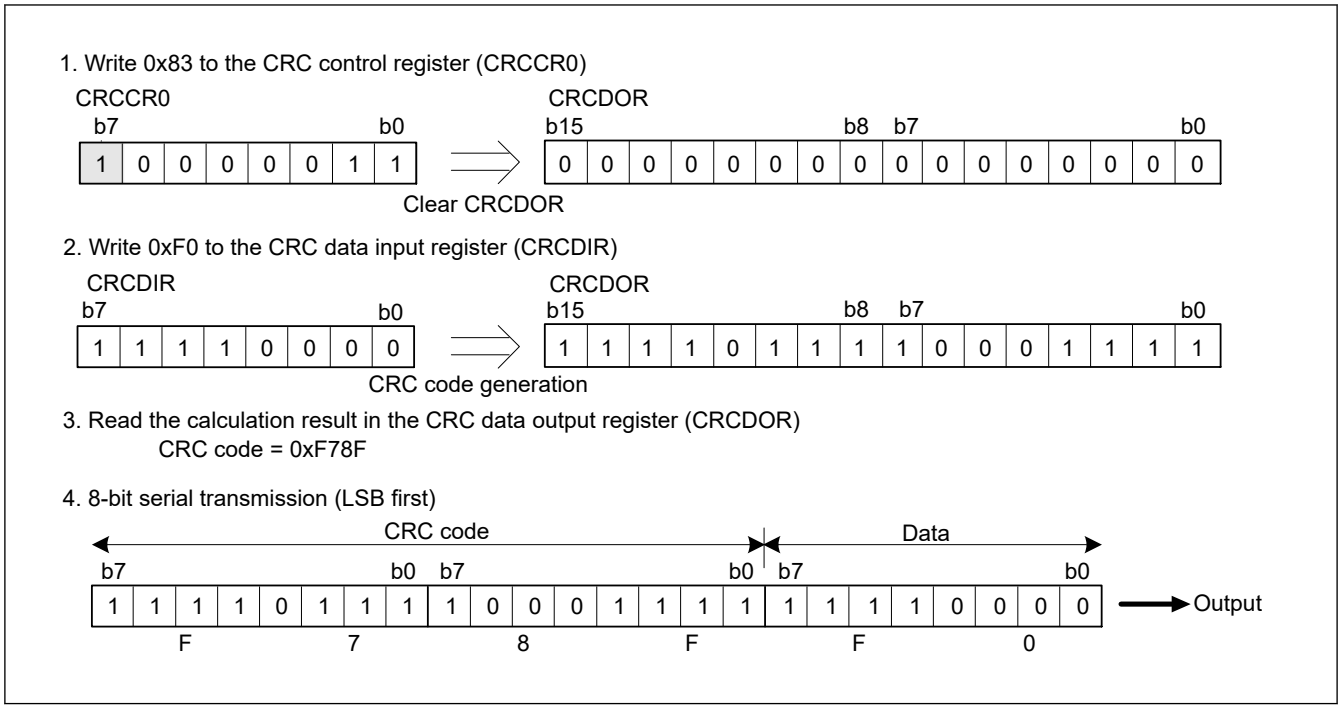


Figure 38.2 LSB first data transmission

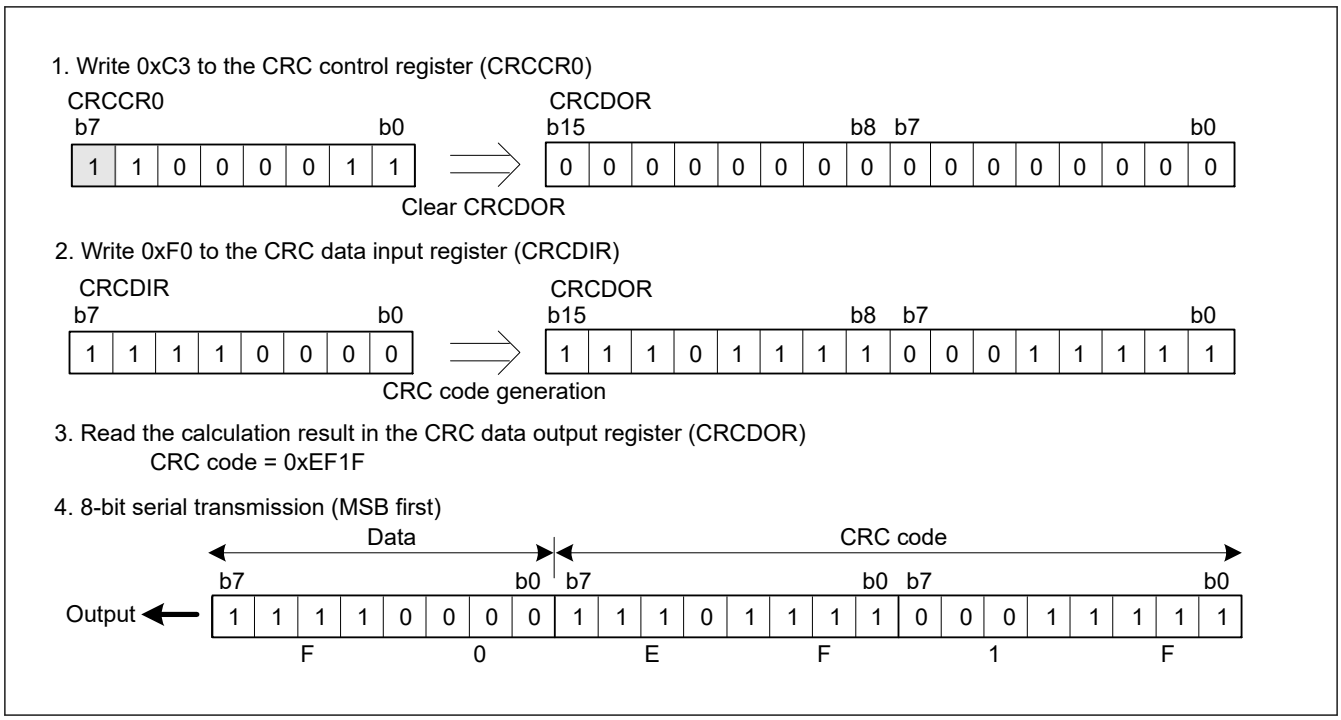


Figure 38.3 MSB first data transmission

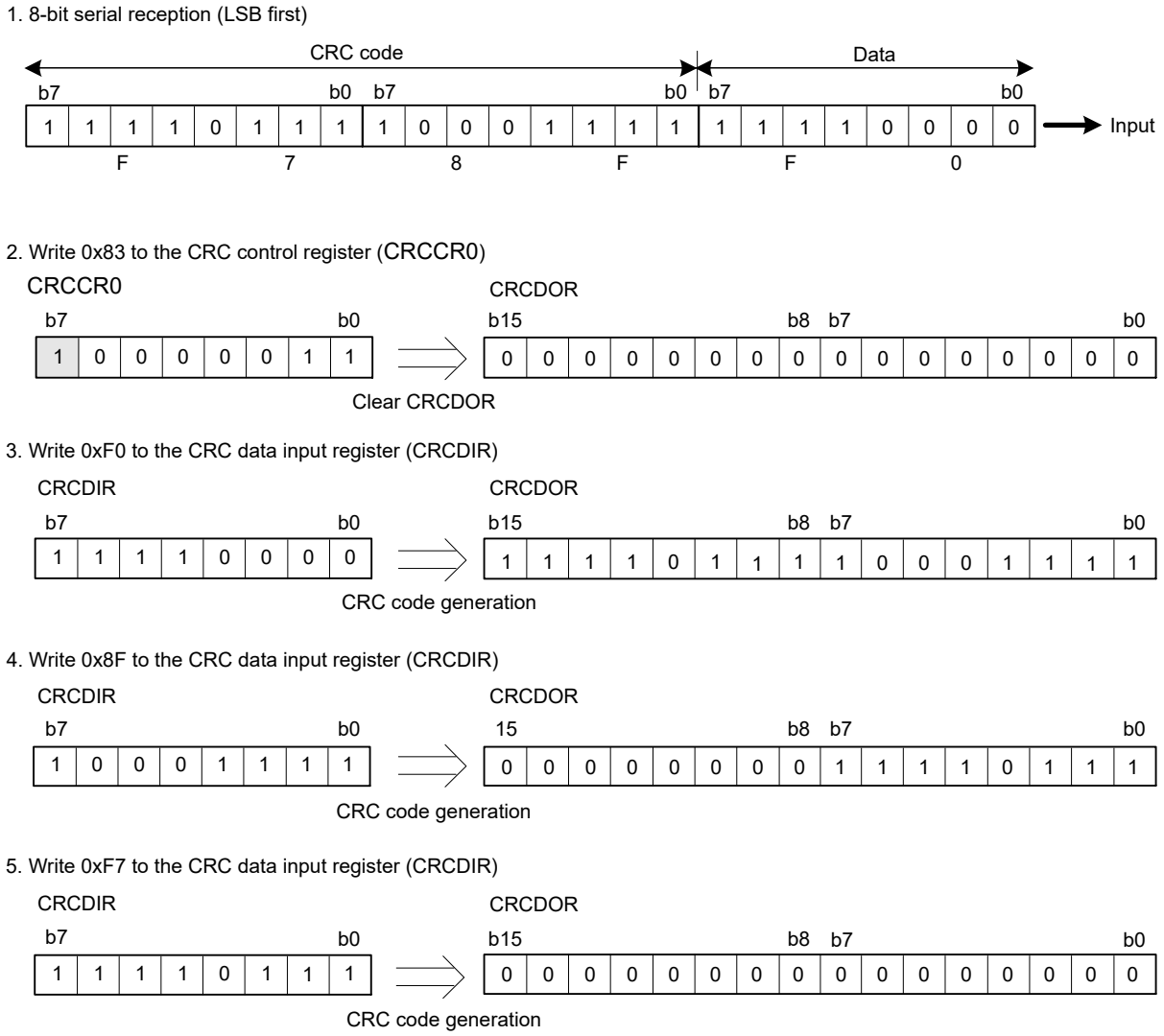


Figure 38.4 LSB first data reception

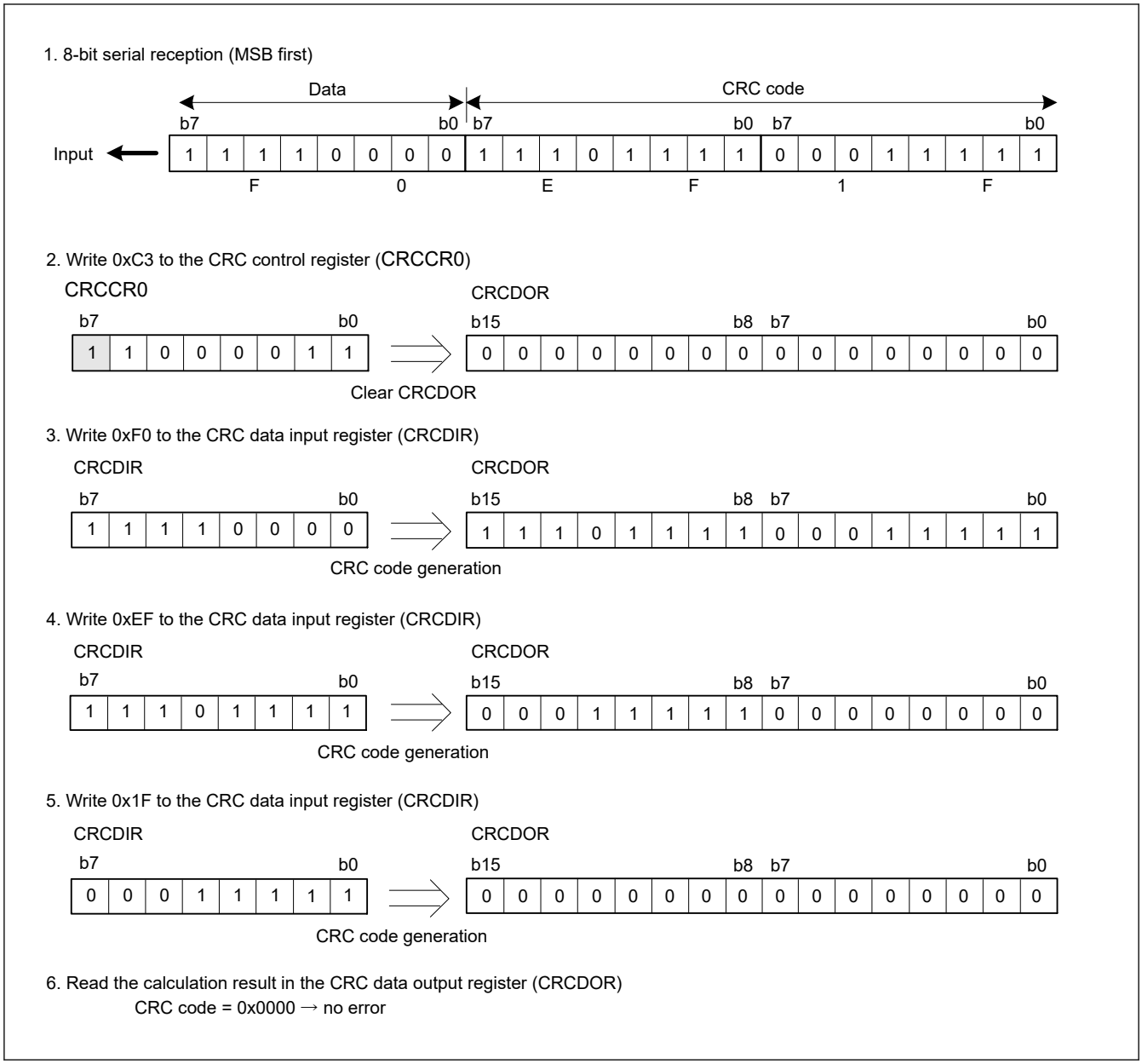


Figure 38.5 MSB first data reception

38.5 Usage Notes

38.5.1 Module-Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the Module Stop Control Register D and G (MSTPCRD and MSTPCRG). After a reset, the CRC is in the module-stop state. Register access is enabled by releasing the module-stop state.

For details, see section 9, Low-Power Consumption Function.

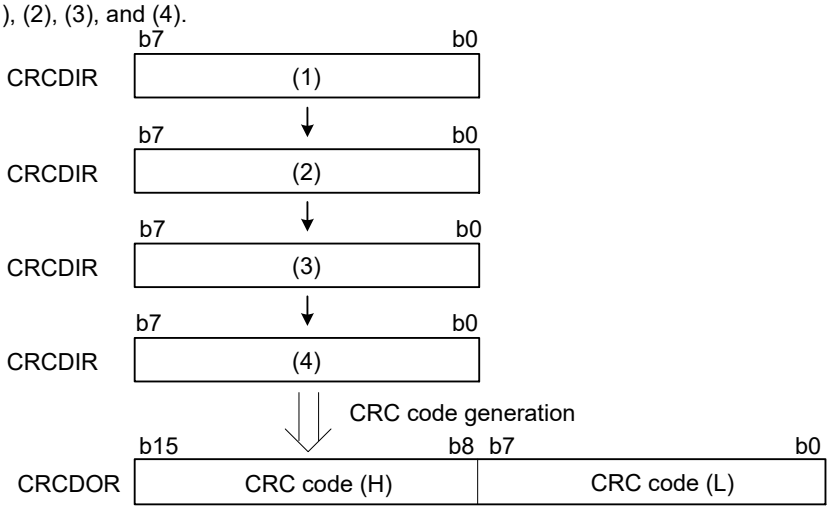
38.5.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

When transmitting 32-bit data (in case operation executed on 8 bits in parallel)

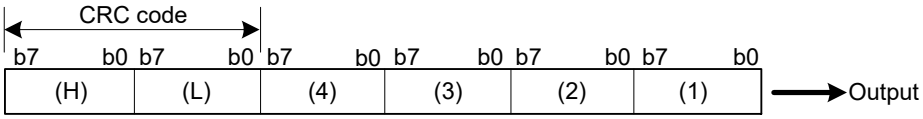
1. CRC code

After specifying the method for generation calculation, write data to the CRCDIR register in the order of (1), (2), (3), and (4).



2. Transmit data

(i) When transmission is LSB first



(ii) When transmission is MSB first

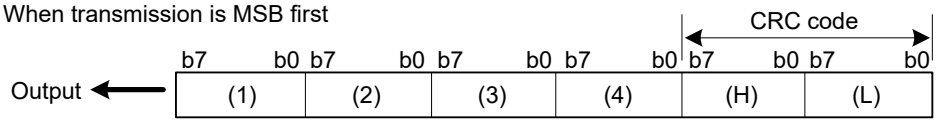


Figure 38.6 LSB first and MSB first data transmission

39. Boundary Scan

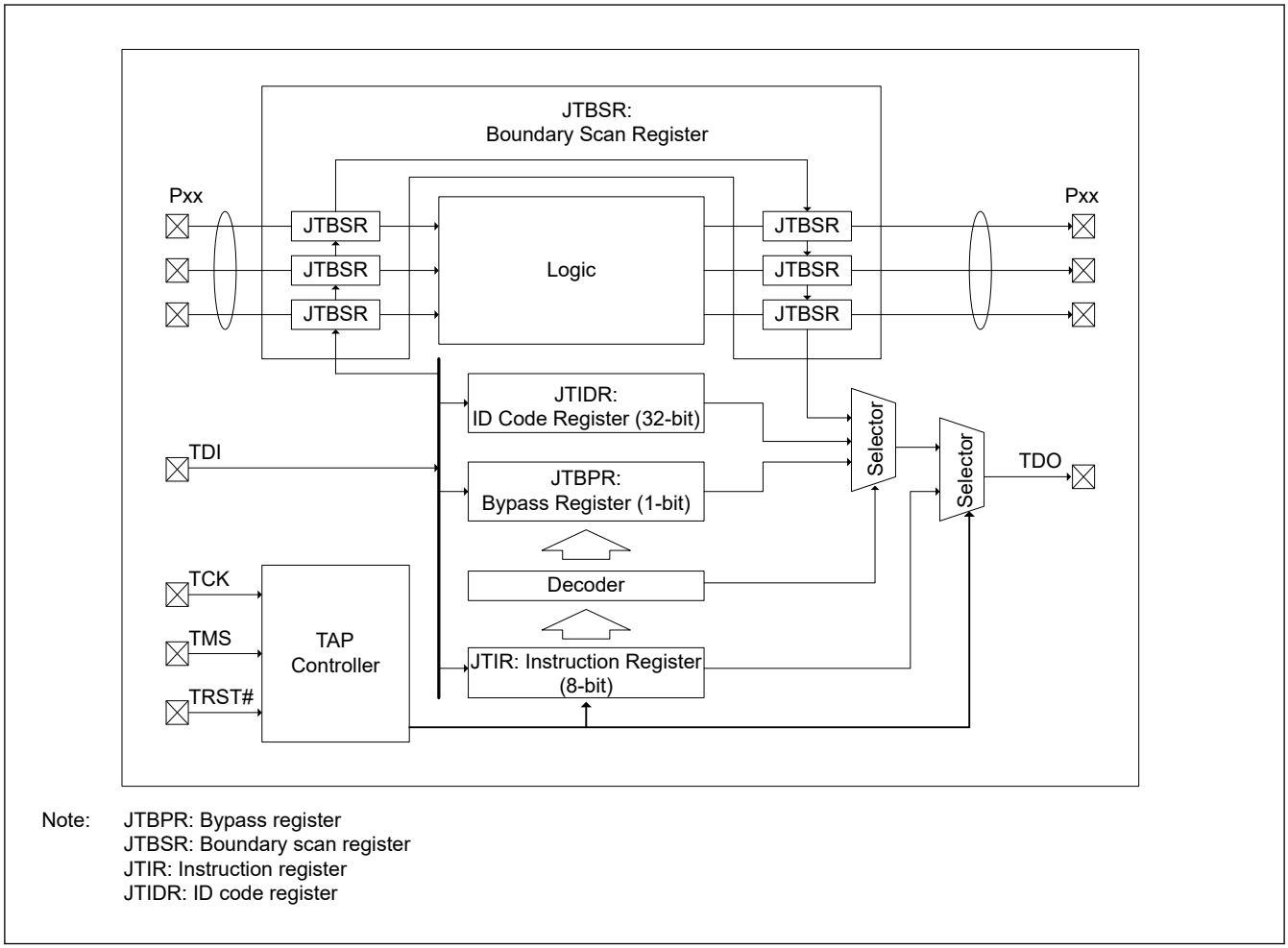
39.1 Overview

The LSI has a boundary scan function. The boundary scan is a serial I/O interface based on the JTAG (Joint Test Action Group), IEEE Std.1149.1, and IEEE Standard Test Access Port and Boundary-Scan Architecture.

Table 39.1 lists the specifications of boundary scan and Figure 39.1 shows a block diagram of the boundary scan function.

Table 39.1 Boundary scan specifications

Parameter	Specifications
Boundary scan enabled/disabled	Boundary scan function is enabled when the BSCANP pin is driven high.
Dedicated boundary scan pins	The TDO, TCK, TDI, TMS, and TRST# pins are dedicated for the JTAG when the boundary scan function is enabled.
Six test modes	• BYPASS mode • EXTEST mode • SAMPLE/PRELOAD mode • CLAMP mode • HIGHZ mode • IDCODE mode



Note: JTBPR: Bypass register
JTBSR: Boundary scan register
JTIR: Instruction register
JTIDR: ID code register

Figure 39.1 JTAG block diagram

Table 39.2 shows the I/O pins used in the boundary scan function.

Table 39.2 Pin configuration of JTAG

Pin name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. Input the clock the duty cycle of which is 50% when boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin
TRST#	Input	Test reset input pin

39.2 Register Descriptions

Instructions can be input to the instruction register via the TDI pin by serial transfer.

The bypass register, which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.

The boundary scan register, which is configured according to the BSDL description, is connected between the TDI and TDO pins when test data are being shifted in.

None of the registers is accessible from the CPU. Table 39.3 shows the availability of serial transfer for the registers.

Table 39.3 Serial transfer for the registers

Register	Serial input	Serial output
Instruction register	Available	Available
ID code register	Available	Available
Bypass register	Available	Available
Boundary scan register	Available	Available

Note: Any serial transfer is available if standards for the boundary scan function are satisfied.

39.2.1 JTIR : Instruction Register

Base address: —

Offset address: —

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TS[7:0]							
Value after reset:	0	1	0	1	0	1	0	1

Bit	Symbol	Function	R/W
7:0	TS[7:0]	Test Bit Set The command configuration is as shown in Table 39.4.	—

The JTIR register is an 8-bit register.

Boundary scan instructions can be transferred to the instruction register by serial input from the TDI pin.

The instruction register is initialized when the TRST# pin is driven to the low level or when the TAP controller is in the Test-Logic-Reset state.

Table 39.4 Command configuration (1 of 2)

TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0	Instruction
0	0	0	0	0	0	0	0	EXTEST
0	1	0	0	0	0	0	0	SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	IDCODE (value after reset)

Table 39.4 Command configuration (2 of 2)

TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0	Instruction
1	1	0	1	0	0	0	0	CLAMP
1	0	0	0	0	0	0	0	HIGHZ
1	1	1	1	1	1	1	1	BYPASS
Other than above								Reserved

39.2.2 JTIDR : ID Code Register

Base address: —

Offset address: —

Bit position:	31																												0
Bit field:	DID[31:0]																												
Value after reset:	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0	1	1	1	1	0	0	1	0	0	0	1	0	0	0	1	1	1

Bit	Symbol	Function	R/W
31:0	DID[31:0]	This register has the fixed value that indicates the device IDCODE.	—

The JTIDR register is a 32-bit register.

ID code register data is output from the TDO pin when the IDCODE instruction is executed.

39.2.3 JTBPR : Bypass Register

The JTBPR register is a 1-bit register.

This register is connected between the TDI and TDO pins when BYPASS mode is set.

The JTBPR register cannot be read from or written to by the CPU.

39.2.4 JTBSR : Boundary Scan Register

The JTBSR register is a shift register to control the external input and output pins of this LSI and is distributed across the pad.

The EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions are issued to apply the JTBSR register in boundary-scan testing.

Refer to the BSDL file for the correspondences between the pins of this LSI and the bits of the boundary scan register.

The value is undefined after a reset.

39.3 Operation

The boundary scan function is valid when the RES# and BSCANP pins are driven high.

39.3.1 TAP Controller

Figure 39.2 shows the state transition diagram of the TAP controller. Table 39.5 describes each state.

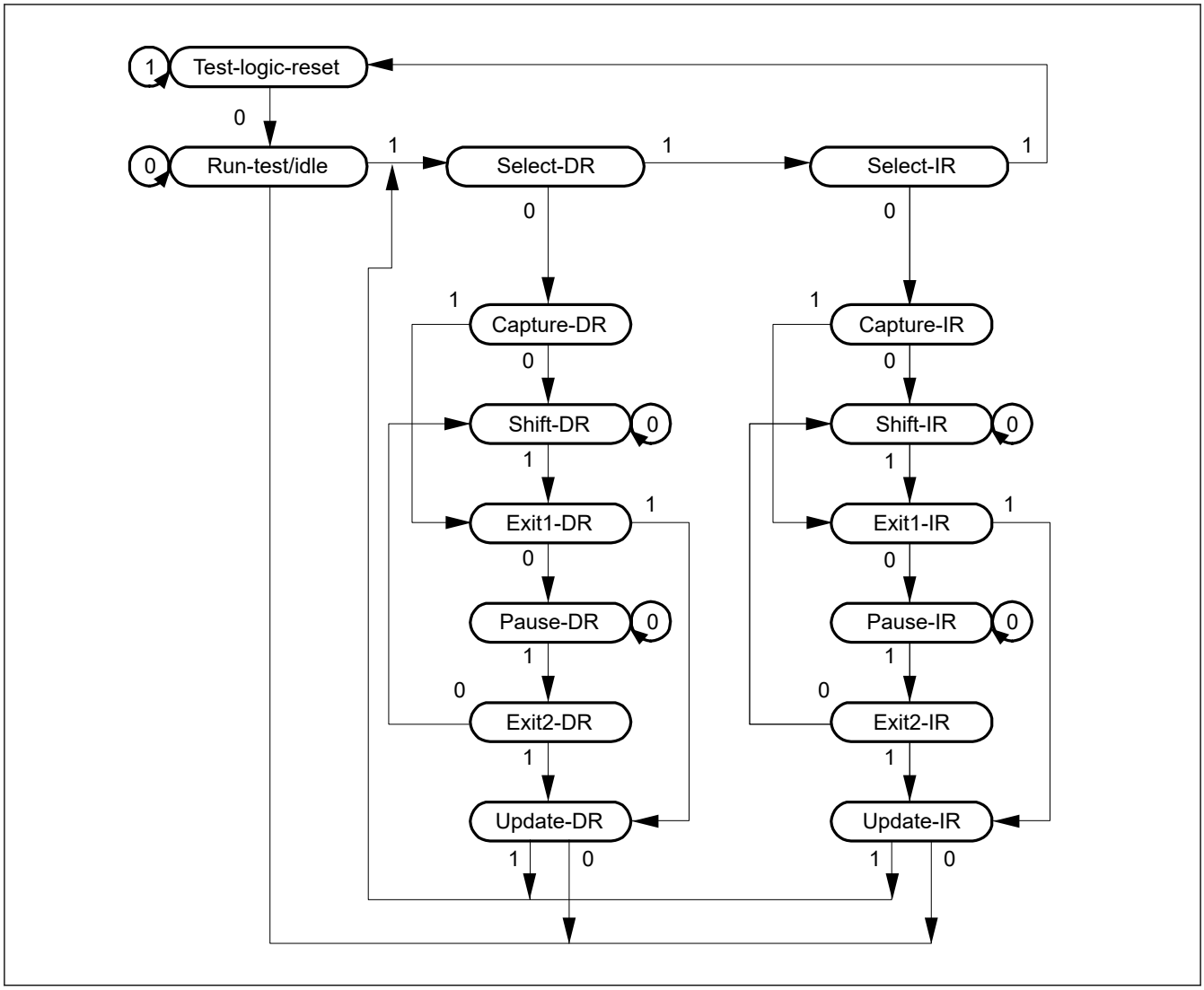


Figure 39.2 State transition diagram of TAP controller

Table 39.5 Explanation of states (1 of 2)

State	Explanation
Test Logic Reset	Reset state of the TAP controller The LSI is in this state during normal operation.
Run Test/Idle	Test execution state
Select DR Scan	Temporary state used for selecting a data register
Select IR Scan	Temporary state used for selecting an instruction register
Capture DR	Data of the test data register corresponding to the current instruction is captured in parallel.
Shift DR	The test data register corresponding to the current instruction is connected between the TDI and TDO pins, and data is transferred serially.
Exit DR	Temporary state
Pause DR	Clocks are applied while the value input in the Shift DR state is retained.
Exit2 DR	Temporary state
Update DR	Output latches of the test data register corresponding to the current instruction are updated.
Capture IR	A fixed value is input to the instruction register.
Shift IR	The instruction register is connected between the TDI and TDO pins, and data is transferred serially.
Exit IR	Temporary state

Table 39.5 Explanation of states (2 of 2)

State	Explanation
Pause IR	Clocks are applied while the value input in the Shift IR state is retained.
Exit2 IR	Temporary state
Update IR	The current instruction is updated to the instruction input in the Shift IR state.

39.3.2 List of Commands

(1) BYPASS [Instruction Code: 1111 1111b]

The BYPASS instruction drives the bypass register. This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed-circuit board at higher speeds. While this instruction is being executed, the test circuit does not affect the system circuit.

The bypass register is connected between the TDI and TDO pins. Bypass operation is initiated from shift-DR operation. The TDO is set to 0 in the first clock cycle in the Shift-DR state. In the subsequent clock cycles, the TDI signal is output on the TDO pin.

(2) EXTEST [Instruction Code: 0000 0000b]

The EXTEST instruction is used to test external circuits when this LSI is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified by the SAMPLE/PRELOAD instruction) from the boundary scan register to the print circuit board, and input pins are used to input test result to the boundary scan register from the print circuit board.

(3) SAMPLE/PRELOAD [Instruction Code: 0100 0000b]

The SAMPLE/PRELOAD instruction is used to input data from this LSI's internal circuits to the boundary scan register, output data from the scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to this LSI and output signals are also directly output to the external circuits. The system circuit of this LSI is not affected by this instruction.

In SAMPLE operation, the boundary scan register latches the snapshot of data transferred from input pins to internal circuit or data transferred from internal circuit to output pins. The latched data is read from the scan path. The scan register latches the snapshot at the rising edge of the TCK in Capture-DR state. The scan register latches snapshots without affecting the LSI normal operation.

In PRELOAD operation, the value after reset is written from the scan path to the parallel output latch of the boundary scan register prior to the EXTEST instruction execution. If the EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In the EXTEST instruction, output parallel latches are always output to the output pins.)

(4) IDCODE [Instruction Code: 0101 0101b]

When the IDCODE instruction is selected, the ID code register value is output from the TDO in LSB-first order in Shift-DR state of the TAP controller. While this instruction is being executed, the test circuit does not affect the system circuit. The instruction register is initialized by the IDCODE instruction in Test-Logic-Reset state of the TAP controller.

(5) CLAMP [Instruction Code: 1101 0000b]

When the CLAMP instruction is selected, output pins output the boundary scan register value which was specified by the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of the boundary scan register is maintained regardless of the TAP controller state.

The bypass register is connected between the TDI and TDO pins, leading to the same operation as when BYPASS mode has been selected.

(6) HIGHZ [Instruction Code: 1000 0000b]

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the status of the boundary scan register is maintained regardless of the TAP controller state. The bypass register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction has been selected.

39.4 Usage Notes

1. For serial transfer, data are input or output in LSB-first order. See Figure 39.3.
2. Pins of the boundary scan (TDI, TMS, and TRST#) have to be pulled up by pull-up resistors.
   However, if an on-chip emulator is in use, handle the TRST# pin according to the manual for the given on-chip emulator.
3. Power supply pins (VDD33, VDD1833_0 to VDD1833_7, VDDP_18_33, VDDP_18_0 to VDDP_18_7, VDD08, VDD18_PLL0 to VDD18_PLL4, VDD08_PLL0 to VDD08_PLL4, VDD33_X, VDDP_18_X, AVDD18A_TSU, DVDD08A_TSU, OTPVDD18, OTPVDD08, VSS, VSS_PLL0 to VSS_PLL4, USB_USVDD33, USB_USVDD18, USB_USDVDD, AVDD_ADC0 to AVDD_ADC2, AVDDIO_ADC0 to AVDDIO_ADC2, AVSS_ADC0 to AVSS_ADC2, AVSSIO_ADC0 to AVSSIO_ADC2, PCIE_VDD18A_CMN, PCIE_VDD18A_L0, PCIE_VDD18A_L1, PCIE_VDD08A_L0, PCIE_VDD08A_L1 DDR_VAA, DDR_VDDQ) cannot be boundary-scanned.
4. Analog reference pins (AVDDREF_ADC0 to AVDDREF_ADC2, USB_TXRTUNE) cannot be boundary-scanned.
5. Analog input pins (AN000 to AN003, AN100 to AN103, AN200 to AN215) cannot be boundary-scanned.
6. Clock pins (EXTAL, XTAL, EXTCLKIN, and XTALSEL) cannot be boundary-scanned.
7. The reset pin (RES#) cannot be boundary-scanned.
8. The operating mode input pin (MDX) cannot be boundary-scanned.
9. USB dedicated pins (USB_QDP, USB_QDM, USB_VUBUSIN, USB_OTG_ID) cannot be boundary-scanned.
10. PCIE dedicated pins (PCIE_REFCLK_P0, PCIE_REFCLK_N0, PCIE_REFCLK_P1, PCIE_REFCLK_N1PCIE_RXDP_L0, PCIE_RXDN_L0, PCIE_RXDP_L1, PCIE_RXDN_L1, PCIE_TXDP_L0, PCIE_TXDN_L0, PCIE_TXDP_L1, PCIE_TXDN_L1) cannot be boundary-scanned.
11. DDR dedicated pins (DDR_ZN, DDR_DTEST, DDR_ATEST, DDR_RESET_N, DDR_CKA_T, DDR_CKA_C, DDR_CKB_T, DDR_CKB_C, DDR_CKEA[1:0], DDR_CKEB[1:0], DDR_CSA[1:0], DDR_CSB[1:0], DDR_CAA[5:0], DDR_CAB[5:0], DDR_DQA[15:0], DDR_DQB[15:0], DDR_DMIA[1:0], DDR_DMIB[1:0], DDR_DQSA_T[1:0], DDR_DQSB_T[1:0], DDR_DQSA_C[1:0], DDR_DQSB_C[1:0]) cannot be boundary-scanned.
12. The boundary-scan pin (BSCANP) cannot be boundary-scanned.
13. The boundary-scan pins (TCK, TMS, TRST#, TDI, and TDO) cannot be boundary-scanned.
14. The boundary scan function is not available when the chip is in the reset state.
15. For a pin that incorporates open-drain functionality and for which the open-drain function is enabled, if the boundary-scan function sets the corresponding bit in the output scan register and output enable register to 1, executing an EXTEST, CLAMP, or SAMPLE/PRELOAD instruction makes the pin output the high level rather than placing it in the high-impedance state.*1

Note 1. For details on boundary-scan target pins, refer to the BSDL file. Boundary-scan is not applicable to some pins and power pins.

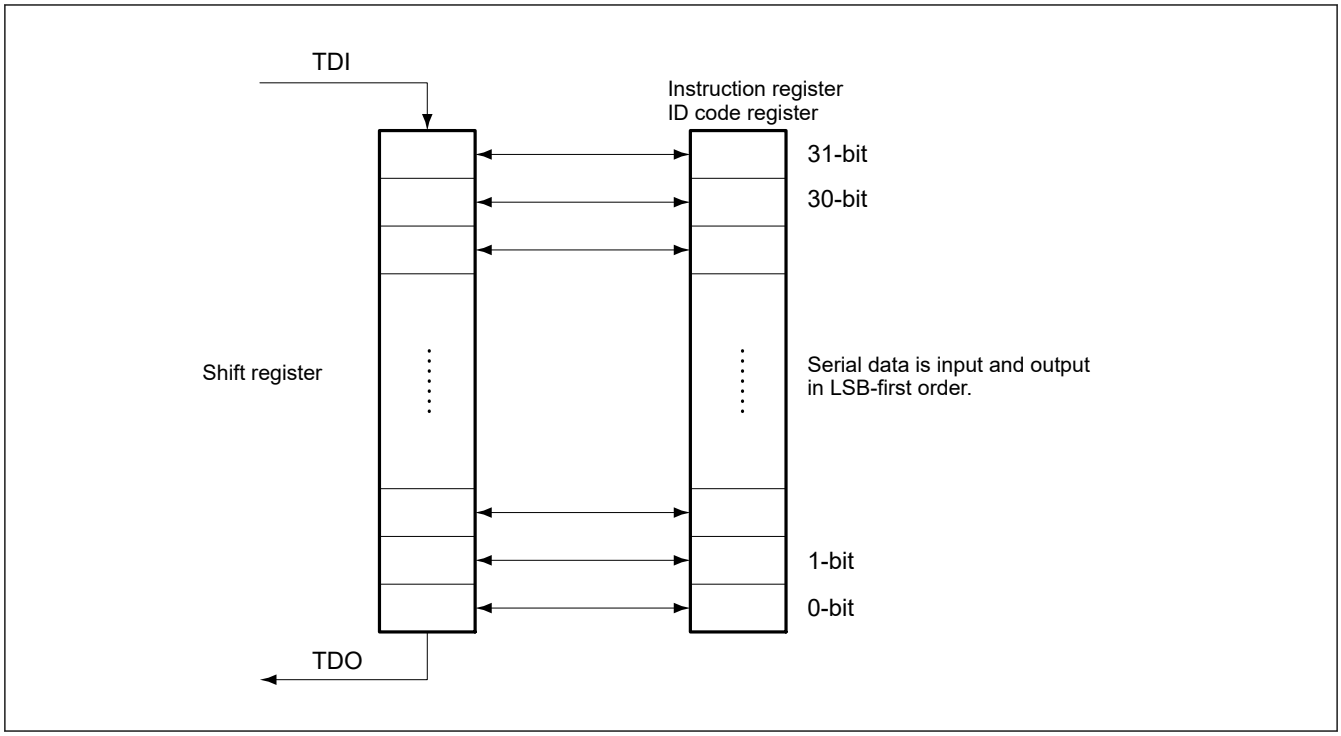


Figure 39.3 Serial data Input/Output

40. ΔΣ Interface (DSMIF)

40.1 Overview

The ΔΣ interfaces (DSMIF) can be connected with up to three external ΔΣ modulators. It is capable of filtering delta-sigma modulated 1-bit digital input data and converting it into 16-bit digital data.

Table 40.1 lists the specifications of DSMIF and Table 40.2 lists the functions of DSMIF. Figure 40.1 shows a block diagram of DSMIF.

Table 40.1 DSMIF specifications

Parameter	Description
Number of units	10 units (6 in LLPP0 and 4 in LLPP1)
Number of channels	3 channels per unit for a total of 30 channels
Decimation filter characteristics	Filter are selectable as Sinc 1st order, Sinc 2nd order, Sinc 3rd order Can select each channel.
Over-sampling rate (decimation rate)	Selectable from 4 to 256 Can select each channel.
Interface clock	Support Master mode (MCLK is used as output)/Slave mode (MCLK is used as input) by channel Master mode supports the following output clock: ● 25/20/12.5/10/6.25/5 MHz (Core Clock: 400 MHz, Bus Clock: 250 MHz) ● 25/20.8/12.5/10.4/6.25/5 MHz (Core Clock: 250 MHz, Bus Clock: 250 MHz) Filter received clock inversion function.
Data capture	Crests and troughs can trigger capture, allowing capture of values for current with the same timing. Can select one capture trigger from six triggers for each trigger input by channel. The PWM synchronization function is realized by initializing the decimation counter for the differential stage of the SINC filter with the trigger output from GPT.
Unit synchronization	Not supported
Reset function of differential filter clock divider	Support channel by channel. Can select one from three triggers.
Error source output (interrupt)	12 kinds of interrupt sources in 1 unit: ● Current data register update interrupt ● Current capture data register A update interrupt ● Current capture data register B update interrupt ● Short-circuit detection error interrupt ● Lower threshold overcurrent detection 0 error interrupt ● Upper threshold overcurrent detection 0 error interrupt ● Lower threshold overcurrent detection 1 error interrupt ● Upper threshold overcurrent detection 1 error interrupt ● Lower threshold overcurrent detection 2 error interrupt ● Upper threshold overcurrent detection 2 error interrupt ● Lower threshold current detection interrupt ● Upper threshold detection interrupt
Low-power consumption function	The module-stop state can be set (clock gating function is implemented)

Table 40.2 DSMIF functions

Parameter			Name
	Clock	I/O	MCLKmn
I/O pin	Data	Input	MDATmn
Interrupt source	Normal	Current Data register Update (Combined with ch0 to ch2)	DSMIFm_CDRUI
		Current Data register Update Common	DSMIFm_CDUPDC
		Current Capture Data register A Update Common	DSMIFm_CDACUPDC
		Current Capture Data register B Update Common	DSMIFm_CDBCUPDC
	Error	Lower threshold current sum error*1	DSMIFm_LTCSE
		Upper threshold current sum error*1	DSMIFm_UTCSE
		Short Circuit Detection CHn*1	DSMIFm_SCDEn
		Overcurrent Lower limit Detection 0 CHn*1	DSMIFm_OVC0ELn
		Overcurrent Upper limit Detection 0 CHn*1	DSMIFm_OVC0EHn
		Overcurrent Lower limit Detection 1 CHn	DSMIFm_OVC1ELn
		Overcurrent Upper limit Detection 1 CHn	DSMIFm_OVC1EHn
		Overcurrent Lower limit Detection 2 CHn	DSMIFm_OVC2ELn
		Overcurrent Upper limit Detection 2 CHn	DSMIFm_OVC2EHn
		Overcurrent Detection Window Notification 0 CHn	DSMIFm_OVC0WNn
		Overcurrent Detection Window Notification 1 CHn	DSMIFm_OVC1WNn
		Overcurrent Detection Window Notification 2 CHn	DSMIFm_OVC2WNn
		Overcurrent Detection Window Notification 3 CHn*2	DSMIFm_OVC3WNn
Trigger input ELC destination	Capture	Current data capture trigger 0	DSMIFm CAP_TRG0
		Current data capture trigger 1	DSMIFm CAP_TRG1
		Current data capture trigger 2	DSMIFm CAP_TRG2
		Current data capture trigger 3	DSMIFm CAP_TRG3
		Current data capture trigger 4	DSMIFm CAP_TRG4
		Current data capture trigger 5	DSMIFm CAP_TRG5
	Counter Reset	Decimation dividing counter initialization trigger 0	DSMIFm CDCNT_INT_TRG0
		Decimation dividing counter initialization trigger 1	DSMIFm CDCNT_INT_TRG1
		Decimation dividing counter initialization trigger 2	DSMIFm CDCNT_INT_TRG2

Note: m = 0 to 9 (unit), n = 0 to 2 (ch)
Note 1. Combined error signal is connected to POE3 and POEG as DSMIF error 0 each unit.
Note 2. Combined error signal is connected to POE3 and POEG as DSMIF error 1 each unit.

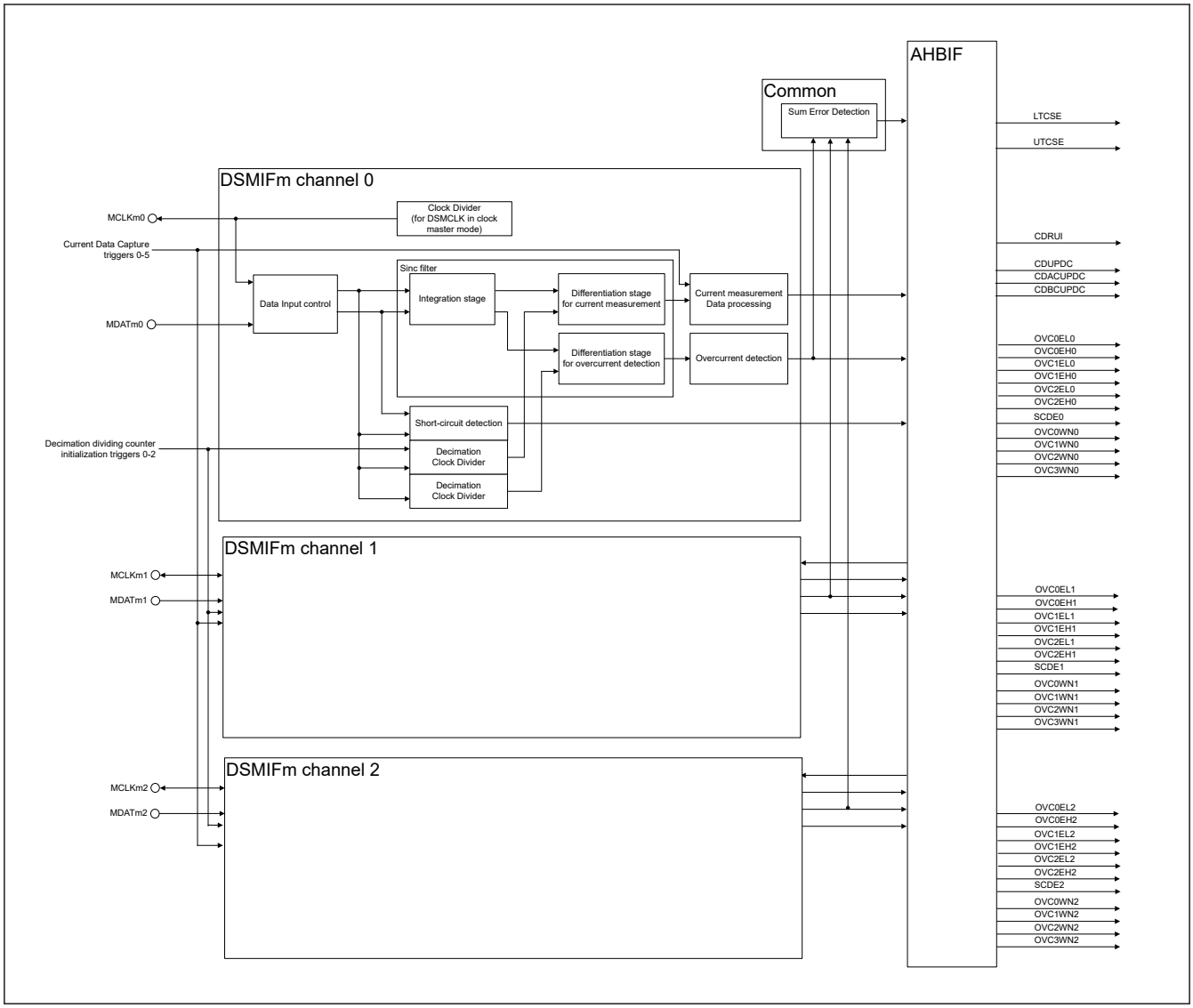


Figure 40.1 Block diagram of DSMIFm

40.2 Register Map

Table 40.3 DSMIF register map (1 of 3)

Address	Register symbol	Register name	Write protection
0x9002_0000 + 0x1000 × m (m = 0 to 5) 0x9012_0000 + 0x1000 × (m - 6) (m = 6 to 9)	DSCCSCR	Core Clock Selection Control Register	—
0x9002_0080 + 0x1000 × m (m = 0 to 5) 0x9012_0080 + 0x1000 × (m - 6) (m = 6 to 9)	DSSEICR	Overcurrent Sum Error Detect Interrupt Control Register	—
0x9002_0084 + 0x1000 × m (m = 0 to 5) 0x9012_0084 + 0x1000 × (m - 6) (m = 6 to 9)	DSSEICR	Overcurrent Sum Error Detect Interrupt Control Register	—
0x9002_0088 + 0x1000 × m (m = 0 to 5) 0x9012_0088 + 0x1000 × (m - 6) (m = 6 to 9)	DSSELTR	Overcurrent Sum Error Detect Low Threshold Register	—
0x9002_008C + 0x1000 × m (m = 0 to 5) 0x9012_008C + 0x1000 × (m - 6) (m = 6 to 9)	DSSEHTR	Overcurrent Sum Error Detect High Threshold Register	—
0x9002_0090 + 0x1000 × m (m = 0 to 5) 0x9012_0090 + 0x1000 × (m - 6) (m = 6 to 9)	DSSECR	Overcurrent Sum Error Detect Control Register	—
0x9002_00A0 + 0x04 × n + 0x1000 × m (m = 0 to 5) 0x9012_00A0 + 0x04 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSSECDRn	Overcurrent Sum Error Detect Capture Data Register n (n = 0 to 2)	—

Table 40.3 DSMIF register map (2 of 3)

Address	Register symbol	Register name	Write protection
0x9002_00C0 + 0x1000 × m (m = 0 to 5) 0x9012_00C0 + 0x1000 × (m - 6) (m = 6 to 9)	DSCMSR	Common Mode Setting Register	—
0x9002_00C4 + 0x1000 × m (m = 0 to 5) 0x9012_00C4 + 0x1000 × (m - 6) (m = 6 to 9)	DSCICR	Common Interrupt Control Register	—
0x9002_0200 + 0x1000 × m (m = 0 to 5) 0x9012_0200 + 0x1000 × (m - 6) (m = 6 to 9)	DSCSTRTR	Channel Software Start Trigger Register	—
0x9002_0204 + 0x1000 × m (m = 0 to 5) 0x9012_0204 + 0x1000 × (m - 6) (m = 6 to 9)	DSCSTPTR	Channel Software Stop Trigger Register	—
0x9002_0220 + 0x1000 × m (m = 0 to 5) 0x9012_0220 + 0x1000 × (m - 6) (m = 6 to 9)	DSCESR	Channel Error Status Register	—
0x9002_0224 + 0x1000 × m (m = 0 to 5) 0x9012_0224 + 0x1000 × (m - 6) (m = 6 to 9)	DSCOCESR	Channel Overcurrent Error Status Register	—
0x9002_0228 + 0x1000 × m (m = 0 to 5) 0x9012_0228 + 0x1000 × (m - 6) (m = 6 to 9)	DSCOCNSR	Channel Overcurrent Notification Status Register	—
0x9002_022C + 0x1000 × m (m = 0 to 5) 0x9012_022C + 0x1000 × (m - 6) (m = 6 to 9)	DSCOCRMR	Channel Overcurrent Comparator Result Monitor Register	—
0x9002_0240 + 0x1000 × m (m = 0 to 5) 0x9012_0240 + 0x1000 × (m - 6) (m = 6 to 9)	DSCSR	Channel Status Register	—
0x9002_0244 + 0x1000 × m (m = 0 to 5) 0x9012_0244 + 0x1000 × (m - 6) (m = 6 to 9)	DSCSSR	Channel State Status Register	—
0x9002_0260 + 0x1000 × m (m = 0 to 5) 0x9012_0260 + 0x1000 × (m - 6) (m = 6 to 9)	DSCESCR	Channel Error Status Clear Register	—
0x9002_0264 + 0x1000 × m (m = 0 to 5) 0x9012_0264 + 0x1000 × (m - 6) (m = 6 to 9)	DSCOCESCR	Channel Overcurrent Error Status Clear Register	—
0x9002_0268 + 0x1000 × m (m = 0 to 5) 0x9012_0268 + 0x1000 × (m - 6) (m = 6 to 9)	DSCOCNSCR	Channel Overcurrent Notification Status Clear Register	—
0x9002_0280 + 0x1000 × m (m = 0 to 5) 0x9012_0280 + 0x1000 × (m - 6) (m = 6 to 9)	DSCSCR	Channel Status Clear Register	—
0x9002_0300 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0300 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSICRCHn	Interrupt Control Register Channel n (n = 0 to 2)	—
0x9002_0304 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0304 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCMCCRCHn	Current Measurement Clock Control Register Channel n (n = 0 to 2)	—
0x9002_0308 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0308 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCMFCRCHn	Current Measurement Filter Control Register Channel n (n = 0 to 2)	—
0x9002_030C + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_030C + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCMCTCRCHn	Current Measurement Capture Trigger Control Register Channel n (n = 0 to 2)	—
0x9002_0320 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0320 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSEDCRCHn	Error Detect Control Register Channel n (n = 0 to 2)	—
0x9002_0330 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0330 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSOCFCRCHn	Overcurrent Detect Filter Control Register Channel n (n = 0 to 2)	—
0x9002_0324 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0324 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSSCTSRCHn	Short Circuit Threshold Setting Register Channel n (n = 0 to 2)	—
0x9002_0334 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0334 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSODCRCHn	Overcurrent Detect Control Register Channel n (n = 0 to 2)	—
0x9002_0338 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0338 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSODWCRCHn	Overcurrent Detect Window Control Register Channel n (n = 0 to 2)	—
0x9002_03A0 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03A0 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSOCLTR0CHn	Overcurrent Low Threshold Register 0 Channel n (n = 0 to 2)	—
0x9002_03A4 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03A4 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSOCHTR0CHn	Overcurrent High Threshold Register 0 Channel n (n = 0 to 2)	—

Table 40.3 DSMIF register map (3 of 3)

Address	Register symbol	Register name	Write protection
0x9002_03A8 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03A8 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSOCLTR1CHn	Overcurrent Low Threshold Register 1 Channel n (n = 0 to 2)	—
0x9002_03AC + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03AC + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSOCHTR1CHn	Overcurrent High Threshold Register 1 Channel n (n = 0 to 2)	—
0x9002_03B0 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03B0 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSOCLTR2CHn	Overcurrent Low Threshold Register 2 Channel n (n = 0 to 2)	—
0x9002_03B4 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03B4 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSOCHTR2CHn	Overcurrent High Threshold Register 2 Channel n (n = 0 to 2)	—
0x9002_03E0 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03E0 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCSTRTRCHn	Software Start Trigger Register Channel n (n = 0 to 2)	—
0x9002_03E4 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03E4 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCSTPTRCHn	Software Stop Trigger Register Channel n (n = 0 to 2)	—
0x9002_03F0 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03F0 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCDRCHn	Current Data Register Channel n (n = 0 to 2)	—
0x9002_03F4 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03F4 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCCDRACHn	Capture Current Data Register A Channel n (n = 0 to 2)	—
0x9002_03F8 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03F8 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCCDRBCHn	Capture Current Data Register B Channel n (n = 0 to 2)	—
0x9002_03FC + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_03FC + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSOCDRCHn	Overcurrent Data Register Channel n (n = 0 to 2)	—
0x9002_0400 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0400 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCOCDR0CHn	Capture Overcurrent Data Register 0 Channel n (n = 0 to 2)	—
0x9002_0404 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0404 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCOCDR1CHn	Capture Overcurrent Data Register 1 Channel n (n = 0 to 2)	—
0x9002_0408 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0408 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCOCDR2CHn	Capture Overcurrent Data Register 2 Channel n (n = 0 to 2)	—
0x9002_0420 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0420 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCSRCHn	Status Register Channel n (n = 0 to 2)	—
0x9002_0430 + 0x140 × n + 0x1000 × m (m = 0 to 5) 0x9012_0430 + 0x140 × n + 0x1000 × (m - 6) (m = 6 to 9)	DSCSCRCHn	Status Clear Register Channel n (n = 0 to 2)	—

Table 40.4 DSMIF related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0 (LLPP0)	—	MSTPCRD.MSTPCRD00	SLVACCCTL7.LLPP_SL*1
1 (LLPP0)	—	MSTPCRD.MSTPCRD01	SLVACCCTL7.LLPP_SL*1
2 (LLPP0)	—	MSTPCRD.MSTPCRD16	SLVACCCTL7.LLPP_SL*1
3 (LLPP0)	—	MSTPCRD.MSTPCRD17	SLVACCCTL7.LLPP_SL*1
4 (LLPP0)	—	MSTPCRD.MSTPCRD18	SLVACCCTL7.LLPP_SL*1
5 (LLPP0)	—	MSTPCRD.MSTPCRD19	SLVACCCTL7.LLPP_SL*1
6 (LLPP1)	—	MSTPCRD.MSTPCRD20	SLVACCCTL7.LLPP_SL*1
7 (LLPP1)	—	MSTPCRD.MSTPCRD21	SLVACCCTL7.LLPP_SL*1
8 (LLPP1)	—	MSTPCRD.MSTPCRD22	SLVACCCTL7.LLPP_SL*1
9 (LLPP1)	—	MSTPCRD.MSTPCRD23	SLVACCCTL7.LLPP_SL*1

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

40.3 Register Descriptions

40.3.1 DSMIF Common Control Registers

40.3.1.1 DSCCSCR : Core Clock Selection Control Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKSEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLKSEL	Core Clock Selection 0: 250 MHz 1: 400 MHz	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

40.3.1.2 DSSEICR : Overcurrent Sum Error Detect Interrupt Control Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISEH	ISEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ISEL	Overcurrent sum error lower limit detection interrupt enable bit 0: Do not output overcurrent sum error lower limit detection interrupt 1: Overcurrent sum error lower limit detection interrupt can be output	R/W
1	ISEH	Overcurrent sum error upper limit detection interrupt enable bit 0: Do not output overcurrent sum error upper limit detection interrupt 1: Overcurrent sum error lower upper detection interrupt can be output	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

40.3.1.3 DSSECSR : Overcurrent Sum Error Detect Channel Setting Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SEDM[2:0]	Overcurrent sum error detect mode setting bit 0 0 0: Detects error of sum value of overcurrent data of CH0, CH1, CH2 0 0 1: Detects error of sum value of overcurrent data of CH0, CH1 0 1 0: Detects error of overcurrent data of CH0 0 1 1: Detects error of overcurrent data of CH1 1 0 0: Detects error of overcurrent data of CH2 Others: Setting prohibited (detects error of sum value of overcurrent data of CH0, CH1, CH2)	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

40.3.1.4 DSSELTR : Overcurrent Sum Error Detect Low Threshold Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x088

Bit position:	31														17																0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCMPTBL[17:0]																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
17:0	SCMPTBL[17:0]	Overcurrent sum error detect lower limit These bits set the lower limit of the active current value for overcurrent sum error detection. Set these bits to the lower limit of the sum value of the overcurrent data for each channel.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

40.3.1.5 DSSEHTR : Overcurrent Sum Error Detect High Threshold Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x08C

Bit position:	31														17																0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCMPTBH[17:0]																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
17:0	SCMPTBH[17:0]	Overcurrent sum error detect upper limit These bits set the upper limit of the active current value for overcurrent sum error detection. Set these bits to the upper limit of the sum value of the overcurrent data for each channel.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

40.3.1.6 DSSECR : Overcurrent Sum Error Detect Control Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEEH	SEEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SEEL	Overcurrent sum error lower limit detection enable bit 0: The lower limit exceeded detection function of overcurrent sum error detection is invalid 1: The lower limit exceeded detection function of overcurrent sum error detection is valid	R/W
1	SEEH	Overcurrent sum error upper limit detection enable bit 0: The upper limit exceeded detection function of overcurrent sum error detection is invalid 1: The upper limit exceeded detection function of overcurrent sum error detection is valid	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

40.3.1.7 DSSECDRn : Overcurrent Sum Error Detect Capture Data Register n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x0A0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SECDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SECDR[15:0]	Overcurrent sum error detect capture data n Captures the overcurrent data of the target channel when the upper limit error or the lower limit error of the over sum current error detection occurs. See Table 40.11 for details of the data to be captured.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

40.3.1.8 DSCMSR : Common Mode Setting Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x0C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CISM[1:0]		DFS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DFS	Data Format Select 0: Left justified 1: Right justified	R/W
2:1	CISM[1:0]	Common Interrupt Synchronous channel Mode 0 0: Common interrupts are not used 0 1: Common interrupts for ch0, ch1, and ch2 can be used 1 0: Common interrupts for ch0 and ch1 can be used 1 1: Setting prohibited	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

40.3.1.9 DSCICR : Common Interrupt Control Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x0C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	IBUCE	IAUCE	IUCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IUCE	Current data register update channel common interrupt enable 0: Do not output capture current data register update channel common interrupt 1: Capture current data register update channel common interrupt can be output	R/W
1	IAUCE	Capture current data register A update channel common interrupt enable 0: Do not output capture current data register A update channel common interrupt 1: Capture current data register A update channel common interrupt can be output	R/W
2	IBUCE	Capture current data register B update channel common interrupt enable 0: Do not output capture current data register B update channel common interrupt 1: Capture current data register B update channel common interrupt can be output	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

40.3.1.10 DSCSTRTR : Channel Software Start Trigger Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	STRTRG2	STRTRG1	STRTRG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STRTRG0	Channel 0 start trigger 0: Do nothing 1: Start channel	R/W
1	STRTRG1	Channel 1 start trigger 0: Do nothing 1: Start channel	R/W
2	STRTRG2	Channel 2 start trigger 0: Do nothing 1: Start channel	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

40.3.1.11 DSCSTPTR : Channel Software Stop Trigger Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x204

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	STPTRG2	STPTRG1	STPTRG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STPTRG0	Channel 0 stop trigger 0: Do nothing 1: Stop channel	R/W
1	STPTRG1	Channel 1 stop trigger 0: Do nothing 1: Stop channel	R/W
2	STPTRG2	Channel 2 stop trigger 0: Do nothing 1: Stop channel	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

40.3.1.12 DSCESR : Channel Error Status Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x220

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SUMERRH	SUMERRL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SCF2	SCF1	SCF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SCF0	Channel 0 short circuit detection flag This bit becomes 1 when 1 is continuously input to the MDATm0 pin, exceeding the number of times set in DSSCTSRCH0.SCNTH, or when 0 is continuously input to the MDATm0 pin, exceeding the number of times set in DSSCTSRCH0.SCNTH. This bit becomes 0 when SCF0 = 1 and writing 1 to the channel 0 short circuit detection flag clear (DSCESCR.CLRSCF0 / DSCSCRCH0.CLRSCF). 0: Indicates that the MDATm0 pin short circuit is not detected 1: Indicates that a short circuit on the MDATm0 pin has been detected	R
1	SCF1	Channel 1 short circuit detection flag This bit becomes 1 when 1 is continuously input to the MDATm1 pin, exceeding the number of times set in DSSCTSRCH1.SCNTH, or when 0 is continuously input to the MDATm1 pin, exceeding the number of times set in DSSCTSRCH1.SCNTH. This bit becomes 0 when SCF1 = 1 and writing 1 to the channel 1 short circuit detection flag clear (DSCESCR.CLRSCF1 / DSCSCRCH1.CLRSCF). 0: Indicates that the MDATm1 pin short circuit is not detected 1: Indicates that a short circuit on the MDATm1 pin has been detected	R
2	SCF2	Channel 2 short circuit detection flag This bit becomes 1 when 1 is continuously input to the MDATm2 pin, exceeding the number of times set in DSSCTSRCH2.SCNTH, or when 0 is continuously input to the MDATm2 pin, exceeding the number of times set in DSSCTSRCH2.SCNTH. This bit becomes 0 when SCF2 = 1 and writing 1 to the channel 2 short circuit detection flag clear (DSCESCR.CLRSCF2 / DSCSCRCH2.CLRSCF). 0: Indicates that the MDATm2 pin short circuit is not detected 1: Indicates that a short circuit on the MDATm2 pin has been detected	R
15:3	—	These bits are read as 0.	R
16	SUMERRL	Overcurrent sum error lower limit detection flag This bit becomes 1 when the sum value of the overcurrent data is updated and this sum value is smaller than the lower limit (DSSELTR.SCMPTBL). This bit becomes 0 when SUMERRL = 1 and writing 1 to the overcurrent sum error lower limit detection flag clear (DSCESCR.CLRSUMERRL). 0: Indicates that the sum value of overcurrent data is not less than the lower limit 1: Indicates that the sum value of overcurrent data is less than the lower limit	R
17	SUMERRH	Overcurrent sum error upper limit detection flag This bit becomes 1 when the sum value of the overcurrent data is updated and this sum value is larger than the upper limit (DSSEHTR.SCMPTBH). This bit becomes 0 when SUMERRH = 1 and writing 1 to the overcurrent sum error upper limit detection flag clear (DSCESCR.CLRSUMERRH). 0: Indicates that the sum value of overcurrent data is not more than the upper limit 1: Indicates that the sum value of overcurrent data is more than the upper limit	R
31:18	—	These bits are read as 0.	R

40.3.1.13 DSCOCESR : Channel Overcurrent Error Status Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x224

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	OC2FH2	OC2FH1	OC2FH0	—	OC2FL2	OC2FL1	OC2FL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	OC1FH2	OC1FH1	OC1FH0	—	OC1FL2	OC1FL1	OC1FL0	—	OC0FH2	OC0FH1	OC0FH0	—	OC0FL2	OC0FL1	OC0FL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OC0FL0	Channel 0 overcurrent lower limit detection 0 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 0. This bit becomes 0 when OC0FL0 = 1 and writing 1 to the channel 0 overcurrent lower limit detection flag 0 clear (DSCOCESCR.CLROC0FL0 / DSCSCRCH0.CLROC0FL). 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
1	OC0FL1	Channel 1 overcurrent lower limit detection 0 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 0. This bit becomes 0 when OC0FL1 = 1 and writing 1 to the channel 1 overcurrent lower limit detection flag 0 clear (DSCOCESCR.CLROC0FL1 / DSCSCRCH1.CLROC0FL). 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
2	OC0FL2	Channel 2 overcurrent lower limit detection 0 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 0. This bit becomes 0 when OC0FL2 = 1 and writing 1 to the channel 2 overcurrent lower limit detection flag 0 clear (DSCOCESCR.CLROC0FL2 / DSCSCRCH2.CLROC0FL). 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
3	—	This bit is read as 0.	R
4	OC0FH0	Channel 0 overcurrent upper limit exceeded 0 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 0. This bit becomes 0 when OC0FH0 = 1 and writing 1 to the channel 0 overcurrent upper limit exceeded flag 0 clear (DSCOCESCR.CLROC0FH0 / DSCSCRCH0.CLROC0FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
5	OC0FH1	Channel 1 overcurrent upper limit exceeded 0 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 0. This bit becomes 0 when OC0FH1 = 1 and writing 1 to the channel 1 overcurrent upper limit exceeded flag 0 clear (DSCOCESCR.CLROC0FH1 / DSCSCRCH1.CLROC0FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
6	OC0FH2	Channel 2 overcurrent upper limit exceeded 0 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 0. This bit becomes 0 when OC0FH2 = 1 and writing 1 to the channel 2 overcurrent upper limit exceeded flag 0 clear (DSCOCESCR.CLROC0FH2 / DSCSCRCH2.CLROC0FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
7	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
8	OC1FL0	Channel 0 overcurrent lower limit detection 1 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 1. This bit becomes 0 when OC1FL0 = 1 and writing 1 to the channel 0 overcurrent lower limit detection flag 1 clear (DSCOCESCR.CLROC1FL0 / DSCSCRCH0.CLROC1FL). 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
9	OC1FL1	Channel 1 overcurrent lower limit detection 1 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 1. This bit becomes 0 when OC1FL1 = 1 and writing 1 to the channel 1 overcurrent lower limit detection flag 1 clear (DSCOCESCR.CLROC1FL1 / DSCSCRCH1.CLROC1FL). 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
10	OC1FL2	Channel 2 overcurrent lower limit detection 1 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 1. This bit becomes 0 when OC1FL2 = 1 and writing 1 to the channel 2 overcurrent lower limit detection flag 1 clear (DSCOCESCR.CLROC1FL2 / DSCSCRCH2.CLROC1FL). 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
11	—	This bit is read as 0.	R
12	OC1FH0	Channel 0 overcurrent upper limit exceeded 1 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 1. This bit becomes 0 when OC1FH0 = 1 and writing 1 to the channel 0 overcurrent upper limit exceeded flag 1 clear (DSCOCESCR.CLROC1FH0 / DSCSCRCH0.CLROC1FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
13	OC1FH1	Channel 1 overcurrent upper limit exceeded 1 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 1. This bit becomes 0 when OC1FH1 = 1 and writing 1 to the channel 1 overcurrent upper limit exceeded flag 1 clear (DSCOCESCR.CLROC1FH1 / DSCSCRCH1.CLROC1FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
14	OC1FH2	Channel 2 overcurrent upper limit exceeded 1 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 1. This bit becomes 0 when OC1FH2 = 1 and writing 1 to the channel 2 overcurrent upper limit exceeded flag 1 clear (DSCOCESCR.CLROC1FH2 / DSCSCRCH2.CLROC1FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
15	—	This bit is read as 0.	R
16	OC2FL0	Channel 0 overcurrent lower limit detection 2 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 2. This bit becomes 0 when OC2FL0 = 1 and writing 1 to the channel 0 overcurrent lower limit detection flag 2 clear (DSCOCESCR.CLROC2FL0 / DSCSCRCH0.CLROC2FL). 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
17	OC2FL1	Channel 1 overcurrent lower limit detection 2 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 2. This bit becomes 0 when OC2FL1 = 1 and writing 1 to the channel 1 overcurrent lower limit detection flag 2 clear (DSCOCESCR.CLROC2FL1 / DSCSCRCH1.CLROC2FL). 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R

Bit	Symbol	Function	R/W
18	OC2FL2	Channel 2 overcurrent lower limit detection 2 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 2. This bit becomes 0 when OC2FL2 = 1 and writing 1 to the channel 2 overcurrent lower limit detection flag 2 clear (DSCOCESCR.CLROC2FL2 / DSCSCRCH2.CLROC2FL). 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
19	—	This bit is read as 0.	R
20	OC2FH0	Channel 0 overcurrent upper limit exceeded 2 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 2. This bit becomes 0 when OC2FH0 = 1 and writing 1 to the channel 0 overcurrent upper limit exceeded flag 0 clear (DSCOCESCR.CLROC2FH0 / DSCSCRCH0.CLROC2FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
21	OC2FH1	Channel 1 overcurrent upper limit exceeded 2 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 2. This bit becomes 0 when OC2FH1 = 1 and writing 1 to the channel 1 overcurrent upper limit exceeded flag 2 clear (DSCOCESCR.CLROC2FH1 / DSCSCRCH1.CLROC2FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
22	OC2FH2	Channel 2 overcurrent upper limit exceeded 2 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 2. This bit becomes 0 when OC2FH2 = 1 and writing 1 to the channel 2 overcurrent upper limit exceeded flag 2 clear (DSCOCESCR.CLROC2FH2 / DSCSCRCH2.CLROC2FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
31:23	—	These bits are read as 0.	R

40.3.1.14 DSCOCNSR : Channel Overcurrent Notification Status Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x228

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	OWD3N2	OWD3N1	OWD3N0	—	OWD2N2	OWD2N1	OWD2N0	—	OWD1N2	OWD1N1	OWD1N0	—	OWD0N2	OWD0N1	OWD0N0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OWD0N0	Channel 0 overcurrent detection window notification 0 flag This bit becomes 1 when the condition of overcurrent detection window notification 0 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD0N0 = 1 and writing 1 to the channel 0 overcurrent detection window notification flag 0 clear (DSCOCNSCR.CLROWD0N0 / DSCSCRCH0.CLROWD0). 0: Indicates that the condition of overcurrent detection window notification 0 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 0 was satisfied	R

Bit	Symbol	Function	R/W
1	OWD0N1	Channel 1 overcurrent detection window notification 0 flag This bit becomes 1 when the condition of overcurrent detection window notification 0 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD0N1 = 1 and writing 1 to the channel 1 overcurrent detection window notification flag 0 clear (DSCOCNSCR.CLROWD0N1 / DSCSCRCH1.CLROWD0). 0: Indicates that the condition of overcurrent detection window notification 0 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 0 was satisfied	R
2	OWD0N2	Channel 2 overcurrent detection window notification 0 flag This bit becomes 1 when the condition of overcurrent detection window notification 0 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD0N2 = 1 and writing 1 to the channel 2 overcurrent detection window notification flag 0 clear (DSCOCNSCR.CLROWD0N2 / DSCSCRCH2.CLROWD0). 0: Indicates that the condition of overcurrent detection window notification 0 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 0 was satisfied	R
3	—	This bit is read as 0.	R
4	OWD1N0	Channel 0 overcurrent detection window notification 1 flag This bit becomes 1 when the condition of overcurrent detection window notification 1 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD1N0 = 1 and writing 1 to the channel 0 overcurrent detection window notification flag 1 clear (DSCOCNSCR.CLROWD1N0 / DSCSCRCH0.CLROWD1). 0: Indicates that the condition of overcurrent detection window notification 1 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 1 was satisfied	R
5	OWD1N1	Channel 1 overcurrent detection window notification 1 flag This bit becomes 1 when the condition of overcurrent detection window notification 1 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD1N1 = 1 and writing 1 to the channel 1 overcurrent detection window notification flag 1 clear (DSCOCNSCR.CLROWD1N1 / DSCSCRCH1.CLROWD1). 0: Indicates that the condition of overcurrent detection window notification 1 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 1 was satisfied	R
6	OWD1N2	Channel 2 overcurrent detection window notification 1 flag This bit becomes 1 when the condition of overcurrent detection window notification 1 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD1N2 = 1 and writing 1 to the channel 2 overcurrent detection window notification flag 1 clear (DSCOCNSCR.CLROWD1N2 / DSCSCRCH2.CLROWD1). 0: Indicates that the condition of overcurrent detection window notification 1 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 1 was satisfied	R
7	—	This bit is read as 0.	R
8	OWD2N0	Channel 0 overcurrent detection window notification 2 flag This bit becomes 1 when the condition of overcurrent detection window notification 2 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD2N0 = 1 and writing 1 to the channel 0 overcurrent detection window notification flag 2 clear (DSCOCNSCR.CLROWD2N0 / DSCSCRCH0.CLROWD2). 0: Indicates that the condition of overcurrent detection window notification 2 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 2 was satisfied	R

Bit	Symbol	Function	R/W
9	OWD2N1	Channel 1 overcurrent detection window notification 2 flag This bit becomes 1 when the condition of overcurrent detection window notification 2 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD2N1 = 1 and writing 1 to the channel 1 overcurrent detection window notification flag 2 clear (DSCOCNSCR.CLROWD2N1 / DSCSCRCH1.CLROWD2). 0: Indicates that the condition of overcurrent detection window notification 2 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 2 was satisfied	R
10	OWD2N2	Channel 2 overcurrent detection window notification 2 flag This bit becomes 1 when the condition of overcurrent detection window notification 2 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD2N2 = 1 and writing 1 to the channel 2 overcurrent detection window notification flag 2 clear (DSCOCNSCR.CLROWD2N2 / DSCSCRCH2.CLROWD2). 0: Indicates that the condition of overcurrent detection window notification 2 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 2 was satisfied	R
11	—	This bit is read as 0.	R
12	OWD3N0	Channel 0 overcurrent detection window notification 3 flag This bit becomes 1 when the condition of overcurrent detection window notification 3 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD3N0 = 1 and writing 1 to the channel 0 overcurrent detection window notification flag 3 clear (DSCOCNSCR.CLROWD3N0 / DSCSCRCH0.CLROWD3). 0: Indicates that the condition of overcurrent detection window notification 3 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 3 was satisfied	R
13	OWD3N1	Channel 1 overcurrent detection window notification 3 flag This bit becomes 1 when the condition of overcurrent detection window notification 3 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD3N1 = 1 and writing 1 to the channel 1 overcurrent detection window notification flag 3 clear (DSCOCNSCR.CLROWD3N1 / DSCSCRCH1.CLROWD3). 0: Indicates that the condition of overcurrent detection window notification 3 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 3 was satisfied	R
14	OWD3N2	Channel 2 overcurrent detection window notification 3 flag This bit becomes 1 when the condition of overcurrent detection window notification 3 is satisfied. See section 40.4.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD3N2 = 1 and writing 1 to the channel 2 overcurrent detection window notification flag 3 clear (DSCOCNSCR.CLROWD3N2 / DSCSCRCH2.CLROWD3). 0: Indicates that the condition of overcurrent detection window notification 3 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 3 was satisfied	R
31:15	—	These bits are read as 0.	R

40.3.1.15 DSCOCRMR : Channel Overcurrent Comparator Result Monitor Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x22C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	OC2CMPH2	OC2CMPH1	OC2CMPH0	—	OC2CMPL2	OC2CMPL1	OC2CMPL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	OC1CMPH2	OC1CMPH1	OC1CMPH0	—	OC1CMPL2	OC1CMPL1	OC1CMPL0	—	OC0CMPH2	OC0CMPH1	OC0CMPH0	—	OC0CMPL2	OC0CMPL1	OC0CMPL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OC0CMPL0	Channel 0 overcurrent detect 0 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
1	OC0CMPL1	Channel 1 overcurrent detect 0 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
2	OC0CMPL2	Channel 2 overcurrent detect 0 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
3	—	This bit is read as 0.	R
4	OC0CMPH0	Channel 0 overcurrent detect 0 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
5	OC0CMPH1	Channel 1 overcurrent detect 0 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
6	OC0CMPH2	Channel 2 overcurrent detect 0 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
7	—	This bit is read as 0.	R
8	OC1CMPL0	Channel 0 overcurrent detect 1 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
9	OC1CMPL1	Channel 1 overcurrent detect 1 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
10	OC1CMPL2	Channel 2 overcurrent detect 1 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
11	—	This bit is read as 0.	R
12	OC1CMPH0	Channel 0 overcurrent detect 1 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
13	OC1CMPH1	Channel 1 overcurrent detect 1 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
14	OC1CMPH2	Channel 2 overcurrent detect 1 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R

Bit	Symbol	Function	R/W
15	—	This bit is read as 0.	R
16	OC2CMPL0	Channel 0 overcurrent detect 2 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
17	OC2CMPL1	Channel 1 overcurrent detect 2 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
18	OC2CMPL2	Channel 2 overcurrent detect 2 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
19	—	This bit is read as 0.	R
20	OC2CMPH0	Channel 0 overcurrent detect 2 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
21	OC2CMPH1	Channel 1 overcurrent detect 2 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
22	OC2CMPH2	Channel 2 overcurrent detect 2 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
31:23	—	These bits are read as 0.	R

The DSCOCRMR register has monitor bits that return the comparison result with the upper/lower limit table for each overcurrent detection every time the overcurrent data is updated. There is no status clear bit of this register because the result updates when the overcurrent data is updated.

40.3.1.16 DSCSR : Channel Status Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x240

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CBUF2	CBUF1	CBUF0	—	CAUF2	CAUF1	CAUF0	—	DUF2	DUF1	DUF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DUF0	Channel 0 Data Update flag This bit is the Current Data Register Update Interrupt Status bit. This bit becomes 0 when DUF0 = 1 and 1 to the Channel n Data Update Clear flag (DSCSCR.CLRDUF0/ DSCSCRCH0.CLRDUF) or when the Current Data Register Channel 0 (DSCDRCH0) is read. 0: Indicates that the Current Data register has not been updated 1: Indicates that the Current Data register has been updated	R
1	DUF1	Channel 1 Data Update flag This bit is the Current Data Register Update Interrupt Status bit. This bit becomes 0 when DUF1 = 1 and 1 to the Channel n Data Update Clear flag (DSCSCR.CLRDUF1/ DSCSCRCH1.CLRDUF) or when the Current Data Register Channel 1 (DSCDRCH1) is read. 0: Indicates that the Current Data register has not been updated 1: Indicates that the Current Data register has been updated	R

Bit	Symbol	Function	R/W
2	DUF2	Channel 2 Data Update flag This bit is the Current Data Register Update Interrupt Status bit. This bit becomes 0 when DUF2 = 1 and 1 to the Channel n Data Update Clear flag (DSCSCR.CLRDUF2/ DSCSCRCH2.CLRDUF) or when the Current Data Register Channel 2 (DSCDRCH2) is read. 0: Indicates that the Current Data register has not been updated 1: Indicates that the Current Data register has been updated	R
3	—	This bit is read as 0.	R
4	CAUF0	Channel 0 capture data A update flag This bit becomes 0 when CAUF0 = 1 and 1 to the capture data update flag A clear (DSCSCR.CLRCAUF00 / DSCESCRCH0.CLRCAUF) or when the capture current data register A channel 0 (DSCCDRACH0) is read. 0: Indicates that the capture register A has not been updated 1: Indicates that the capture register A has been updated	R
5	CAUF1	Channel 1 capture data A update flag This bit becomes 0 when CAUF1 = 1 and 1 to the capture data update flag A clear (DSCSCR.CLRCAUF01 / DSCESCRCH1.CLRCAUF) or when the capture current data register A channel 1 (DSCCDRACH1) is read. 0: Indicates that the capture register A has not been updated 1: Indicates that the capture register A has been updated	R
6	CAUF2	Channel 2 capture data A update flag This bit becomes 0 when CAUF2 = 1 and 1 to the capture data update flag A clear (DSCSCR.CLRCAUF02 / DSCESCRCH2.CLRCAUF) or when the capture current data register A channel 2 (DSCCDRACH2) is read. 0: Indicates that the capture register A has not been updated 1: Indicates that the capture register A has been updated	R
7	—	This bit is read as 0.	R
8	CBUF0	Channel 0 capture data B update flag This bit becomes 0 when CBUF0 = 1 and 1 to the capture data update flag B clear (DSCSCR.CLRCBUF00 / DSCESCRCH0.CLRCBUF) or when the capture current data register B channel 0 (DSCCDRBCH0) is read. 0: Indicates that the capture register B has not been updated 1: Indicates that the capture register B has been updated	R
9	CBUF1	Channel 1 capture data B update flag This bit becomes 0 when CBUF1 = 1 and 1 to the capture data update flag B clear (DSCSCR.CLRCBUF01 / DSCESCRCH1.CLRCBUF) or when the capture current data register B channel 1 (DSCCDRBCH1) is read. 0: Indicates that the capture register B has not been updated 1: Indicates that the capture register B has been updated	R
10	CBUF2	Channel 2 capture data B update flag This bit becomes 0 when CBUF2 = 1 and 1 to the capture data update flag B clear (DSCSCR.CLRCBUF02 / DSCESCRCH2.CLRCBUF) or when the capture current data register B channel 2 (DSCCDRBCH2) is read. 0: Indicates that the capture register B has not been updated 1: Indicates that the capture register B has been updated	R
31:11	—	These bits are read as 0.	R

40.3.1.17 DSCSSR : Channel State Status Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x244

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CHSTATE2	—	—	—	CHSTATE1	—	—	—	CHSTATE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CHSTATE0	Channel 0 state 0: Channel stop 1: Channel in operation	R
3:1	—	These bits are read as 0.	R
4	CHSTATE1	Channel 1 state 0: Channel stop 1: Channel in operation	R
7:5	—	These bits are read as 0.	R
8	CHSTATE2	Channel 2 state 0: Channel stop 1: Channel in operation	R
31:9	—	These bits are read as 0.	R

40.3.1.18 DSCESCR : Channel Error Status Clear Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x260

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRSUMERRH	CLRSUMERRL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRSCF2	CLRSCF1	CLRSCF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLRSCF0	Channel 0 Overcurrent Lower Limit Detection Flag Clear 0: No effect 1: Clear DSCESR.OCFL0 / DSCSRCH0.OCFL	W
1	CLRSCF1	Channel 1 Overcurrent Lower Limit Detection Flag Clear 0: No effect 1: Clear DSCESR.OCFL1 / DSCSRCH1.OCFL	W
2	CLRSCF2	Channel 2 Overcurrent Lower Limit Detection Flag Clear 0: No effect 1: Clear DSCESR.OCFL2 / DSCSRCH2.OCFL	W

Bit	Symbol	Function	R/W
15:3	—	The write value should be 0.	W
16	CLRSUMERRL	Overcurrent Sum Error Lower Limit Detection Flag Clear 0: No effect 1: Clear DSCESR.SUMERRL	W
17	CLRSUMERRH	Overcurrent Sum Error Upper Limit Detection Flag Clear 0: No effect 1: Clear DSCESR.SUMERRH	W
31:18	—	The write value should be 0.	W

40.3.1.19 DSCOCESCR : Channel Overcurrent Error Status Clear Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x264

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	CLRO C2FH2	CLRO C2FH1	CLRO C2FH0	—	CLRO C2FL2	CLRO C2FL1	CLRO C2FL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CLRO C1FH2	CLRO C1FH1	CLRO C1FH0	—	CLRO C1FL2	CLRO C1FL1	CLRO C1FL0	—	CLRO C0FH2	CLRO C0FH1	CLRO C0FH0	—	CLRO C0FL2	CLRO C0FL1	CLRO C0FL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLROC0FL0	Channel 0 overcurrent lower limit detection 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FL0 / DSCSRCH0.OC0FL	W
1	CLROC0FL1	Channel 1 overcurrent lower limit detection 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FL1 / DSCSRCH1.OC0FL	W
2	CLROC0FL2	Channel 2 overcurrent lower limit detection 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FL2 / DSCSRCH2.OC0FL	W
3	—	The write value should be 0.	W
4	CLROC0FH0	Channel 0 overcurrent upper limit exceeded 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FH0 / DSCSRCH0.OC0FH	W
5	CLROC0FH1	Channel 1 overcurrent upper limit exceeded 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FH1 / DSCSRCH1.OC0FH	W
6	CLROC0FH2	Channel 2 overcurrent upper limit exceeded 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FH2 / DSCSRCH2.OC0FH	W
7	—	The write value should be 0.	W
8	CLROC1FL0	Channel 0 overcurrent lower limit detection 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FL0 / DSCSRCH0.OC1FL	W
9	CLROC1FL1	Channel 1 overcurrent lower limit detection 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FL1 / DSCSRCH1.OC1FL	W

Bit	Symbol	Function	R/W
10	CLROC1FL2	Channel 2 overcurrent lower limit detection 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FL2 / DSCSRCH2.OC1FL	W
11	—	The write value should be 0.	W
12	CLROC1FH0	Channel 0 overcurrent upper limit exceeded 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FH0 / DSCSRCH0.OC1FH	W
13	CLROC1FH1	Channel 1 overcurrent upper limit exceeded 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FH1 / DSCSRCH1.OC1FH	W
14	CLROC1FH2	Channel 2 overcurrent upper limit exceeded 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FH2 / DSCSRCH2.OC1FH	W
15	—	The write value should be 0.	W
16	CLROC2FL0	Channel 0 overcurrent lower limit detection 2 flag clear 0: No effect 1: Clear DSCOCESR.OC2FL0 / DSCSRCH0.OC2FL	W
17	CLROC2FL1	Channel 1 overcurrent lower limit detection 2 flag clear 0: No effect 1: Clear DSCOCESR.OC2FL1 / DSCSRCH1.OC2FL	W
18	CLROC2FL2	Channel 2 overcurrent lower limit detection 2 flag clear 0: No effect 1: Clear DSCOCESR.OC2FL2 / DSCSRCH2.OC2FL	W
19	—	The write value should be 0.	W
20	CLROC2FH0	Channel 0 overcurrent upper limit exceeded 2 flag clear 0: No effect 1: Clear DSCOCESR.OC2FH0 / DSCSRCH0.OC2FH	W
21	CLROC2FH1	Channel 1 overcurrent upper limit exceeded 2 flag 0: No effect 1: Clear DSCOCESR.OC2FH1 / DSCSRCH1.OC2FH	W
22	CLROC2FH2	Channel 2 overcurrent upper limit exceeded 2 flag clear 0: No effect 1: Clear DSCOCESR.OC2FH2 / DSCSRCH2.OC2FH	W
31:23	—	The write value should be 0.	W

40.3.1.20 DSCOCNSCR : Channel Overcurrent Notification Status Clear Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x268

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CLROWD3N2	CLROWD3N1	CLROWD3N0	—	CLROWD2N2	CLROWD2N1	CLROWD2N0	—	CLROWD1N2	CLROWD1N1	CLROWD1N0	—	CLROWD0N2	CLROWD0N1	CLROWD0N0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLROWD0N0	Channel 0 overcurrent detection window notification 0 flag clear 0: No effect 1: Clear DSCOCNSR.OWD0N0 / DSCSRCH0.OWD0N	W
1	CLROWD0N1	Channel 1 overcurrent detection window notification 0 flag clear 0: No effect 1: Clear DSCOCNSR.OWD0N1 / DSCSRCH1.OWD0N	W
2	CLROWD0N2	Channel 2 overcurrent detection window notification 0 flag clear 0: No effect 1: Clear DSCOCNSR.OWD0N2 / DSCSRCH2.OWD0N	W
3	—	The write value should be 0.	W
4	CLROWD1N0	Channel 0 overcurrent detection window notification 1 flag clear 0: No effect 1: Clear DSCOCNSR.OWD1N0 / DSCSRCH0.OWD1N	W
5	CLROWD1N1	Channel 1 overcurrent detection window notification 1 flag clear 0: No effect 1: Clear DSCOCNSR.OWD1N1 / DSCSRCH1.OWD1N	W
6	CLROWD1N2	Channel 2 overcurrent detection window notification 1 flag clear 0: No effect 1: Clear DSCOCNSR.OWD1N2 / DSCSRCH2.OWD1N	W
7	—	The write value should be 0.	W
8	CLROWD2N0	Channel 0 overcurrent detection window notification 2 flag clear 0: No effect 1: Clear DSCOCNSR.OWD2N0 / DSCSRCH0.OWD2N	W
9	CLROWD2N1	Channel 1 overcurrent detection window notification 2 flag clear 0: No effect 1: Clear DSCOCNSR.OWD2N1 / DSCSRCH1.OWD2N	W
10	CLROWD2N2	Channel 2 overcurrent detection window notification 2 flag clear 0: No effect 1: Clear DSCOCNSR.OWD2N2 / DSCSRCH2.OWD2N	W
11	—	The write value should be 0.	W
12	CLROWD3N0	Channel 0 overcurrent detection window notification 3 flag clear 0: No effect 1: Clear DSCOCNSR.OWD3N0 / DSCSRCH0.OWD3N	W
13	CLROWD3N1	Channel 1 overcurrent detection window notification 3 flag clear 0: No effect 1: Clear DSCOCNSR.OWD3N1 / DSCSRCH1.OWD3N	W
14	CLROWD3N2	Channel 2 overcurrent detection window notification 3 flag clear 0: No effect 1: Clear DSCOCNSR.OWD3N2 / DSCSRCH2.OWD3N	W
31:15	—	The write value should be 0.	W

40.3.1.21 DSCSCR : Channel Status Clear Register

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x280

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CLRCBUF2	CLRCBUF1	CLRCBUF0	—	CLRCAUF2	CLRCAUF1	CLRCAUF0	—	CLRDUF2	CLRDUF1	CLRDUF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLRDUF0	Channel 0 Data Update Flag Clear 0: No effect 1: Clear DSCSR.DUF0 / DSCSRCH0.DUF	W
1	CLRDUF1	Channel 1 Data Update Flag Clear 0: No effect 1: Clear DSCSR.DUF1 / DSCSRCH1.DUF	W
2	CLRDUF2	Channel 2 Data Update Flag Clear 0: No effect 1: Clear DSCSR.DUF2 / DSCSRCH2.DUF	W
3	—	The write value should be 0.	W
4	CLRCAUF0	Channel 0 Capture Data A Update Flag Clear 0: No effect 1: Clear DSCSR.CAUF0 / DSCSRCH0.CAUF	W
5	CLRCAUF1	Channel 1 Capture Data A Update Flag Clear 0: No effect 1: Clear DSCSR.CAUF1 / DSCSRCH1.CAUF	W
6	CLRCAUF2	Channel 2 Capture Data A Update Flag Clear 0: No effect 1: Clear DSCSR.CAUF2 / DSCSRCH2.CAUF	W
7	—	The write value should be 0.	W
8	CLRCBUF0	Channel 0 Capture Data B Update Flag Clear 0: No effect 1: Clear DSCSR.CBUF0 / DSCSRCH0.CBUF	W
9	CLRCBUF1	Channel 1 Capture Data B Update Flag Clear 0: No effect 1: Clear DSCSR.CBUF1 / DSCSRCH1.CBUF	W
10	CLRCBUF2	Channel 2 Capture Data B Update Flag Clear 0: No effect 1: Clear DSCSR.CBUF2 / DSCSRCH2.CBUF	W
31:11	—	The write value should be 0.	W

40.3.2 DSMIF Channel Control Registers

40.3.2.1 DSICRCHn : Interrupt Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x300 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	OWNE3	OWNE2	OWNE1	OWNE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IOEH2	IOEL2	IOEH1	IOEL1	IOEH0	IOEL0	—	—	—	—	ISE	IBUE	IAUE	IUE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IUE	Current data register update interrupt enable 0: Do not output current data register update interrupt 1: Current data register update interrupt can be output	R/W
1	IAUE	Capture current data register A update interrupt enable 0: Do not output capture current data register A update interrupt 1: Capture current data register A update interrupt can be output	R/W
2	IBUE	Capture current data register B update interrupt enable 0: Do not output capture current data register B update interrupt 1: Capture current data register B update interrupt can be output	R/W
3	ISE	Short circuit detection error interrupt enable bit 0: Do not output short circuit detection interrupt 1: Short circuit detection interrupt can be output	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	IOEL0	Overcurrent lower limit detection interrupt 0 0: Do not output the overcurrent lower limit detection interrupt 0 1: Overcurrent lower limit detection interrupt 0 can be output	R/W
9	IOEH0	Overcurrent upper limit exceeded detection interrupt 0 0: Do not output the overcurrent upper limit detection interrupt 0 1: Overcurrent upper limit exceeded detection interrupt 0 can be output	R/W
10	IOEL1	Overcurrent lower limit detection interrupt 1 0: Do not output the overcurrent lower limit detection interrupt 1 1: Overcurrent lower limit detection interrupt 1 can be output	R/W
11	IOEH1	Overcurrent upper limit exceeded detection interrupt 1 0: Do not output the overcurrent upper limit detection interrupt 1 1: Overcurrent upper limit exceeded detection interrupt 1 can be output	R/W
12	IOEL2	Overcurrent lower limit detection interrupt 2 0: Do not output the overcurrent lower limit detection interrupt 2 1: Overcurrent lower limit detection interrupt 2 can be output	R/W
13	IOEH2	Overcurrent upper limit exceeded detection interrupt 2 0: Do not output the overcurrent upper limit detection interrupt 2 1: Overcurrent upper limit exceeded detection interrupt 2 can be output	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	OWNE0	Overcurrent detection window notification 0 output enable 0: Do not output overcurrent detection window notification 0 1: Overcurrent detection window notification 0 can be output	R/W

Bit	Symbol	Function	R/W
17	OWNE1	Overcurrent detection window notification 1 output enable 0: Do not output overcurrent detection window notification 1 1: Overcurrent detection window notification 1 can be output	R/W
18	OWNE2	Overcurrent detection window notification 2 output enable 0: Do not output overcurrent detection window notification 2 1: Overcurrent detection window notification 2 can be output	R/W
19	OWNE3	Overcurrent detection window notification 3 output enable 0: Do not output overcurrent detection window notification 3 1: Overcurrent detection window notification 3 can be output	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.2 DSCMCCRCHn : Current Measurement Clock Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x304 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CKDIV[5:0]						SEDGE	—	—	—	—	—	—	CKDIR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CKDIR	A/D conversion clock master/slave switching 0: MCLKmn pin is input (slave operation) 1: MCLKmn pin is output (master operation)	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	SEDGE	Sampling edge selection 0: Capture MDATmn at the negative edge of MCLKmn 1: Capture MDATmn at the positive edge of MCLKmn	R/W
13:8	CKDIV[5:0]	A/D conversion clock division ratio Set the division ratio in clock master mode. Set to a division ratio of 2 × (CKDIV[5:0] + 1) with the core clock	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.3 DSCMFCRCHn : Current Measurement Filter Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x308 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CMSH[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMDEC[7:0]								—	—	—	—	—	—	CMSINC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMSINC[1:0]	Current measurement filter order setting 0 0: Sinc3 (3rd order) 0 1: Sinc1 (1st order) 1 0: Sinc2 (2nd order) 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	CMDEC[7:0]	Decimation ratio selection for current measurement M indicates the decimation ratio. Decimation ratio M is set as follows: M = CMDEC + 1 The decimation ratio M can be set within the following range. 4 ≤ M ≤ 256 (0x3 ≤ CMDEC ≤ 0xFF) Setting less than M = 4 is prohibited. When these value is set, Sinc filter operation is not guaranteed.	R/W
20:16	CMSH[4:0]	Data shift setting for current measurement*1 0x00: [23:8] 0x01: [22:7] 0x02: [21:6] 0x03: [20:5] 0x04: [19:4] 0x05: [18:3] 0x06: [17:2] 0x07: [16:1] 0x08: [15:0] 0x09: [14:0]<<1 0x0A: [13:0]<<2 0x0B: [12:0]<<3 0x0C: [11:0]<<4 0x0D: [10:0]<<5 0x0E: [9:0]<<6 0x0F: [8:0]<<7 0x10: [7:0]<<8 0x11: [6:0]<<9 0x12: [5:0]<<10 0x13: [4:0]<<11 Others: Setting prohibited	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In CMSH[4:0] = 0x09 to 0x13, the lower bits are filled with 0s.

40.3.2.4 DSCMCTCRCHn : Current Measurement Capture Trigger Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x30C + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DEDGE	—	—	—	—	—	DITSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CTSELB[2:0]			—	—	—	—	—	CTSELA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CTSELA[2:0]	Current capture trigger A selection bit 0x0: Do not capture 0x1: Select current data capture trigger 0 0x2: Select current data capture trigger 1 0x3: Select current data capture trigger 2 0x4: Select current data capture trigger 3 0x5: Select current data capture trigger 4 0x6: Select current data capture trigger 5 0x7: Setting prohibited (do not capture)	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	CTSELB[2:0]	Current capture trigger B selection bit 0x0: Do not capture 0x1: Select current data capture trigger 0 0x2: Select current data capture trigger 1 0x3: Select current data capture trigger 2 0x4: Select current data capture trigger 3 0x5: Select current data capture trigger 4 0x6: Select current data capture trigger 5 0x7: Setting prohibited (do not capture)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
17:16	DITSEL[1:0]	Current measurement filter initialization trigger selection bit for frequency division counter for decimation. 0 0: Do not initialize 0 1: Select decimation dividing counter initialization trigger 0 1 0: Select decimation dividing counter initialization trigger 1 1 1: Select decimation dividing counter initialization trigger 2	R/W
22:18	—	These bits are read as 0. The write value should be 0.	R/W
23	DEDGE	Current measurement filter initialization trigger for division counter for decimation edge selection bit. The trigger from ELC is usually used positive edge. Change from the initial value if necessary. 0: The frequency division counter for decimation is initialized at the negative edge of initialization trigger 1: The frequency division counter for decimation is initialized at the positive edge of initialization trigger	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.5 DSEDCRCHn : Error Detect Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x320 + 0x140 × n

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SDE	Short circuit detection enable bit 0: Short circuit detection is invalid 1: Short circuit detection is valid	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.6 DSOCFCRCHn : Overcurrent Detect Filter Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x330 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	OCSH[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OCDEC[7:0]								—	—	—	—	—	—	OCSINC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OCSINC[1:0]	Overcurrent detection filter order setting 0 0: Sinc3 (3rd order) 0 1: Sinc1 (1st order) 1 0: Sinc2 (2nd order) 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	OCDEC[7:0]	Decimation ratio selection for overcurrent detection M indicates the decimation ratio. Decimation ratio M is set as follows: M = OCDEC + 1 The decimation ratio M can be set within the following range. 4 ≤ M ≤ 256 (0x3 ≤ OCDEC ≤ 0xFF) Setting less than M = 4 is prohibited. When these value is set, Sinc filter operation is not guaranteed.	R/W

Bit	Symbol	Function	R/W
20:16	OCSH[4:0]	Data shift setting for overcurrent detection*1 0x00: [23:8] 0x01: [22:7] 0x02: [21:6] 0x03: [20:5] 0x04: [19:4] 0x05: [18:3] 0x06: [17:2] 0x07: [16:1] 0x08: [15:0] 0x09: [14:0]<<1 0x0A: [13:0]<<2 0x0B: [12:0]<<3 0x0C: [11:0]<<4 0x0D: [10:0]<<5 0x0E: [9:0]<<6 0x0F: [8:0]<<7 0x10: [7:0]<<8 0x11: [6:0]<<9 0x12: [5:0]<<10 0x13: [4:0]<<11 Others: Setting prohibited	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In OCSH[4:0] = 0x09 to 0x13, the lower bits are filled with 0s.

40.3.2.7 DSSCTSRCHn : Short Circuit Threshold Setting Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x324 + 0x140 × n

Bit position:	31			28												16			12												0
Bit field:	—	—	—	SCNTH[12:0]												—	—	—	SCNTL[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	SCNTL[12:0]	Short circuit detection low continuous detection count Set the upper limit for the number of times 0 is input continuously to the MDATmn pin.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	SCNTH[12:0]	Short circuit detection high continuous detection count Set the upper limit for the number of times 1 is input continuously to the MDATmn pin.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.8 DSODCRCHn : Overcurrent Detect Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x334 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	OWFE3	OWFE2	OWFE1	OWFE0	—	—	ODEH2	ODEL2	ODEH1	ODEL1	ODEH0	ODEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ODEL0	Overcurrent lower limit detection 0 enable 0: Overcurrent lower limit detection 0 is invalid 1: Overcurrent lower limit detection 0 is valid	R/W
1	ODEH0	Overcurrent upper limit exceeded detection 0 enable 0: Overcurrent upper limit exceeded detection 0 is invalid 1: Overcurrent upper limit exceeded detection 0 is valid	R/W
2	ODEL1	Overcurrent lower limit detection 1 enable 0: Overcurrent lower limit detection 1 is invalid 1: Overcurrent lower limit detection 1 is valid	R/W
3	ODEH1	Overcurrent upper limit exceeded detection 1 enable 0: Overcurrent upper limit exceeded detection 1 is invalid 1: Overcurrent upper limit exceeded detection 1 is valid	R/W
4	ODEL2	Overcurrent lower limit detection 2 enable 0: Overcurrent lower limit detection 2 is invalid 1: Overcurrent lower limit detection 2 is valid	R/W
5	ODEH2	Overcurrent upper limit exceeded detection 2 enable 0: Overcurrent upper limit exceeded detection 2 is invalid 1: Overcurrent upper limit exceeded detection 2 is valid	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	OWFE0	Overcurrent lower limit detection interrupt 0 0: Do not output the overcurrent lower limit detection interrupt 0 1: Overcurrent lower limit detection interrupt 0 can be output	R/W
9	OWFE1	Overcurrent detection window function 0 enable 0: Overcurrent detection window function 0 is invalid 1: Overcurrent detection window function 0 is valid	R/W
10	OWFE2	Overcurrent detection window function 1 enable 0: Overcurrent detection window function 1 is invalid 1: Overcurrent detection window function 1 is valid	R/W
11	OWFE3	Overcurrent detection window function 2 enable 0: Overcurrent detection window function 2 is invalid 1: Overcurrent detection window function 2 is valid	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.9 DSODWCRCHn : Overcurrent Detect Window Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x338 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	OWNM3[3:0]				OWNM2	OWNM1	OWNM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OWNM0	Channel n overcurrent detection window notification 0 mode select 0: (overcurrent data channel n < lower limit 0) or (overcurrent data channel n > upper limit 0) 1: lower limit 0 channel n ≤ overcurrent data channel n ≤ upper limit 0 channel n	R/W
1	OWNM1	Channel n overcurrent detection window notification 1 mode select 0: (overcurrent data channel n < lower limit 1) or (overcurrent data channel n > upper limit 1) 1: lower limit 1 channel n ≤ overcurrent data channel n ≤ upper limit 1 channel n	R/W
2	OWNM2	Channel n overcurrent detection window notification 2 mode select 0: (overcurrent data channel n < lower limit 2) or (overcurrent data channel n > upper limit 2) 1: lower limit 2 channel n ≤ overcurrent data channel n ≤ upper limit 2 channel n	R/W
6:3	OWNM3[3:0]	Channel n overcurrent detection window notification 3 mode select 0x0: Overcurrent detection window notification 0 0x1: Overcurrent detection window notification 1 0x2: Overcurrent detection window notification 2 0x3: Overcurrent detection window notification 0 OR overcurrent detection window notification 1 0x4: Overcurrent detection window notification 0 AND overcurrent detection window notification 1 0x5: Overcurrent detection window notification 0 OR overcurrent detection window notification 2 0x6: Overcurrent detection window notification 0 AND overcurrent detection window notification 1 0x7: Overcurrent detection window notification 1 OR overcurrent detection window notification 2 0x8: Overcurrent detection window notification 1 AND overcurrent detection window notification 2 0x9: Overcurrent detection window notification 0 OR overcurrent detection window notification 1 OR overcurrent detection window notification 2 0xA: Overcurrent detection window notification 0 AND overcurrent detection window notification 1 AND overcurrent detection window notification 2 Others: Setting prohibited	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.10 DSOCLTR0CHn : Overcurrent Low Threshold Register 0 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3A0 + 0x140 × n

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBL0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBL0[15:0]	Overcurrent detection lower limit 0 This bit sets the lower limit of the active current value for overcurrent detection 0.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.11 DSOCHTR0CHn : Overcurrent High Threshold Register 0 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3A4 + 0x140 × n

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBH0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBH0[15:0]	Overcurrent detection upper limit 0 This bit sets the upper limit of the active current value for overcurrent detection 0.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.12 DSOCLTR1CHn : Overcurrent Low Threshold Register 1 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3A8 + 0x140 × n

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBL1[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBL1[15:0]	Overcurrent detection lower limit 1 This bit sets the lower limit of the active current value for overcurrent detection 1.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.13 DSOCHTR1CHn : Overcurrent High Threshold Register 1 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3AC + 0x140 × n

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBH1[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBH1[15:0]	Overcurrent detection upper limit 1 This bit sets the upper limit of the active current value for overcurrent detection 1.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.14 DSOCLTR2CHn : Overcurrent Low Threshold Register 2 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3B0 + 0x140 × n

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBL2[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBL2[15:0]	Overcurrent detection lower limit 2 This bit sets the lower limit of the active current value for overcurrent detection 2.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.15 DSOCHTR2CHn : Overcurrent High Threshold Register 2 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3B4 + 0x140 × n

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBH2[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBH2[15:0]	Overcurrent detection upper limit 2 This bit sets the upper limit of the active current value for overcurrent detection 2.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.16 DSCSTRTRCHn : Software Start Trigger Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3E0 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STRTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STRTRG	Channel start trigger Mirror register of DSCSTRTR.STRTRGn 0: Do nothing 1: Start channel n	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.17 DSCSTPTRCHn : Software Stop Trigger Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3E4 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STPTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STPTRG	Channel stop trigger Mirror register of DSCSTRTR.STPTRGn 0: Do nothing 1: Stop channel n	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

40.3.2.18 DSCDRCHn : Current Data Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3F0 + 0x140 × n

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADDR[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADDR[15:0]	Current data Stores the current value generated by the current measurement data block.	R
31:16	—	These bits are read as 0.	R

40.3.2.19 DSCCDRACHn : Capture Current Data Register A Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3F4 + 0x140 × n

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CDRA[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CDRA[15:0]	Capture current data A The current data value when the capture trigger A is input is stored.	R
31:16	—	These bits are read as 0.	R

40.3.2.20 DSCCDRBCHn : Capture Current Data Register B Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3F8 + 0x140 × n

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CDRB[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CDRB[15:0]	Capture current data B The current data value when the capture trigger B is input is stored.	R
31:16	—	These bits are read as 0.	R

40.3.2.21 DSOCDRCHn : Overcurrent Data Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x3FC + 0x140 × n

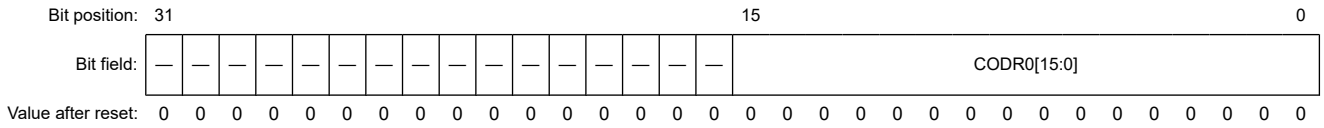
Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ODR[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ODR[15:0]	Overcurrent data Stores the current value generated by the overcurrent data block.	R
31:16	—	These bits are read as 0.	R

40.3.2.22 DSCOCDR0CHn : Capture Overcurrent Data Register 0 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x400 + 0x140 × n

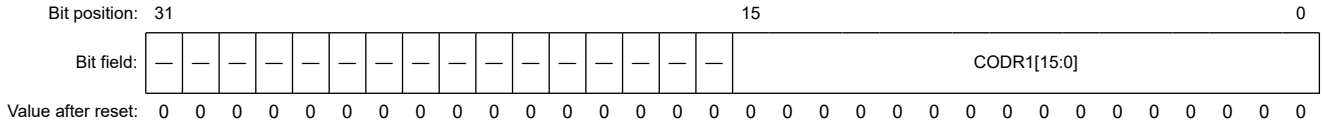


Bit	Symbol	Function	R/W
15:0	CODR0[15:0]	Capture overcurrent data 0 When either the overcurrent upper limit or the overcurrent lower limit is detected by the overcurrent data block, the overcurrent data 0 used for comparison is stored.	R
31:16	—	These bits are read as 0.	R

40.3.2.23 DSCOCDR1CHn : Capture Overcurrent Data Register 1 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x404 + 0x140 × n

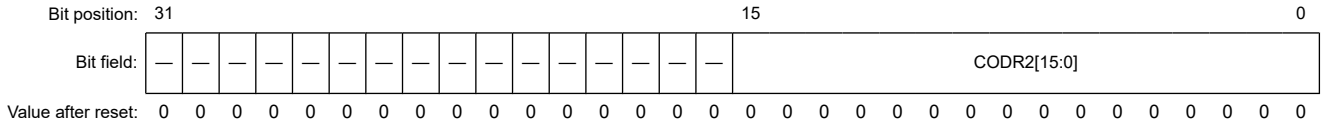


Bit	Symbol	Function	R/W
15:0	CODR1[15:0]	Capture overcurrent data 1 When either the overcurrent upper limit or the overcurrent lower limit is detected by the overcurrent data block, the overcurrent data 1 used for comparison is stored.	R
31:16	—	These bits are read as 0.	R

40.3.2.24 DSCOCDR2CHn : Capture Overcurrent Data Register 2 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x408 + 0x140 × n



Bit	Symbol	Function	R/W
15:0	CODR2[15:0]	Capture overcurrent data 2 When either the overcurrent upper limit or the overcurrent lower limit is detected by the overcurrent data block, the overcurrent data 2 used for comparison is stored.	R
31:16	—	These bits are read as 0.	R

40.3.2.25 DSCSRCHn : Status Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x420 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	OC2CMPH	OC2CMPL	OC1CMPH	OC1CMPL	OC0CMPH	OC0CMPL	—	—	—	—	OWD3N	OWD2N	OWD1N	OWD0N
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	OC2FH	OC2FL	OC1FH	OC1FL	OC0FH	OC0FL	CHSTATE	—	—	—	SCF	CBUF	CAUF	DUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DUF	Channel n data update flag Mirror register of DSCSR.DUFLn 0: Indicates that the current data register has not been updated 1: Indicates that the current data register has been updated	R
1	CAUF	Channel n capture data A update flag Mirror register of DSCSR.CAUFn 0: Indicates that the capture data register A has not been updated 1: Indicates that the capture data register A has been updated	R
2	CBUF	Channel n capture data B update flag Mirror register of DSCSR.CBUFn 0: Indicates that the capture data register B has not been updated 1: Indicates that the capture data register B has been updated	R
3	SCF	Channel n short circuit detection flag Mirror register of DSCESR.SCFn 0: Indicates that the MDATmn pin short circuit is not detected 1: Indicates that a short circuit on the MDATmn pin has been detected	R
6:4	—	These bits are read as 0.	R
7	CHSTATE	Channel n state Mirror register of DSCSSR.CHSTATEn 0: Channel stop 1: Channel in operation	R
8	OC0FL	Channel n overcurrent lower limit detection 0 flag Mirror register of DSCOCESR.OC0FLn 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
9	OC0FH	Channel n overcurrent upper limit exceeded 0 flag Mirror register of DSCOCESR.OC0FHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
10	OC1FL	Channel n overcurrent lower limit detection 1 flag Mirror register of DSCOCESR.OC1FLn 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
11	OC1FH	Channel n overcurrent upper limit exceeded 1 flag Mirror register of DSCOCESR.OC1FHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
12	OC2FL	Channel n overcurrent lower limit detection 2 flag Mirror register of DSCOCESR.OC2FLn 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R

Bit	Symbol	Function	R/W
13	OC2FH	Channel n overcurrent upper limit exceeded 2 flag Mirror register of DSCOCESR.OC2FHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
15:14	—	These bits are read as 0.	R
16	OWD0N	Channel n overcurrent detection window notification 0 Mirror register of DSCOCNSR.OWD0Nn 0: Indicates that the condition of overcurrent detection window notification 0 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 0 was satisfied	R
17	OWD1N	Channel n overcurrent detection window notification 1 Mirror register of DSCOCNSR.OWD1Nn 0: Indicates that the condition of overcurrent detection window notification 1 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 1 was satisfied	R
18	OWD2N	Channel n overcurrent detection window notification 2 Mirror register of DSCOCNSR.OWD2Nn 0: Indicates that the condition of overcurrent detection window notification 2 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 2 was satisfied	R
19	OWD3N	Channel n overcurrent detection window notification 3 Mirror register of DSCOCNSR.OWD3Nn 0: Indicates that the condition of overcurrent detection window notification 3 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 3 was satisfied	R
23:20	—	These bits are read as 0.	R
24	OC0CMPL	Channel n overcurrent detect 0 lower limit compare result Mirror register of DSCOCRMR.OC0CMPLn 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has fallen the lower limit 0	R
25	OC0CMPH	Channel n overcurrent detect 0 upper limit compare result Mirror register of DSCOCRMR.OC0CMPHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
26	OC1CMPL	Channel n overcurrent detect 1 lower limit compare result Mirror register of DSCOCRMR.OC1CMPLn 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has fallen the lower limit 1	R
27	OC1CMPH	Channel n overcurrent detect 1 upper limit compare result Mirror register of DSCOCRMR.OC1CMPHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
28	OC2CMPL	Channel n overcurrent detect 2 lower limit compare result Mirror register of DSCOCRMR.OC2CMPLn 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has fallen the lower limit 2	R
29	OC2CMPH	Channel n overcurrent detect 2 upper limit compare result Mirror register of DSCOCRMR.OC2CMPHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
31:30	—	These bits are read as 0.	R

The DSCSRCHn register is a mirror register of various registers in section 40.3.1. DSMIF Common Control Registers.

40.3.2.26 DSCSCRCHn : Status Clear Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9002_0000 + 0x1000 × m (m = 0 to 5)
DSMIFi = 0x9012_0000 + 0x1000 × (i - 6) (i = 6 to 9)

Offset address: 0x430 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CLROWD3N	CLROWD2N	CLROWD1N	CLROWD0N
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CLROC2FH	CLROC2FL	CLROC1FH	CLROC1FL	CLROC0FH	CLROC0FL	—	—	—	—	CLRSCF	CLRCBUF	CLRCAUF	CLRDUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLRDUF	Channel n data update flag clear Mirror register of DSCSCR.CLRDUFn 0: Do nothing 1: Clear DSCSR.DUFn /DSCSRCHn.DUF	W
1	CLRCAUF	Channel n capture data A update flag clear Mirror register of DSCSCR.CLRCAUFn 0: Do nothing 1: Clear DSCSR.CAUFn /DSCSRCHn.CAUF	W
2	CLRCBUF	Channel n capture data B update flag clear Mirror register of DSCSCR.CLRCBUFn 0: Do nothing 1: Clear DSCSR.CBUFn /DSCSRCHn.CBUF	W
3	CLRSCF	Channel n short circuit detection flag clear Mirror register of DSCESCR.CLRSCFn 0: Do nothing 1: Clear DSCESR.SCFn / DSCSRCHn.SCF	W
7:4	—	The write value should be 0.	W
8	CLROC0FL	Channel n overcurrent lower limit detection flag 0 clear Mirror register of DSCOCESCR.CLROC0FLn 0: Do nothing 1: Clear DSCOCESR.OSC0FLn / DSCSRCHn.OC0FL	W
9	CLROC0FH	Channel n overcurrent upper limit exceeded flag 0 clear Mirror register of DSCOCESCR.CLROC0FHn 0: Do nothing 1: Clear DSCOCESR.OSC0FHn / DSCSRCHn.OC0FH	W
10	CLROC1FL	Channel n overcurrent lower limit detection flag 1 clear Mirror register of DSCOCESCR.CLROC1FLn 0: Do nothing 1: Clear DSCOCESR.OSC1FLn / DSCSRCHn.OC1FL	W
11	CLROC1FH	Channel n overcurrent upper limit exceeded flag 1 clear Mirror register of DSCOCESCR.CLROC1FHn 0: Do nothing 1: Clear DSCOCESR.OSC1FHn / DSCSRCHn.OC1FH	W
12	CLROC2FL	Channel n overcurrent lower limit detection flag 2 clear Mirror register of DSCOCESCR.CLROC2FLn 0: Do nothing 1: Clear DSCOCESR.OSC2FLn / DSCSRCHn.OC2FL	W
13	CLROC2FH	Channel n overcurrent upper limit exceeded flag 2 clear Mirror register of DSCOCESCR.CLROC2FHn 0: Do nothing 1: Clear DSCOCESR.OSC2FHn / DSCSRCHn.OC2FH	W

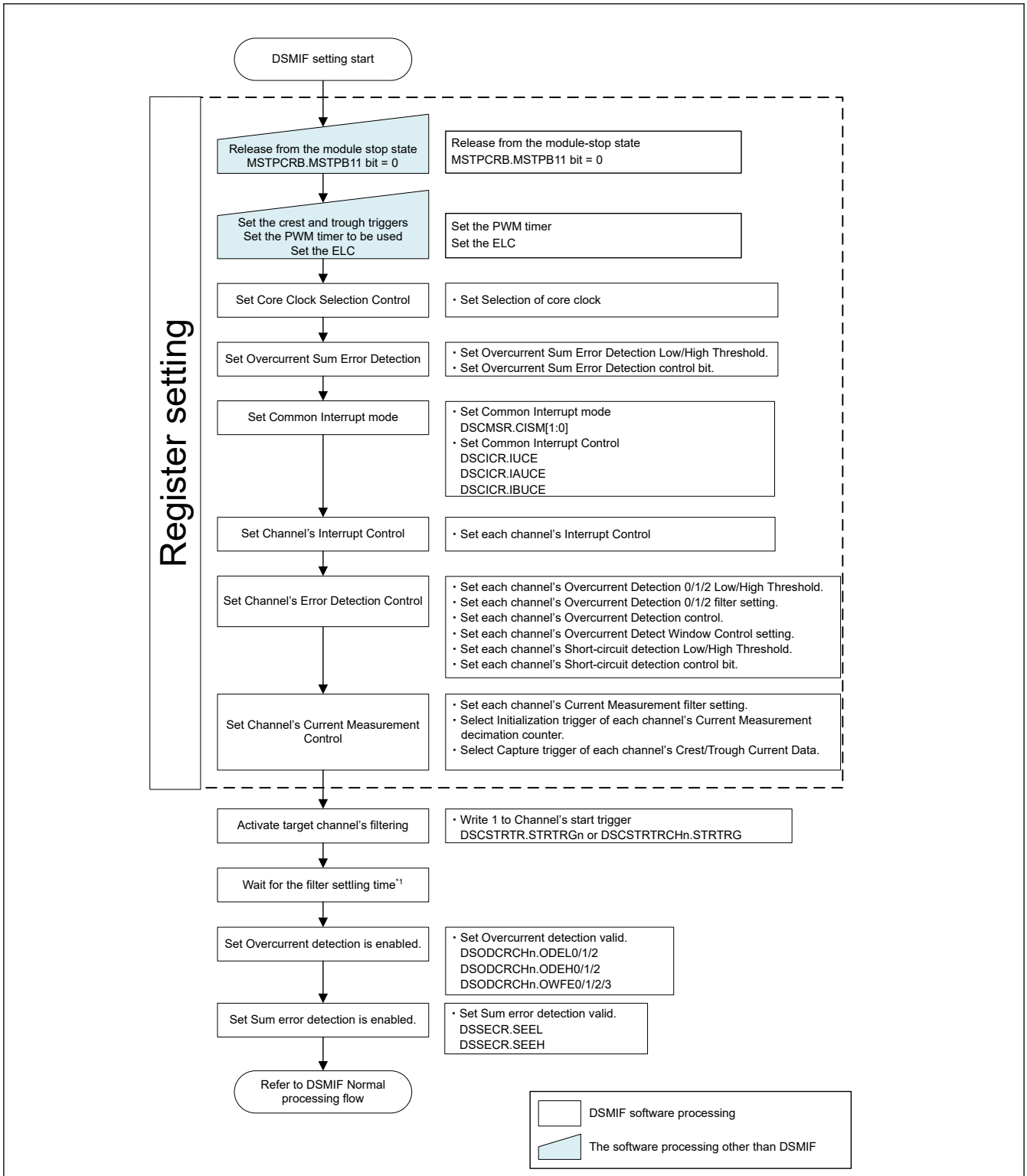
Bit	Symbol	Function	R/W
15:14	—	The write value should be 0.	W
16	CLROWD0N	Channel n overcurrent detection window notification 0 flag clear Mirror register of DSCOCNSCR.CLROWD0n 0: Do nothing 1: Clear DSCOCNSR.OWD0Nn / DSCSRCHn.OWD0N	W
17	CLROWD1N	Channel n overcurrent detection window notification 1 flag clear Mirror register of DSCOCNSCR.CLROWD1n 0: Do nothing 1: Clear DSCOCNSR.OWD1Nn / DSCSRCHn.OWD1N	W
18	CLROWD2N	Channel n overcurrent detection window notification 2 flag clear Mirror register of DSCOCNSCR.CLROWD2n 0: Do nothing 1: Clear DSCOCNSR.OWD2Nn / DSCSRCHn.OWD2N	W
19	CLROWD3N	Channel n overcurrent detection window notification 3 flag clear Mirror register of DSCOCNSCR.CLROWD3n 0: Do nothing 1: Clear DSCOCNSR.OWD3Nn / DSCSRCHn.OWD3N	W
31:20	—	The write value should be 0.	W

The DSCSCRCHn register is a mirror register of various registers in section 40.3.1. DSMIF Common Control Registers.

40.4 Operation

40.4.1 Start Flow

Figure 40.2 shows an example of a start procedure for DSMIF.

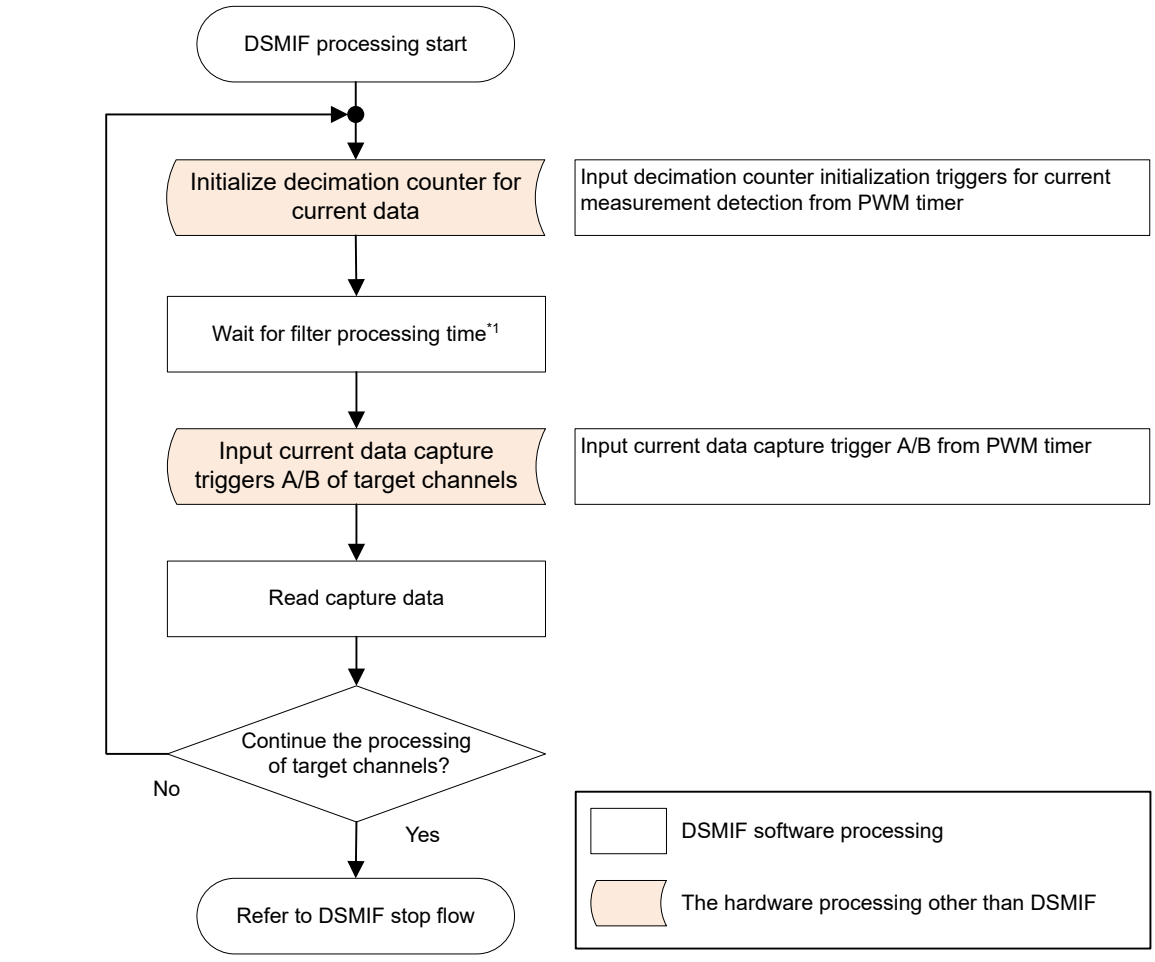


Note 1. See section 40.4.15.1. Settling Time of Channel Activation.

Figure 40.2 Start flow of DSMIF

40.4.2 Normal Processing Flow

Figure 40.3 shows an example of a normal processing procedure for DSMIF.



Note: The filter result during settling time is invalid. It is prohibited to use this data for capture data.
Note 1. See section 40.4.15.2. Settling Time after PWM Synchronization.

Figure 40.3 Normal processing flow for DSMIF

40.4.3 Stop Flow

Figure 40.4 shows an example of a stop procedure in channel synchronization control mode for DSMIF. Figure 40.5 shows the example of the stop procedure in channel individual control mode of DSMIF.

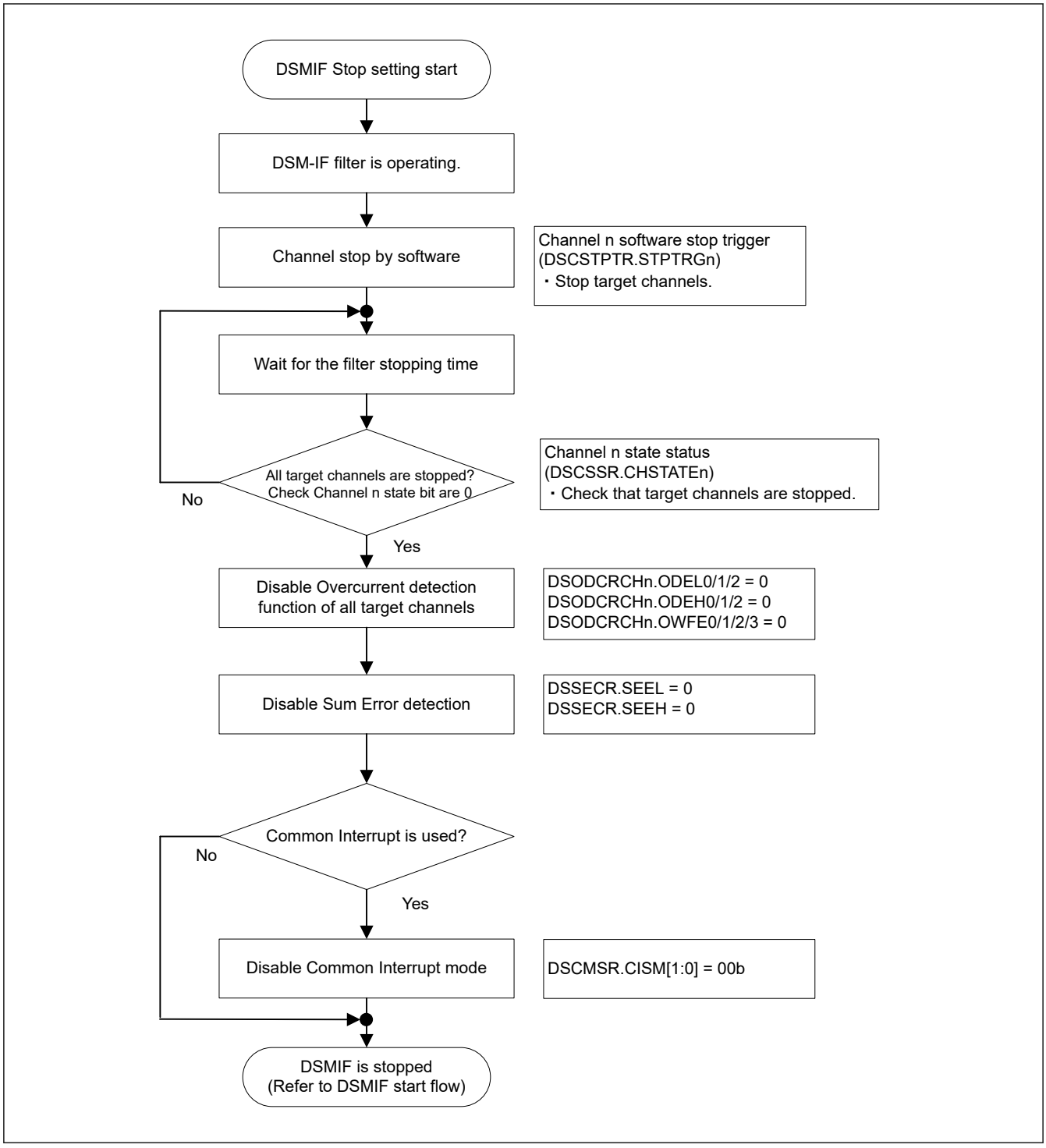


Figure 40.4 Stop flow for DSMIF (Channel synchronization control mode)

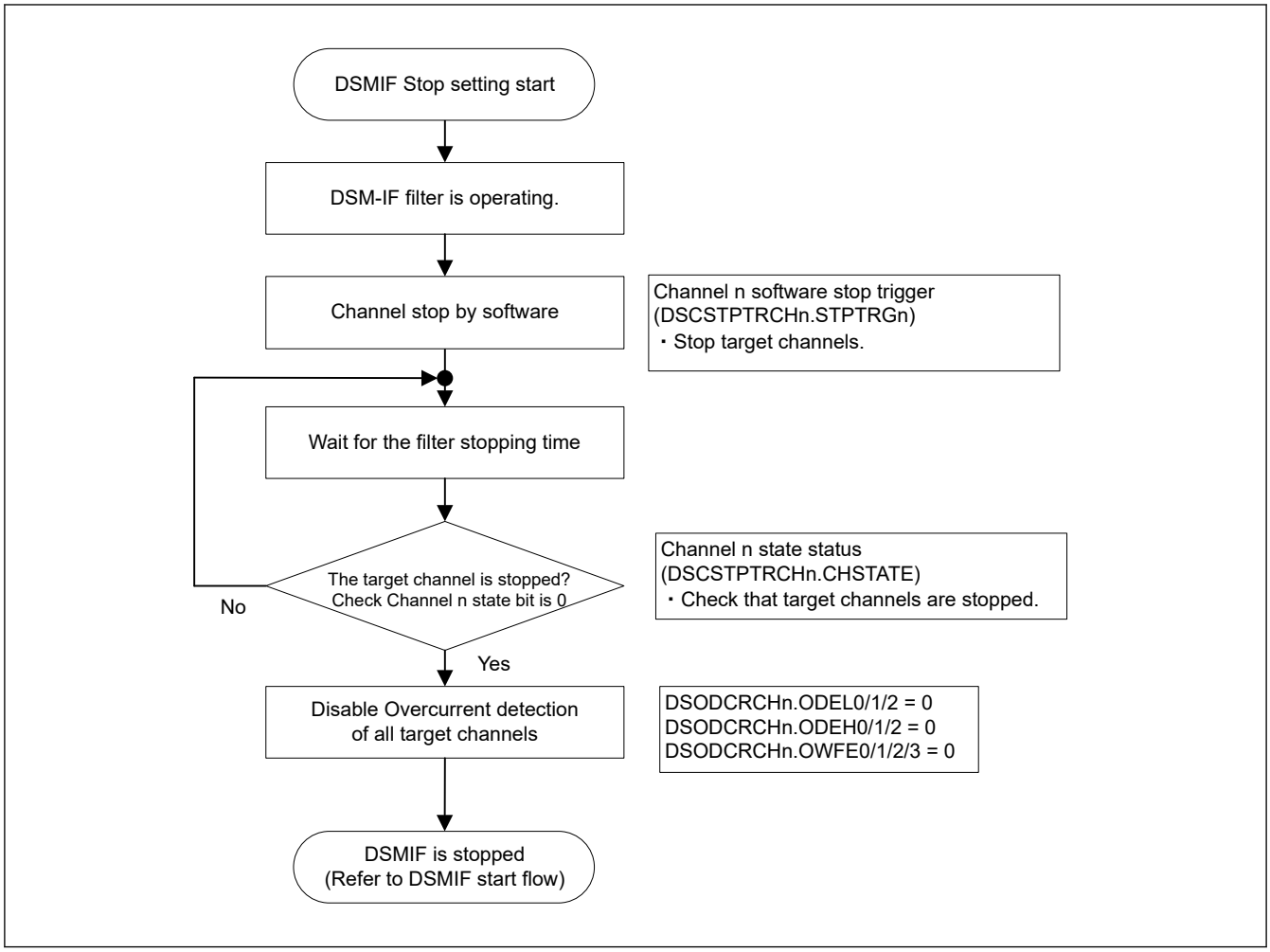
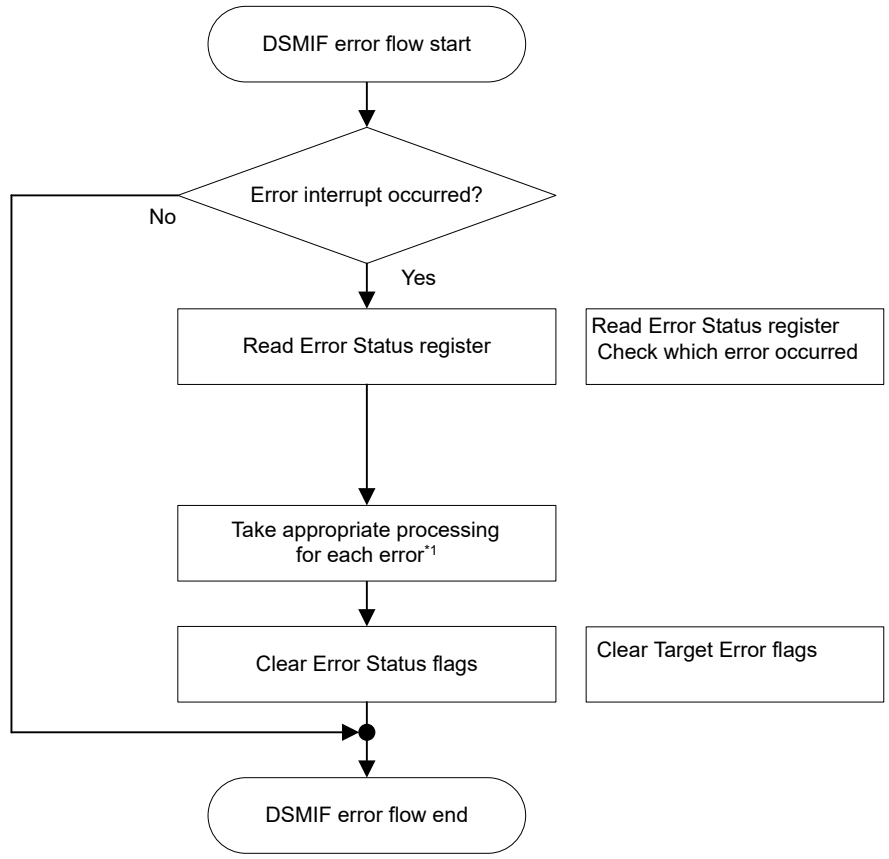


Figure 40.5 Stop flow for DSMIF (Channel individual control mode)

40.4.4 Error Flow

Figure 40.6 shows an example of an error procedure for DSMIF.



Note 1. The processing after each error detection depends on the user's system. Therefore, execute the error processing after the error detection according to each system.

Figure 40.6 Error flow for DSMIF

40.4.5 Clock Specification

When DSCMCCRCHn.CKDIR is set to 0 (clock slave mode), an external delta-sigma modulator clock is input to MCLKmn. When DSCMCCRCHn.CKDIR is set to 1 (clock master mode), the frequency-divided core clock is output to MCLKmn.

The frequency of the MCLKmn can be selected by the DSCCSCR.CLKSEL bit and the DSCMCCRCHn.CKDIV[5:0] bits.

Table 40.5 shows the clock division ratio of master mode in DSMIF.

Table 40.5 Clock division ratio of Master mode in DSMIF (1 of 2)

No.	Core clock (MHz)	MCLKmn (MHz)	Supported core clock division ratio	Setting value (DSCMCCRCHn.CKDIV[5:0])
1	250 (DSCCSCR.CLKSEL = 0b)	25	10	00_0100b
2		20.8	12	00_0101b
3		12.5	20	00_1001b
4		10.4	24	00_1011b
5		6.25	40	01_0011b
6		5	50	01_1000b

Table 40.5 Clock division ratio of Master mode in DSMIF (2 of 2)

No.	Core clock (MHz)	MCLKmn (MHz)	Supported core clock division ratio	Setting value (DSCMCCRCHn.CKDIV[5:0])
7	400 (DSCCSCR.CLKSEL = 1b)	25	16	00_0111b
8		20	20	00_1001b
9		12.5	32	00_1111b
10		10	40	01_0011b
11		6.25	64	01_1111b
12		5	80	10_0111b

40.4.6 Channel Start and Channel Stop

The channel start and channel stop are controlled by the register in Table 40.6. Set the channels to be synchronized to 1 simultaneously. Set the channel start according to section 40.4.1. Start Flow. Set the channel stop according to section 40.4.3. Stop Flow.

Table 40.6 Channel start trigger and channel stop trigger

Channel mode	Category	Register bit name
Channel synchronization control mode	Channel start	DSCSTRTR.STRTRG0
		DSCSTRTR.STRTRG1
		DSCSTRTR.STRTRG2
	Channel stop	DSCSTPTR.STPTRG0
		DSCSTPTR.STPTRG1
		DSCSTPTR.STPTRG2
Channel individual control mode	Channel start	DSCSTRTRCHn.STRTRG
	Channel stop	DSCSTPTRCHn.STPTRG

40.4.7 Channel Synchronization

To synchronize the Sinc filters of multiple channels, use the following procedure. To allow for multiple channels to be synchronized, follow the procedure in section 40.4.1. Start Flow.

1. Set the same clock settings.
2. Select the same trigger for division counter initialization trigger.
   Set the same value to Division counter initialization trigger select bits (DSCMCTCRCHn.DITSEL[1:0]).
3. Select the same trigger for capture trigger A and B.
   (a) Set the same value to the Current Capture Trigger A Select bits (DSCMCTCRCHn.CTSELA[2:0]).
   (b) Set the same value to the Current Capture Trigger B Select bits (DSCMCTCRCHn.CTSELB[2:0]).
4. Activate the channels simultaneously
   Write 1 to Channel n start trigger (DSCSTRTR.STRTRGn) at the same time.

40.4.8 Data Input Control

By using the DSCMCCRCHn.SEDGE bit, data receive can be selected to receive on the falling edge of MCLKmn or the rising edge of MCLKmn. Data input control receives clock and data from DSM. Synchronized data and clock are output to the sinc filter and short-circuit detection.

40.4.9 Sinc Filter

The differentiation stage is operated by decimation clock (the frequency is 1/M of MCLKmn). M is the decimation ratio and is set in the DSCMFCRCHn.CMDEC[7:0] bits for current measurement or in the DSOCFCRCHn.OCDEC[7:0] bits for overcurrent detection. The filter result is stored in register every decimation clock. In the case of current data, an interrupt is output at the same time when the filter result is stored.

Figure 40.7 shows a block diagram for the Sinc Filter.

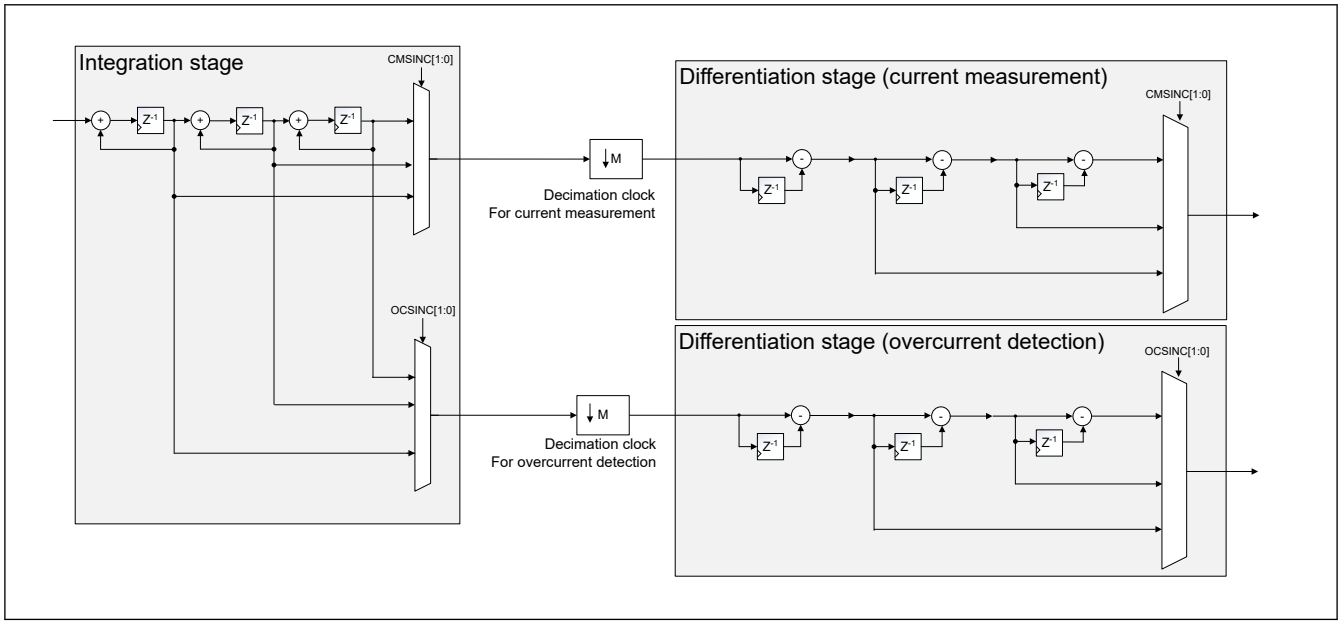


Figure 40.7 Block diagram of Sinc filter

40.4.10 Available Settings of Sinc Filter

Table 40.7 lists the available settings of the sinc filter and data shifting.

Table 40.7 Available settings of Sinc filter (1 of 2)

Sinc filter order	Decimation rate	Effective bits	Peak value (Decimal)	Peak value (Hexadecimal)	Stored data		CMSH[4:0] OCSH[4:0]	Resolution
					DFS = 0	DFS = 1		
3 (Sinc3)	4	[6:0]	63*1	000003F	[5:0], 00_0000_0000	00_0000_0000, [5:0]	0x12	6 bits
	5	[6:0]	125	000007D	[6:0], 0_0000_0000	0_0000_0000, [6:0]	0x11	7 bits
	6	[7:0]	216	00000D8	[7:0], 0000_0000	0000_0000, [7:0]	0x10	8 bits
	7	[8:0]	343	0000157	[8:0], 000_0000	000_0000, [8:0]	0x0F	9 bits
	8	[9:0]	511*1	00001FF	[8:0], 000_0000	000_0000, [8:0]	0x0F	9 bits
	9-10	[9:0]	1000	00003E8	[9:0], 00_0000	00_0000, [9:0]	0x0E	10 bits
	11-12	[10:0]	1728	00006C0	[10:0], 0_0000	0_0000, [10:0]	0x0D	11 bits
	13-15	[11:0]	3375	0000D2F	[11:0], 0000	0000, [11:0]	0x0C	12 bits
	16	[12:0]	4095*1	0000FFF	[11:0], 0000	0000, [11:0]	0x0C	12 bits
	17-20	[12:0]	8000	0001F40	[12:0], 000	000, [12:0]	0x0B	13 bits
	21-25	[13:0]	15625	0003D09	[13:0], 00	00, [13:0]	0x0A	14 bits
	26-31	[14:0]	29791	000745F	[14:0], 0	0, [14:0]	0x09	15 bits
	32	[15:0]	32767*1	0007FFF	[14:0], 0	0, [14:0]	0x09	15 bits
	33-40	[15:0]	64000	000FA00	[15:0]		0x08	16 bits
	41-50	[16:0]	125000	001E848	[16:1]		0x07	16 bits
	51-63	[17:0]	250047	003D0BF	[17:2]		0x06	16 bits
	64	[18:0]	262143*1	003FFFF	[17:2]		0x06	16 bits
	65-80	[18:0]	512000	007D000	[18:3]		0x05	16 bits
	81-101	[19:0]	1030301	00FB89D	[19:4]		0x04	16 bits
	102-127	[20:0]	2048383	01F417F	[20:5]		0x03	16 bits
	128	[21:0]	2097151*1	01FFFFF	[20:5]		0x03	16 bits
	129-161	[21:0]	4173281	03FADE1	[21:6]		0x02	16 bits
	162-203	[22:0]	8365427	07FA573	[22:7]		0x01	16 bits
	204-255	[23:0]	16581375	0FD02FF	[23:8]		0x00	16 bits
	256	[24:0]	16777215*1	0FFFFFF	[23:8]		0x00	16 bits
2 (Sinc2)	16	[7:0]	255*1	00000FF	[7:0], 0000_0000	0000_0000, [7:0]	0x10	8 bits
	17-20	[8:0]	400	0000190	[8:0], 000_0000	000_0000, [8:0]	0x0F	9 bits
	21-31	[9:0]	961	00003C1	[9:0], 00_0000	00_0000, [9:0]	0x0E	10 bits
	32	[10:0]	1023*1	00003FF	[9:0], 00_0000	00_0000, [9:0]	0x0E	10 bits
	33-40	[10:0]	1600	0000640	[10:0], 0_0000	0_0000, [10:0]	0x0D	11 bits
	41-63	[11:0]	3969	0000F81	[11:0], 0000	0000, [11:0]	0x0C	12 bits
	64	[12:0]	4095*1	0000FFF	[11:0], 0000	0000, [11:0]	0x0C	12 bits
	65-90	[12:0]	8100	0001FA4	[12:0], 000	000, [12:0]	0x0B	13 bits
	91-127	[13:0]	16129	0003F01	[13:0], 00	00, [13:0]	0x0A	14 bits
	128	[14:0]	16383*1	0003FFF	[13:0], 00	00, [13:0]	0x0A	14 bits
	129-181	[14:0]	32761	0007FF9	[14:0], 0	0, [14:0]	0x09	15 bits
	182-255	[15:0]	65025	000FE01	[15:0]		0x08	16 bits
	256	[16:0]	65535*1	000FFFF	[15:0]		0x08	16 bits

Table 40.7 Available settings of Sinc filter (2 of 2)

Sinc filter order	Decimation rate	Effective bits	Peak value (Decimal)	Peak value (Hexadecimal)	Stored data		CMSH[4:0] OCSH[4:0]	Resolution
					DFS = 0	DFS = 1		
1 (Sinc1)	32	[5:0]	31*1	000001F	[4:0], 000_0000_0000	000_0000_0000, [4:0]	0x13	5 bits
	33-63	[5:0]	63	000003F	[5:0], 00_0000_0000	00_0000_0000, [5:0]	0x12	6 bits
	64	[6:0]	63*1	000003F	[5:0], 00_0000_0000	00_0000_0000, [5:0]	0x12	6 bits
	65-127	[6:0]	127	000007F	[6:0], 0_0000_0000	0_0000_0000, [6:0]	0x11	7 bits
	128	[7:0]	127*1	000007F	[6:0], 0_0000_0000	0_0000_0000, [6:0]	0x11	7 bits
	129-255	[7:0]	255	00000FF	[7:0], 0000_0000	0000_0000, [7:0]	0x10	8 bits
	256	[8:0]	255*1	00000FF	[7:0], 0000_0000	0000_0000, [7:0]	0x10	8 bits

Note 1. When the filter result is the maximum value, it is rounded to the maximum value of -1.

Figure 40.8 shows the relationship between the filter settings and the stored values.

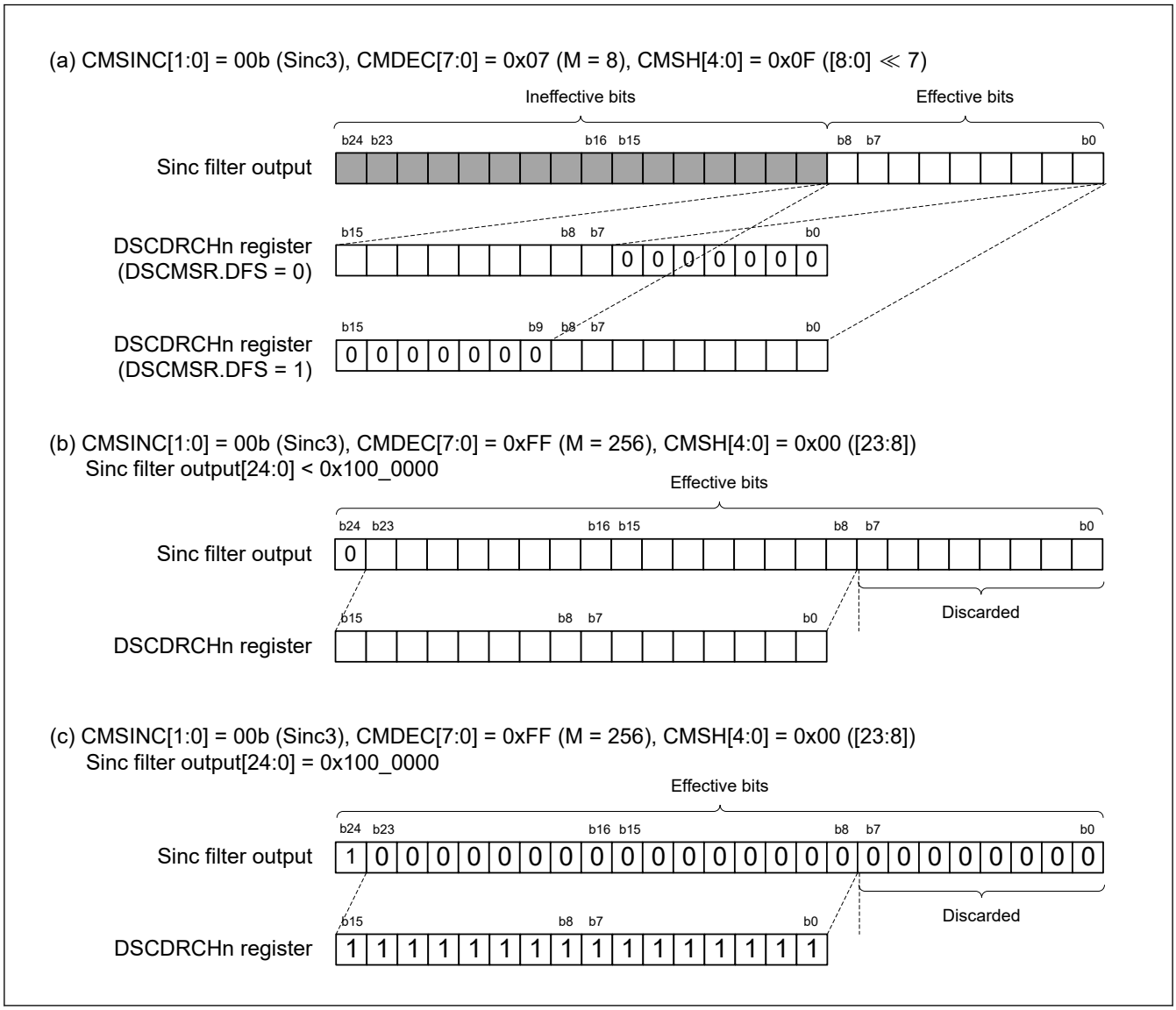


Figure 40.8 Relationship between the filter settings and the stored values

40.4.11 Current Measurement Data Processing

Each channel of DSMIF has the data registers specified in Table 40.8. Capture trigger A is selected by DSCMCTCRCHn.CTSELA[2:0] and capture trigger B is selected by DSCMCTCRCHn.CTSELB[2:0].

The trigger capturing function allows the capture of the values for current at the time of capture trigger A and capture trigger B using the event link controller (ELC) to set the capture trigger A or capture trigger B.

Table 40.8 Current measurement data list

No.	Register name	Symbol	Stored data
1	Current Data register	DSCDRCHn	The measured values for current
2	Capture Current Data Register A	DSCCDRACHn	The captured value of Current Data register by capture trigger A from the internal timer module
3	Capture Current Data Register B	DSCCDRBCHn	The captured value of Current Data register by capture trigger B from the internal timer module

40.4.12 Short-circuit Detection

When the short-circuit detection enable (DSEDCRCHn.SDE) is valid, the short-circuit detection can operate. A dedicated 13-bit counter counts consecutive 0's or 1's input from dsmdata n (n = 0 to 2). When the number of consecutive 0's exceeds the value set in the DSSCTSRCHn.SCNTL[12:0] register or the number of consecutive 1's exceeds the value set in the DSSCTSRCHn.SCNTH[12:0] register, a short-circuit detection interrupt request is generated.

40.4.13 Overcurrent Detection

The overcurrent detection function is used for abnormally detection. Overcurrent data is generated by the sinc filter result that is setting in section 40.4.9. Sinc Filter. The overcurrent data is compared with the value of the upper limit value or the lower limit value. The interrupt of overcurrent detection is output by these results.

It has 3 sets of upper limit value and lower limit value for each channel. And the interrupts can be output from each comparison results.

In addition, the overcurrent detection function also has overcurrent detection window notification function that notifies within range or out of range from the comparison result of the upper limit and the lower limit.

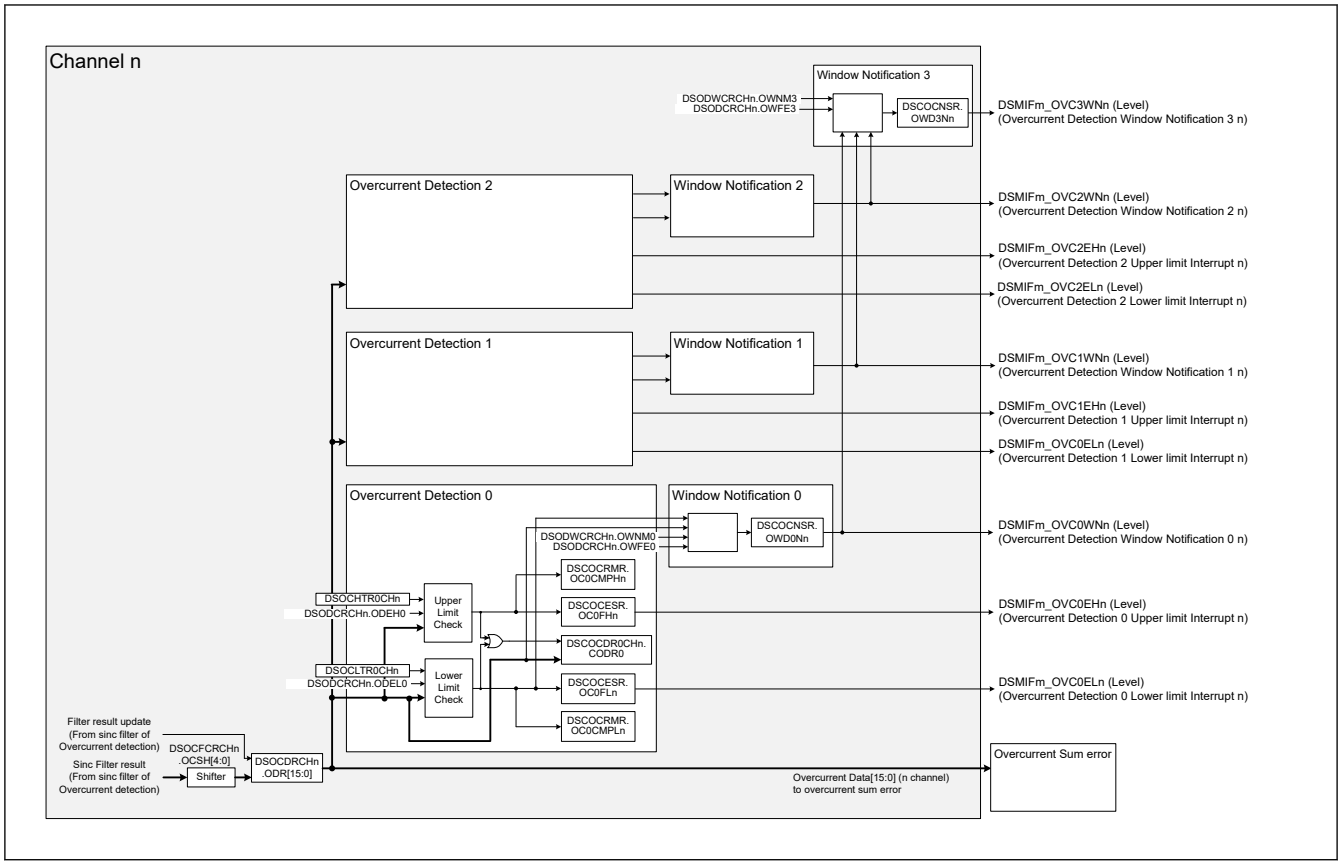


Figure 40.9 Configuration of overcurrent data registers

The measure values from current are stored as 16-bit values obtained by shifting the 25-bit data output from the sinc-filter in accordance with the setting of the DSOCFCRCHn.OCSH[4:0] bits.

The format of stored data can be selected by the DSCMSR.DFS bit. Either left-justified or right-justified can be selected.

40.4.13.1 Overcurrent Detection 0 to 2

Overcurrent detection has three upper limit excess judgments and three lower limit judgments for each channel. Table 40.9 shows the list of overcurrent detection.

Table 40.9 List of overcurrent detection

Function	Upper Lower	Enable	Comparator table	Error flag	Comparator result	Interrupt	Error capture data
Overcurrent Detection 0	Upper	DSODCRCHn.ODEH0	DSOCHTR0CHn.OCMPTBH0	DSCOCESR.OC0FHn	DSCOCRMR.OC0CMPHn	OVC0EHn	DSCOCDR0CHn.CODR0
	Lower	DSODCRCHn.ODEL0	DSOCHTR0CHn.OCMPTBL0	DSCOCESR.OC0FLn	DSCOCRMR.OC0CMPLn	OVC0ELn	
Overcurrent Detection 1	Upper	DSODCRCHn.ODEH1	DSOCHTR1CHn.OCMPTBH1	DSCOCESR.OC1FHn	DSCOCRMR.OC1CMPHn	OVC1EHn	DSCOCDR1CHn.CODR1
	Lower	DSODCRCHn.ODEL1	DSOCHTR1CHn.OCMPTBL1	DSCOCESR.OC1FLn	DSCOCRMR.OC1CMPLn	OVC1ELn	
Overcurrent Detection 2	Upper	DSODCRCHn.ODEH2	DSOCHTR2CHn.OCMPTBH2	DSCOCESR.OC2FHn	DSCOCRMR.OC2CMPHn	OVC2EHn	DSCOCDR2CHn.CODR2
	Lower	DSODCRCHn.ODEL2	DSOCHTR2CHn.OCMPTBL2	DSCOCESR.OC2FLn	DSCOCRMR.OC2CMPLn	OVC2ELn	

The overcurrent detection function is explained using overcurrent detection 0 as an example.

When the overcurrent upper limit detection 0 enable (DSODCRCHn.ODEH0) is valid, the overcurrent upper limit detection can operate. And when the overcurrent lower limit detection 0 enable (DSODCRCHn.ODEL0) is valid, the overcurrent lower limit detection can operate.

The current data is stored in the DSOCDRCHn as a 16-bit data shifted from 25-bit data output from sinc filter for overcurrent detection in accordance with the setting of the DSOCFCRCHn.OCSH[4:0] bits.

An overcurrent upper limit detection interrupt is requested when the value in the DSOCDRCHn register is upper than the value of OCMPTBH0.

An overcurrent lower limit detection interrupt is requested when the value in the DSOCDRCHn register is lower than the value of OCMPTBL0.

Overcurrent detection has the following two statuses.

1. Error flags that are set when the upper limit is exceeded / less than the lower limit is detected and cleared with the clear bit.
2. Monitor bits that return the comparison result for each decimation cycle.

40.4.13.2 Overcurrent Detection Window Notification

Overcurrent detection window notification is event notification function to notify within a range or out of range by comparison with upper limit and lower limit of overcurrent detection. There are four kinds of notification for each channel. Table 40.10 shows a list of overcurrent detection notification.

Table 40.10 List of overcurrent detection notification

Function	Enable	Mode bit	Flag	Notification
Overcurrent detection Window notification 0	DSODCRCHn.OWFE0	DSODWCRCHn.OWNM0	DSCOCNSR.OWD0Nn	OVC0WNn
Overcurrent detection Window notification 1	DSODCRCHn.OWFE1	DSODWCRCHn.OWNM1	DSCOCNSR.OWD1Nn	OVC1WNn
Overcurrent detection Window notification 2	DSODCRCHn.OWFE2	DSODWCRCHn.OWNM2	DSCOCNSR.OWD2Nn	OVC2WNn
Overcurrent detection Window notification 3	DSODCRCHn.OWFE3	DSODWCRCHn.OWNM3	DSCOCNSR.OWD3Nn	OVC3WNn

Overcurrent detection window notification 0 to 2 will be generated by comparison with upper limit and lower limit for overcurrent detection 0 to 2. The notification can be set for within a range or out of range as Figure 40.10.

1. OWNMm = 0: (overcurrent data channel n < lower limit m) or (overcurrent data channel n > upper limit m)
2. OWNMm = 1: lower limit m ≤ overcurrent data channel n ≤ upper limit m

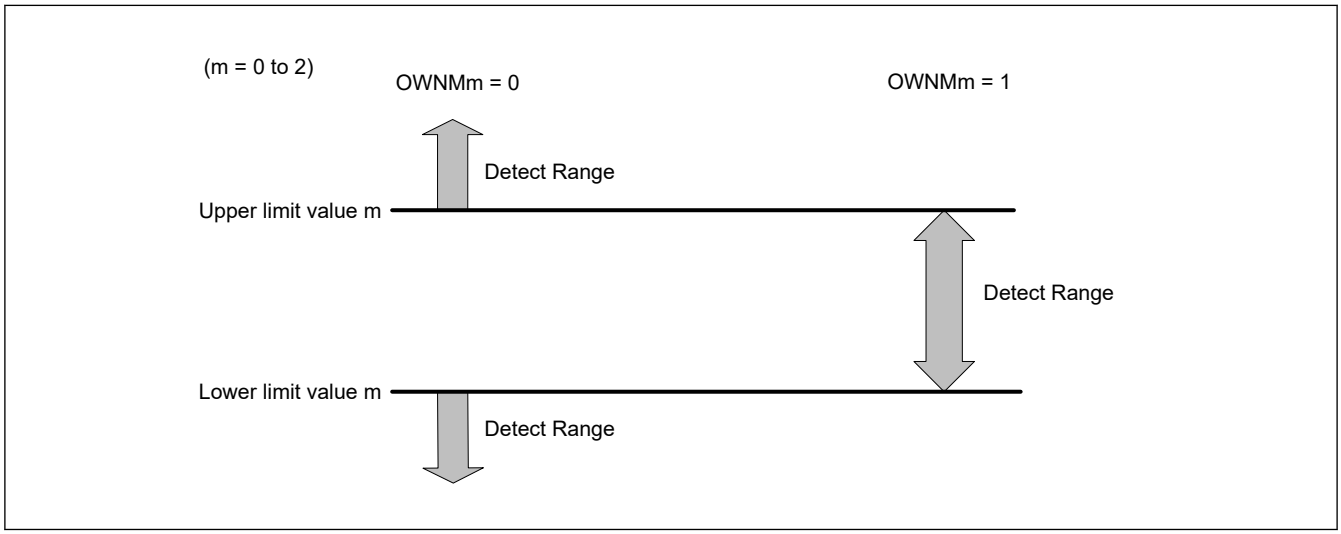


Figure 40.10 Detect range of overcurrent detection window notification

40.4.14 Overcurrent Sum Error Detection

When the overcurrent sum error upper limit detection enable (DSSECR.SEEH) is valid, the overcurrent sum error upper limit detection can operate. When the overcurrent sum error lower limit detection enable (DSSECR.SEEL) is valid, the

overcurrent sum error lower limit detection can operate. An overcurrent upper limit sum error detection interrupt is requested when the sum of the values for each mode (see Table 40.11) is higher than the value of SCMPTBH.

An overcurrent lower limit sum error detection interrupt is requested when the sum of the values for each mode is lower than the value of SCMPTBL.

Table 40.11 List of sum error mode detect mode and capture data

SEDM[2:0]	Sum error detect mode	CH0	CH1	CH2	DSSECDR0	DSSECDR1	DSSECDR2
000b	Three value sum error detect	✓	✓	✓	DSOCDRCH0	DSOCDRCH1	DSOCDRCH2
001b	Two value sum error detect	✓	✓	—	DSOCDRCH0	DSOCDRCH1	—*1
010b	One channel error detect	✓	—	—	DSOCDRCH0	—	—
011b		—	✓	—	DSOCDRCH1	—	—
100b		—	—	✓	DSOCDRCH2	—	—

Note 1. Do not store data, hold previous value.

40.4.15 Settling Time

40.4.15.1 Settling Time of Channel Activation

Settling time of channel activation is the time from when the channel is activated until the filter result is stable and output.

The settling time is calculated as follows.

Table 40.12 Settling time of channel activation

Core clock (MHz)	Sinc order	Processing time of settling time of channel activation
250 (DSCCSCR.CLKSEL = 0)	$Sinc^1$	(MCLK period × Decimation ratio) × sinc order + MCLK period × 1 cycle + PCLKH period × 18 cycles
	$Sinc^3$ $Sinc^2$	(MCLK period × Decimation ratio) × (sinc order + 1) + MCLK period × 1 cycle + PCLKH period × 18 cycles
400 (DSCCSCR.CLKSEL = 1)	$Sinc^1$	(MCLK period × Decimation ratio) × sinc order + MCLK period × 1 cycle + PCLKH period × 11 cycles + Core Clock period × 21 cycles
	$Sinc^3$ $Sinc^2$	(MCLK period × Decimation ratio) × (sinc order + 1) + MCLK period × 1 cycle + PCLKH period × 11 cycles + Core Clock period × 21 cycles

40.4.15.2 Settling Time after PWM Synchronization

Settling time after PWM synchronization is the time from the input of the decimation dividing counter initialization trigger to the stable output of the filter result.

Table 40.13 Settling time after PWM synchronization

Core clock (MHz)	Processing time of settling time of PWM synchronization
250 (DSCCSCR.CLKSEL = 0)	(MCLK period × Decimation ratio) × sinc order + PCLKH period × 13 cycles
400 (DSCCSCR.CLKSEL = 1)	(MCLK period × Decimation ratio) × sinc order + PCLKH period × 7 cycles + Core Clock period × 15 cycles

40.4.16 Common Interrupt

The common interrupt is a function that outputs a single common interrupt that is output for each channel. Common interrupts are provided for the following data update interrupts.

When using common interrupts, it is recommended that channel-specific data update interrupts are disabled.

Table 40.14 Data update common interrupt (1 of 2)

Function	Interrupt	Pulse or Level	Enable
Current data	Current data register update common interrupt	Pulse	DSCICR.IUCE

Table 40.14 Data update common interrupt (2 of 2)

Function	Interrupt	Pulse or Level	Enable
Capture data A	Current capture data register A update common interrupt	Pulse	DSCICR.IAUCE
Capture data B	Current capture data register B update common interrupt	Pulse	DSCICR.IBUCE

Common interrupts have two modes, 3ch synchronization mode and 2ch synchronization mode.

As shown in Table 40.15, common interrupts can be output when synchronization mode is enabled and all data on the target channel has been updated.

Table 40.15 Mode of common interrupt

CISM[1:0]	Function	Target channel			Target function		
		CH0	CH1	CH2	Current data	Capture data A	Capture data B
00b	Common interrupts can not be used	—	—	—	—	—	—
01b	Three channel synchronous mode	✓	✓	✓	✓	✓	✓
10b	Two channel synchronous mode	✓	✓	—	✓	✓	✓
11b	Setting prohibited	—	—	—	—	—	—

The operation timing of the common interrupt in the three channel synchronization mode is explained using the current data update common interrupt as an example. When the current data is updated for each of the three channels, the current data common interrupt is output at the same timing as the latest channel current data update interrupt.

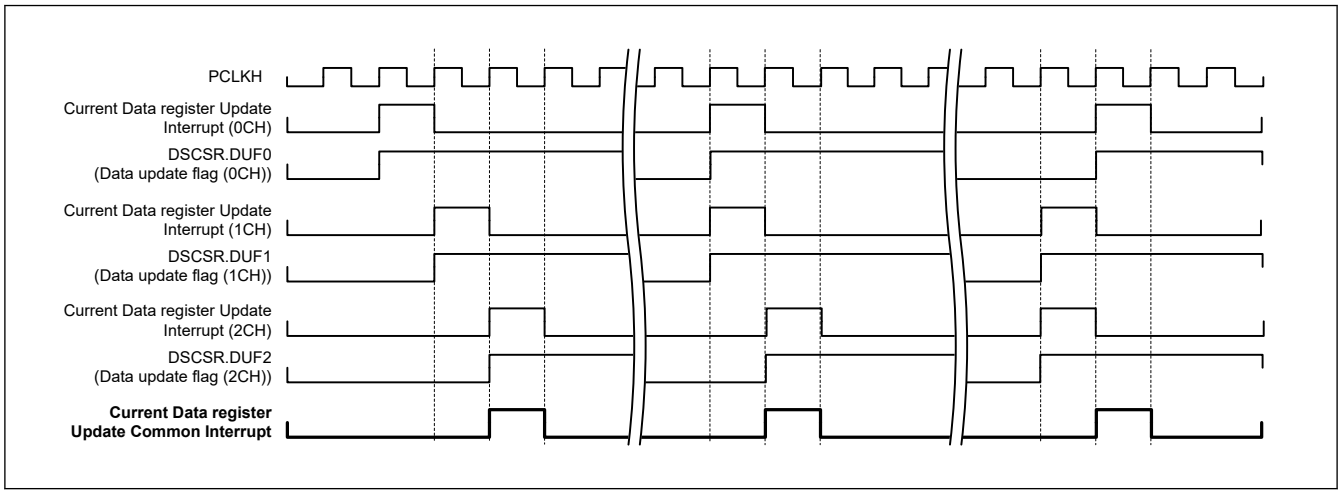


Figure 40.11 Common interrupt output timing (Three channel synchronous mode)

Similarly, in the two channel synchronous mode, the current data common interrupt is output at the same timing as the latest channel current data update interrupt.

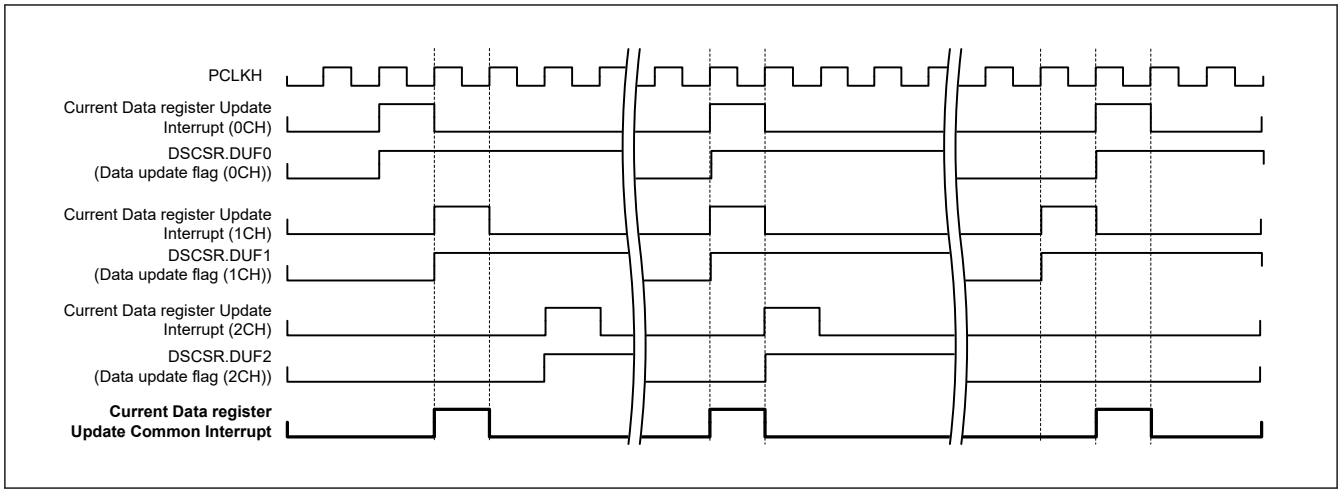


Figure 40.12 Common interrupt output timing (Two channel synchronous mode)

41. 12-Bit A/D converter (ADC12)

41.1 Overview

This LSI incorporates three unit of a 12-bit successive approximation A/D converter. Analog inputs of up to 4 channels are selectable for unit 0 and unit 1 (LLPP of Cortex-R52), analog inputs of up to 15 channels are selectable for unit 2 (NONSAFETY).

The 12-bit A/D converter converts analog inputs of up to 4/15 selected channels to a 12-bit digital value through successive approximation.

The 12-bit A/D converter has three operating modes: single scan mode in which analog inputs selected arbitrarily are converted only once in ascending order of channel number; continuous scan mode in which analog inputs selected arbitrarily are continuously converted in ascending order of channel number; and group scan mode in which analog inputs are arbitrarily selected and divided into two groups (group A and group B) or three groups (group A, group B and group C), and then analog inputs of channels selected for each group are converted in ascending order of channel number.

In group scan mode, the conditions for scanning start of each groups (A, B or A, B, C) (synchronous trigger) can be independently selected, thus allowing A/D conversion of each groups to be started independently. In addition to the above actions, the group priority operation in group scan mode accepts the start of scanning the priority group during the scan of the low priority group, interrupts the scan of the low priority group, and starts the scan of the priority group. The priority of group priority operations is group A > group B > group C. In group priority operation, if the scan start of group B is accepted during the scan of group C, the scan of group C is interrupted and the scan of group B is started. If the scan start of group A is accepted during the scan of group C, the scan of group A is started. The scan of Group C is interrupted and the scan of Group A is started. Similarly, if the scan start of Group A is accepted during the scan of Group B, the scan of Group B is interrupted and the scan of Group A is started. The interrupted group scan is able to resume after the priority group scan is complete also.

In double trigger mode, one analog input channel arbitrarily selected is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second synchronous triggers are stored into different registers (duplication of A/D conversion data).

The compare function is a function that outputs an interrupt (ADCn_CMPAI/ADCn_CMPBI) when the A/D conversion value of the selected channel by specifying the upper and lower reference values for each window A/B matches the comparison conditions. The event (ADCn_WCMPM/ADCn_WCMPUM) is output according to the event conditions (A or B, A and B, A exor B). In addition, a comparator operation that compares the A/D conversion value with the lower reference value is also possible.

Table 41.1 lists the specifications of the 12-bit A/D converter and Table 41.2 indicates the functions of the 12-bit A/D converter. Figure 41.1 shows a block diagram of the 12-bit A/D converter.

Table 41.1 ADC12 specifications (1 of 2)

Item	Description
Number of units	Three units, 0, 1, and 2
Input channels	• Unit 0, 1: Up to 4 channels in LLPP of Cortex-R52 • Unit 2: Up to 15 channels in NONSAFETY
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	0.32 µs: Channel-dedicated sample-and-hold circuits not in use 0.64 µs: Channel-dedicated sample-and-hold circuits in use At PCLKADC in the 12-bit A/D converter: 62.5 MHz
A/D conversion clock	PCLKADC (62.5 MHz)

Table 41.1 ADC12 specifications (2 of 2)

Item	Description
Operating modes	• Single scan mode: A/D conversion is performed only once on the analog inputs of arbitrarily selected channels • Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs of arbitrarily selected channels • Group scan mode: Two (groups A and B) or three (groups A, B, and C) can be selected as the number of the groups to be used. Only the combination of groups A and B can be selected when the number of the groups is two. Analog inputs that are arbitrarily selected are divided into two groups (group A and B) or three groups (group A, B, and C), and A/D conversion of the analog input selected on a group basis is performed only once. The conditions for scanning start of groups A, B, and C (synchronous trigger) can be independently selected, thus allowing A/D conversion of each group to be started independently. • Group scan mode (when Group A is given priority): If a priority-group trigger is input during scanning of the low-priority group, scan of the low-priority group is stopped and scan of the priority group is started. The priority order is group A (highest) > group B > group C (lowest). Whether or not to restart scanning of the low-priority group after processing for the high-priority group completes, is selectable. Rescan can also be set to start either from the beginning of the selected channel or the channel on which A/D conversion is not completed.
Conditions for A/D conversion start	• Software trigger • Synchronous triggers Trigger by the multi-function timer pulse unit (MTU3) and event link controller (ELC) • Asynchronous trigger A/D conversion can be triggered by the external trigger ADTRGn#.
Functions	• Dedicated sample-and-hold function with optional constant sampling and 3 channels in each unit • Variable sampling state count • Selectable A/D-converted value addition mode or average mode • Double-trigger mode (duplication of A/D conversion data) • Automatic clear function for A/D data registers • Digital comparison of values in the comparison and data registers, and between values in the data registers • Overwrite checking function of the A/D data register
Interrupt sources and ELC events	• ADCn_ADI (for ELC, trigger of interrupt is slightly different for CPU SPI) • ADCn_GBADI: A/D scan end interrupt for Group B • ADCn_GCADI: A/D scan end interrupt for Group C • ADCn_CMPAI: Window A compare match • ADCn_CMPBI: Window B compare match • ADCn_WCMPM: compare match • ADCn_WCMPUM: compare mismatch
ELC interface	• An ELC event is generated upon completion of all scans • Able to start scanning by a trigger from the ELC
Module-stop function	Module-stop state can be set to reduce power consumption

Table 41.2 Function of ADC12 (1 of 3)

Parameter	Unit 0 (ADC120)	Unit 1 (ADC121)	Unit 2 (ADC122)
Analog input channel	AN000 to AN003	AN100 to AN103	AN200 to AN214*2

Table 41.2 Function of ADC12 (2 of 3)

Parameter			Unit 0 (ADC120)	Unit 1 (ADC121)	Unit 2 (ADC122)
Condition for A/D conversion start	Software	Software trigger	Enabled	Enabled	Enabled
	Asynchronous trigger	Trigger input pin (Use only synchronous trigger as the scan start in group scan mode)	ADTRG0#	ADTRG1#	ADTRG2#
	Synchronous	Compare match with or input capture to MTU0.TGRA	TRGA0N	TRGA0N	—
		Compare match with or input capture to MTU1.TGRA	TRGA1N	TRGA1N	—
		Compare match with or input capture to MTU2.TGRA	TRGA2N	TRGA2N	—
		Compare match with or input capture to MTU3.TGRA	TRGA3N	TRGA3N	—
		Compare match with or input capture to MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	TRGA4N	TRGA4N	—
		Compare match with or input capture to MTU6.TGRA	TRGA6N	TRGA6N	—
		Compare match with or input capture to MTU7.TGRA or underflow (trough) of MTU7.TCNT in complementary PWM mode	TRGA7N	TRGA7N	—
		Compare match with MTU0.TGRE	TRG0N	TRG0N	—
		Compare match between MTU4.TADCORA and MTU4.TCNT	TRG4AN	TRG4AN	—
		Compare match between MTU4.TADCORB and MTU4.TCNT	TRG4BN	TRG4BN	—
		Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT	TRG4AN or TRG4BN	TRG4AN or TRG4BN	—
		Compare match between MTU4.TADCORA and MTU4.TCNT and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	TRG4ABN	TRG4ABN	—
		Compare match between MTU7.TADCORA and MTU7.TCNT	TRG7AN	TRG7AN	—
		Compare match between MTU7.TADCORB and MTU7.TCNT	TRG7BN	TRG7BN	—
		Compare match between MTU7.TADCORA and MTU7.TCNT or compare match between MTU7.TADCORB and MTU7.TCNT	TRG7AN or TRG7BN	TRG7AN or TRG7BN	—
		Compare match between MTU7.TADCORA and MTU7.TCNT and compare match between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is used)	TRG7ABN	TRG7ABN	—
		ELC trigger	ADC0 A, ADC0 B	ADC1 A, ADC1 B	ADC2 A, ADC2 B
Channel-dedicated sample-and-hold function	Target channel		AN000 to AN002	AN100 to AN102	AN200 to AN202

Table 41.2 Function of ADC12 (3 of 3)

Parameter	Unit 0 (ADC120)	Unit 1 (ADC121)	Unit 2 (ADC122)
Interrupt	ADC0_ADI ADC0_GBADI ADC0_GCADI ADC0_CMPAI ADC0_CMPBI	ADC1_ADI ADC1_GBADI ADC1_GCADI ADC1_CMPAI ADC1_CMPBI	ADC2_ADI ADC2_GBADI ADC2_GCADI ADC2_CMPAI ADC2_CMPBI
Output to ELC	ADC0_ADI*1 ADC0_WCMPM ADC0_WCMPUM	ADC1_ADI*1 ADC1_WCMPM ADC1_WCMPUM	ADC2_ADI*1 ADC2_WCMPM ADC2_WCMPUM
Module-stop function setting	Available	Available	Available

Note 1. Trigger of event is slightly different from normal interrupt for CPU SPI.
Note 2. RZ/T2H (729-pin FCBGA) supports AN200 to AN205.

Table 41.3 ADC12 interrupt sources

Name	Interrupt sources
ADCn_ADI*1	A/D scan end interrupt
ADCn_GBADI	A/D scan end interrupt for Group B
ADCn_GCADI	A/D scan end interrupt for Group C
ADCn_CMPAI	Window A compare match
ADCn_CMPBI	Window B compare match
ADCn_WCMPM	Compare match
ADCn_WCMPUM	Compare mismatch

Note: n = 0 to 2.
Note 1. Trigger of event is slightly different from normal interrupt for CPU SPI.

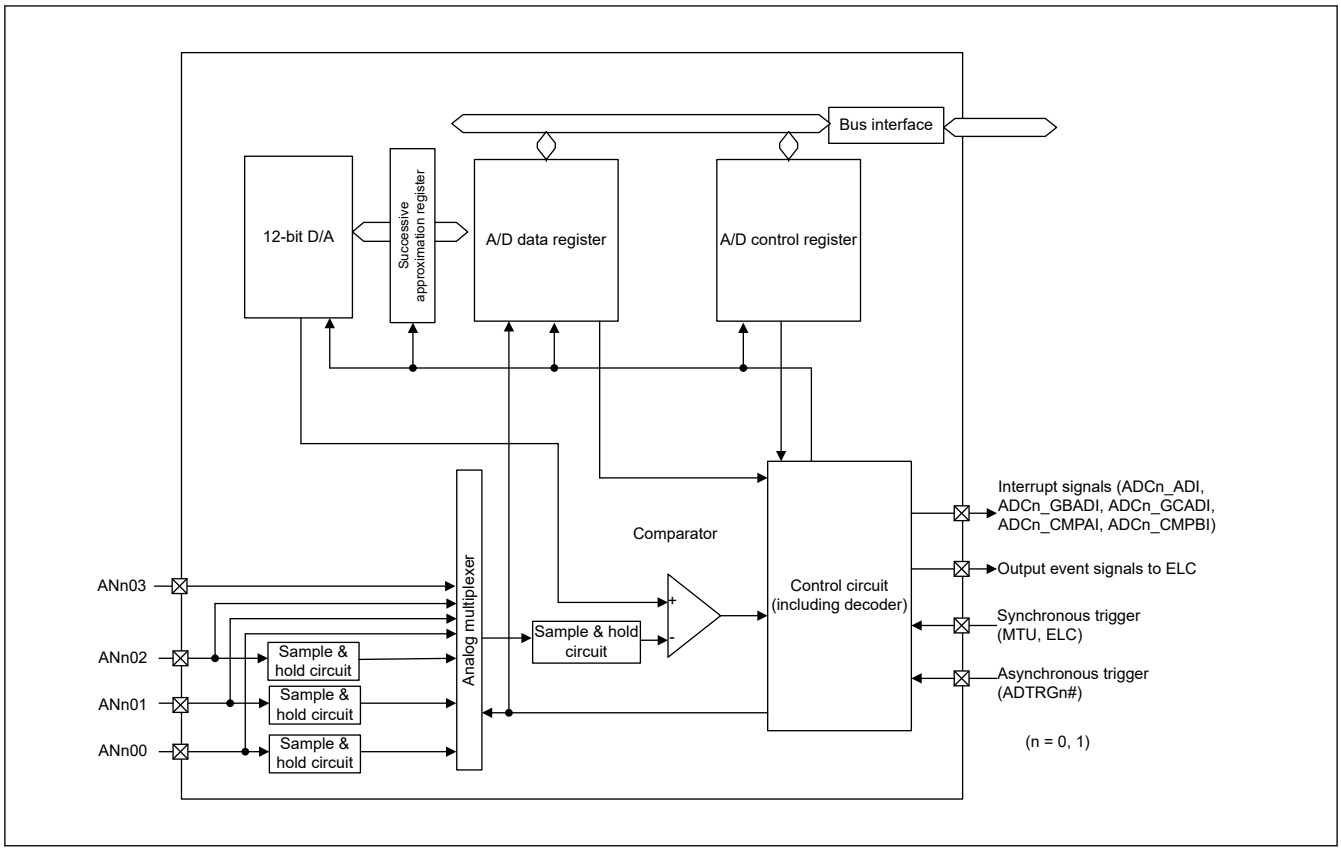
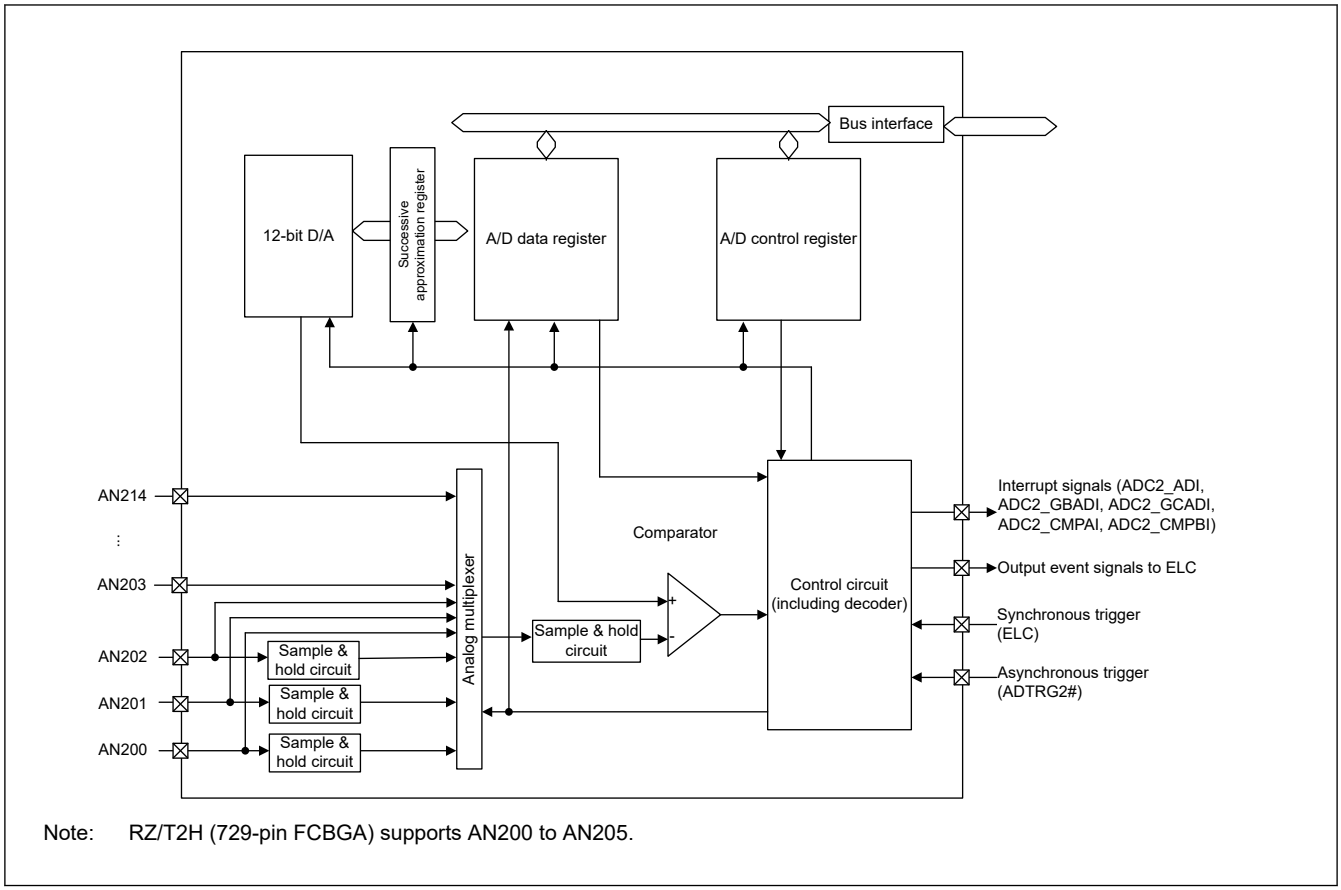


Figure 41.1 Block diagram of 12-bit A/D converter (Unit 0, Unit 1)



Note: RZ/T2H (729-pin FCBGA) supports AN200 to AN205.

Figure 41.2 Block diagram of 12-bit A/D converter (Unit 2)

41.2 Register Map

Table 41.4 ADC12 register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x9001_4000 + 0x400 x m (m = 0, 1) 0x8000_8000 (m = 2)	ADCSR	A/D Control Register	—
0x9001_4004 + 0x400 x m (m = 0, 1) 0x8000_8004 (m = 2)	ADANSA0	A/D Channel Select Register A0	—
0x9001_4008 + 0x400 x m (m = 0, 1) 0x8000_8008 (m = 2)	ADADS0	A/D-Converted Value Addition/Average Function Channel Select Register 0	—
0x9001_400C + 0x400 x m (m = 0, 1) 0x8000_800C (m = 2)	ADADC	A/D-Converted Value Addition/Average Count Select Register	—
0x9001_400E + 0x400 x m (m = 0, 1) 0x8000_800E (m = 2)	ADCER	A/D Control Extended Register	—
0x9001_4010 + 0x400 x m (m = 0, 1) 0x8000_8010 (m = 2)	ADSTRGR	A/D Conversion Start Trigger Select Register	—
0x9001_4014 + 0x400 x m (m = 0, 1) 0x8000_8014 (m = 2)	ADANSB0	A/D Channel Select Register B0	—
0x9001_4018 + 0x400 x m (m = 0, 1) 0x8000_8018 (m = 2)	ADDBLDR	A/D Data Duplication Register	—
0x9001_4020 + 0x400 x m + 0x02 × n (m = 0, 1) 0x8000_8020 + 0x02 × n (m = 2)	ADDRn	A/D Data Register n (n = 0 to 3 for unit 0, 1, n = 0 to 15 for unit 2)	—
0x9001_4066 + 0x400 x m (m = 0, 1) 0x8000_8066 (m = 2)	ADSHCR	A/D Sample and Hold Control Register	—

Table 41.4 ADC12 register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x9001_407D + 0x400 x m (m = 0, 1) 0x8000_807D (m = 2)	ADELCCR	A/D Event Link Control Register	—
0x9001_4080 + 0x400 x m (m = 0, 1) 0x8000_8080 (m = 2)	ADGSPCR	A/D Group Scan Priority Control Register	—
0x9001_4084 + 0x400 x m (m = 0, 1) 0x8000_8084 (m = 2)	ADDBLDRA	A/D Data Duplication Register A	—
0x9001_4086 + 0x400 x m (m = 0, 1) 0x8000_8086 (m = 2)	ADDBLDRB	A/D Data Duplication Register B	—
0x9001_408C + 0x400 x m (m = 0, 1) 0x8000_808C (m = 2)	ADWINMON	A/D Compare Function Window A/B Status Monitoring Register	—
0x9001_4090 + 0x400 x m (m = 0, 1) 0x8000_8090 (m = 2)	ADCMPCR	A/D Compare Function Control Register	—
0x9001_4094 + 0x400 x m (m = 0, 1) 0x8000_8094 (m = 2)	ADCMPANSR0	A/D Compare Function Window A Channel Select Register 0	—
0x9001_4098 + 0x400 x m (m = 0, 1) 0x8000_8098 (m = 2)	ADCMPLR0	A/D Compare Function Window A Comparison Condition Setting Register 0	—
0x9001_409C + 0x400 x m (m = 0, 1) 0x8000_809C (m = 2)	ADCMPDR0	A/D Comparison Function Window A Lower Level Setting Register	—
0x9001_409E + 0x400 x m (m = 0, 1) 0x8000_809E (m = 2)	ADCMPDR1	AD Comparison Function Window A Upper Level Setting Register	—
0x9001_40A0 + 0x400 x m (m = 0, 1) 0x8000_80A0 (m = 2)	ADCMPSR0	A/D Comparison Function Window A Channel Status Register 0	—
0x9001_40A6 + 0x400 x m (m = 0, 1) 0x8000_80A6 (m = 2)	ADCMPBNSR	A/D Compare Function Window B Channel Select Register	—
0x9001_40A8 + 0x400 x m (m = 0, 1) 0x8000_80A8 (m = 2)	ADWINLLB	A/D Compare Function Window B Lower-Side Level Setting Register	—
0x9001_40AA + 0x400 x m (m = 0, 1) 0x8000_80AA (m = 2)	ADWINULB	A/D Compare Function Window B Upper-Side Level Setting Register	—
0x9001_40AC + 0x400 x m (m = 0, 1) 0x8000_80AC (m = 2)	ADCMPBSR	A/D Compare Function Window B Status Register	—
0x9001_40D4 + 0x400 x m (m = 0, 1) 0x8000_80D4 (m = 2)	ADANSC0	A/D Channel Select Register C0	—
0x9001_40D9 + 0x400 x m (m = 0, 1) 0x8000_80D9 (m = 2)	ADGCTRGR	A/D Group C Trigger Select Register	—
0x9001_40E0 + 0x400 x m + 0x01 × n (m = 0, 1) 0x8000_80E0 + 0x01 × n (m = 2)	ADSSTRn	A/D Sampling State Register n (n = 0 to 3 for unit 0, 1, n = 0 to 15 for unit 2)	—
0x9001_41F0 + 0x400 x m (m = 0, 1) 0x8000_81F0 (m = 2)	ADCALCTL	A/D Calibration Control Register	—

Table 41.5

Unit	Module reset- control register	Module stop control register	Slave access control register
Unit 0	—	MSTPCRC.MSTPCRC06	SLVACCCTL7.LLPP_SL*1
Unit 1	—	MSTPCRC.MSTPCRC07	SLVACCCTL7.LLPP_SL*1
Unit 2	—	MSTPCRC.MSTPCRC25	SLVACCCTL04.ADC2_SL

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

41.3 Register Descriptions

41.3.1 ADCSR : A/D Control Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADST	ADCS[1:0]		ADIE	—	—	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DBLANS[4:0]	Double Trigger Channel Select These bits select one analog input channel for double triggered operation. The setting is only valid while double trigger mode is selected.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	GBADIE	Group B Scan End Interrupt Enable 0: Disables ADCn_GBADI interrupt generation upon group B scan completion 1: Enables ADCn_GBADI interrupt generation upon group B scan completion	R/W
7	DBLE	Double Trigger Mode Select 0: Deselects double trigger mode 1: Selects double trigger mode	R/W
8	EXTRG	Trigger Select 0: A/D conversion is started by a synchronous trigger (MTU3, ELC) 1: A/D conversion is started by the asynchronous trigger (ADTRGn#)	R/W
9	TRGE	Trigger Start Enable 0: Disables A/D conversion to be started by the synchronous or asynchronous trigger 1: Enables A/D conversion to be started by the synchronous or asynchronous trigger	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	ADIE	Scan End Interrupt Enable 0: Disables ADCn_ADI interrupt generation upon scan completion 1: Enables ADCn_ADI interrupt generation upon scan completion	R/W
14:13	ADCS[1:0]	Scan Mode Select 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
15	ADST	A/D conversion Start 0: Stops A/D conversion process 1: Starts A/D conversion process	R/W

41.3.2 ADANSA0 : A/D Channel Select Register A0

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSA0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSA0[15:0]	A/D conversion Analog input Channel Select ANSA0[0] bit corresponds to ch0 and ANSA0[15] corresponds ch15. 0: ch0-ch15 are not subjected to conversion. 1: ch0-ch15 are subjected to conversion.	R/W

41.3.3 ADADS0 : A/D-Converted Value Addition/Average Function Channel Select Register 0

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x08

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADS0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADS0[15:0]	A/D-Converted Value Addition/Average Channel Select 0: A/D-converted value addition/average mode for ch0 to ch15 is not selected 1: A/D-converted value addition/average mode for ch0 to ch15 is selected	R/W

41.3.4 ADADC : A/D-Converted Value Addition/Average Count Select Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	AVEE	—	—	—	—	ADC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ADC[2:0]	Addition Count Select 0 0 0: 1-time conversion (no addition; same as normal conversion) 0 0 1: 2-time conversion (addition once) 0 1 0: 3-time conversion (addition twice)*1 0 1 1: 4-time conversion (addition three times) 1 0 1: 16-time conversion (addition 15 times)*1 Others: Setting is prohibited	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	AVEE	Average Mode Enable 0: Addition mode is selected 1: Average mode is selected	R/W

Note 1. When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the Addition count to 3 times and 16times (ADADC.ADC[1:0] = 010b, 101b).

41.3.5 ADCER : A/D Control Extended Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x0E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADRFMT	—	—	—	—	—	—	—	—	—	ACE	—	—	ADPRC[1:0]		—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
2:1	ADPRC[1:0]	A/D Conversion Accuracy Specify 0 0: A/D conversion is performed with 12-bit accuracy Others: Setting prohibited	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	ACE	A/D Data Register Automatic Clearing Enable 0: Disables automatic clearing 1: Enables automatic clearing	R/W
14:6	—	These bits are read as 0. The write value should be 0.	R/W
15	ADRFMT	A/D Data Register Format Select 0: Right-alignment is selected for the A/D data register format 1: Left-alignment is selected for the A/D data register format	R/W

41.3.6 ADSTRGR : A/D Conversion Start Trigger Select Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x10

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TRSA[5:0]						—	—	TRSB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	TRSB[5:0]	A/D Conversion Start Trigger Select for Group B Select the A/D conversion start trigger for group B in group scan mode. See Table 41.6 for details.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TRSA[5:0]	A/D Conversion Start Trigger Select Select the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for group A is selected. See Table 41.7 for details.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Table 41.6 Selection of A/D activation sources by the TRSB[5:0] bits (for group B only) (1 of 2)

Module	Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source de-selection state			1	1	1	1	1	1
MTU3	TRGA0N	Compare match with or input capture to MTU0.TGRA	0	0	0	0	0	1
MTU3	TRGA1N	Compare match with or input capture to MTU1.TGRA	0	0	0	0	1	0

Table 41.6 Selection of A/D activation sources by the TRSB[5:0] bits (for group B only) (2 of 2)

Module	Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
MTU3	TRGA2N	Compare match with or input capture to MTU2.TGRA	0	0	0	0	1	1
MTU3	TRGA3N	Compare match with or input capture to MTU3.TGRA	0	0	0	1	0	0
MTU3	TRGA4N	Compare match with or input capture to MTU4.TGRA, or an underflow of MTU4.TCNT (in the trough) in complementary PWM mode	0	0	0	1	0	1
MTU3	TRGA6N	Compare match with MTU6.TGRA	0	0	0	1	1	0
MTU3	TRGA7N	Compare match with MTU7.TGRA	0	0	0	1	1	1
MTU3	TRG0N	Compare match with MTU0.TGRE	0	0	1	0	0	0
MTU3	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
MTU3	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
MTU3	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
MTU3	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is in use)	0	0	1	1	0	0
MTU3	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	1	1	0	1
MTU3	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	0
MTU3	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	1
MTU3	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is in use)	0	1	0	0	0	0
ELC	ADCn A		0	1	0	0	0	1
ELC	ADCn B		0	1	0	0	1	0

Note: Trigger source from MTU3 is not available in ADC12 unit 2.

Table 41.7 Selection of A/D activation sources by the TRSA[5:0] bits (1 of 2)

Module	Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source de-selection state			1	1	1	1	1	1
External	ADTRGn#	Input pin for the trigger	0	0	0	0	0	0
MTU3	TRGA0N	Compare match with or input capture to MTU0.TGRA	0	0	0	0	0	1
MTU3	TRGA1N	Compare match with or input capture to MTU1.TGRA	0	0	0	0	1	0
MTU3	TRGA2N	Compare match with or input capture to MTU2.TGRA	0	0	0	0	1	1
MTU3	TRGA3N	Compare match with or input capture to MTU3.TGRA	0	0	0	1	0	0

Table 41.7 Selection of A/D activation sources by the TRSA[5:0] bits (2 of 2)

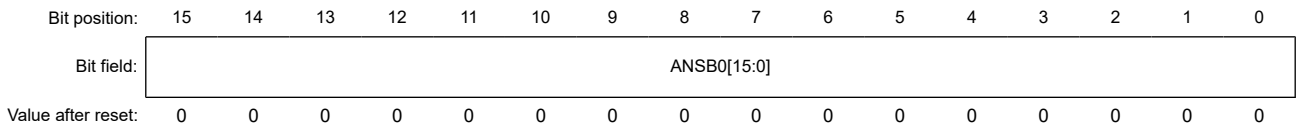
Module	Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
MTU3	TRGA4N	Compare match with or input capture to MTU4.TGRA, or in complementary PWM mode, an underflow of MTU4.TCNT (in the trough)	0	0	0	1	0	1
MTU3	TRGA6N	Compare match with MTU6.TGRA	0	0	0	1	1	0
MTU3	TRGA7N	Compare match with MTU7.TGRA	0	0	0	1	1	1
MTU3	TRG0N	Compare match with MTU0.TGRE	0	0	1	0	0	0
MTU3	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
MTU3	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
MTU3	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
MTU3	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is in use)	0	0	1	1	0	0
MTU3	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	1	1	0	1
MTU3	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	0
MTU3	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	1
MTU3	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is in use)	0	1	0	0	0	0
ELC	ADCn A		0	1	0	0	0	1
ELC	ADCn B		0	1	0	0	1	0

Note: Trigger source from MTU3 is not available in ADC12 unit 2.

41.3.7 ADANSB0 : A/D Channel Select Register B0

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x14



Bit	Symbol	Function	R/W
15:0	ANSB0[15:0]	A/D Conversion Analog Input Channel Select ANSB0[0] bit corresponds to ch0 and ANSB0[15] corresponds ch15. 0: ch0-ch15 are not subjected to conversion 1: ch0-ch15 are subjected to conversion	R/W

41.3.8 ADDBLDR : A/D Data Duplication Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x18

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DBLDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DBLDR[15:0]	The result of A/D conversion in response to the second trigger in double trigger mode.	R

A/D-converted value is 12-bit accuracy. The formats for data in the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers vary according to the following conditions.

- The setting of the A/D data register format select bit (ADCER.ADRFMT) (determining whether the data are flush-left or flush-right in the registers)
- The setting of the Addition count select bits (ADADC.ADC[2:0]) (once, twice, three times, four times, or sixteen times is selectable.)
- The setting of the average mode enable bit (ADADC.AVEE) (Addition or average is selectable.)

When the number of the conversion count is set one to four in the addition mode, the A/D conversion addition result is expanded by 2 bits and stored in the A/D data register. When the number of the conversion count is sixteen in the addition mode, the A/D conversion addition result is expanded by 4 bits and stored in the A/D data register.

41.3.9 ADDRn : A/D Data Register n (n = 0 to 3 for unit 0, 1, n = 0 to 14 for unit 2)

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x020 + 0x002 × n (ADC120, ADC121: n = 0 to 3, ADC122: n = 0 to 14)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DR[15:0]	The result of A/D conversion (n: Number of channel)	R

A/D-converted value is 12-bit accuracy. The formats for data in the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers vary according to the following conditions.

- The setting of the A/D data register format select bit (ADCER.ADRFMT) (determining whether the data are flush-left or flush-right in the registers)
- The setting of the Addition count select bits (ADADC.ADC[2:0]) (once, twice, three times, four times, or sixteen times is selectable.)
- The setting of the average mode enable bit (ADADC.AVEE) (Addition or average is selectable.)

When the number of the conversion count is set one to four in the addition mode, the A/D conversion addition result is expanded by 2 bits and stored in the A/D data register. When the number of the conversion count is sixteen in the addition mode, the A/D conversion addition result is expanded by 4 bits and stored in the A/D data register.

41.3.10 ADSHCR : A/D Sample and Hold Control Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x66

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SHANS[2:0]			SSTSH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Bit	Symbol	Function	R/W
7:0	SSTSH[7:0]	Sample and hold period setting These bits set the sample and hold period in the range from 8 to 255 states. S/H period is set 12 + this register value.	R/W
10:8	SHANS[2:0]	Sample and hold use or bypass select for ch0-2 SHANS[2] corresponds to ch2, SHANS[0] corresponds to ch0. When sample and hold is used, In case of 1 channel: set 001b (use channel 0) In case of 2 channels: set 011b (use channel 0 and 1) In case of 3 channels: set 111b (use channel 0 to 2) Other settings are prohibited. 0: Bypass S/H 1: Use S/H	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

41.3.11 ADELCCR : A/D Event Link Control Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x7D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GCELC	ELCC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ELCC[1:0]	Event link control bits Control the end of scan event (ADCn_ADI). To enable the ELCC[1:0] setting, set GCELC to 0. 0 0: Output event signal at the end of scan except the end of group scan of group A, B 0 1: Output event signal at the end of group scan of group B 1 x: Output event signal at the end of scan	R/W
2	GCELC	Event control bit for Group C 0: Enable the setting of ELCC[1:0] 1: Disable the setting of ELCC[1:0] and output event signal at the end of group scan of group C	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

41.3.12 ADGSPCR : A/D Group Scan Priority Control Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x80

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GBRP	LGRRS	—	—	—	—	—	—	—	—	—	—	—	—	GBRSCN	PGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PGS	Group Priority Control Setting*1 0: Operation is without group priority control 1: Operation is with group priority control	R/W
1	GBRSCN	Group B Restart Setting (Enabled only when PGS = 1. Set 0 when PGS = 0.) 0: Scanning for group is not restarted when priority control is performed 1: Scanning for group is restarted when priority control is performed	R/W
13:2	—	These bits are read as 0. The write value should be 0.	R/W
14	LGRRS	Restart Channel Select (Enabled only when PGS = 1 and GBRSCN = 1, Set 0 when PGS = 0.) 0: Restart scan with first channel 1: Restart scan with suspended channel	R/W
15	GBRP	Group B Single Scan Continuous Start*2 (Enabled only when PGS = 1. Set 0 when PGS = 0.) 0: Single scan for group B is not continuously activated 1: Single scan for group B is continuously activated	R/W

Note 1. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode). If the bits are set to any other values, proper operation cannot be guaranteed.
Note 2. When the GBRP bit has been set to 1, single scan is performed continuously for low priority group regardless of the setting of the GBRSCN bit.

41.3.13 ADDBLDRA : A/D Data Duplication Register A

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x84

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DBLDRA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DBLDRA[15:0]	The result of A/D conversion during extended operation in double trigger mode	R

A/D-converted value is 12-bit accuracy. The formats for data in the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers vary according to the following conditions.

- The setting of the A/D data register format select bit (ADCER.ADRFMT) (determining whether the data are flush-left or flush-right in the registers)
- The setting of the Addition count select bits (ADADC.ADC[2:0]) (once, twice, three times, four times, or sixteen times is selectable.)
- The setting of the average mode enable bit (ADADC.AVEE) (Addition or average is selectable.)

When the number of the conversion count is set one to four in the addition mode, the A/D conversion addition result is expanded by 2 bits and stored in the A/D data register. When the number of the conversion count is sixteen in the addition mode, the A/D conversion addition result is expanded by 4 bits and stored in the A/D data register.

41.3.14 ADDBLDRB : A/D Data Duplication Register B

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x86

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DBLDRB[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DBLDRB[15:0]	The result of A/D conversion during extended operation in double trigger mode	R

A/D-converted value is 12-bit accuracy. The formats for data in the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers vary according to the following conditions.

- The setting of the A/D data register format select bit (ADCER.ADRFMT) (determining whether the data are flush-left or flush-right in the registers)
- The setting of the Addition count select bits (ADADC.ADC[2:0]) (once, twice, three times, four times, or sixteen times is selectable.)
- The setting of the average mode enable bit (ADADC.AVEE) (Addition or average is selectable.)

When the number of the conversion count is set one to four in the addition mode, the A/D conversion addition result is expanded by 2 bits and stored in the A/D data register. When the number of the conversion count is sixteen in the addition mode, the A/D conversion addition result is expanded by 4 bits and stored in the A/D data register.

41.3.15 ADWINMON : A/D Compare Function Window A/B Status Monitoring Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x8C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MONCMPB	MONCMPA	—	—	—	MONCOMB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MONCOMB	Combination result monitor 0: Window A/B combination condition does not match. 1: Window A/B combination condition match.	R
3:1	—	These bits are read as 0.	R
4	MONCMPA	Comparing result monitor for window A 0: Window A comparing condition does not match. 1: Window A comparing condition match.	R
5	MONCMPB	Comparing result monitor for window B 0: Window B comparing condition does not match. 1: Window B comparing condition match.	R
7:6	—	These bits are read as 0.	R

41.3.16 ADCMPCR : A/D Compare Function Control Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x90

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPAIE	WCMPE	CMPBIE	—	CMPAE	—	CMPBE	—	—	—	—	—	—	—	CMPAB[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMPAB[1:0]	Window A/B combination condition setting This bit is valid when both windows A/B are valid (CMPAE = 1 and CMPBE = 1). 0 0: Output ADCn_WCMPM when window A comparison condition match OR window B comparison condition match, otherwise output ADCn_WCMPUM 0 1: Output ADCn_WCMPM when window A comparison condition match EXOR window B comparison condition match, otherwise output ADCn_WCMPUM 1 0: Output ADCn_WCMPM when window A comparison condition match AND window B comparison condition match, otherwise output ADCn_WCMPUM 1 1: Setting prohibited.	R/W
8:2	—	These bits are read as 0. The write value should be 0.	R/W
9	CMPBE	Window B operation permission 0: Window B is not in operation. Output of ADCn_WCMPM/ADCn_WCMPUM is not permitted. 1: Window B is in operation.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	CMPAE	Window A operation permission 0: Window A is not in operation. Output of ADCn_WCMPM/ADCn_WCMPUM is not permitted. 1: Window A is in operation.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	CMPBIE	Compare window B Interrupt Enable 0: Generation of an ADCn_CMPBI interrupt in response to matches with a condition for comparison is disabled. 1: Generation of an ADCn_CMPBI interrupt in response to matches with a condition for comparison is enabled.	R/W
14	WCMPE	Window Function enable 0: Window function disabled 1: Window function enabled	R/W
15	CMPAIE	Compare window A Interrupt Enable 0: Generation of an ADCn_CMPAI interrupt in response to matches with a condition for comparison is disabled. 1: Generation of an ADCn_CMPAI interrupt in response to matches with a condition for comparison is enabled.	R/W

41.3.17 ADCMPANSR0 : A/D Compare Function Window A Channel Select Register 0

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x94

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPCHA0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPCHA0[15:0]	Window A Channel Select Set the CMPCHA0[15:0] bits when DCSR.ADST = 0. 0: Channel is not target 1: Channel is target	R/W

41.3.18 ADCMPLR0 : A/D Compare Function Window A Comparison Condition Setting Register 0

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x98

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPLCHA0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPLCHA0[15:0]	Window A comparison condition for target channel (ch0-15) setting CMPLCHA0[15] corresponds channel 15 and CMPLCHA0[0] corresponds channel 0. When the comparison result of each channel input matches the set condition, the ADCMPSR0.CMPSTCHA0n bit is set to 1 and a compare interrupt (ADCn_CMPAI) is generated. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 register value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 register value or ADCMPDR1 register value < A/D-converted value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 register value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W

41.3.19 ADCMPDR0 : A/D Comparison Function Window A Lower Level Setting Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x9C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPLLA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPLLA[15:0]	Reference data setting when using the compare function window A	R/W

41.3.20 ADCMPDR1 : AD Comparison Function Window A Upper Level Setting Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x9E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPULA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPULA[15:0]	Reference data setting when using the compare function window A	R/W

41.3.21 ADCMPSR0 : A/D Comparison Function Window A Channel Status Register 0

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0xA0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPSTCHA0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPSTCHA0[15:0]	Window A Status Flag This flag shows the comparison result of each targeted channel, when window A is operating (ADCMPCR.CMPAE = 1b). If this flag is set while ADCMPCR.CMPAIE = 1, a compare interrupt (ADCn_CMPAI) request is generated. 0: Condition does not match 1: Condition matches	R/W

41.3.22 ADWINLLB : A/D Compare Function Window B Lower-Side Level Setting Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0xA8

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPLLB[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPLLB[15:0]	Reference lower data setting when using the compare function window B	R/W

41.3.23 ADWINULB : A/D Compare Function Window B Upper-Side Level Setting Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0xAA

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPULB[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPULB[15:0]	Reference upper data setting when using the compare function window B	R/W

41.3.24 ADCMPBSR : A/D Compare Function Window B Status Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0xAC

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CMPSTB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPSTB	Window B Flag When window B operation is enabled (ADCMPCR.CMPBE = 1b), this bit indicates the comparison result of channels to which window B comparison conditions are applied. 0: Comparison conditions are not met 1: Comparison conditions are met	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

41.3.25 ADCMPBNSR : A/D Compare Function Window B Channel Select Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0xA6

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CMPLB	—	CMPCHB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CMPCHB[5:0]	Window B Channel Select Select channels for comparing with window B condition. Set the CMPCHB[5:0] bits when the ADCSR.ADST bit is 0. 0x00: ch0 0x01: ch1 0x02: ch2 ⋮ 0x0F: ch15 0x3F: No channel is selected Others: Setting prohibited	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	CMPLB	Window B Comparison Condition Setting When the comparison result of target input matches the set condition, the ADCMPBSR.CMPSTB bit is set to 1 and a compare interrupt (ADCn_CMPBI) is generated. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADWINLLB register value > A/D-converted value. When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADWINLLB register value or ADWINULB register value < A/D-converted value. 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADWINLLB register value < A/D-converted value. When window function is enabled (ADCMPCR.WCMPE = 1): ADWINLLB register value < A/D-converted value < ADWINULB register value.	R/W

41.3.26 ADANSC0 : A/D Channel Select Register C0

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0xD4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSC0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSC0[15:0]	A/D-Converted Channel Select for Group C in Group Scan Mode ANSC0[15] corresponds ch15, and ANSC0[0] corresponds ch0. 0: Channel is not target 1: Channel is target	R/W

41.3.27 ADGCTRGR : A/D Group C Trigger Select Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0xD9

Bit position:	7	6	5	4	3	2	1	0
Bit field:	GRCE	GCADIE	TRSC[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	TRSC[5:0]	Group C A/D Conversion Start Trigger Select Select starting trigger for A/D conversion of group C in group scan mode. See Table 41.8 for more details.	R/W
6	GCADIE	Group C Scan Completion Interrupt Enable 0: Group C scan completed interrupt (ADCn_GCADI) is disabled 1: Group C scan completed interrupt (ADCn_GCADI) is enabled	R/W
7	GRCE	Group C A/D Conversion Enable 0: Group C A/D conversion is disabled 1: Group C A/D conversion is enabled	R/W

Table 41.8 A/D conversion start trigger for Group C in group scan mode (1 of 2)

Module	Trigger	Description	TRSC					
			[5]	[4]	[3]	[2]	[1]	[0]
No Trigger is selected			1	1	1	1	1	1
MTU3	TRGA0N	MTU0.TGRA Compare match/Input capture	0	0	0	0	0	1
MTU3	TRGA1N	MTU1.TGRA Compare match/Input capture	0	0	0	0	1	0
MTU3	TRGA2N	MTU2.TGRA Compare match/Input capture	0	0	0	0	1	1
MTU3	TRGA3N	MTU3.TGRA Compare match/Input capture	0	0	0	1	0	0
MTU3	TRGA4N	MTU4.TGRA Compare match/Input capture, or underflow (trough) of MTU4.TCNT in PWM mode	0	0	0	1	0	1
MTU3	TRGA6N	MTU6.TGRA Compare match/Input capture	0	0	0	1	1	0
MTU3	TRGA7N	MTU7.TGRA Compare match/Input capture, or underflow (trough) of MTU7.TCNT in PWM mode	0	0	0	1	1	1
MTU3	TRG0N	MTU0.TGRE Compare match	0	0	1	0	0	0
MTU3	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1

Table 41.8 A/D conversion start trigger for Group C in group scan mode (2 of 2)

Module	Trigger	Description	TRSC					
			[5]	[4]	[3]	[2]	[1]	[0]
MTU3	TRG4BN	Compare match between MTU4.TADCORB and NTU4.TCNT	0	0	1	0	1	0
MTU3	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
MTU3	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	1	0	0
MTU3	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	1	1	0	1
MTU3	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	0
MTU3	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	1
MTU3	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and Compare match between MTU7.TADCORB and MTU7.TCNT	0	1	0	0	0	0
ELC	ADCn A		0	1	0	0	0	1
ELC	ADCn B		0	1	0	0	1	0

Note: Trigger source from MTU3 is not available in ADC12 unit 2.

41.3.28 ADSSTRn : A/D Sampling State Register n (n = 0 to 3 for unit 0, 1, n = 0 to 14 for unit 2)

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0xE0 + 0x01 × n (ADC120, ADC121: n = 0 to 3, ADC122: n = 0 to 14)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SST[7:0]							
Value after reset:	0	0	0	0	1	0	1	1

Bit	Symbol	Function	R/W
7:0	SST[7:0]	Sampling Time Setting These bits set the sampling time in the range from 7 to 255 states.	R/W

41.3.29 ADCALCTL : A/D Calibration Control Register

Base address: ADC12m = 0x9001_4000 + 0x0400 × m (m = 0, 1)
ADC122 = 0x8000_8000

Offset address: 0x1F0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CAL_ERR	CAL_RDY	CAL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CAL	Calibration Start 0: No calibration. 1: Start calibration.	R/W
1	CAL_RDY	Calibration Status 0: Calibration is not started, or on progress. 1: Calibration is completed.	R

Bit	Symbol	Function	R/W
2	CAL_ERR	Calibration Result 0: Normal end. 1: Abnormal end.	R
15:3	—	These bits are read as 0. The write value should be 0.	R/W

The ADCALCTL register controls calibration of A/D Converter. After reset is released, make sure to calibrate A/D Converter before A/D conversion.

41.4 Operation

41.4.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

There are three operating modes: single scan mode, continuous scan mode, and group scan mode. In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADCSR.ADST bit is cleared to 0 from 1. In group scan mode, the selected channels of group A and the selected channels of group B (and the selected channels of group C) are scanned once after starting to be scanned according to the respective synchronous triggers (MTU3, ELC).

In single scan mode and continuous scan mode, A/D conversion is performed for ANn pins selected by the ADANSA0 register, starting from the pin with the smallest number n. In group scan mode, A/D conversion is performed for ANn pins selected by the ADANSA0 register for group A, and performed for ANn pins selected by the ADANSB0 register for group B and performed for ANn pins selected by the ADANSC0 register for group C, respectively, starting from the pin with the smallest number n.

Double trigger mode is to be used with single scan mode or group scan mode. With double trigger mode being enabled, A/D conversion data of a channel selected by the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by any of the synchronous triggers (MTU3, ELC) selected by the ADSTRGR.TRSA[5:0] bits. In group scan mode, only group A can use the double trigger.

To enable double trigger extended mode, select double trigger mode, and then select the synchronous trigger by the ADSTRGR.TRSA[5:0].

- ADSTRGR.TRSA[5:0] = 001011b
  TRG4AN or TRG4BN
- ADSTRGR.TRSA[5:0] = 001111b
  TRG7AN or TRG7BN

In the double trigger extended mode, in addition to the normal double trigger mode operation, the A/D conversion data activated by the odd number trigger (TRG4AN, TRG7AN) is stored in the A/D data duplex register A (ADDBLDRA), and the A/D conversion data activated by the even number trigger (TRG4BN, TRG7BN) is stored in the A/D Data Duplication register B (ADDBLDRB).

If two types of trigger source occur at the same time, the data is not sorted by the trigger factor and the A/D conversion data is stored in the Data Duplication register B (ADDBLDRB). If one trigger is performing A/D conversion and the other trigger is input, the other trigger will be ignored.

Analog input AN000 to AN002 can use the S/H circuit by setting of ADSHCR.SHANS[2:0].

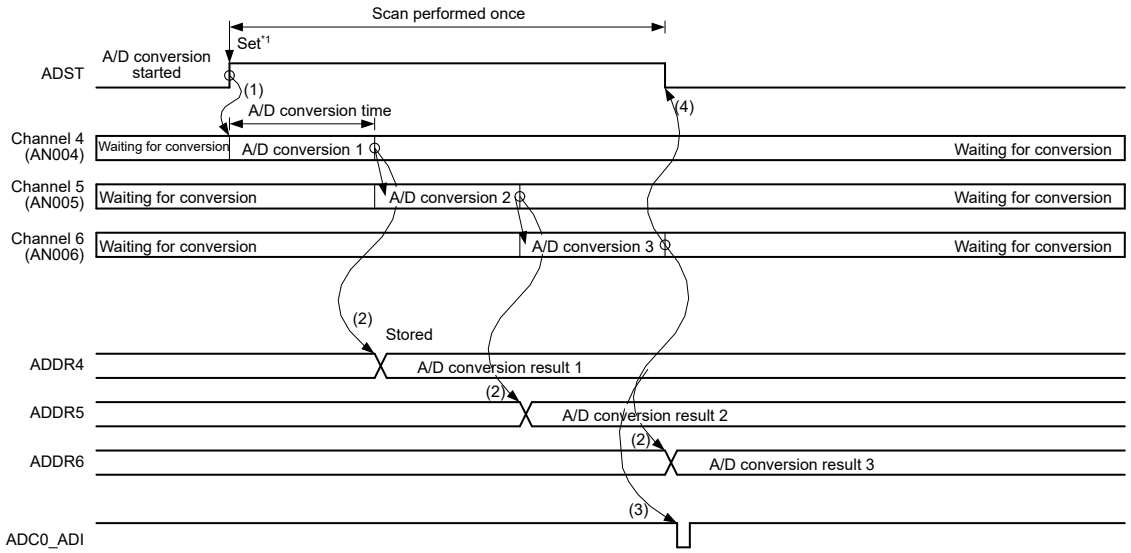
41.4.2 Single Scan Mode

41.4.2.1 Basic Operation (Not use S/H circuits)

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) due to software or synchronous trigger (MTU3, ELC) input, A/D conversion starts in ascending order of n of ANn selected in the ADANSA0 register.
2. When the A/D conversion of one channel is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).

3. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning) after A/D conversion of all selected channels is completed, an ADC0_ADI interrupt request (pulse output) is generated.
4. ADCSR.ADST bit holds 1 (A/D conversion start) during A/D conversion and is automatically cleared when A/D conversion of all selected channels is completed, and the 12-bit A/D converter goes into a standby state.



Note 1. The arrows in this figure indicate instruction execution by software.

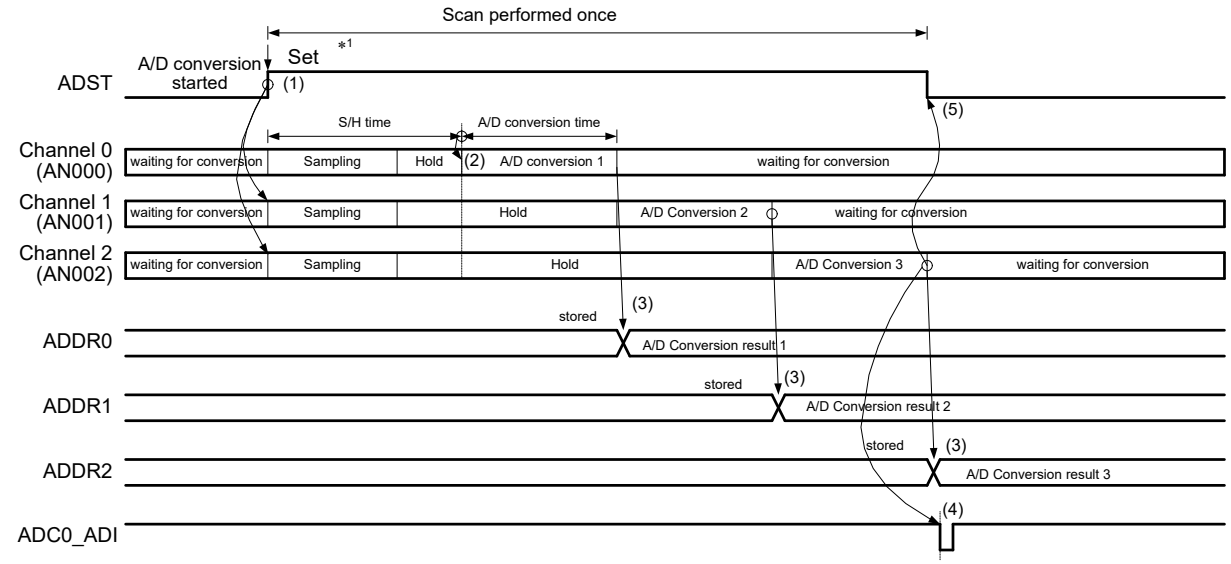
Figure 41.3 Example of operation in single scan mode (basic operation AN004 to AN006 selected)

41.4.2.2 Basic Operation (Use S/H circuits)

The S/H circuit is used to A/D Convert the analog inputs of all selected channels after S/H is performed only once.

The channel that uses the S/H circuit is selected with the ADSHCR.SHANS[2:0] bits.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by software or synchronization trigger (MTU3, ELC), all channels using the S/H circuit start sampling analog inputs.
2. After sampling and holding, A/D conversion is started in ascending order of n of the channel ANn selected in the ADANSA0 register.
3. When the A/D conversion of the channel is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
4. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning) after A/D conversion of all selected channels is completed, an ADC0_ADI interrupt request (pulse output) is generated.
5. The ADCSR.ADST bit holds 1 (A/D conversion start) during A/D conversion and is automatically cleared when A/D conversion of all selected channels is completed, and the 12-bit A/D converter goes into a standby state.



Note 1. The arrows in this figure indicate instruction execution by software.

Figure 41.4 Example of operation in single scan mode (basic operation with S/H, AN000 to AN002 selected)

41.4.2.3 A/D conversion in Double Trigger Mode

When double trigger mode is selected in single scan mode, A/D conversion is performed for two rounds of single scan operation started by a synchronous trigger (MTU3, ELC).

A/D conversion Data Duplication is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting ADCSR.DBLE to 1. When ADCSR.DBLE is set to 1, the channel selection of the ADANSA0 register is disabled.

To select the double trigger mode, use ADSTRGR.TRSA[5:0] to select the synchronous trigger (MTU3, ELC), set the ADCSR.EXTRG bit to 0, and set the ADCSR.TRGE bit to 1. Also, do not use software trigger.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) due to the synchronous trigger (MTU3, ELC) input, the A/D conversion of the 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
2. When the A/D conversion is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
3. ADCSR.ADST is automatically cleared and the 12-bit A/D converter is in the standby state. At this time, the ADC0_ADI interrupt does not occur regardless of the setting of the ADCSR.ADIE bit (ADC0_ADI interrupt enabled by the end of scanning).
4. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by the second synchronous trigger input, A/D conversion of the 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
5. When the A/D conversion is completed, the A/D conversion result is stored in the A/D Data Duplication register (ADDBLDR) dedicated to the double trigger mode.
6. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning), an ADC0_ADI interrupt request (pulse output) is generated.
7. The ADCSR.ADST bit holds 1 (A/D conversion start) during A/D conversion, is automatically cleared when A/D conversion is completed, and the 12-bit A/D converter is in the standby state.

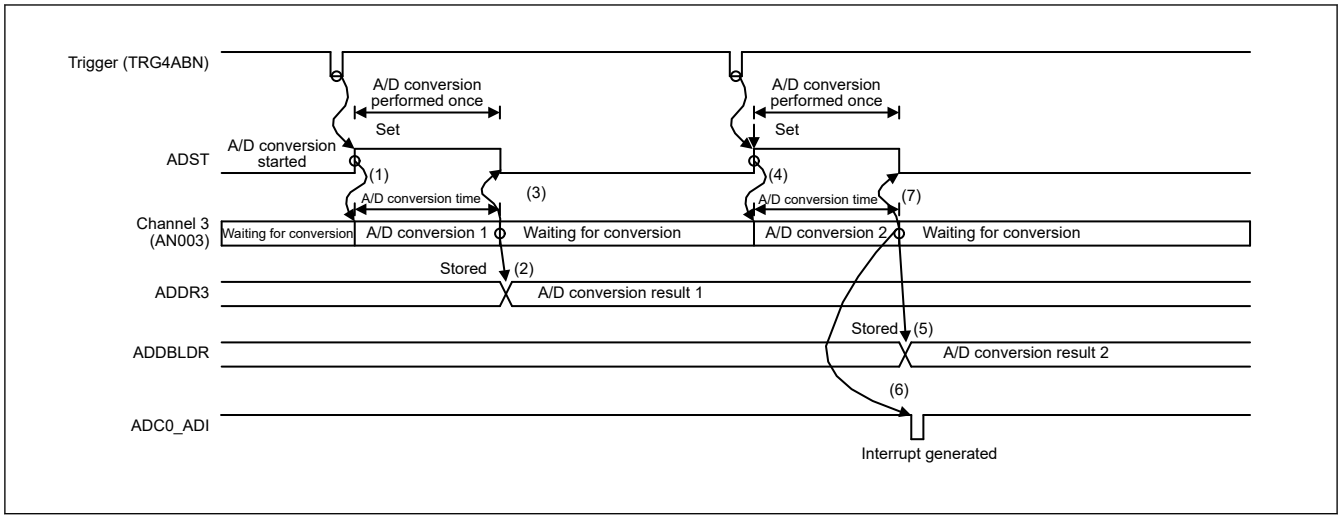


Figure 41.5 Example of operation in single scan mode (double trigger mode, AN003 duplicated, TRG4ABN is selected as the trigger)

41.4.2.4 Extended Operations When Double Trigger Mode is Selected

When double trigger mode is selected in single scan mode, single scan is performed twice.

To enable double trigger extended mode, and then select the following synchronous trigger by the ADSTRGR.TRSA[5:0].

- ADSTRGR.TRSA[5:0] = 001011b
  TRG4AN or TRG4BN
- ADSTRGR.TRSA[5:0] = 001111b
  TRG7AN or TRG7BN

And set the ADCSR.EXTRG bit to 0 and the ADCSR.TRGE bit to 1. Also, do not use software trigger.

A/D conversion Data Duplication is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting ADCSR.DBLE to 1. When ADCSR.DBLE is set to 1, the channel selection of the ADANSA0 register is disabled.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by TRG4AN, A/D conversion of 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
2. When the A/D conversion is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy) and A/D Data Duplication register A (ADDBLDRA).
3. ADCSR.ADST is automatically cleared and the 12-bit A/D converter is in the standby state. At this time, the ADC0_ADI interrupt does not occur regardless of the setting of the ADCSR.ADIE bit (ADC0_ADI interrupt enabled by the end of scanning).
4. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by TRG4BN input, A/D conversion of 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
5. When the A/D conversion is completed, the A/D conversion result is stored in the A/D Data Duplication register (ADDBLDR) and the A/D Data Duplication register B (ADDBLDRB).
6. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning), an ADC0_ADI interrupt request (pulse output and level output) is generated.
7. The ADCSR.ADST bit holds 1 (A/D conversion start) during A/D conversion, is automatically cleared when A/D conversion is completed, and the 12-bit A/D converter is in the standby state.

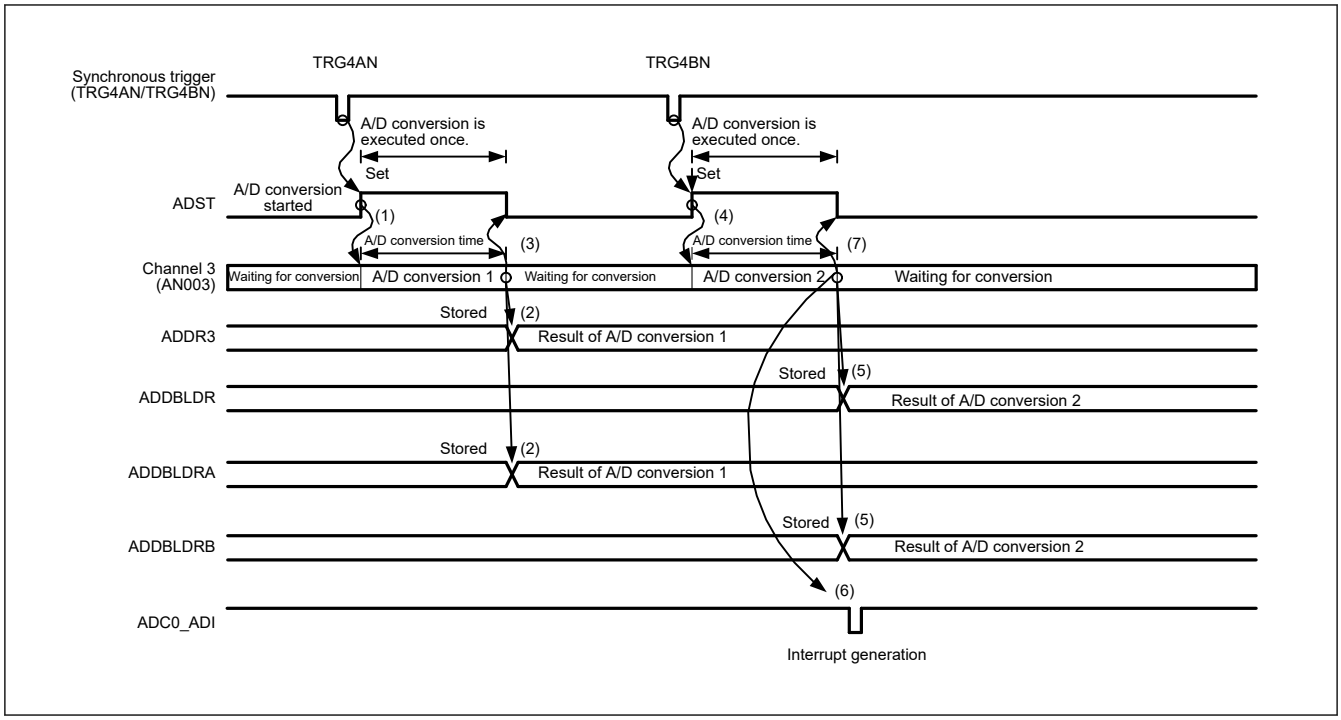


Figure 41.6 Example of extended operation in double trigger mode (duplication selected for AN003, TRG4AN, TRG4BN selected)

41.4.3 Continuous Scan Mode

41.4.3.1 Basic Operation (Not use S/H circuits)

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the channels selected by the ADANSA0 register.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by software, synchronous trigger (MTU3, ELC), or asynchronous trigger, A/D conversion starts in ascending order of n of ANn selected in the ADANSA0 register.
2. When the A/D conversion of one channel is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
3. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning) after A/D conversion of all selected channels is completed, an ADC0_ADI interrupt request (pulse output) is generated. In addition, the 12-bit A/D converter continuously starts A/D conversion in ascending order of n of ANn selected in the ADANSA0 register.
4. The ADCSR.ADST bit is not cleared automatically, and (2) to (3) are repeated during 1 (A/D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stopped), A/D conversion is stopped and the 12-bit A/D converter is in the standby state.
5. After that, when the ADCSR.ADST bit becomes 1 (A/D conversion starts), A/D conversion starts again in ascending order of n of ANn selected in the ADANSA0 register.

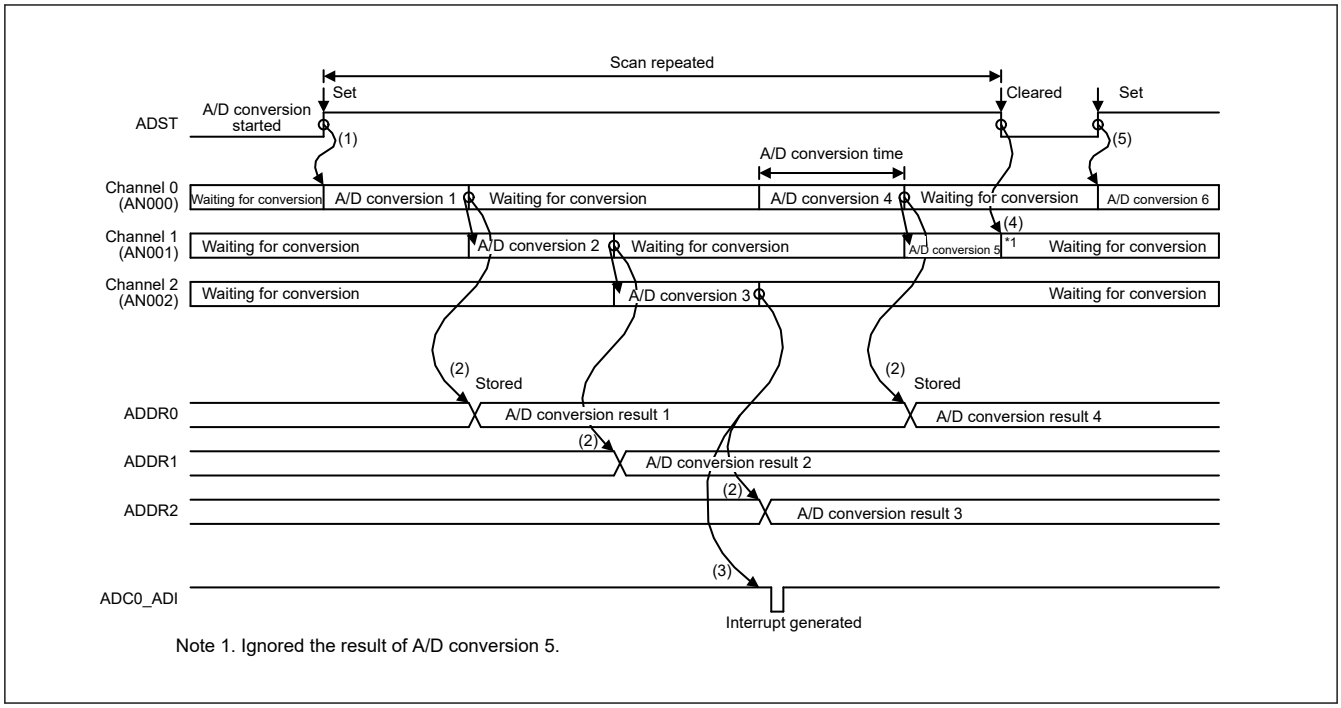


Figure 41.7 Example of operation in continuous scan mode (basic operation AN000 to AN002 selected)

41.4.3.2 Basic Operation (Use S/H circuits)

When the S/H circuit is used, the operation of A/D conversion of the analog inputs of all the selected channels is repeated after the sample and hold is performed. The channel that uses the S/H circuit is selected with the ADSHCR.SHANS[2:0] bits.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by software, synchronization trigger (MTU3, ELC), or asynchronization trigger, all channels using the S/H circuit start sampling analog inputs.
2. After sampling and holding, A/D conversion is started in ascending order of n of the channel ANn selected in the ADANSA0 register.
3. When the A/D conversion of the channel is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
4. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning) after A/D conversion of all selected channels is completed, ADC0_ADI interrupt request (pulse output) is generated. It also begins sampling the analog inputs of all channels that use the S/H circuitry.
5. The ADCSR.ADST bit is not cleared automatically, and (2) to (4) are repeated while it is set to 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stopped), A/D conversion is stopped and the 12-bit A/D converter is in the standby state.
6. After that, when the ADCSR.ADST bit becomes 1 (A/D conversion starts), sampling of analog inputs of all channels using the S/H circuit starts again.

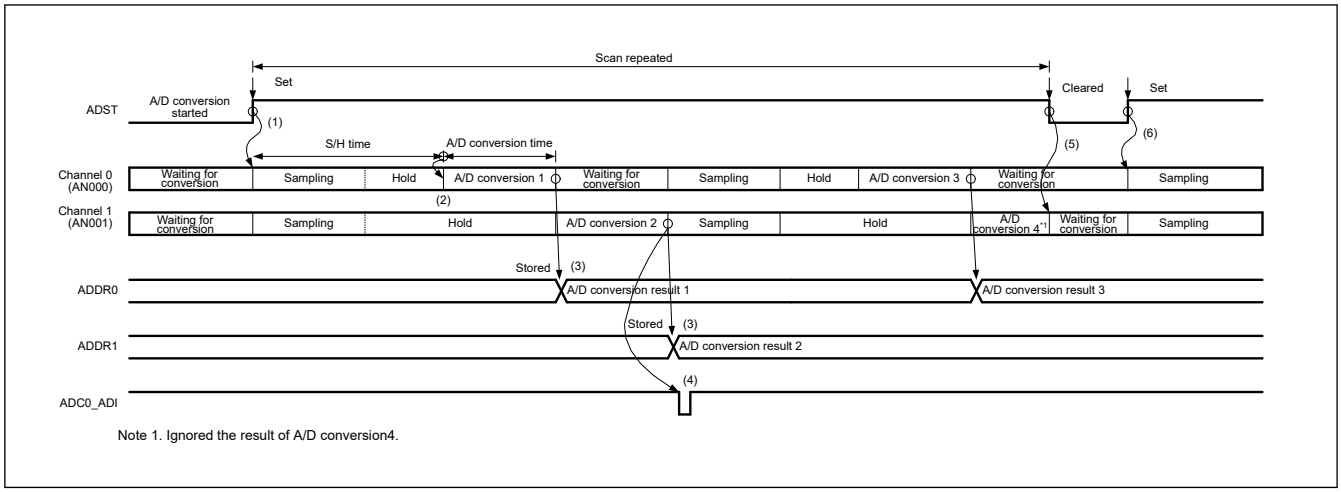


Figure 41.8 Example of operation in continuous scan mode (basic operation AN000 to AN001 selected, use S/H circuit)

41.4.4 Group Scan Mode

41.4.4.1 Basic Operation

In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in each group (A and B or A, B and C) after scanning is started by a synchronous trigger (MTU3, ELC). Scan operation of each group is similar to the scan operation in single scan mode.

To set the trigger for group scan mode, select the trigger for group A with the ADSTRGR.TRSA[5:0] bits, select the trigger for group B with the ADSTRGR.TRSB[5:0] bits, and select the trigger for group C with the ADGCTRGR.TRSC[5:0] bits. Triggers for groups A, B, and C should be separate triggers so that scans for groups A, B, and C do not occur at the same time. Also, do not use software triggers. For the channels to be scanned, select the group A channel in the ADANSA0 register, select the group B channel in the ADANSB0 register, and select the group C channel in the ADANSC0 register.

The following is an example of the operation of the group scan mode by the synchronous trigger from the MTU3. Group A is set to start conversion with a TRG4AN trigger from MTU3, Group B is set to start conversion with a TRG4BN trigger from MTU3, and Group C is set to start conversion with a TRG4ABN trigger from MTU3.

1. Start scanning of group A with TRG4AN trigger from MTU3.
2. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled) at the end of group A scanning, the ADC0_ADI interrupt is output.
3. Start scanning Group B with TRG4BN trigger from MTU3.
4. If the ADCSR.GBADIE bit is set to 1 (ADC0_GBADI interrupt enabled) at the end of Group B scanning, an ADC0_GBADI interrupt is output.
5. Trigger TRG4ABN from MTU3 to start scanning for Group C.
6. If the ADGCTRGR.GCADIE bit is set to 1 (ADC0_GCADI interrupt enabled) at the end of Group C scanning, an ADC0_GCADI interrupt is output.

In group scan mode, different channels and triggers should be selected for group A and group B.

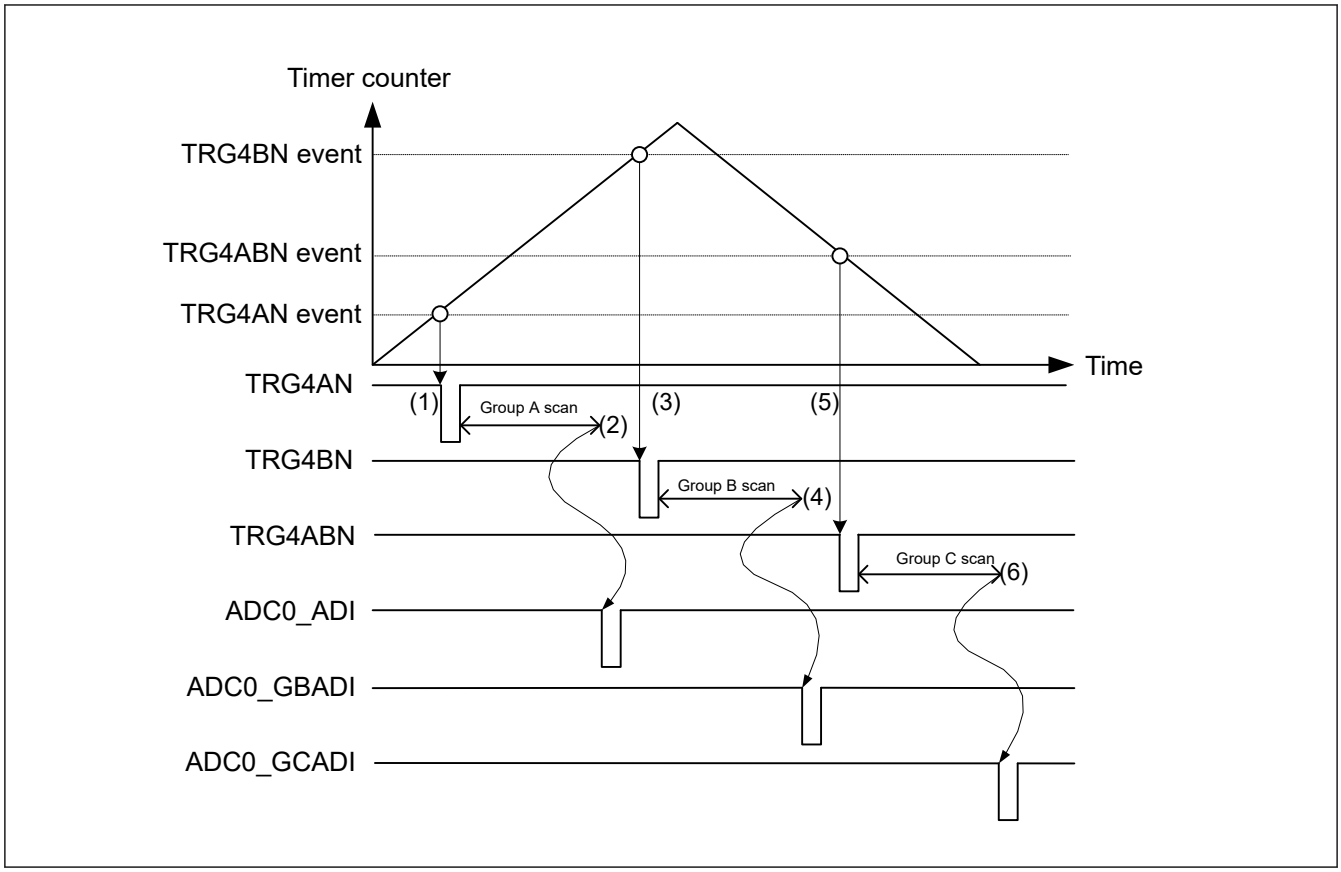


Figure 41.9 Example of operation in group scan mode (basic operation trigger from MTU3)

41.4.4.2 A/D conversion in Double Trigger Mode

When double trigger mode is selected in group scan mode, group A controls two single scan mode executions starting with a synchronous trigger (MTU3, ELC) as a series of operations. Group B and Group C behave in the same way as single scan mode started with a synchronous trigger (MTU3, ELC).For group scan mode trigger settings, the ADSTRGR.TRSA[5:0] bits select the trigger for group A, the ADSTRGR.TRSB[5:0] bits select the trigger for group B, and ADGCTRGR.TRSC[5:0] bits selects the trigger for Group C. Triggers for groups A, B, and C should be separate triggers so that scans for groups A, B, and C do not occur at the same time. Also, do not use software triggers.

If you select "TRG4AN" or "TRG4BN", "TRG7AN" or "TRG7BN" as the trigger for group A in the ADSTRGR.TRSA[5:0] bits, it operates in double trigger extended mode.

For the channels to be scanned, select the group A channel with the ADCSR.DBLANS[4:0] bits, select the group B channel with the ADANSB0 register, and select the group C channel with the ADANSC0 register.

A/D conversion Data Duplication is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.

The following shows an operation example when the group scan mode and double trigger mode are set by the synchronous trigger from the MTU3. Group A is set to start conversion with a TRG4ABN trigger from MTU3, Group B is set to start conversion with a TRGA0N trigger from MTU3, and Group C is set to start conversion with a TRGA1N trigger from MTU3.

1. Start scanning Group C with the TRGA1N trigger from the MTU3.
2. If the ADGCTRGR.GCADIE bit is set to 1 (ADC0_GCADI interrupt enabled) at the end of Group C scanning, the ADC0_GCADI interrupt is output.
3. The TRGA0N trigger from the MTU3 starts scanning for group B.
4. If the ADCSR.GBADIE bit is set to 1 (ADC0_GBADI interrupt enabled) at the end of Group B scanning, an ADC0_GBADI interrupt is output.
5. The first TRG4ABN trigger from the MTU3 will start the first scan of Group A.

6. At the end of the first scan of Group A, the converted data is stored in ADDRy, and the ADC0_ADI interrupt request does not occur regardless of the setting of the ADCSR.ADIE bit.
7. The second TRG4ABN trigger from the MTU3 will start the second scan of Group A.
8. At the end of the second scan of group A, the conversion data is stored in ADDBLDR, and if the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled), the ADC0_ADI interrupt is output.
9. The second TRGA0N trigger from the MTU3 starts the second scan of group B.
10. At the end of the second scan of Group B, if the ADCSR.GBADIE bit is set to 1 (ADC0_GBADI interrupt enabled), an ADC0_GBADI interrupt is output.
11. The second TRGA1N trigger from the MTU3 starts the second scan of Group C.
12. At the end of the second scan of Group C, if the ADGCTRGR.GCADIE bit is set to 1 (ADC0_GCADI interrupt enabled), an ADC0_GCADI interrupt is output.

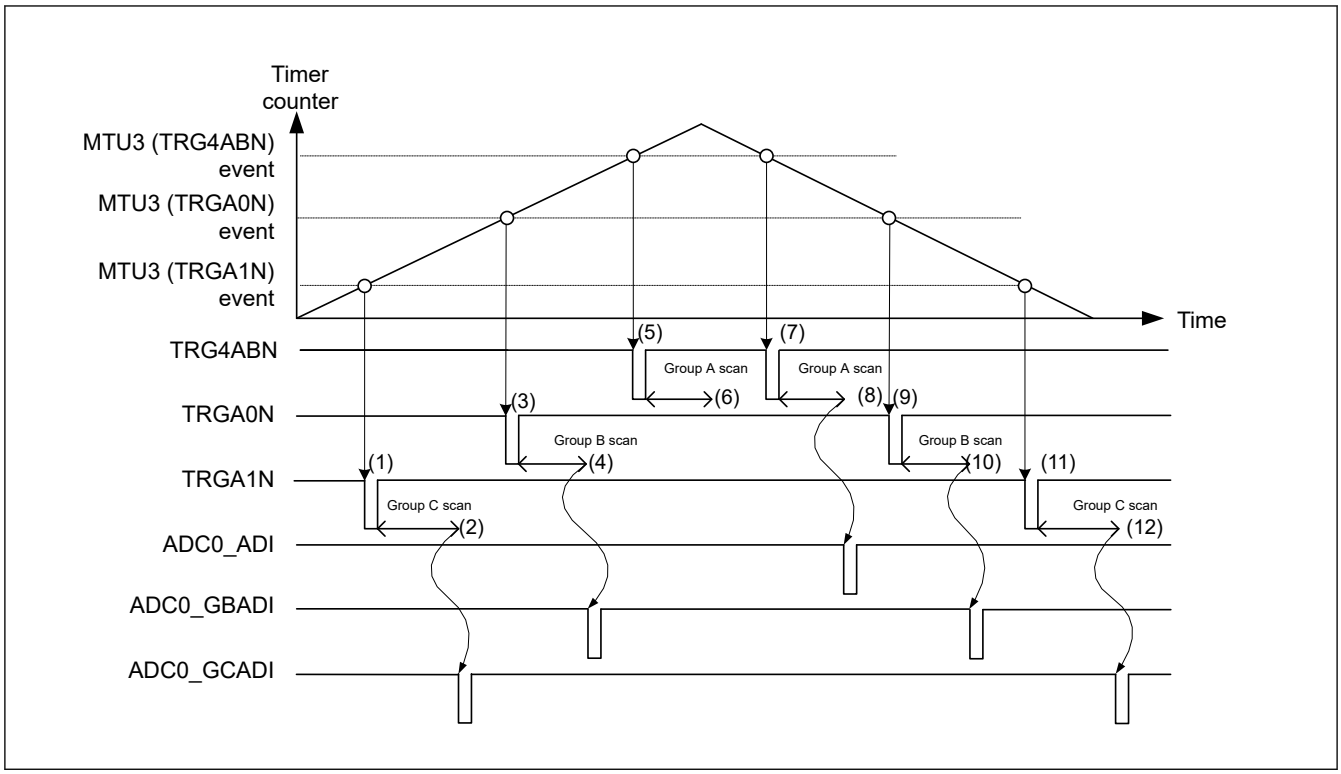


Figure 41.10 Example of operation in group scan mode with double trigger (trigger from MTU3)

41.4.4.3 Operation under Group Priority Control

When the ADGSPCR.PGS bit is set to 1 in group scan mode, group priority operation is performed. Group priority is in the order of group A > group B > group C. The number of groups used in group scan mode can be selected from either 2 (Groups A, B) or 3 (Groups A, B, C) depending on the setting of ADGCTRGR.GRCE. When setting the PGS bit in the ADPGSCR register to 1, follow the procedure in Figure 41.11 to set the relevant registers. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.

In the basic operation of group scan mode, the trigger input generated during scanning of groups A, B, and C is ignored, and the scanning operation of groups A, B, and C is the same as that of single scan mode.

In the group priority operation, if there is a trigger input of the priority group during the scan of the low priority group, the scan of the low priority group is interrupted and the priority group is scanned.

When the ADGSPCR.GBRSCN bit is 0, the low priority group is in the standby state after the priority group scan is completed. Also, the low priority group trigger input that occurs during the scan is ignored.

When the ADGSPCR.GBRSCN bit is 1, the scan of the low priority group is automatically re-executed after the scan of the priority group is completed. In addition, the trigger input of the low priority group that occurred during the scan of the

priority group is valid, and the scan of the low priority group is automatically executed after the scan of the priority group is completed.

When the ADGSPCR.GBRSCN bit is 1 and the ADGSPCR.LGRRS is 0, the scan of the low priority group is restarted from the beginning. Also, when ADGSPCR.LGRRS is 1, the scan of the low priority group is re-executed from the interrupted channel.

Table 41.9 shows the setting of the ADGSPCR.GBRSCN bit and the operation when a trigger is input during scanning.

When the ADGSPCR.GBRP bit is set to 1, the scan operation of the lowest priority group is the operation of executing a single scan continuously.

For the trigger setting in group scan mode, select the group A synchronous trigger with the ADSTRGR.TRSA[5:0] bits, and set the group B synchronous trigger different from the group A trigger with the ADSTRGR.TRSB[5:0] bits. Select a Group C synchronous trigger that is different from the Group A and B triggers on the ADGCTRGR.TRSC[5:0] bits.

When the group scan mode is set to 2 groups (ADGCTRGR.GRCE bit is set to 0) and the ADGSPCR.GBRP bit is set to 1, the ADSTRGR.TRSB[5:0] bits is set to 0x3F.

If the group scan mode is set to 3 groups (ADGCTRGR.GRCE bit is set to 1) and the ADGSPCR.GBRP bit is set to 1, set the ADGCTRGR.TRSC[5:0] bits to 0x3F.

For the channels to be scanned, select the group A channel in the ADANSA0 register, select the group B channel in the ADANSB0 register, and select the group C channel in the ADANSC0 register.

Table 41.10 shows Group priority operation setting and operation mode of 2 groups, and Table 41.11 shows Group priority operation setting and operation mode of 3 groups.

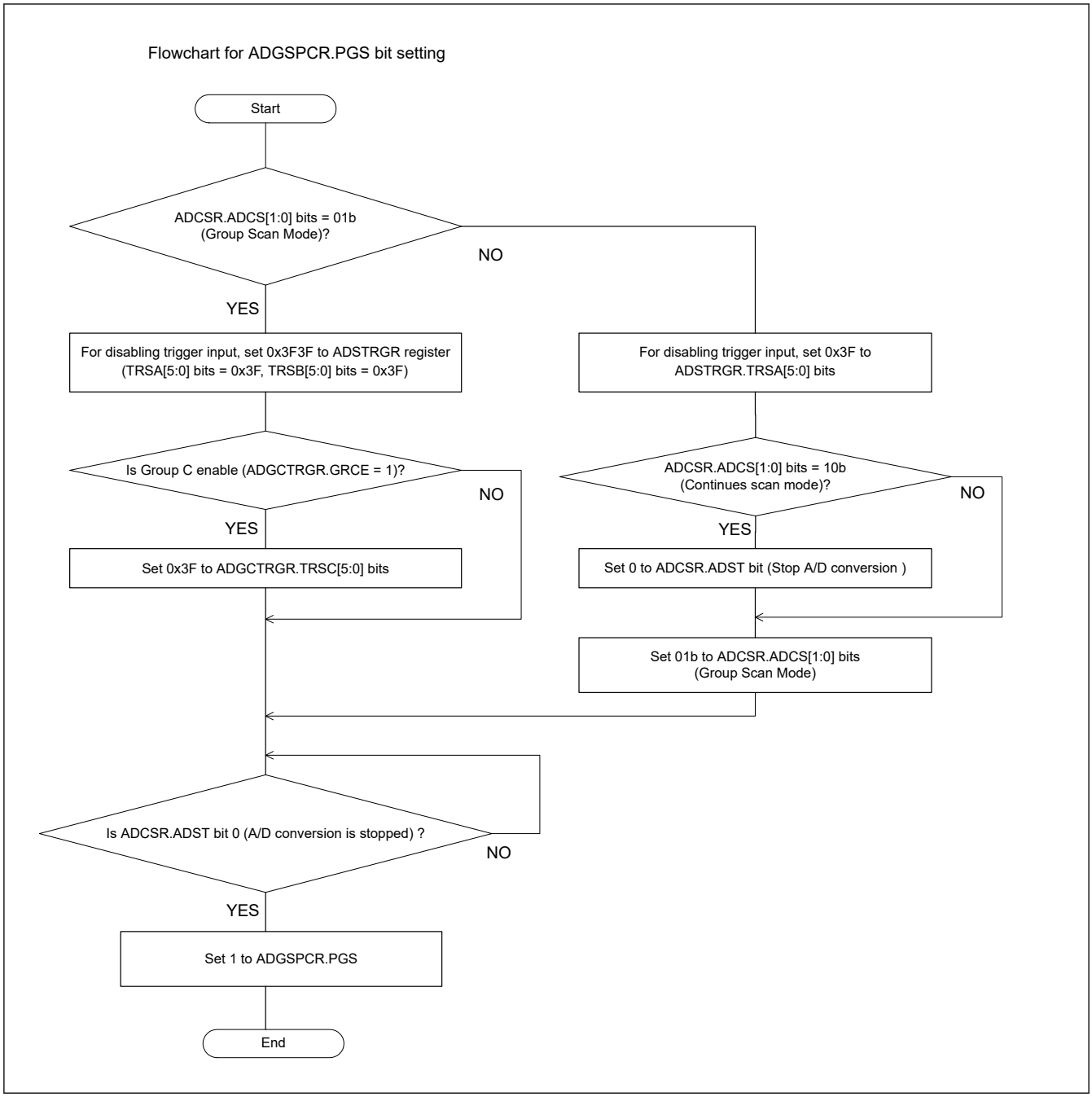


Figure 41.11 Flowchart for ADGSPCR.PGS bit setting

Table 41.9 Control of A/D conversion operations according to the settings of the ADGSPCR.GBRSCN bit (1 of 2)

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion is performed on group B after A/D conversion on group A is completed.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion is performed on group C after A/D conversion on group A is completed.

Table 41.9 Control of A/D conversion operations according to the settings of the ADGSPCR.GBRSCN bit (2 of 2)

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group B is in progress	Input of trigger for group A	Conversion for group B that is in progress is discontinued and conversion for group A starts.	• Conversion in progress for group B is discontinued and conversion for group A starts. • Conversion for group B starts after conversion for group A is completed.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion is performed on group C after A/D conversion on group B is completed.
When A/D conversion for group C is in progress	Input of trigger for group A	Conversion for group C that is in progress is discontinued and conversion for group A starts.	• Conversion in progress for group C is discontinued and conversion for group A starts. • Conversion for group C starts after conversion for group A is completed.
	Input of trigger for group B	Conversion for group C that is in progress is discontinued and conversion for group B starts.	• Conversion in progress for group C is discontinued and conversion for group B starts. • Conversion for group C starts after conversion for group B is completed.
	Input of trigger for group C	Trigger input is ineffective.	Trigger input is ineffective.

Table 41.10 Group priority operation setting and operation mode for 2 groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 0)

ADGSPCR			Operation mode
GBRSCN	LGRRS	GBRP	
0	—	0	Group priority operation of 2 groups (groups A and B). When the trigger of group A is input, the scan of group B ends (does not re-execute).
1	0	0	Group priority operation of 2 groups (groups A and B). After the scan of group B is interrupted, group B restarts the scan from the beginning of the channel specified by ADANSB0 after the scan of group A is completed.
1	1	0	Group priority operation of 2 groups (groups A and B). After the scan of group B is interrupted, group B resumes the scan from the interrupted channel among the channels specified by ADANSB0 after the scan of group A is completed.
—	0	1	Group priority operation of 2 groups (groups A and B). Group B continuously performs a single scan without inputting a start trigger. After the group B scan is interrupted, the single scan is restarted from the beginning of the channel specified by ADANSB0 after the group A scan is completed.
1	1	1	Group priority operation of 2 groups (groups A and B). Group B continuously performs a single scan without inputting a start trigger. After the scan of group B is interrupted, the single scan is restarted from the interrupted channel among the channels specified by ADANSB0 after the scan of group A is completed.

Table 41.11 Group priority operation setting and operation mode for 3 groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1) (1 of 2)

ADGSPCR			Operation mode
GBRSCN	LGRRS	GBRP	
0	—	0	Group priority operation of 3 groups (groups A, B, C). When the trigger of group A is input, the scan of group B ends (does not re-execute). When the trigger of group A or group B is input, the scan of group C ends (does not re-execute).

Table 41.11 Group priority operation setting and operation mode for 3 groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1) (2 of 2)

ADGSPCR			
GBRSCN	LGRRS	GBRP	Operation mode
0	—	1	Group priority operation of 3 groups (groups A, B, C). When the trigger of group A is input, the scan of group B ends (does not re-execute). Group C continuously performs a single scan without inputting a start trigger. After the group C scan is interrupted, the scan is restarted from the beginning of the channel specified by ADANSC0 after the group A/B scan is completed.
1	0	0	Group priority operation of 3 groups (groups A, B, C). After the scan of group B is suspended, the scan restarts from the top channel of the channels specified by ADANSB0 after the group A scan is completed. After the scan of group C is suspended, the scan restarts from the top channel of the channels specified by ADANSC0 after the group A/B scan is completed.
1	1	0	Group priority operation of 3 groups (groups A, B, C). After the scan of group B is suspended, the scan restarts from the interrupted channel of the channels specified by ADANSB0 after the group A scan is completed. After the scan of group C is suspended, the scan restarts from the interrupted channel of the channels specified by ADANSC0 after the group A/B scan is completed.
1	0	1	Group priority operation of 3 groups (groups A, B, C). After the scan of group B is interrupted, the scan is restarted from the beginning of the channel specified by ADANSB0 after the scan of group A is completed. Group C continuously performs a single scan without inputting a start trigger. After the group C scan is interrupted, the single scan is restarted from the beginning of the channel specified by ADANSC0 after the group A/B scan is completed.
1	1	1	Group priority operation of 3 groups (groups A, B, C). After the A/D conversion process of group B is interrupted, the interrupted channel is restarted from the channels specified by ADANSB0 after the scan of group A is completed. Group C continuously performs a single scan without inputting a start trigger. After the group C scan is interrupted, the single scan is restarted from the interrupted channel among the channels specified by ADANSC0 after the group A/B scan is completed.

41.4.5 Compare Function (Window A, Window B)

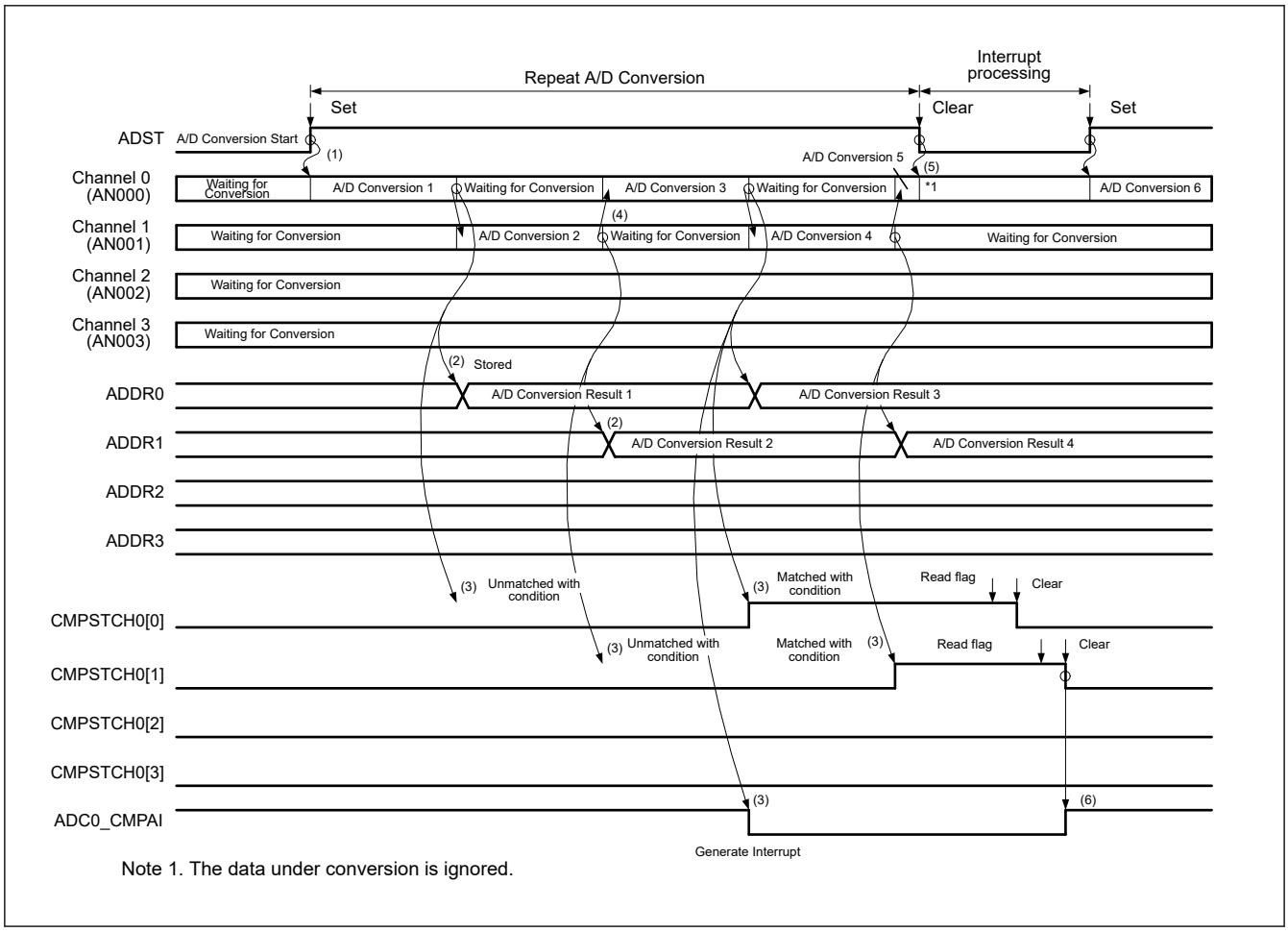
41.4.5.1 Compare Function Window A/B

The compare function is a function that compares the reference value set in the register with the A/D conversion result, and the reference value can be set for each window (A/B). Double trigger mode cannot be used when using the compare function. The major difference between window A and window B is that window B has one selectable channel and different interrupt output signals.

Operation using compare function in combination with continuous scan mode is described below.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by software, synchronous trigger (MTU3, ELC), or asynchronous trigger, A/D conversion of the selected channel is started.
2. When the A/D conversion is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy). When ADCMPCR.CMPAE = 1b, if it is set for window A in the ADCMPANSR0 register, it is compared with the ADCMPDR0 register setting value. When ADCMPCR.CMPBE = 1b, if the ADCMPBNSR register is set for window B, it will be compared with the ADWINULB/ADWINLLB register setting value.
3. As a result of comparison, when the condition set in ADCMPLR0 is matched in window A, the flag (ADCMPSR0. CMPSTCHA0n) bit of the compare function window A is set to 1. At this time, if the ADCMPCR.CMPAIE bit is set to 1, an ADC0_CMPAI interrupt request (level) is generated. Similarly, when window B matches the conditions set in ADCMPBNSR.CMPLB, the compare function window B flag (ADCMPBSR.CMPSTB) bit is set to 1. At this time, if the ADCMPCR.CMPBIE bit is set to 1, an ADC0_CMPBI interrupt request (level) is generated.
4. When all the selected A/D conversions and comparisons are completed, scanning is performed again.
5. After accepting the ADC0_CMPAI/ADC0_CMPBI interrupt, set the ADCSR.ADST bit to 0 (A/D conversion stopped) and execute processing for the channel for which the compare flag is set.

6. When all the compare flags in window A are cleared, the ADC0_CMPAI interrupt request is canceled. Similarly, clearing the compare window B flag cancels the ADC0_CMPBI interrupt request. If you want to perform the compare again, start the A/D conversion again.



Note 1. The data under conversion is ignored.

Figure 41.12 Example of operation of compare function (compare target: AN000-AN003)

41.4.5.2 Event output in Compare Function

For the event output of the compare function, specify the upper and lower reference values for each of windows A/B, compare the A/D conversion value of the selected channel with the upper/lower reference values, and then the event (ADCn_WCMPM/ADCn_WCMPUM) is output according to the event condition (A or B, A and B, A exor B) from the result of satisfying/failing the comparison condition of windows A and B. When using this function, perform A/D conversion in single scan mode.

In window A, any channel of ANn00-ANn15 can be selected. When multiple channels are selected in window A and the comparison condition is satisfied in one of the channels, the comparison condition is satisfied in window A. In window B, one channel of ANn00-ANn15 can be selected.

The setting procedure and setting example when using the event output of the compare function are shown below.

1. ADCSR.ADCS bit must be 00b (single scan mode).
2. Select the channel to be used for window A in the ADCMPANSR0 register, set the comparison condition of the compare function window in the ADCMPLR0 register, and set the upper/lower reference value in the ADCMPDR0/1 register.
3. Select the channel to be used for window B and the comparison condition in the ADCMPBNSR register, and set the upper/lower reference values in the ADWINULB/ADWINLLB registers.
4. In the ADCMPCR register, set the window A/B compound condition, window A/B operation permission, and interrupt output permission.

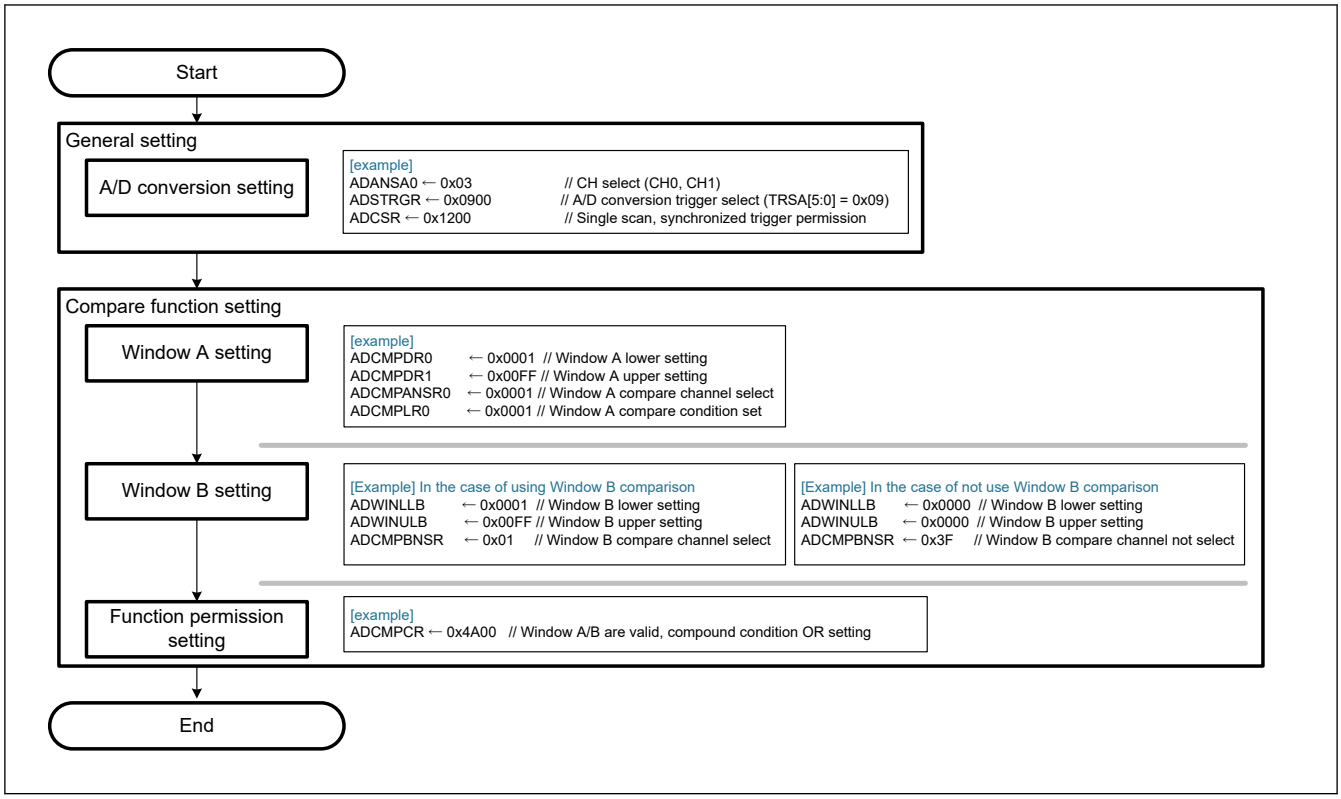


Figure 41.13 Example of setting for using event output of compare function

Precautions when using event output using only window A of the compare function

- Enable both windows A and B. (ADCMPCR.CMPAE = 1, ADCMPCR.CMPBE = 1)
- Set the compound condition setting for windows A and B to "OR condition". (ADCMPCR.CMPAB[1:0] = 00b)
- Set the comparison target channel in window B to "Unselected". (ADCMPBNSR.CMPCHB[5:0] = 111111b).
- Set the comparison condition of window B to 0 < conversion result < 0, which is "always inconsistent". (ADCMPCR.WCMPE = 1, ADWINLLB.CMPLLB[15:0] = ADWINULB.CMPULB[15:0] = 0x0000, ADCMPBNSR.CMPLB = 1)

Figure 41.14 shows example of event output operation of compare function.

The scan end event is output when one single scan ends. Also, depending on the setting of ADCMPCR.CMPAB[1:0], the match/unmatch event (ADCn_WCMPM/ADCn_WCMPUM) is output with a delay of 1 PCLK (Unit 0, 1: PCLKH, Unit 2: PCLKM).

Note: Match/unmatch events are exclusive outputs, and both events are not output at the same time.

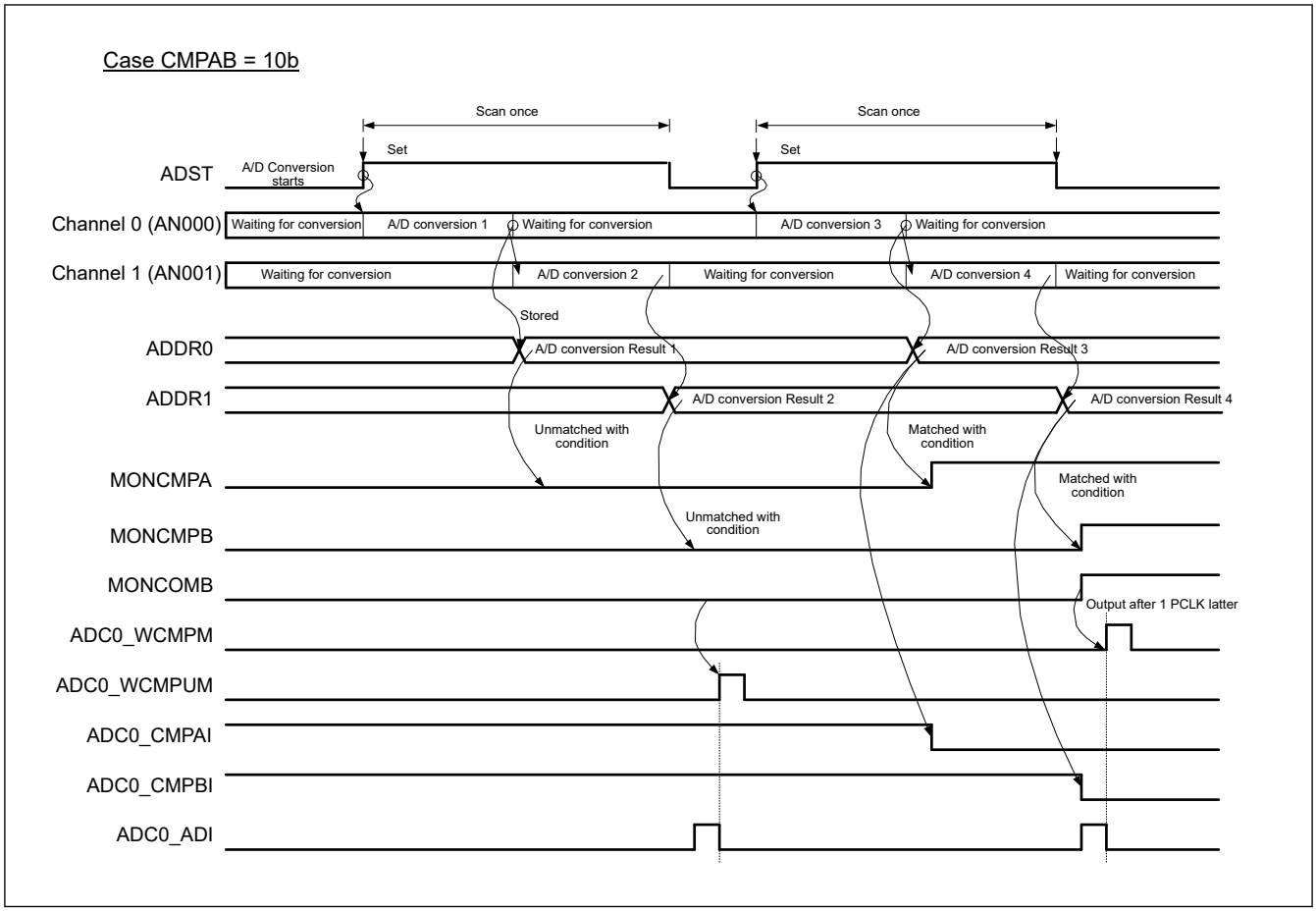


Figure 41.14 Example of operation for event output of compare function

Precautions when using event output of the compare function

The event output of the compare function outputs match/unmatch according to the setting of ADCMPCR.CMPAB[1:0] from the comparison result of window A and the comparison result of window B. The comparison result of window A is the logical sum of the comparison result of the channel to be compared in window A.

The comparison result of window A and the comparison result of window B are updated every A/D conversion of the channel to be compared, and the comparison result is retained even after the single scan is completed. To clear the comparison result (0), set ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0.

41.4.5.3 Compare function constraint

The compare function has the following restrictions.

- Double trigger mode cannot be used together.
  (ADDBLDR, ADDBLDRA, ADDBLDRB are not subject to the compare function.)
- If you want to use match/unmatch event output, set the single scan mode.
- The same CH cannot be set in window A and window B.
- Set so that the upper reference value ≥ the lower reference value.

41.4.6 A/D Data Register Automatic Clearing Function

By setting the ADCER.ACE bit to 1, these are used when the CPU and DMAC read the A/D data registers (ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB). The register can be cleared automatically to 0x0000.

By using this function, it is possible to detect an un-updated failure of the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers. The following is an example when the automatic clear function of the ADDRy register is disabled/enabled.

When the ADCER.ACE bit is 0 (automatic clearing prohibited), the old data (0x0111) becomes the value of the ADDRy register when the A/D conversion result (0x0222) is not written to the ADDRy register for some reason. Furthermore, when the value of this ADDRy register is read to the general-purpose register by using the A/D conversion end interrupt, the old data (0x0111) can be saved in the general-purpose register. However, when checking for un-updated data, it is necessary to check while holding old data in RAM and general-purpose registers one by one.

When the ADCER.ACE bit is 1 (automatic clearing enabled), the ADDRy register is automatically cleared to 0x0000 when ADDRy = 0x0111 is read by the CPU and DMAC. After that, when 0x0222 of the A/D conversion result cannot be transferred to the ADDRy register for some reason, the cleared data (0x0000) remains as the ADDRy register value. If the value of this ADDRy register is read to a general-purpose register by using the A/D conversion end interrupt, 0x0000 is held in the general-purpose register. By simply checking that the read data value is 0x0000, it can be determined that there was an un-updated failure of the ADDRy register.

41.4.7 A/D-converted Value Addition/Average Function

In A/D-converted value addition function, the same channel is A/D-converted two, three, four or sixteen consecutive times and the sum of the converted values is stored in the data register. In A/D-converted value average function, the same channel is A/D-converted two or four consecutive times and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that affect A/D conversion. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.

The A/D-converted value addition/average mode can be specified when A/D conversion of the analog input of the selected channels is selected.

41.4.8 Starting A/D Conversion with Asynchronous Trigger

The A/D conversion can be started by the input of an asynchronous trigger. To start up the A/D converter by an asynchronous trigger, the A/D conversion start trigger select bits (ADSTRGR.TRSA[5:0]) should be set to 000000b and a low-level signal should be input to the asynchronous trigger (ADTRGn#). Both the ADCSR.TRGE and ADCSR.EXTRG bits then should be set to 1. Figure 41.15 shows a timing of the asynchronous trigger input.

Asynchronous triggers cannot be selected for groups B and C used in group scan mode.

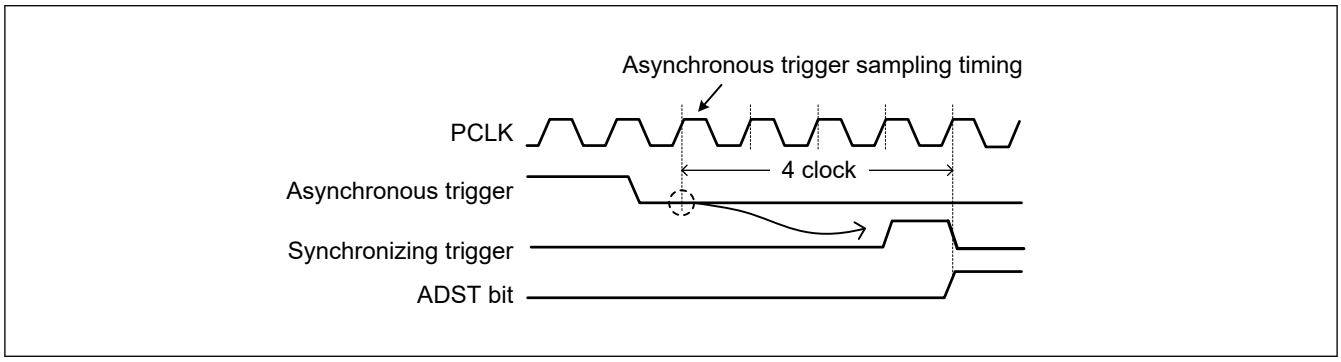


Figure 41.15 Timing of asynchronous trigger input

41.4.9 Starting A/D Conversion with Synchronous Trigger from Peripheral Module

A/D conversion can be started by the synchronization trigger from MTU3 and ELC (1 pulse of "L" in PCLK synchronization). When starting A/D conversion with a synchronous trigger, the ADCSR.TRGE bit is set to 1, the ADCSR.EXTRG bit is set to 0, the ADSTRGR.TRSA[5:0], ADSTRGR.TRSB[5:0], and ADGCTRGR.TRSC[5:0] bits are set to the appropriate trigger factor.

41.4.10 Event Link Function

41.4.10.1 Event output to ELC

ELC can use the ADCn_ADI interrupt request signal as an event signal (ADCn_ADI) to link to a preset module (ADCn_GBADI/ADCn_GCADI interrupt, ADCn_CMPAI/ADCn_CMPBI interrupt cannot be used as an event signal). The event signal can be output regardless of the setting of the corresponding interrupt request permission bit.

The 12-bit A/D converter outputs the scan end event (ADCn_ADI), compare function match event (ADCn_WCMPM), and unmatch event (ADCn_WCMPUM). The scan end event (ADCn_ADI) outputs a "High pulse" for 1 PCLK at the same output timing as the interrupt output (ADCn_ADI) regardless of the setting of ADCSR.ADIE.

The compare match/unmatch event (ADCn_WCMPM/ADCn_WCMPUM) to ELC outputs a "High pulse" for 1 PCLK at a timing delayed by 1 cycle (PCLK) from the interrupt output (ADCn_ADI) regardless of the setting of ADCSR.ADIE. When using the compare match/unmatch event to ELC (ADCn_WCMPM/ADCn_WCMPUM), set it to single scan mode.

41.4.10.2 Starting A/D Conversion with ELC event

The 12-bit A/D converter can start A/D conversion by a preset event by setting ELC_SSEL in ELC.

41.5 Usage Notes

41.5.1 Calibration

After reset is released, need to calibrate A/D Converter before A/D conversion. Calibration flow is the following:

- Release module stop of ADC12.
- Wait for at least 0.5 µs.
- Write ADCCALCTL.CAL bit to 1 to start calibration.
- Poll ADCCALCTL.CAL_RDY bit until it is changed to 1.
- After confirming ADCCALCTL.CAL_RDY is 1, confirm ADCCALCTL.CAL_ERR bit is 0.
- Write ADCCALCTL.CAL bit to 0.

41.5.2 Notes on Reading Data Registers

The A/D data registers, A/D data duplication registers, A/D data duplication register A, A/D data duplication register B, must be read in 16-bit word units, to avoid unmatch from upper value and lower value due to reading by byte units separately.

41.5.3 Notes on Stopping A/D Conversion

41.5.3.1 A/D Conversion Stop Procedure

If using asynchronous or synchronous trigger as the A/D conversion start condition, follow the procedure in the flowchart in Figure 41.16 to stop the A/D conversion.

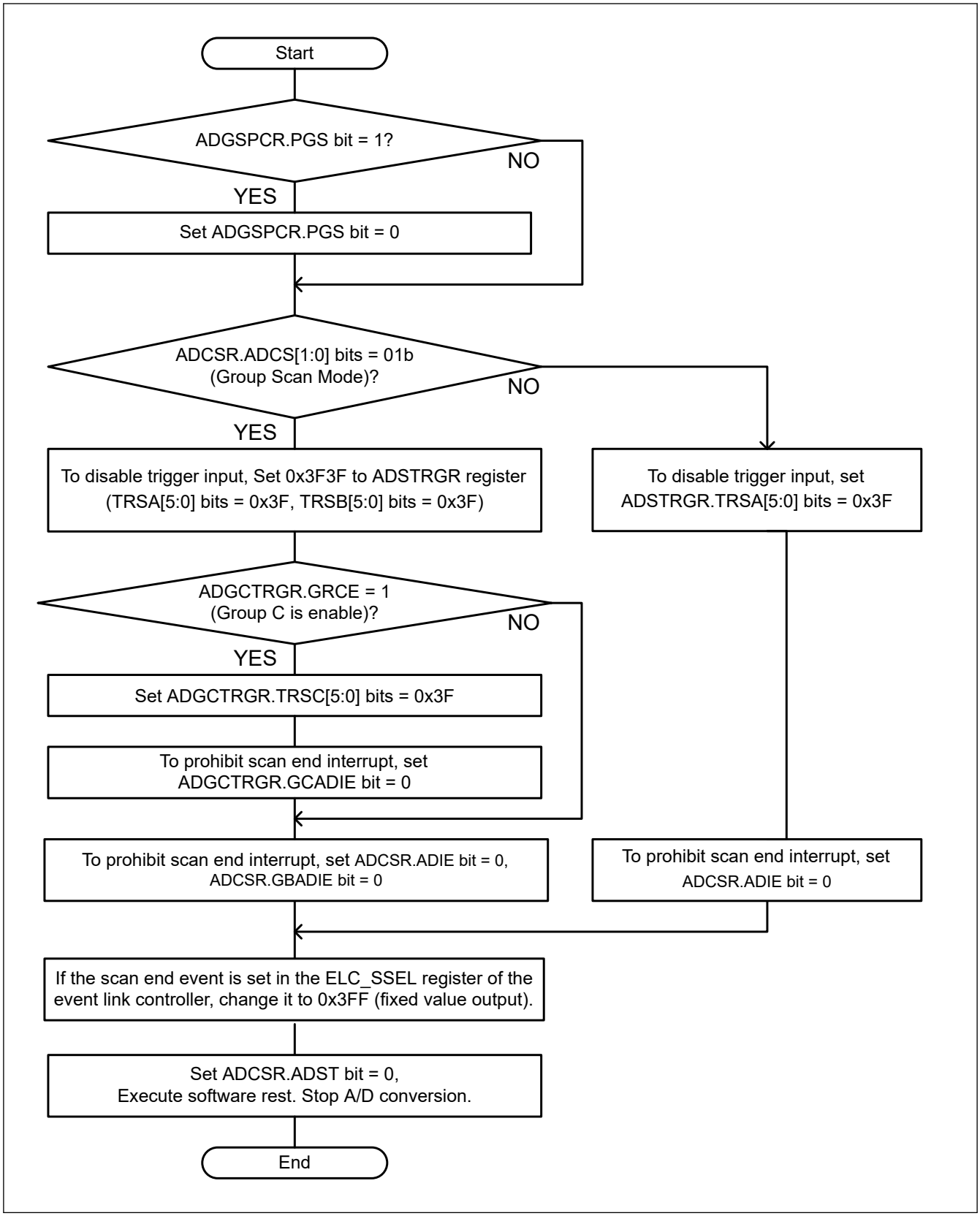


Figure 41.16 A/D Conversion stop procedure

It takes 2 PCLKADC cycles from a software clear until the soft macro stops scanning. If you want to set the following after executing a software clear, enter a wait of 2 PCLKADC cycles or more.

- Scan end interrupt enabled
- Event link controller scan end event valid setting

- Start of A/D conversion by software
- Trigger input valid setting

41.5.3.2 Notes on Mode/Status bits

Initialize or reset the odd/even judgment and monitor bits of the compare function in the double trigger mode as necessary.

- Double trigger mode starts from the first scan operation when ADCSR.DBLE is set from 0 to 1.
- To initialize the monitor bits (MONCMPA, MONCMPB, MONCOMB) of the compare function, set ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0.

41.5.4 A/D conversion forced stop and operation timing at starts

Up to 6 PCLKADCs time is necessary that the A/D converter starts by ADCSR.ADST bit = 1 from stopped status.

When the ADCSR.ADST bit is set to 0 and the A/D conversion is forcibly stopped, the analog part of the 12-bit A/D converter requires a maximum of 2 PCLKADCs to stop operating.

41.5.5 Notes on Scan End Interrupt Processing

When scanning the same analog input twice using any trigger, if the CPU does not complete reading out the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated, the first A/D-converted data is overwritten with the second A/D-converted data and an overwrite error occurs.

41.5.6 Module-Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled using the Module Stop Control register. The initial setting is for operation of the 12-bit A/D converter to be halted. Releasing the module-stop state enables access to the registers. After release from the module-stop state, wait for at least 0.5 µs before starting A/D conversion.

41.5.7 Notes on Entering Low Power Consumption States

When transitioning to module stop mode, stop A/D conversion. When stopping the A/D conversion, it is necessary to secure the time until the analog part of the 12-bit A/D converter stops after setting the ADCSR.ADST bit to 0.

41.5.8 Notes on A/D-converted Value Addition/Average Function

When using channel-dedicated sample-and-hold circuits, it is prohibited to use A/D-converted value addition/average function.

42. Temperature Sensor Unit (TSU)

42.1 Overview

The temperature sensor unit (TSU) indicates the junction temperature as a 12-bit binary digital output code. A temperature sensor unit is used to monitor the LSI junction temperature.

Table 42.1 lists the TSU specifications. Figure 42.1 shows a block diagram of TSU.

Table 42.1 TSU specifications

Item	Description
Number of units	1 unit
Temperature Code Read	• Temperature code averaging function • Comparison with the upper and lower limit settings • Single scan / Continuous scan
Interrupt source	• Conversion Complete • Comparison Result
Low power	• Power down mode

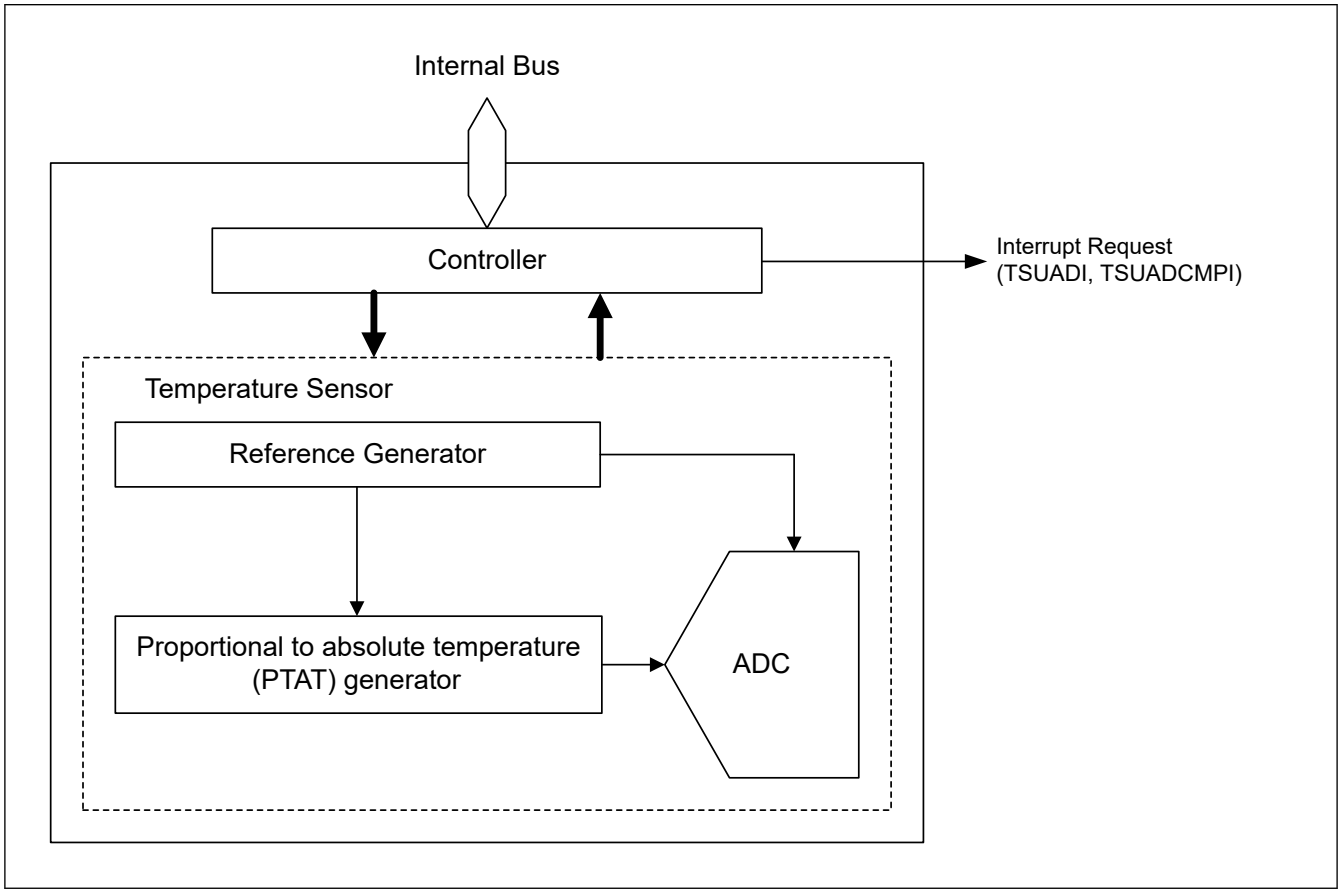


Figure 42.1 TSU block diagram

42.2 Register Map

Table 42.2 TSU register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x8008_6000	SSUSR	Sensor Suspend Register	—
0x8008_6004	STRGR	Sensor Trigger Register	—

Table 42.2 TSU register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x8008_6008	SOSR1	Sensor Operation Setting 1 Register	—
0x8008_6010	SCRR	Sensor Code Read Register	—
0x8008_6014	SSR	Sensor Status Register	—
0x8008_6018	CMSR	Compare Mode Setting Register	—
0x8008_601C	LLSR	Lower Limit Setting Register	—
0x8008_6020	ULSR	Upper Limit Setting Register	—
0x8008_6030	SISR	Sensor Interrupt Status Register	—
0x8008_6034	SIER	Sensor Interrupt Enable Register	—
0x8008_6038	SICR	Sensor Interrupt Clear Register	—

Table 42.3 TSU related system control register map

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
All the registers	—	MSTPCRD.MSTPCRD07	SLVACCCTL1.TSU_SL

42.3 Register Descriptions

42.3.1 SSUSR : Sensor Suspend Register

Base address: TSU = 0x8008_6000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SOC_TS_EN	ADC_PD	EN_TS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
0	EN_TS	Temperature Sensor Enable 0: Temperature Sensor power down 1: Temperature Sensor normal operation	R/W
1	ADC_PD	ADC Power Down 0: Normal operation 1: Power down	R/W
2	SOC_TS_EN	SOC_TS (Start signal of the conversion process) Enable 0: Disable 1: Enable	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The SSUSR register controls settings related to the power-down of TSU.

42.3.2 STRGR : Sensor Trigger Register

Base address: TSU = 0x8008_6000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEND	ADST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

Bit	Symbol	Function	R/W
0	ADST	TSU ADC Start Trigger 0: No effect 1: Conversion starts	W
1	ADEND	TSU ADC Stop Trigger 0: No effect 1: conversion ends	W
31:2	—	The write value should be 0.	W

The STRGR register controls a start and an end trigger to ADC in the TSU.

42.3.3 SOSR1 : Sensor Operation Setting 1 Register

Base address: TSU = 0x8008_6000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	OUTSEL	—	—	—	—	ADCS	—	—	ADCT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

Bit	Symbol	Function	R/W
1:0	ADCT[1:0]	AD Count Setting 0 0: 1 0 1: 2 1 0: 4 1 1: 8	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	ADCS	ADC Scan Mode Select 0: Single scan mode 1: Continuous scan mode	R/W
8:5	—	These bits are read as 0. The write value should be 0.	R/W
9	OUTSEL	Select data read from Sensor Code Read register (SCRR) 0: Latest read data 1: Average data	R

Bit	Symbol	Function	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The SOSR1 register controls the operation setting of ADC.

42.3.4 SCRR : Sensor Code Read Register

Base address: TSU = 0x8008_6000

Offset address: 0x10

Bit position:	31														11															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OUT12BIT_TS[11:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	OUT12BIT_TS[11:0]	Temperature Sensor Code Data	R
31:12	—	These bits are read as 0.	R

The SCRR register is used to read acquired code data.

42.3.5 SSR : Sensor Status Register

Base address: TSU = 0x8008_6000

Offset address: 0x14

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CONV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CONV	TSU Operating Status 0: Conversion is completed. Single scan mode: AD conversion of the number of times needed for average data is completed. Continuous mode: AD conversion of the number of times needed for average data is completed after ADEND is asserted. 1: Conversion is in progress. Set when ADST bit of Sensor Trigger Register is asserted.	R
31:1	—	These bits are read as 0.	R

The SSR register is used to read the TSU operating status.

42.3.6 CMSR : Compare Mode Setting Register

Base address: TSU = 0x8008_6000

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CMPCOND	CMPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	CMPEN	Compare Enable 0: Disable 1: Enable	R/W
1	CMPCOND	Compare Condition to assert Compare Result Flag 0: Outside of the range Lower Limit (LLILM) > Code Data, or Code Data > Upper Limit (ULIM) 1: Inside of the range Lower Limit (LLILM) < Code Data < Upper Limit (ULIM)	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The CMSR register controls the compare operation setting.

42.3.7 LLSR : Lower Limit Setting Register

Base address: TSU = 0x8008_6000

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LLIM[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	LLIM[11:0]	Lower Limit Setting Value	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The LLSR register controls the lower limit value to be compared with acquired code data.

42.3.8 ULSR : Upper Limit Setting Register

Base address: TSU = 0x8008_6000

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ULIM[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	ULIM[11:0]	Upper Limit Setting Value	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The ULSR register controls the upper limit value to be compared with acquired code data.

42.3.9 SISR : Sensor Interrupt Status Register

Base address: TSU = 0x8008_6000

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CMPF	ADF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADF	AD Conversion Complete Flag Cleared when ADCLR bit of Sensor Interrupt Clear Register is set to 1. 0: Not completed 1: Completed	R
1	CMPF	Compare Result Flag Cleared when CMPCLR bit of Sensor Interrupt Clear Register is set to 1. 0: Not matched 1: Matched	R
31:2	—	These bits are read as 0.	R

The SISR register is used to monitor the interrupt status.

42.3.10 SIER : Sensor Interrupt Enable Register

Base address: TSU = 0x8008_6000

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CMPIE	ADIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADIE	AD Conversion Complete Interrupt Enable 0: Disable 1: Enable	R/W
1	CMPIE	Compare Result Interrupt Enable 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The SIER register enables or disables output to the interrupt request output signal (TSUADI, TSUADCMPI).

42.3.11 SICR : Sensor Interrupt Clear Register

Base address: TSU = 0x8008_6000

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CMPCLR	ADCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

Bit	Symbol	Function	R/W
0	ADCLR	AD Conversion Complete Interrupt Clear 0: No effect 1: Interrupt is cleared.	W
1	CMPCLR	Compare Result Interrupt Clear 0: No effect 1: Interrupt is cleared.	W
31:2	—	The write value should be 0.	W

The SICR register clears the interrupt status.

42.4 Function Description

42.4.1 Acquiring Temperature Code

Conversion starts on reception of the conversion start trigger (when the STRGR.ADST bit is set to 1). When the SCRR.OUT12BIT_TS bit is read after the completion of conversion, a 12-bit temperature code can be acquired.

If the conversion start trigger comes within 120T (internal clock period (320 ns)) after release of the temperature sensor and ADC from the power-down state (setting the sensor suspend register so that the SSUSR.ADC_PD bit is 0 and the SSUSR.EN_TS bit is 1), conversion only starts after 120T has elapsed following release.

42.4.2 Temperature Code Averaging Function

The temperature codes acquired within the TSU are averaged the number of times set by the SOSR1.ADCT[1:0] bits.

42.4.3 Temperature Code Reading Function

The temperature code can be read from the SCRR register through the internal bus. When SOSR1.OUTSEL = 0, the latest converted value is read. When SOSR1.OUTSEL = 1, the average value is read.

42.4.4 Temperature Code Comparison Functions

This function compares the average temperature code and the values set by the LLSR.LLIM[11:0] bits and the ULSR.ULIM[11:0] bits when CMSR.CMPEN = 1.

The temperature code for use in the comparison function is the average temperature code, so the SOSR1.OUTSEL bit should be set to 1 (average data output) when reading the temperature code for use in comparison. When the averaging count setting is 1, however, the latest value is the same as the average value, so the temperature code for use in comparison will be read regardless of the setting of the SOSR1.OUTSEL bit.

The SISR.CMPF bit is set to 1 when the following conditions (1) or (2) is satisfied.

(1) CMSR.CMPCOND = 0 and either one of the below formula is satisfied.

- Acquired data > ULIM setting value of ULSR register
- LLIM setting value of LLSR register > Acquired data

(2) CMSR.CMPCOND = 1 and both two below formulas are satisfied.

- ULIM setting value of ULSR register > Acquired data
- Acquired data > LLIM setting value of LLSR register

42.4.5 Interrupt Output Function

An interrupt is output when conversion is completed and the result of temperature code comparison satisfies the condition. The TSU has two interrupt signals, a conversion complete interrupt (TSUADI) and a comparison result interrupt (TSUADCMPI).

An interrupt enable/disable switching bit, status confirmation bit, and status clearing bit are in place for each of these signals.

42.4.5.1 Conversion Complete Interrupt

A conversion complete interrupt is generated on completion of conversion.

- When the interrupt is enabled (SIER.ADIE = 1)
  - The conversion complete interrupt (TSUADI) is generated on completion of conversion by the TSU.
  - The SISR.ADF bit is set to 1 on completion of conversion by TSU.
  - Writing 1 to the SICR.ADCLR bit sets the SISR.ADF bit to 0 to clear the status flag. The SISR.ADF bit retains the value 1 unless 1 is written to the SICR.ADCLR bit.
- When the interrupt is disabled (SIER.ADIE = 0)
  - The conversion complete interrupt (TSUADI) is not generated on completion of conversion by TSU.
  - The SISR.ADF bit is set to 1 on completion of conversion by TSU.
  - Writing 1 to the SICR.ADCLR bit sets the SISR.ADF bit to 0 to clear the status flag. The SISR.ADF bit retains the value 1 unless 1 is written to the SICR.ADCLR bit.

42.4.5.2 Comparison Result Interrupt

A comparison result interrupt is generated according to the selected condition for comparison when the result of comparison satisfies the condition.

- When the interrupt is enabled (SIER.CMPIE = 1)
  - The comparison result interrupt (TSUADCMPI) is generated when comparison result satisfies the condition in section 42.4.4. Temperature Code Comparison Functions.
  - The SISR.CMPF bit is set to 1 when comparison result satisfies the condition in section 42.4.4. Temperature Code Comparison Functions.
  - Writing 1 to the SICR.CMPCLR bit sets the SISR.CMPF bit to 0 to clear the status flag. The SISR.CMPF bit retains the value 1 unless 1 is written to the SICR.CMPCLR bit.
- When the interrupt is disabled (SIER.CMPIE = 0)
  - The comparison result interrupt (TSUADCMPI) is not generated when comparison result satisfies the condition in section 42.4.4. Temperature Code Comparison Functions.
  - The SISR.CMPF bit is set to 1 when comparison result satisfies the condition in section 42.4.4. Temperature Code Comparison Functions.
  - Writing 1 to the SICR.CMPCLR bit sets the SISR.CMPF bit to 0 to clear the status flag. The SISR.CMPF bit retains the value 1 unless 1 is written to the SCIR.CMPCLR bit.

42.4.6 Single Scan Mode and Continuous Scan Mode

Single scan mode and continuous scan mode are supported. In single scan mode, conversion is only performed once with averaging the number of sampling set by the SOSR1.ADCT[1:0] bits. In continuous scan mode, conversion continues repeatedly until the STRGR.ADEND bit is set to 1. The SOSR1.ADCS bit selects one of the two scan modes. SOSR1.ADCS = 0 is single scan mode and SOSR1.ADCS = 1 is continuous scan mode.

42.5 Operation

The TSU is mainly used in the following two ways.

- Using an interrupt for each completion of the conversion
- Using the comparison result interrupt

42.5.1 Using Interrupt for Each Completion of Conversion

Conversion is started by software and the interrupt is generated on completion of the conversion. After the generation of the interrupt, the temperature code is read.

The comparison function is enabled by default. The upper and lower limit settings for use in comparison are both all 0s by default and the conditions for comparison are not satisfied even while the comparison function is enabled, but the conditions may inadvertently be satisfied if operation proceeds after the upper and lower limits have been set. If the comparison function is not to be used, set the CMSR.CMPEN bit to 0 to disable the comparison function.

Table 42.4 is an example of the procedure for operations when an interrupt is to be used to indicate each completion of the conversion.

Table 42.4 Example of the operation procedure when using interrupt for each completion of the conversion

No	Item	Example of the operation procedure
1	Release from suspended state	Set the SSUSR register to 0x00000005. ● Temperature Sensor is normal operation. ● ADC is normal operation. ● SOC_TS is enabled.
2	Setting of the operating mode	Set 0x0000020B to the SOSR1 register. ● The number of samples for averaging is 8. ● Single scan mode ● Average data output Set SIER register to 0x00000001. ● Enable the conversion complete interrupt. ● Mask the comparison result interrupt.
3	Check the conversion status	Read the SSR.CONV bit. ● Confirm whether conversion is not in progress.
4	Start conversion	Set the STRGR register to 0x00000001. ● Start conversion.
5	Check the interrupt	Read the SISR.ADF bit. ● Confirm AD conversion complete flag is set to 1. Set SICR register to 0x00000001. ● Clear AD conversion complete flag.
6	Check the conversion status	Read the SSR.CONV bit. ● Confirm whether conversion is not in progress.
7	Obtain the temperature code	Read the SCRR register. ● Obtain the temperature code
8	Temperature compensation	Calculate compensated temperature. For details, see section 42.5.3. Temperature Compensation Calculation.

42.5.2 Using Comparison Result Interrupt

Conversion is started by software after the conditions for comparison (within or outside the range, and the upper and lower limits) have been set and the interrupt is generated when the result of comparison with the settings satisfies the conditions. After the generation of the interrupt, the temperature code is read.

Table 42.5 is an example of the procedure for operations when the comparison result interrupt is to be used.

Table 42.5 Example of the operation procedure when using comparison result interrupt

No	Item	Example of the operation procedure
1	Release from suspended state	Set the SSUSR register to 0x00000005. ● Temperature Sensor is normal operation. ● ADC is normal operation. ● SOC_TS is enabled.
2	Setting of the operating mode	Set the SOSR1 register to 0x0000021B. ● The number of samples for averaging is 8. ● Continuous scan mode ● Average data output Set the SIER register to 0x00000002. ● Mask the conversion complete interrupt. ● Enable the comparison result interrupt. Set a lower limit to the LLSR register and an upper limit to the ULSR register. ● Set an upper and lower limit for the temperature. See section 42.5.4. Comparison Setting Value Calculation for the calculation of the setting value.
3	Check the conversion status	Read the SSR.CONV bit. ● Confirm whether conversion is not in progress.
4	Start conversion	Set 0x0000_0001 to STRGR register. ● Start conversion.
5	Check the interrupt	Read the SISR.CMPF bit. ● Confirm comparison result flag is set to 1. Set SICR register to 0x00000002. ● Clear comparison result flag.
6	Check the conversion status	Read the SSR.CONV bit. ● Confirm whether conversion is not in progress.
7	Obtain the temperature code	Read the SCRR register. ● Obtain the temperature code
8	Temperature compensation	Calculate compensated temperature. For details, see section 42.5.3. Temperature Compensation Calculation.

42.5.3 Temperature Compensation Calculation

After acquiring the temperature code data, the compensated temperature T (°C) should be calculated from the value (a) of the SCRR.OUT12BIT_TS bits, the value (b) at −40 (°C) and the value (c) at 125 (°C) stored in OTP.

[Calculation formula]

$$\text{Compensated temperature T (°C)} = \left(\frac{e - d}{c - b}\right) \times (a - b) + d$$

$d = -40, e = 125$

42.5.4 Comparison Setting Value Calculation

The comparison setting value (a) set to the LLSR and ULSR registers should be calculated from each temperature T (°C) you want to set, the value (b) at −40 (°C) and the value (c) at 125 (°C) stored in OTP.

[Calculation formula]

$$\text{Setting value a} = \frac{(T - d) \times (c - b)}{(e - d)} + b$$

42.6 Usage Notes

42.6.1 Release from Power-down

Set the SSUSR.SOC_TS_EN bit to 1 simultaneously or before setting the SSUSR.EN_TS bit to 1 and the SSUSR.ADC_PD bit to 0 to release the TSU from the power-down state.

42.6.2 Conversion Start Trigger

Before starting conversion by the STRGR.ADST bit, check that the SSR.CONV bit is 0.

43. Data Operation Circuit (DOC)

43.1 Overview

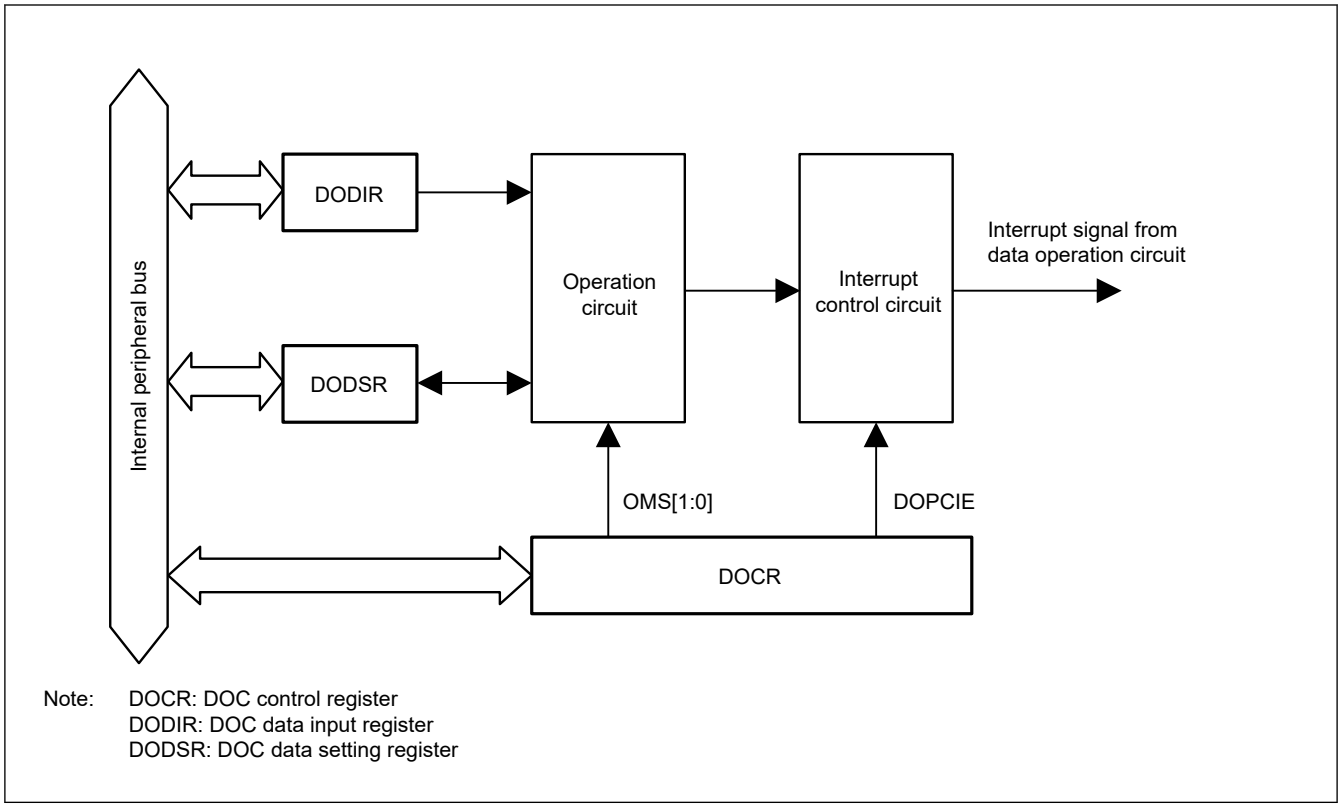
The data operation circuit (DOC) is used to compare, add, and subtract 16-bit data.

Table 43.1 lists the data operation circuit specifications and Figure 43.1 shows a block diagram of the data operation circuit.

- 16-bit data is compared and an interrupt can be generated when a selected condition applies.
- 16-bit data can be added.
- 16-bit data can be subtracted.

Table 43.1 DOC specifications

Parameter	Description
Data operation function	16-bit data comparison, addition, and subtraction
Lower power consumption function	Module-stop state can be set.
Interrupts	An interrupt occurs at the following timings: • The compared values either match or mismatch • The result of data addition is greater than 0xFFFF • The result of data subtraction is less than 0x0000



Note: DOCR: DOC control register
DODIR: DOC data input register
DODSR: DOC data setting register

Figure 43.1 Block diagram of data operation circuit (DOC)

43.2 Register Map

Table 43.2 DOC register map

Address	Register symbol	Register name	Write protection
0x8008_4000	DOCR	DOC Control Register	—
0x8008_4002	DODIR	DOC Data Input Register	—
0x8008_4004	DODSR	DOC Data Setting Register	—

Table 43.3 DOC related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	—	MSTPCRD.MSTPCRD08	SLVACCCTL1.DOC_SL

43.3 Register Descriptions

43.3.1 DOCR : DOC Control Register

Base address: DOC = 0x8008_4000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	DOPCFCL	DOPCF	DOPCIE	—	DCSEL	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OMS[1:0]	Operating Mode Select Selects an operating mode of the data operation circuit. 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
2	DCSEL	Detection Condition Select Selects a condition for detecting data comparison result. This bit is valid only in the data comparison mode. 0: Data mismatch is detected. 1: Data match is detected.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	DOPCIE	Data Operation Circuit Interrupt Enable Specifies whether to enable or disable data operation circuit interrupt requests. 0: Disables interrupts from the data operation circuit. 1: Enables interrupts from the data operation circuit.	R/W
5	DOPCF	Data Operation Circuit Flag [Setting conditions] When any of the following is met: ● The condition selected by the DCSEL bit is met. ● A result of data addition is greater than 0xFFFF. ● A result of data subtraction is less than 0x0000. [Clearing condition] Writing 1 to the DOPCFCL bit	R
6	DOPCFCL	DOPCF Flag Clear 0: No effect 1: Clear the DOPCF flag	W
7	—	This bit is read as 0. The write value should be 0.	R/W

43.3.2 DODIR : DOC Data Input Register

Base address: DOC = 0x8008_4000

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The DODIR register is a 16-bit readable/writable register in which 16-bit data for use in operations are stored.	R/W

43.3.3 DODSR : DOC Data Setting Register

Base address: DOC = 0x8008_4000

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The DODSR register is a 16-bit readable/writable register. This register stores 16-bit data for use as a reference in data comparison mode. This register also stores the results of operations in data addition and data subtraction modes.	R/W

43.4 Operation

43.4.1 Data Comparison Mode

Figure 43.2 shows an example of the steps involved in data comparison mode operation by the data operation circuit. The following is an example of operation when DCSEL is set to 0 (data mismatch is detected as a result of data comparison).

1. Writing 00b to the DOCR.OMS[1:0] bits selects data comparison mode.
2. The 16-bit reference data is set in DODSR.
3. 16-bit data for comparison is written to DODIR.
4. Writing of 16-bit data continues until all data for comparison have been written to DODIR.
5. If a value written to DODIR does not match that in DODSR, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note: When the DODIR register has the value after reset (0x0000), the DOCR.DOPCF flag does not change to 1 if a value other than 0x0000 is set in the DODSR register. To perform data comparison, set the DODSR register and write a value to the DODIR register. If the DODSR register is set again after data comparison, the value remaining in the DODIR register will not be compared again with the value set in the DODSR register. That is, comparison proceeds in response to writing to the DODIR register but not in response to writing to the DODSR register.

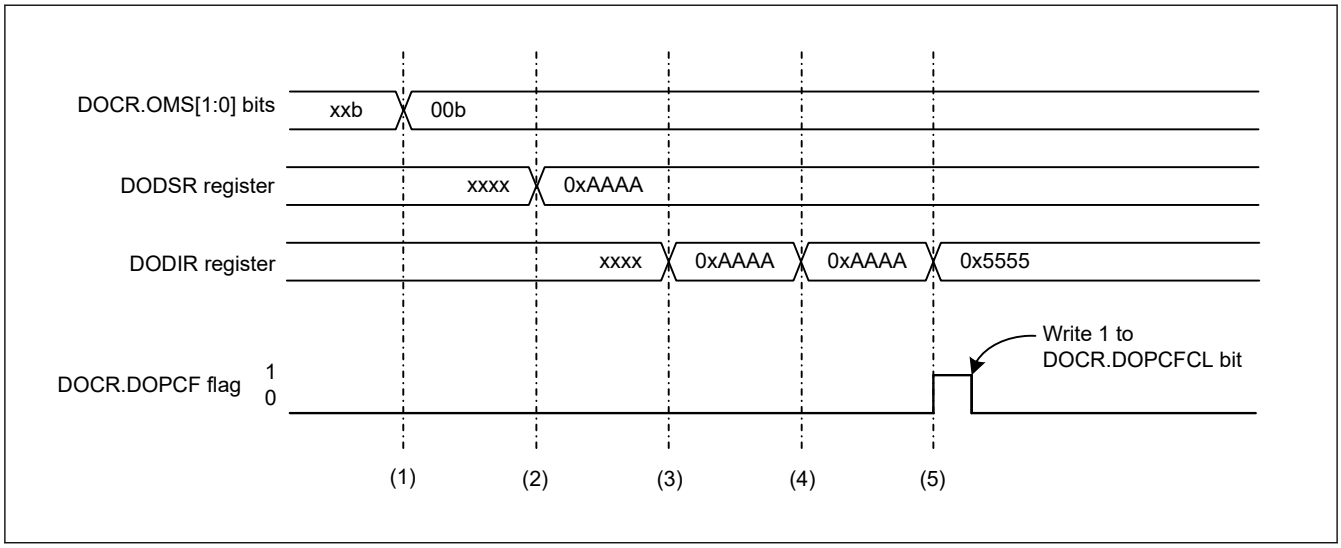


Figure 43.2 Example of operation in data comparison mode

43.4.2 Data Addition Mode

Figure 43.3 shows an example of the steps involved in data addition mode operation by the data operation circuit.

1. Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
2. 16-bit data is set in the DODSR register as the value after reset.
3. 16-bit data to be added is written to DODIR. The result of the operation is stored in DODSR.
4. Writing of 16-bit data continues until all data for addition have been written to DODIR.
5. If the result of an operation is greater than 0xFFFF, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note: When the DODIR register has the value after reset (0x0000), data addition is not performed if a value other than 0x0000 is set in the DODSR register. To perform data addition, set the DODSR register and write a value to be added into the DODIR register. If the DODSR register is set again after data addition, the value remaining in the DODIR register will not be added again to the value set in the DODSR register. That is, addition proceeds in response to writing to the DODIR register but not in response to writing to the DODSR register.

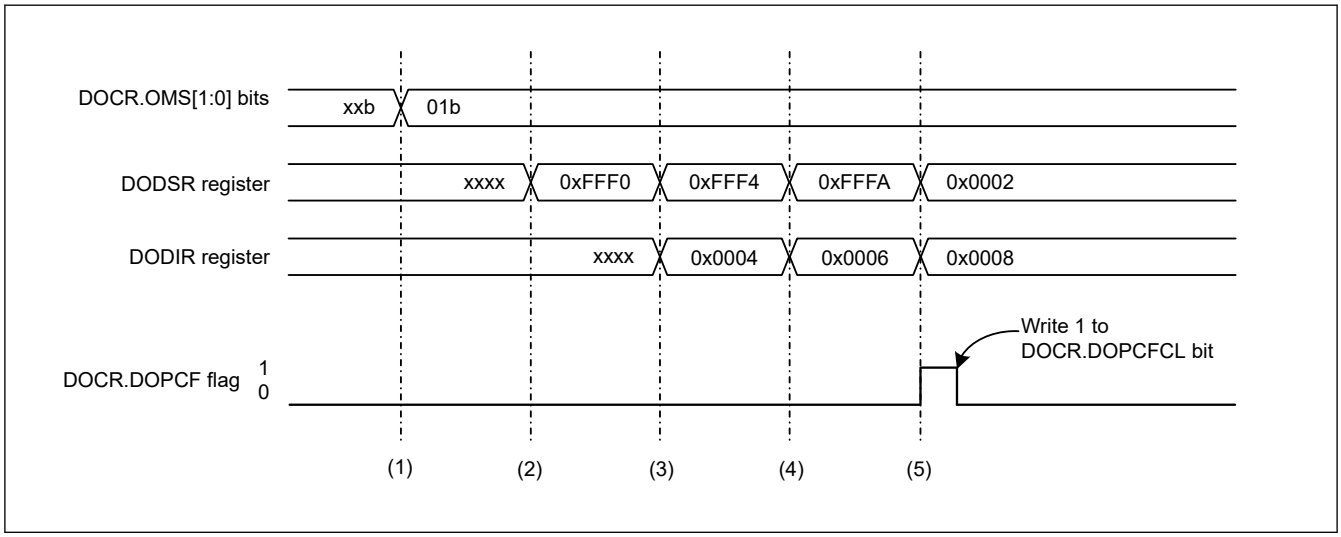


Figure 43.3 Example of operation in data addition mode

43.4.3 Data Subtraction Mode

Figure 43.4 shows an example of the steps involved in data subtraction mode operation by the data operation circuit.

1. Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
2. 16-bit data is set in the DODSR register as the value after reset.
3. 16-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR.
4. Writing of 16-bit data continues until all data for subtraction have been written to DODIR.
5. If the result of an operation is less than 0x0000, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note: When the DODIR register has the value after reset (0x0000), data subtraction is not performed if a value other than 0x0000 is set in the DODSR register. To perform data subtraction, set the DODSR register and write a value to be subtracted into the DODIR register. If the DODSR register is set again after data subtraction, the value remaining in the DODIR register will not be subtracted again from the value set in the DODSR register. That is, subtraction proceeds in response to writing to the DODIR register but not in response to writing to the DODSR register.

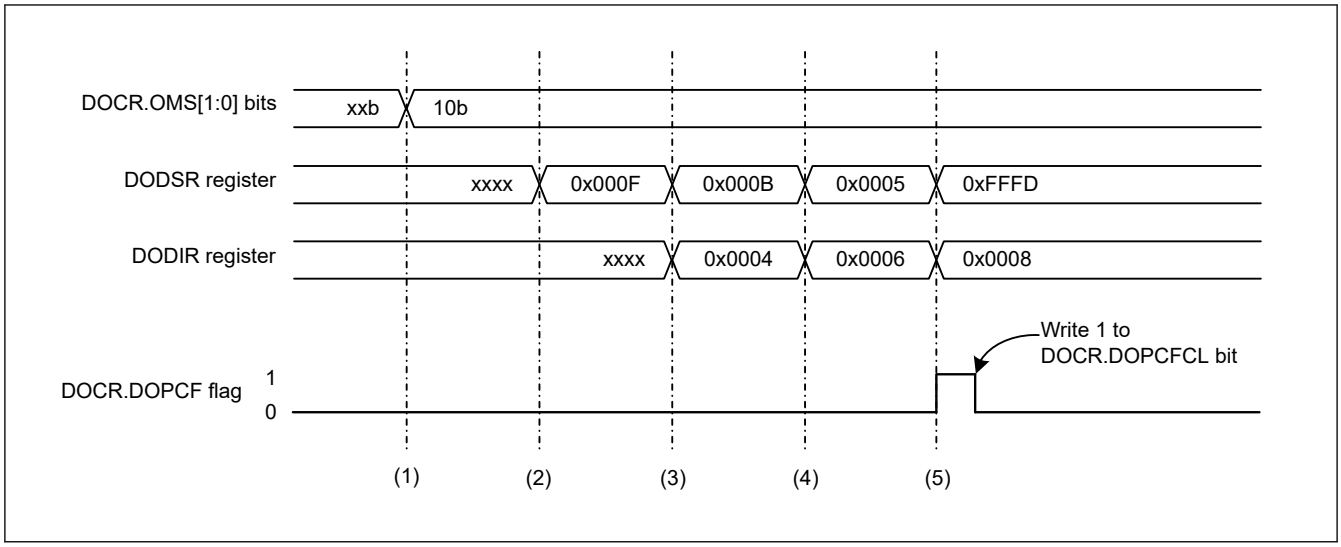


Figure 43.4 Example of operation in data subtraction mode

43.5 Interrupt Requests

The data operation circuit generates the data operation circuit interrupt as an interrupt request. When an interrupt source is generated, the data operation circuit flag (DOCR.DOPCF) corresponding to the interrupt is set to 1. Table 43.4 describes the interrupt request.

Table 43.4 Interrupt request from data operation circuit

Interrupt request	Data operation circuit flag	Interrupt generation timing
Data operation circuit interrupt	DOPCF	• The result of data addition is greater than 0xFFFF • The result of data subtraction is less than 0x0000 • The compared values either match or mismatch

43.6 Usage Note

43.6.1 Module Stop Function Setting

Operation of the data operation circuit can be disabled or enabled using module stop control register D (MSTPCRD). The value after reset indicates that the data operation circuit is in the stop state. Register access is enabled by canceling the module-stop state. For details, see section 9, Low-Power Consumption Function.

44. System SRAM (SYSRAM)

44.1 Overview

The LSI has an on-chip high-speed RAM (with ECC error correction). The RAM, whose capacity is 2.0 MB (512 KB × 4), is installed. Table 44.1 lists the System SRAM (SYSRAM) specifications and Figure 44.1 shows a block diagram of SYSRAM.

Table 44.1 System SRAM specifications

Parameter	Description
RAM capacity	2.0 MB (512 KB × 4)
Operating frequency	250 MHz (PCLKH)
RAM address	SYSRAM0: 0x1000_0000 to 0x1007_FFFF SYSRAM1: 0x1008_0000 to 0x100F_FFFF SYSRAM2: 0x1010_0000 to 0x1017_FFFF SYSRAM3: 0x1018_0000 to 0x101F_FFFF Each RAM module consists of 4 pages of 128 KB
Shortest Latency	• ECC enabled Writing 32/64/96/128-bit transfer: Latency 0 (write to RAM in 1 cycle) 8/16-bit transfer: Latency 2 (write to RAM in 3 cycles) Reading All data transfer: Latency 2 (read from RAM in 3 cycles) • ECC disabled Writing All data transfer: Latency 0 (write to RAM in 1 cycle) Reading All data transfer: Latency 1 (read from RAM in 2 cycles)
Error checking	7-bit ECC data to 32-bit RAM data 1-bit error correction and 2-bit error detection are supported.
Interrupts	3 interrupts are available for each SYSRAMn. They can be used as error events.
Bus error response	SLVERR is returned to the access to the System SRAM. • Write access occurs when write access is disabled by VLWEN bit. • Write or read access occurs when access is disabled by VCEN bit. • Write access occurs when write access is disabled by VRWEN bit to corresponding page. • 2-bit ECC error occurs when read access is done. This response is maskable.
Test function (by error injection)	• Register values can be used as test data to be written to RAM, those in ECEDB[31:0] to the RAM data section and ECERDB[6:0] to the ECC redundancy bit section, respectively. • The ECC redundancy bits included in the RAM data are retained and read as a register value. • The ECC redundancy bits (encoder circuit) generated from the input data and the syndrome code (decoder circuit) can be checked.
Initializing function	Supported

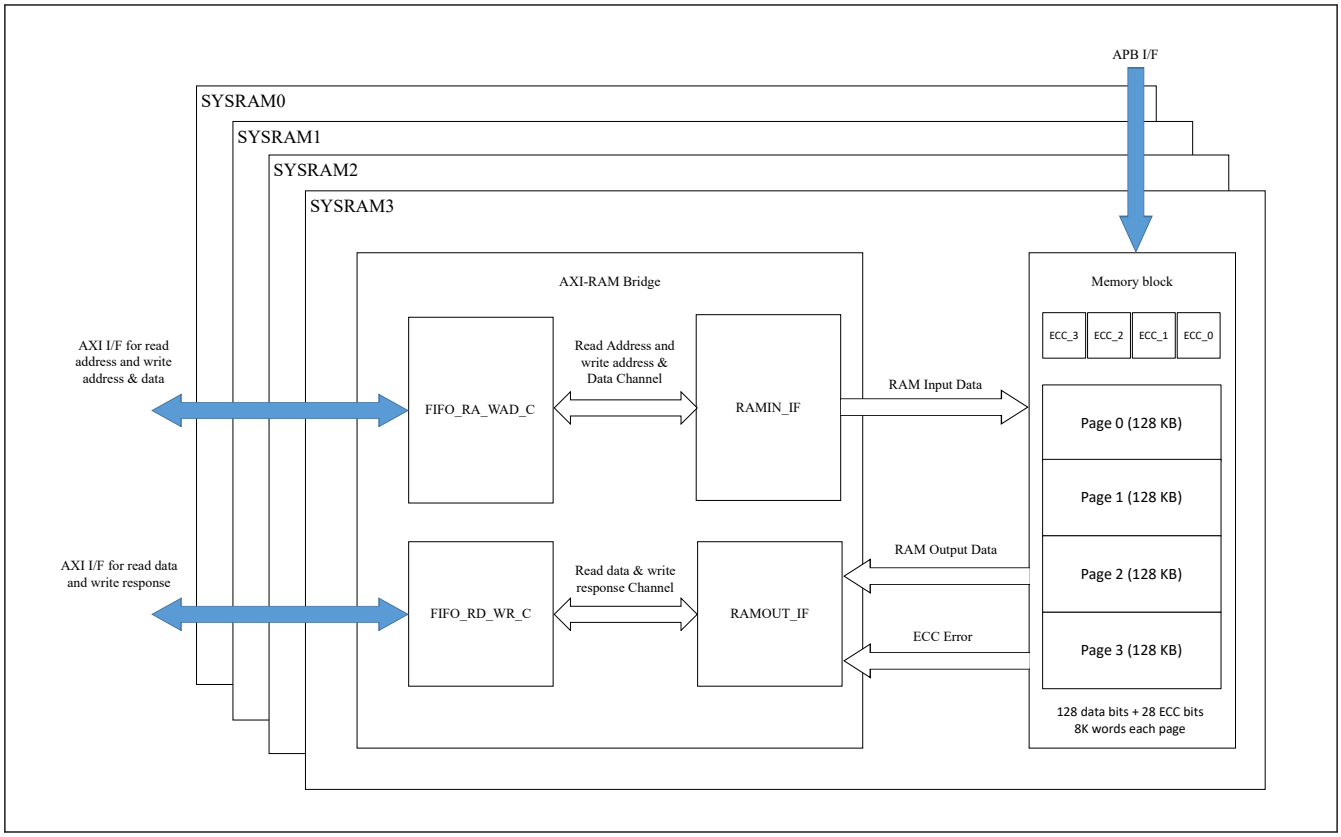


Figure 44.1 Block diagram of System SRAM

44.2 Register Map

Table 44.2 SYSRAM register map

Address	Register symbol	Register name	Write protection
0x8129_3000 + 0x010 × n	SYSRAM_CTRLn	System SRAM Control Register n	PRCRN.PRC3
0x8104_0000 + 0x1000 × i + 0x040 × n (i = 0 to 3, n = 0 to 3)	Wn_EC710CTL	ECC Control Register n	—
0x8104_0004 + 0x1000 × i + 0x040 × n (i = 0 to 3, n = 0 to 3)	Wn_EC710TMC	ECC Test Mode Control Register n	—
0x8104_0008 + 0x1000 × i + 0x040 × n (i = 0 to 3, n = 0 to 3)	Wn_EC710TRC	ECC Redundancy Bit Data Control Test Register n	—
0x8104_000C + 0x1000 × i + 0x040 × n (i = 0 to 3, n = 0 to 3)	Wn_EC710TED	ECC Encode/Decode Input/Output Switchover Test Register n	—
0x8104_0010 + 0x1000 × i + 0x040 × n + 0x004 × m (i = 0 to 3, n = 0 to 3, m = 0 to 7)	Wn_EC710EADm	ECC Error Address m Register n	—

Note: n for SYSRAM_CTRLn and i for the others stand for unit number.

Table 44.3 SYSRAM related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register*1
0	—	—	SLVACCCTL6.SYSRAM0_SL
1	—	—	SLVACCCTL6.SYSRAM1_SL
2	—	—	SLVACCCTL6.SYSRAM2_SL
3	—	—	SLVACCCTL6.SYSRAM3_SL

Note 1. Access to memory area is controlled by SLVACCCTL8.TZCSYSRAM_SL.

44.3 Register Descriptions

44.3.1 SYSRAM_CTRLn : System SRAM Control Register n (n = 0 to 3)

Base address: SYSRAM_CTL = 0x8129_3000

Offset address: 0x000 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	MKICCAXIERR	—	—	VLWEN	VCEN	VRWEN[3:0]			
Value after reset:	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VECEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	VECEN	Enables or disables error correction with ECC 0: Disabled 1: Enabled	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
19:16	VRWEN[3:0]	Enables write for each page of RAM The bit n corresponds to the page n (n = 0 to 3). 0: Disabled 1: Enabled	R/W
20	VCEN	Enables access to RAM 0: Disabled 1: Enabled	R/W
21	VLWEN	Enables write for RAM 0: Disabled 1: Enabled	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	MKICCAXIERR	Controls AXI-SLVERR issuance for ECC 2-bit errors 0: Issue AXI-SLEVER 1: Do not issue AXI-SLEVER	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The SYSRAM_CTRLn register is a 32-bit register that controls ECC of the SYSRAMn.

44.3.2 Wn_EC710CTL : ECC Control Register n (n = 0 to 3)

Base address: SYSRAMi = 0x8104_0000 + 0x1000 × i (i = 0 to 3)

Offset address: 0x000 + 0x040 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ECEDF7[1:0]		ECEDF6[1:0]		ECEDF5[1:0]		ECEDF4[1:0]		ECEDF3[1:0]		ECEDF2[1:0]		ECEDF1[1:0]		ECEDF0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EMCA[1:0]		—	—	ECOVFF	ECER2C	ECER1C	—	ECTHM	ECERVF	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	ECEMF	ECC Error Indicate Flag Indicates that an error is detected in the current read data. This bit is updated for each 32-bit RAM output data. When ECC error determination is disabled (ECERVF = 0) or the ECC function through mode is set (ECTHM = 1), this bit is always 0. 0: No error is detected in the current read data 1: An error is detected in the current read data	R
1	ECER1F	1-Bit ECC Error Detection/Correction Flag Indicates that a 1-bit error is detected and corrected in the 32-bit read data plus 7 ECC bits (total 39 bits) while the ECC error determination is enabled. 0: 1-bit error has not occurred 1: 1-bit error has occurred*1	R
2	ECER2F	2-Bit ECC Error Detection Flag Indicates that a 2-bit error is detected in the 32-bit read data plus 7 ECC bits (total 39 bits) while the ECC error determination is enabled. 0: 2-bit error has not occurred 1: 2-bit error has occurred*2	R
3	EC1EDIC	1-Bit ECC Error Detection Interrupt Control Enables or disables to output an interrupt signal (SRAMn_IE1) on detection of a 1-bit ECC error. 0: Disables to output an interrupt signal 1: Enables to output an interrupt signal	R/W
4	EC2EDIC	2-Bit ECC Error Detection Interrupt Control Enables or disables to output an interrupt signal (SRAMn_IE2) on detection of a 2-bit ECC error. 0: Disables to output an interrupt signal 1: Enables to output an interrupt signal	R/W
5	EC1ECP	1-Bit ECC Error Correction Enable Enables or disables error correction when a 1-bit ECC error is detected. 0: Error correction is performed on detection of a 1-bit ECC error 1: Error correction is not performed on detection of a 1-bit ECC error	R/W
6	ECERVF	ECC Error Determination Enable To change this bit, it is necessary to write 01b to bits 15 and 14 at the same time to permit writing. 0: The error determination is disabled 1: The error determination is enabled. The ECC function according to EC1EDIC, EC2EDIC, and EC1ECP bits are enabled.	R/W
7	ECTHM	ECC Function Through Mode Enable Enables or disables ECC function Through mode. If Through mode is enabled, no ECC error determination and no error correction is performed. To change this bit, it is necessary to write 01b to bits 15 and 14 at the same time to permit writing. 0: Disables Through mode 1: Enables Through mode. No effect on encoder. Decoder does not determine the error and does not perform error correction.	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	ECER1C	1-Bit ECC Error Detection Clear Clears the 1-bit ECC error detection flag and the error address related flags for the 32-bit read data. Writing 1 to this bit clears the ECER1F, ECOVFF, and ECEDF7[1:0] to ECEDF0[1:0] bits. When a conflict occurs between 1-bit ECC error detection and writing 1 to this bit, writing to this bit is given priority and the 1-bit ECC error detection flag is not set.	R/W
10	ECER2C	2-Bit ECC Error Detection Clear Clears the 2-bit ECC error detection flag and the error address related flags for the 32-bit read data. Writing 1 to this bit clears the ECER2F, ECOVFF, and ECEDF7[1:0] to ECEDF0[1:0] bits. When a conflict occurs between 2-bit ECC error detection and writing 1 to this bit, writing to this bit is given priority and the 2-bit ECC error detection flag is not set.	R/W

Bit	Symbol	Function	R/W
11	ECOVFF	ECC Error Address Capture Overflow Flag Indicates that the error address fails to be stored in the ECC error address register when an ECC error occurs. When an ECC error address capture overflow occurs, SRAMn_OVF interrupt is output. If an ECC error address capture overflow occurs again while this flag is 1, SRAMn_OVF interrupt is output again. 0: ECC error address capture overflow has not occurred 1: ECC error address capture overflow has occurred	R
13:12	—	These bits are read as 0. The write value should be 0.	R/W
15:14	EMCA[1:0]	Access Control to ECC Mode Selection Setting these bits to 01b enables writing to bits 7 and 6 of this register. Otherwise, writing to bits 7 and 6 is ignored. These bits are always read as 00b. 0 1: Writing is enabled Others: Writing is disabled	R/W
17:16	ECEDF0[1:0]	ECC Error Address Capture Flag m (m = 0) Indicates the error address register and the error type. When an ECC error occurs and its error type can be identified, the following value is read from these bits. 0 0: ECC error address is not captured in Wn_EC710EADm 0 1: 1-bit ECC error address is captured in Wn_EC710EADm 1 0: 2-bit ECC error address is captured in Wn_EC710EADm	R
19:18	ECEDF1[1:0]	ECC Error Address Capture Flag m (m = 1) This function is the same as ECEDF0.	R
21:20	ECEDF2[1:0]	ECC Error Address Capture Flag m (m = 2) This function is the same as ECEDF0.	R
23:22	ECEDF3[1:0]	ECC Error Address Capture Flag m (m = 3) This function is the same as ECEDF0.	R
25:24	ECEDF4[1:0]	ECC Error Address Capture Flag m (m = 4) This function is the same as ECEDF0.	R
27:26	ECEDF5[1:0]	ECC Error Address Capture Flag m (m = 5) This function is the same as ECEDF0.	R
29:28	ECEDF6[1:0]	ECC Error Address Capture Flag m (m = 6) This function is the same as ECEDF0.	R
31:30	ECEDF7[1:0]	ECC Error Address Capture Flag m (m = 7) This function is the same as ECEDF0.	R

Note 1. When a 1-bit ECC error is detected while EC1EDIC bit is 1 and ECER1F bit is 0, SRAMn_IE1 interrupt is output. If a 1-bit ECC error is detected while the ECER1F bit is already 1, the SRAMn_IE1 interrupt is not output.

Note 2. When a 2-bit ECC error is detected while EC2EDIC bit is 1 and ECER2F bit is 0, SRAMn_IE2 interrupt is output. If a 2-bit ECC error is detected while the ECER2F bit is already 1, the SRAMn_IE2 interrupt is not output.

The Wn_EC710CTL register indicates operation settings and error status at the time of an ECC error occurrence.

Because RAM block consists of 128-bit data width but ECC is controlled on a 32-bit basis, the highest-order 32 bits are controlled by the W3_EC710CTL register and the lowest-order 32 bits by the W0_EC710CTL register. Do not write to the control bits (bits 7 to 3) of this register while AXI access is in progress.

44.3.3 Wn_EC710TMC : ECC Test Mode Control Register n (n = 0 to 3)

Base address: SYSRAMi = 0x8104_0000 + 0x1000 × i (i = 0 to 3)

Offset address: 0x004 + 0x040 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ETMA[1:0]		—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECREIS	ECC Redundancy Bit Input Data Select Selects whether the ECERDB[6:0] bits value or the ECC data in RAM is to be input to the decoder. This bit is writable only when the ECTMCE bit is 1, and synchronously cleared when the ECTMCE bit becomes 0. 0: ECC data read from RAM is input to the decoder 1: The value of the ECERDB[6:0] bits is input to the decoder	R/W
1	ECDCS	ECC Decode Input Select Selects whether the Wn_EC710TED register value or the RAM output data is to be input to the decoder. This bit is writable only when the ECTMCE bit is 1, and synchronously cleared when the ECTMCE bit becomes 0. 0: RAM output data is input to the decoder 1: The value of the Wn_EC710TED register is input to the decoder	R/W
2	ECENS	ECC Encode Input Select Selects whether the input data to RAM or the Wn_EC710TED register value is to be input to the encoder. This bit is writable only when the ECTMCE bit is 1, and synchronously cleared when the ECTMCE bit becomes 0. 0: Data input to RAM is selected 1: The value of the Wn_EC710TED register is selected	R/W
3	ECREOS	ECC Redundancy Bit Output Data Select Selects whether the ECC encoder output data or the value of the ECERDB[6:0] bits in the Wn_EC710TRC register is to be output as the ECC output. This bit is writable only when the ECTMCE bit is 1, and synchronously cleared when the ECTMCE bit becomes 0. 0: The encoded result of the encoder is output 1: The value of the ECERDB[6:0] bits in the Wn_EC710TRC register is output	R/W
4	ECTRRS	RAM Read Test Mode Select If 1 is set, read value of Wn_EC710TED register is 32-bit output data from RAM and read value of the Wn_EC710TRC.ECERDB[6:0] bits is 7 redundancy bits input to RAM. This bit is writable only when the ECTMCE bit is 1, and synchronously cleared when the ECTMCE bit becomes 0. 0: Read value is the written value for the Wn_EC710TED register and the Wn_EC710TRC.ECERDB[6:0] bits 1: Read value of Wn_EC710TED register is 32-bit output data from RAM and read value of the Wn_EC710TRC.ECERDB[6:0] bits is 7 redundancy bits input to RAM	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	ECTMCE	Test Mode Enable Enables or disables access to the test register and the test control register. When writing to this bit, it is necessary to set bits 15 and 14 of this register to 10b. If bits 15 and 14 are not 10b, writing to this bit is ignored. 0: Disables access to the test mode register and its bits 1: Enables access to the test mode register and its bits	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:14	ETMA[1:0]	ECTMCE Write Enable When these bits are 10b, writing to the ECTMCE bit is enabled. Any settings other than 10b disables writing to the ECTMCE bit. These bits are always read as 00b. 1 0: Writing is enabled Others: Writing is disabled	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The Wn_EC710TMC register is used to switch over to and control the ECC function test mode. Do not write to this register while AXI access is in progress.

44.3.4 Wn_EC710TRC : ECC Redundancy Bit Data Control Test Register n (n = 0 to 3)

Base address: SYSRAMi = 0x8104_0000 + 0x1000 × i (i = 0 to 3)

Offset address: 0x008 + 0x040 × n

Bit position:	31	30	24	22	16	14	8	6	0
Bit field:	—	ECSYND[6:0]		—	ECHORD[6:0]	—	ECECRD[6:0]	—	ECERDB[6:0]
Value after reset:	0	0 0 0 0 0 0 0	0	0	0 0 0 0 0 0 0	0	0 0 0 0 0 0 0	0	0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
6:0	ECERDB[6:0]	ECC Redundancy Bit Input/Output Substitute Buffer Register Handling of this register differs according to the setting of the associated register bits as follows: • Wn_EC710TMC.ECTMCE = 1: This register holds the substitute data for use as the value for the redundancy bit. • Wn_EC710TMC.ECREOS = 1: The value in this register substitutes for the redundancy bit value from the encoder and is written to the RAM. • Wn_EC710TMC.ECREIS = 1: The value in this register substitutes for the redundancy bit value from the RAM and is input to the decoder. • Wn_EC710TMC.ECTRRS = 1: The value read from this register is the input data to be supplied to the RAM. The value written in this register is not read.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
14:8	ECECRD[6:0]	ECC Encode Test Register When Wn_EC710TMC.ECTMCE = 1, the redundancy bit value generated in the encoder circuit is read from this register. Note that the read value is that from the encoder as is, for example, the encoded result of the data which was input to the encoder circuit. On the other hand, the value to be written to the RAM is the data output from the encoder circuit with its lower-order 2 bits inverted.	R
15	—	This bit is read as 0. The write value should be 0.	R/W
22:16	ECHORD[6:0]	ECC 7-Redundancy-Bit Data Retain Test Register When Wn_EC710TMC.ECTMCE = 1, 7 redundancy bits read from the RAM are stored in this register at the next clock cycle for the RAM read access. In addition, ECC 7-Redundancy-Bit Data is stored in the next clock cycle when Wn_EC710TED register is read if Wn_EC710TMC.ECTRRS = 1.	R
23	—	This bit is read as 0. The write value should be 0.	R/W
30:24	ECSYND[6:0]	ECC Decode Syndrome Register When Wn_EC710TMC.ECTMCE = 1, the syndrome code generated in the decoder circuit is read from this register. When Wn_EC710TMC.ECTMCE = 0, all 0s are read. The value of this register varies according to changes of decoder input value.	R
31	—	This bit is read as 0. The write value should be 0.	R/W

The Wn_EC710TRC register controls ECC redundancy bits. The value read from this register is valid only when Wn_EC710TMC.ECTMCE = 1, and always 0s when Wn_EC710TMC.ECTMCE = 0.

44.3.5 Wn_EC710TED : ECC Encode/Decode Input/Output Switchover Test Register n (n = 0 to 3)

Base address: SYSRAMi = 0x8104_0000 + 0x1000 × i (i = 0 to 3)

Offset address: 0x00C + 0x040 × n

Bit position:	31																												0
Bit field:	ECEDB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ECEDB[31:0]	32-Bit Data Test Register for ECC Encode/Decode In the test mode, the data of this register is used as the input to the encoder/decoder circuit. When Wn_EC710TMC.ECTRRS = 1, the value read from this register is the value of the data output from RAM. This register is readable and writable when Wn_EC710TMC.ECTMCE = 1. When Wn_EC710TMC.ECTMCE = 0, synchronous clearing occurs.	R/W

The Wn_EC710TED register handles ECC redundancy bits.

44.3.6 Wn_EC710EADm : ECC Error Address m Register n (n = 0 to 3, m = 0 to 7)

Base address: SYSRAMi = 0x8104_0000 + 0x1000 × i (i = 0 to 3)

Offset address: 0x010 + 0x040 × n + 0x004 × m

Bit position:	31															14													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECEAD[14:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	ECEAD[14:0]	Bit Error Address Retain the address where a 1-bit or 2-bit ECC error was detected.	R
31:15	—	These bits are read as 0.	R

The Wn_EC710EADm registers are read-only registers that retain the values of addresses where ECC errors have occurred.

44.4 Operation

44.4.1 Initialization

RAM initialization function initializes the data and redundancy bit area during reset sequence automatically.

44.4.2 Access Cycle to System SRAM with ECC Function Enabled

Read

Read access cycle when the ECC function is enabled is one cycle longer than the case when the ECC function is disabled regardless of access size.

Write

8/16-bit write access cycle when the ECC function is enabled is two cycles longer than the case when the ECC function is disabled. 32/64/96/128-bit write access cycle when the ECC function is enabled is the same as the case when the ECC function is disabled.

44.4.3 ECC Error Address Calculation Method

An ECC error address is stored as a relative address in the ECC Error Address Register (Wn_EC710EADm). To convert a relative address to an absolute address, follow the calculation formula shown in Table 44.4.

Table 44.4 ECC error address calculation formula

Register	ECC error address calculation formula
W0_EC710EAD0 to W0_EC710EAD7	SYSRAMi start address + (register value × 0x10) + 0x0
W1_EC710EAD0 to W1_EC710EAD7	SYSRAMi start address + (register value × 0x10) + 0x4
W2_EC710EAD0 to W2_EC710EAD7	SYSRAMi start address + (register value × 0x10) + 0x8
W3_EC710EAD0 to W3_EC710EAD7	SYSRAMi start address + (register value × 0x10) + 0xC

Note: i = 0 to 3

44.4.4 Notes on ECC Error Address Register

When 1-bit or 2-bit errors occur, up to 8 addresses where such errors have occurred are saved in the Wn_EC710EAD0 to Wn_EC710EAD7 registers in this order.

If the retained error address is detected again due to the same error cause, it is not retained this time. When the same address is detected due to a different error cause, it is saved if there is an empty register.

If a 2-bit error is detected when all the error address registers are filled with addresses of 1-bit errors, the Wn_EC710EAD7 register is overwritten with the error address where the 2-bit error occurred. At this time, the status register indicating the error flag information is not overwritten.

When storage of 8 or more error addresses is attempted, an overflow interrupt occurs. Clear the flag when the overflow interrupt occurs or every time a 1-bit error occurs.

44.4.5 Notes on ECC Error Detection

When a 1-bit or 2-bit error is detected, information of up to 8 errors can be stored in the ECC error information storage registers Wn_EC710CTL.ECEDFm[1:0] (m = 0 to 7), Wn_EC710CTL.ECOVFF, Wn_EC710CTL.ECER2F, Wn_EC710CTL.ECER1F, and Wn_EC710EADm (m = 0 to 7) in the ascending order of register numbers.

Note that operation differs depending on the storage state of ECC error information.

Note:
1. The ECC error information storage state is determined by every 32 bits. When an ECC error occurs in the least significant 32 bits, ECC error storage state is determined by the W0_EC710CTL register bits and the W0_EC710EADm register. When an ECC error occurs in the most significant 32 bits, ECC error storage state is determined by the W3_EC710CTL register bits and the W3_EC710EADm register.
2. When a 2-bit error is detected, immediately conduct the error processing and operate the ECER2C and ECER1C registers to clear error flags and the registers related to error address.
3. When storage of 8 or more error addresses is attempted, an overflow interrupt occurs. When an overflow interrupt occurs, immediately conduct the error processing and operate the ECER2C and ECER1C registers to clear error flags and the registers related to error address.

(1) When there are empty ECC error information storage registers

The error type and error address are stored in an empty register with the smallest number. At this time, the ECC error occurred causes the 1-bit ECC error detection/correction flag (ECER1F) or the 2-bit ECC error detection flag (ECER2F) to change from 0 to 1, and an interrupt signal to be output if the corresponding Interrupt Control register (EC1EDIC or EC2EDIC) enables interrupt output.

Error type:	Error type is stored in the ECC Error Address Capture flag (ECEDFm[1:0])
Error address:	ECC error address is stored in the ECC Error Address register (Wn_EC710EADm)
Error detection flag:	When 1-bit ECC error is detected: 1-bit ECC Error Detection/Correction flag (ECER1F) is set to 1. When 2-bit ECC error is detected: 2-bit ECC Error Detection flag (ECER2F) is set to 1.

Note: n = 0 to 3, m = 0 to 7.

(2) When there are no empty ECC error information storage registers (when 1-bit ECC error is detected)

When a 1-bit ECC error is detected, only the ECC Error Address Capture Overflow flag (ECOVFF) is set to 1 in the ECC Error Information Storage register.

At this time, the 1-bit ECC error occurred causes an ECC error address capture overflow interrupt to be generated.

Error type:	Error type is not stored in the ECC Error Address Capture flag (ECEDFm[1:0])
Error address:	ECC error address is not stored in the ECC Error Address register (Wn_EC710EADm)
Error detection flag:	ECC Error Address Capture Overflow flag (ECOVFF) is set to 1

Note: n = 0 to 3, m = 0 to 7.

(3) When there are no empty ECC error information storage registers (when 2-bit ECC error is detected)

When a 2-bit ECC error is detected, operation depends on the state of the 2-bit Error Detection flag (ECER2F) at the detection time.

When ECER2F = 0 on detection of a 2-bit ECC error

If the 2-bit ECC Error Detection flag (ECER2F) is 0 when the error is detected, all ECC error information storage registers are full of 1-bit ECC errors. In this case, only the error address is overwritten in the ECC Error Address register (Wn_EC710EADm).

At this time, the 2-bit ECC error occurred causes the following to occur:

- ECC error capture overflow interrupt
- 2-bit ECC error detection interrupt (EC2EDIC = 1: Interrupt output enabled)

Error type:	Error type is not stored in the ECC Error Address Capture flag (ECEDFm[1:0])
Error address:	2-bit ECC error address is overwritten in the ECC Error Address register (Wn_EC710EADm)
Error detection flag:	2-bit ECC Error Detection flag (ECER2F) is set to 1. ECC Error Address Capture Overflow flag (ECOVFF) is set to 1.

When ECER2F = 1 on detection of a 2-bit ECC error

If the 2-bit Error Detection flag (ECER2F) is 1 when the error is detected, only the ECC Error Address Capture Overflow flag (ECOVFF) is set to 1 in the ECC Error Information Storage register.

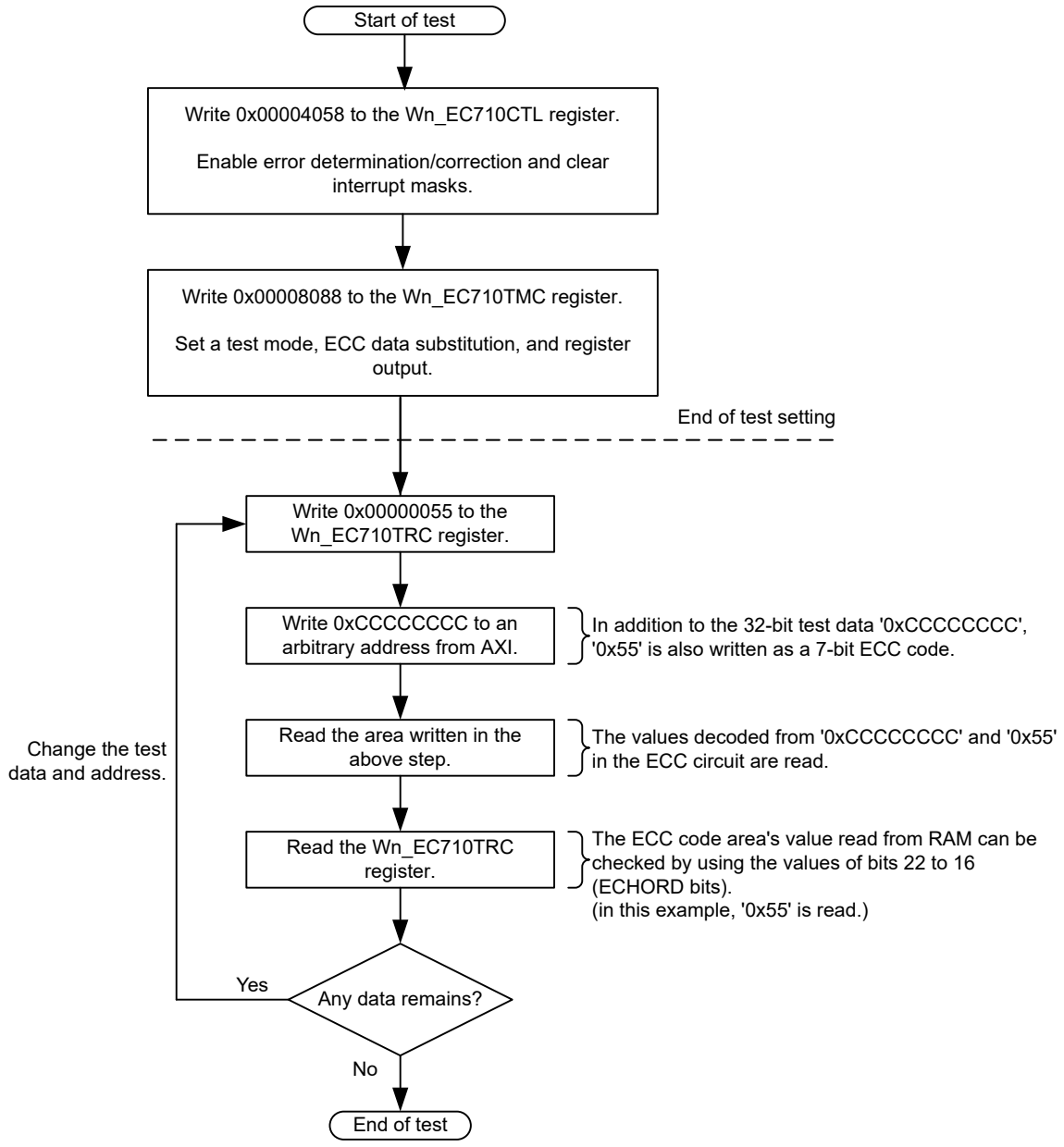
At this time, the 2-bit ECC error occurred causes an ECC error capture overflow interrupt to occur.

Error type:	Error type is not stored in the ECC Error Address Capture flag (ECEDFm[1:0])
Error address:	2-bit ECC error address is not stored in the ECC Error Address register (Wn_EC710EADm)
Error detection flag:	ECC Error Address Capture Overflow flag (ECOVFF) is set to 1

Note: n = 0 to 3, m = 0 to 7.

44.4.6 Test Function (Error Injection)

System SRAM has the error injection function, a facility to test the ECC function. The ECC errors intentionally generated by using the test registers are written to the target RAM area for checking the ECC function without involving the ECC circuit. Figure 44.2 shows an example of test flowchart.



Note: The value of the Wn_EC710CTL register is an example. Use an appropriate value according to the test conditions and purpose.

Figure 44.2 Flowchart example of ECC error injection test

44.4.7 Interrupt

When ECC error occurs, interrupt can be output. Table 44.5 lists the System SRAM interrupt sources. These interrupts are combined to peripheral error interrupts (PERI_ERR0 and PERI_ERR1) as error events with other error interrupts.

Table 44.5 System SRAM interrupt sources

Name	Interrupt sources
SRAMn_IE1	System SRAMn ECC 1 bit error
SRAMn_IE2	System SRAMn ECC 2 bits error
SRAMn_OVF	System SRAMn ECC error address capture overflow

Note: n = 0 to 3

44.5 Usage Notes

44.5.1 Release of Register Write Protection

SYSRAM_CTRLn registers are write protected. These registers are used to release write protection, use the following steps.

1. Write 0x0000A508 to PRCRS register (release write protection by PRC3 bit).
2. Write value to protection register.
3. Write 0x0000A500 to PRCRS register (set write protection by PRC3 bit).

44.5.2 Note on Setting the Register

All bus masters must not access System SRAM while a register of this module is being set.

44.5.3 Access in Bit Size Units

This module performs ECC determination process on a 32-bit basis. If access is made on an 8/16-bit basis, the module performs the following operations:

8/16-bit reading

1. After reading 32 bits of data, this module performs ECC determination.
2. To the read requesting bus master, this module returns 8/16-bit data.

8/16-bit writing

1. This module reads 32 bits of data from the write-target address and performs ECC determination.
2. On the read-out 32 bits, this module overwrites new 8/16-bit data.
3. After attaching ECC data to the write-target 32-bit data of step 2., this module writes them to the System SRAM.

45. One-Time Programmable Memory (OTP)

45.1 Overview

This LSI has one-time programmable memory (OTP). Various information is stored or can be additionally stored in OTP. Data written in OTP can't be overwritten. OTP is shared with product configuration parameters which cannot be accessed by user. Then, all the areas are not accessible from CPU.

Table 45.1 lists the OTP specifications and Figure 45.1 shows a block diagram of the OTP.

Table 45.1 OTP Specifications

Item	Description
Number of units	1 unit
Functionality	• Bit width: 32-bit • Word size: 1024-word • Register I/F to read or write OTP • WORDLOCK available each address (32-bit data) • Overwrite protection • Power down mode support
Available information (Non-security support product)	• Unique ID • JTAG authentication ID and setting • SCI/USB boot setting • Anti-rollback counter • Part number • Product Version • User area

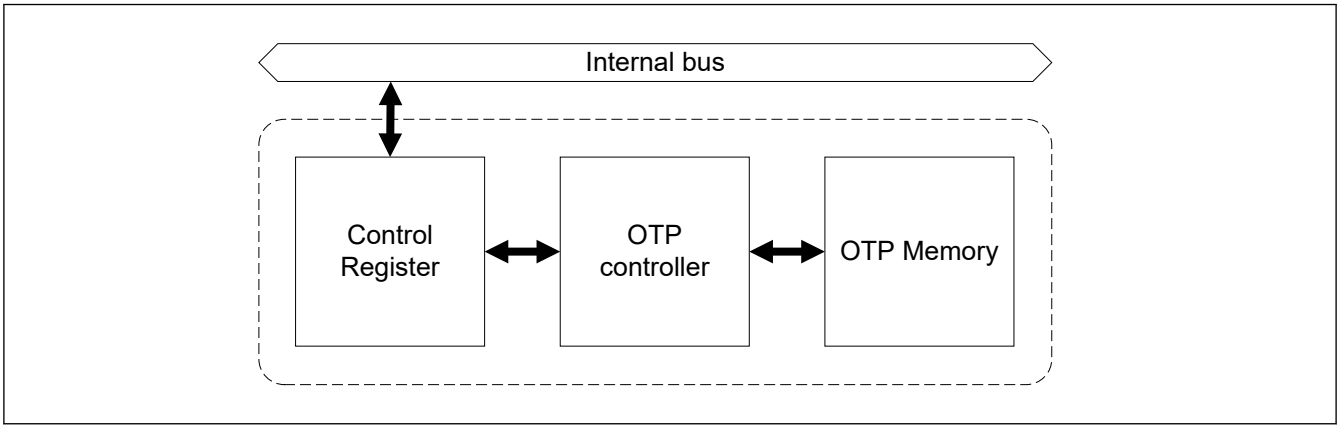


Figure 45.1 OTP Block Diagram

45.2 Register Map

Table 45.2 OTP register map

Address	Register symbol	Register name	Write protection
0x810C_0000	OTPPWR	OTP Power Control Register	—
0x810C_0004	OTPSTR	OTP Access Status Register	—
0x810C_0008	OTPSTAWR	OTP Write Command Register	—
0x810C_000C	OTPADRWR	OTP Write Address Register	—
0x810C_0010	OTPDATAWR	OTP Write Data Register	—
0x810C_0014	OTPADRRD	OTP Read Address Register	—
0x810C_0018	OTPDATARD	OTP Read Data Register	—

Table 45.3 OTP related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
All the registers	—	—	SLVACCCTL7.OTP_SL

45.3 Register Descriptions

45.3.1 OTPPWR : OTP Power Control Register

Base address: OTP = 0x810C_0000

Offset address: 0x00

Bit position:	31																										4				0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ACCL	—	—	—	PWR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PWR	OTP power on/off setting When starting OTP access, set PWR bit to 1 regardless of OTP_PWOK bit in OTPSTR register. 0: Power off 1: Power on	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	ACCL	Selects OTP access I/F 0: Not selected 1: Selected	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

OTPPWR register controls the power of OTP and provides a valid access path.

If 0 is written to PWR bit during write access to OTP, OTP powers off after write access is complete. PWR bit cannot be changed during read access to OTP.

When changing ACCL bit to 1, confirm the following:

- PWR bit is 0.
- CMD_RDY bit in OTPSTR register is 0.

When changing ACCL bit to 0, confirm CMD_RDY bit in OTPSTR register is 1.

When OTP access is requested while ACCL bit is 0, the access request is ignored and ERR_RP bit in OTPSTR register is not asserted.

45.3.2 OTPSTR : OTP Access Status Register

Base address: OTP = 0x810C_0000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OTP_PWOK	—	—	ERR_RDY_RD	ERR_RDY_WR	—	—	—	ERR_RP	ERR_WP	ERR_WR[1:0]		CMD_RDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMD_RDY	Indicates whether OTP controller is ready to receive command or not. 0: Not ready 1: Ready	R
2:1	ERR_WR[1:0]	OTP write status Cleared at the start of the next OTP write command. OTP macro is written each 1-bit internally even if multiple bits write. This bit shows the error status each 1-bit write. The status is not reserved and cleared to 0 before next 1-bit write. When multiple bits write, this bit shows the status of the last 1-bit write only. Status of other bits write is not shown. 0 0: PASS 0 1: Fail due to overwrite to the area where WORDLOCK bit is set to 1 1 0: 1 bit write error 1 1: Reserved	R
3	ERR_WP	Write protection error This bit is set to 1 when OTP write access is issued to the write prohibited area. Cleared at the start of the next OTP write command unless the same error condition occurs.	R
4	ERR_RP	Read protection error This bit is set to 1 when OTP read access is issued to the read prohibited area. Cleared at the start of the next OTP read command unless the same error condition occurs.	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	ERR_RDY_WR	OTP write command ready error This bit is set to 1 when OTP write access is issued during CMD_RDY = 0. Cleared by writing 0 on this bit. Writing 1 is ignored.	R/W
9	ERR_RDY_RD	OTP read command ready error This bit is set to 1 when OTP read access is issued during CMD_RDY = 0. Cleared by writing 0 on this bit. Writing 1 is ignored.	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	OTP_PWOK	Indicates power on/off status of OTP Memory 0: Power Off 1: Power On	R
31:13	—	These bits are read as 0. The write value should be 0.	R/W

OTPSTR register indicates status of command ready and error.

45.3.3 OTPSTAWR : OTP Write Command Register

Base address: OTP = 0x810C_0000

Offset address: 0x08

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STAWR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STAWR	OTP write start By writing 1 to this bit, the OTP write command is issued and writing data to OTP is started. This bit is automatically cleared after the OTP write command is completed. Only 1 can be written to this bit. Writing 0 is ignored.	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

OTPSTAWR register is used to issue the OTP write command.

If OTPSTAWR register is set to 1 while write prohibited area is set in the OTPADRWR register, the STAWR bit does not change to 1 and no OTP write access occurs. In this case, ERR_WP bit in OTPSTR register is set to 1.

45.3.4 OTPADRWR : OTP Write Address Register

Base address: OTP = 0x810C_0000

Offset address: 0x0C

Bit position:	31																				9									0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADRWR[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	ADRWR[9:0]	OTP write address Set the OTP address where data is written. The address to be set is OTP address in a word (32-bit) unit.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

OTPADRWR register is used to set the write address to OTP.

45.3.5 OTPDATAWR : OTP Write Data Register

Base address: OTP = 0x810C_0000

Offset address: 0x10

Bit position:	31																													0
Bit field:	DATAWR[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DATAWR[31:0]	OTP write data Set the data to be written to the OTP.	R/W

OTPDATAWR register is used to set the write data to OTP.

45.3.6 OTPADRRD : OTP Read Address Register

Base address: OTP = 0x810C_0000

Offset address: 0x14

Bit position:	31																				9									0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADRRD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	ADRRD[9:0]	OTP read address Set the OTP address where data is read. The address to be set is OTP address in a word (32-bit) unit. After reading data from the OTP, the address set in this register is automatically incremented. By reading this register, the incremented address can be read. However, if read error (ERR_RP in OTPSTR register) occurs, it will not be incremented. Set a different address.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

OTPADRRD register is used to set the read addresses to OTP.

45.3.7 OTPDATARD : OTP Read Data Register

Base address: OTP = 0x810C_0000

Offset address: 0x18

Bit position:	31																												0
Bit field:	DATARD[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DATARD[31:0]	OTP read data OTP data in address specified by the OTPADRRD register can be read.	R

OTPDATARD register stores read data from OTP.

OTP read access is not performed when the OTPDATARD register is read while read prohibited area is set in the OTPADRRD register. In this case, the ERR_RP bit of the OTPSTR register is set.

45.4 OTP Memory Map

Data width of OTP is 32-bit, and its address is not memory mapped. Dedicated address (0x000 to 0x3FF) is defined to access OTP. Table 45.4 shows OTP memory map in non-security support product. Various information is stored or can be stored in OTP. Address not listed in Table 45.4 is reserved and do not access reserved area.

In OTP, unwritten bit is 0 and written bit is 1. WORDLOCK function is available each address to prevent the area from being overwritten. Control area for WORDLOCK function is assigned in 0x001 to 0x01F. Target area (0x020 to 0x3FF) is protected from overwrite by setting 1 to the corresponding address/bit of the WORDLOCK control area. The relation of target area and corresponding address/bit of the WORDLOCK control area is expressed as following formula.

Address in WORDLOCK control area: Integer part of Target_address / 32

Bit of the address in WORDLOCK control area: Target_address % 32, where % is modulo operation.

Do not write to the corresponding address/bit of WORDLOCK control area for the target area not listed in Table 45.4.

WORDLOCK function for WORDLOCK control area is not available. That means writing to 0x000 is impossible.

Table 45.4 OTP memory map in non-security support product (1 of 2)

Area	Bit width	Start address	End address	WORDLOCK control		Initial property
				Address	Bit	
WORDLOCK control area	1024	0x001	0x01F	—	—	R/W
Unique ID	64	0x0E5	0x0E6	0x007	5 to 6	R
JTAG authentication mode	9	0x0E9	0x0E9	0x007	9	R/W
JTAG authentication level 1 plaintext ID	128	0x0EB	0x0EE	0x007	11 to 14	W
JTAG authentication level 2 plaintext ID	128	0x0EF	0x0F2	0x007	15 to 18	W
SCI/USB boot authentication mode	6	0x1CB	0x1CB	0x00E	11	R/W
SCI/USB boot authentication plaintext ID	128	0x0F3	0x0F6	0x007	19 to 22	W
Anti-rollback counter	320	0x1B1	0x1BA	0x00D	17 to 26	R/W
Part number	32	0x1D9	0x1D9	0x00E	25	R
Product Version	32	0x1DA	0x1DA	0x00E	26	R
Temperature Sensor Code High	12	0x1DC	0x1DC	0x00E	28	R
Temperature Sensor Code Low	12	0x1DD	0x1DD	0x00E	29	R

Table 45.4 OTP memory map in non-security support product (2 of 2)

Area	Bit width	Start address	End address	WORDLOCK control		Initial property
				Address	Bit	
User area	16256	0x204	0x3FF	0x010 to 0x01F	4 to 31 (0x010) 0 to 31 (0x011 to 0x01F)	R/W

45.4.1 Unique ID (UID)

Indicates 64-bit unique ID. This area is read-only.

Address(es): 0x0E5 to 0x0E6

Bit position:	31																												0
Bit field:	ID																												
Value after reset:	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	Symbol	Function	R/W
31:0	ID	64-bit unique ID 0x0E5: Unique ID [31:0] 0x0E6: Unique ID [63:32]	R

Note 1. Value after reset depends on the product variant.

45.4.2 JTAG Authentication Mode (AUTHMODEJ)

Specifies how to authenticate JTAG connection.

Address(es): 0x0E9

Bit position:	31																						8							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MODE[8:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	MODE[8:0]	Specifies JTAG authentication mode Significant bit consists of 3-bit OR. Here, B2 = OR(MODE[8:6]), B1 = OR(MODE[5:3]), B0 = OR(MODE[2:0]). Then, B2 B1 B0: 0 0 0: No authentication 0 0 1: Authentication level 1 (Both EL2 and EL1/EL0 access permission) 0 1 x: Authentication level 2 (Only EL1/EL0 access permission) 1 x x: Permanent prohibition of JTAG connection	R/W
31:9	—	Reserved	—

45.4.3 JTAG Authentication Level 1 Plaintext ID (JID1P)

Specifies plaintext ID for JTAG authentication level 1. This area is write-only. Read value is 0. ID must include at least one '1' programmed bit each 32-bit. It is recommended to set WORDLOCK to this area after writing ID to avoid unintentional ID change.

Address(es): 0x0EB to 0x0EE

Bit position:	31																												0
Bit field:	ID																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ID	128-bit JTAG authentication level 1 plaintext ID 0x0EB: ID [31:0] 0x0EC: ID [63:32] 0x0ED: ID [95:64] 0x0EE: ID [127:96]	W

45.4.4 JTAG Authentication Level 2 Plaintext ID (JID2P)

Specifies plaintext ID for JTAG authentication level 2. This area is write-only. Read value is 0. ID must include at least one '1' programmed bit each 32-bit. It is recommended to set WORDLOCK to this area after writing ID to avoid unintentional ID change.

Address(es): 0x0EF to 0x0F2

Bit position:	31																												0
Bit field:	ID																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ID	128-bit JTAG authentication level 2 plaintext ID 0x0EF: ID [31:0] 0x0F0: ID [63:32] 0x0F1: ID [95:64] 0x0F2: ID [127:96]	W

45.4.5 SCI/USB Boot Authentication Mode (AUTHMODES)

Specifies how to authenticate SCI/USB boot connection.

Address(es): 0x1CB

Bit position:	31																										5			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MODE[5:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	MODE[5:0]	Specifies SCI/USB boot authentication mode Significant bit consists of 3-bit OR. Here, B1 = OR(MODE[5:3]), B0 = OR(MODE[2:0]). Then, B1 B0: 0 0: No authentication 0 1: Authentication 1 x: Permanent prohibition of SCI/USB boot connection	R/W
31:6	—	Reserved	—

45.4.6 SCI/USB Boot Authentication Plaintext ID (SIDP)

Specifies plaintext ID for SCI/USB boot authentication. This area is write-only. Read value is 0. ID must include at least one '1' programmed bit each 32-bit. It is recommended to set WORDLOCK to this area after writing ID to avoid unintentional ID change.

Address(es): 0x0F3 to 0x0F6

Bit position:	31																												0
Bit field:	ID																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ID	128-bit SCI/USB boot authentication plaintext ID 0x0F3: ID [31:0] 0x0F4: ID [63:32] 0x0F5: ID [95:64] 0x0F6: ID [127:96]	W

45.4.7 Anti-rollback Counter (ARBC)

This area is used as anti-rollback counter. User can freely use this area to control version of their application. For example, five 64-bit counter for version control of five program modules at firmware update.

Address(es): 0x1B1 to 0x1BA

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BIT31	BIT30	BIT29	BIT28	BIT27	BIT26	BIT25	BIT24	BIT23	BIT22	BIT21	BIT20	BIT19	BIT18	BIT17	BIT16	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	BIT31 to BIT0	Uses as version control	R/W

45.4.8 Part Number (PN)

Indicates part number of the product. This area is read-only.

Address(es): 0x1D9

Bit position:	31																												0
Bit field:	PN																												
Value after reset:	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	Symbol	Function	R/W
31:0	PN	Part number of the product 0x0000_3000: R9A09G077M48GBG 0x0000_3001: R9A09G077M28GBG 0x0000_3002: R9A09G077M08GBG 0x0000_3003: R9A09G077M44GBG 0x0000_3004: R9A09G077M24GBG 0x0000_3005: R9A09G077M04GBG 0x0000_4000: R9A09G087M48GBG 0x0000_4001: R9A09G087M28GBG 0x0000_4002: R9A09G087M08GBG 0x0000_4003: R9A09G087M44GBG 0x0000_4004: R9A09G087M24GBG 0x0000_4005: R9A09G087M04GBG	R

Note 1. Depends on part number of the product

45.4.9 Product Version (VER)

Indicates product version. This area is read-only.

Address(es): 0x1DA

Bit position:	31																												0
Bit field:	VER																												
Value after reset:	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	Symbol	Function	R/W
31:0	VER	Product version	R

Note 1. Depends on version of the product

45.4.10 Temperature Sensor Code High (TEMPHI)

Indicates 12-bit temperature sensor code data measured at 125±1°C. This area is read-only.

Address(es): 0x1DC

Bit position:	31																				11										0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCODE[11:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	Symbol	Function	R/W
11:0	TSCODE[11:0]	Temperature sensor code data measured at 125±1°C	R
31:12	—	Reserved	—

Note 1. Value after reset depends on the product variant.

45.4.11 Temperature Sensor Code Low (TEMPLO)

Indicates 12-bit temperature sensor code data measured at −40±1°C. This area is read-only.

Address(es): 0x1DD

Bit position:	31																				11										0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCODE[11:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	Symbol	Function	R/W
11:0	TSCODE[11:0]	Temperature sensor code data measured at −40±1℃	R
31:12	—	Reserved	—

Note 1. Value after reset depends on the product variant.

45.4.12 User Area (USERAREA)

This area can be used by user freely. Since WORDLOCK is applied for this area, one time write of 32-bit data is needed each address.

Address(es): 0x204 to 0x3FF

Bit position:	31																												0
Bit field:	AREA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	AREA[31:0]	Free area which user can use.	R/W

45.5 Operation

45.5.1 Write procedure

Table 45.5 Write procedure

Steps	What to do
A	It is assumed that OTP is in a power-down state. Confirm completion of the power-down process by checking that the PWR bit in OTPPWR register is 0 and the CMD_RDY bit in OTPSTR register is 0.
B	Set the PWR and ACCL bits in OTPPWR register to 1. Confirm if CMD_RDY bit in OTPSTR register is 1 after setting.
C	Set the write address to the OTPADRWR register.
D	Set the data to be written to the OTPDATAWR register.
E	Set 1 to the STAWR bit in OTPSTAWR register.
F	OTP controller starts the OTP write process. CMD_RDY bit changes from 1 to 0. If setting the write-prohibited area to OTPADRWR register, the STAWR bit is not set to 1 and the CMD_RDY bit keeps 1. In this case, ERR_WP bit in the OTPSTR register is set to 1.
G	OTP controller writes the value of OTPDATAWR register to OTP. Note) The values of OTPADRWR and OTPDATAWR registers must be retained until this procedure is completed.
H	Poll the STAWR bit until changing to 0 in order to detect the completion of the write command acceptance.
I	Poll the CMD_RDY bit until changing to 1 in order to detect the completion of the write.
J	Once detecting the CMD_RDY bit is 1, confirm that the ERR_WP and ERR_WR bit in the OTPSTR register is not set to 1. Note) If set to 1, OTP write error occurs.
K	When continuing OTP write, repeat step C) and later. When issuing OTP read, refer to step C) and later of Table 45.6. When completing OTP access: • When OTP Read is performed after releasing power down (PWR = 1), dummy read is needed. After dummy OTP Read to the unused user area, set PWR and ACCL bits in the OTPPWR register to 0, and wait until CMD_RDY bit is set to 0. • When OTP Read is not performed after releasing power down (PWR = 1), dummy read is not needed. Set PWR and ACCL bits in the OTPPWR register to 0, and wait until CMD_RDY bit is set to 0. Note) Be sure to power down by this procedure if no more access is needed.

Note: WORDLOCK can be applied for some areas. The OTP write to the WORDLOCK applied area cannot be performed.
For written bits, the bits retain 1 no matter whether they are written with 0 or 1 again.
Reset while writing OTP is prohibited. That operation is not guaranteed.

45.5.2 Read procedure

Table 45.6 Read procedure

Steps	What to do
A	It is assumed that OTP is in a power-down state. Confirm completion of the power-down process by checking that the PWR bit in OTPPWR register is 0 and the CMD_RDY bit in OTPSTR register is 0.
B	Set the PWR and ACCL bits in OTPPWR register to 1. Confirm if CMD_RDY bit in OTPSTR register is 1 after setting.
C	Set the read address to the OTPADRRD register.
D	Read the OTPDATARD register.
E	OTP controller starts the OTP read process. CMD_RDY bit changes from 1 to 0. If setting the read-prohibited area to OTPADRRD register, OTP read process is not performed and the CMD_RDY bit keeps 1. In this case, ERR_RP bit of the OTPSTR register is set to 1.
F	OTP controller reads the data of address set in the OTPADRRD register from OTP. After completion of the process, CPU can receive read data. CMD_RDY bit changes from 0 to 1. Confirm that the ERR_WP bit in the OTPSTR register is not set to 1. If set to 1, OTP read error occurs. Note) Since CPU is waited until completion of the process, polling of CMD_RDY bit is not needed.
G	When continuing OTP read, repeat step C) and later. ● When reading a continuous address (continuous read), repeat step D) and later. The OTPADRRD register is incremented automatically, so no need to change. ● When reading a different address from a continuous address, repeat step C) and later. When issuing OTP write, refer to step C) and later of Table 45.5. When completing OTP access, perform dummy OTP read to the unused user area. Then set PWR and ACCL bits in the OTPPWR registers to 0, and wait until CMD_RDY bit is set to 0. Note) Be sure to power down by this procedure if no more access is needed.

Note: Continuous read is valid for readable areas. If the read address is changed to the read-prohibited area, OTP read process is not performed and the CMD_RDY bit keeps 1. ERR_RP bit of the OTPSTR register is set to 1.

46. Serial Host Interface (SHOSTIF)

46.1 Overview

This LSI has a serial host interface to connect an external host CPU (SPI master) with up to eight input/output pins. The serial host interface allows an external SPI master to access the internal resources (mainly System SRAM) of this LSI directly.

Figure 46.1 shows a block diagram of the SHOSTIF and Table 46.1 lists the serial host interface specifications.

Table 46.1 SHOSTIF specifications

Parameter	Description
Number of units	1 unit
Functionality	• Serial communication is possible in slave mode • Supported interface: – Motorola Serial Peripheral Interface (4-wire SPI) – Enhanced SPI modes with Dual, Quad or Octal SPI. • Serial clock polarity switching • Serial clock phase switching • Transfer mode: – Single data transfer (SDR) supported – No DDR, data strobe nor data mask supported • All the SPI transfer consists of Instruction phase, Address phase and Data phase: – Command phase: 8 bits — Support read/write, transfer type, wait cycles and data length – Address phase: 24 bits — Aligned to 32-bit data boundary – Data phase: 32 bits × burst length up to 64 • Hardware protocol state machine for transfer management • 32 bits × 64 stage FIFO transmit and receive buffers • Serial clock (sclk_in): Up to 40 MHz
Interrupt source	Maskable interrupt sources. All sources are combined to one interrupt. External output pin is available: • SPI master error • AHB error • Transmit FIFO Empty • Transmit FIFO Underflow • Receive FIFO Full • Receive FIFO Overflow
Accessible area	• System SRAM • Mailbox and Semaphore registers • Serial Host Interface registers
Module-stop function	Module-stop state can be set to reduce power consumption

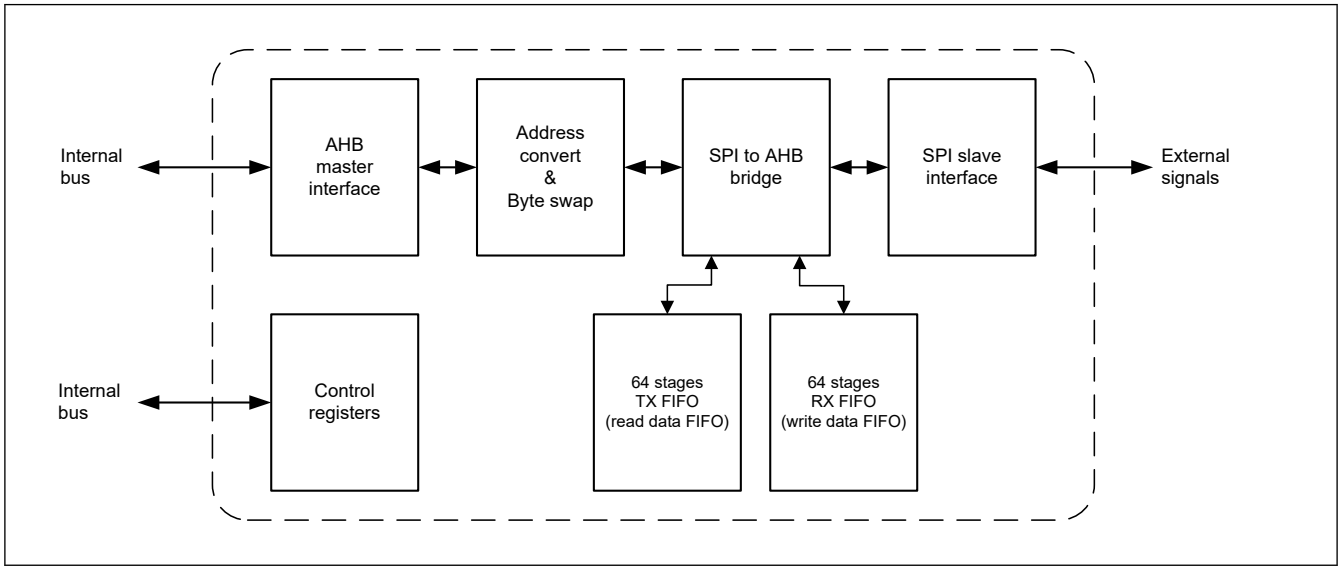


Figure 46.1 SHOSTIF block diagram

Table 46.2 lists the input/output pins of the serial host interface.

Table 46.2 SHOSTIF I/O pins

Name	I/O	Function
HSPI_CK	Input	SHOSTIF Clock input pin
HSPI_CS#	Input	SHOSTIF Chip select input pin
HSPI_IO0	I/O	SHOSTIF Data0 input/output pin
HSPI_IO1	I/O	SHOSTIF Data1 input/output pin
HSPI_IO2	I/O	SHOSTIF Data2 input/output pin
HSPI_IO3	I/O	SHOSTIF Data3 input/output pin
HSPI_IO4	I/O	SHOSTIF Data4 input/output pin
HSPI_IO5	I/O	SHOSTIF Data5 input/output pin
HSPI_IO6	I/O	SHOSTIF Data6 input/output pin
HSPI_IO7	I/O	SHOSTIF Data7 input/output pin
HSPI_INT#	Output	SHOSTIF Interrupt output pin

46.2 Register Map

Table 46.3 SHOSTIF register map

Address*1	Register symbol	Register name	Write protection
0x8024_1000	CTRLR0	Control Register 0	—
0x8024_1008	ENR	Enable Register	—
0x8024_1014	RXFBTR	Receive FIFO Burst Threshold Register	—
0x8024_1018	TXFTLR	Transmit FIFO Threshold Level Register	—
0x8024_101C	RXFTLR	Receive FIFO Threshold Level Register	—
0x8024_1028	SR	Status Register	—
0x8024_102C	IMR	Interrupt Mask Register	—
0x8024_1030	ISR	Interrupt Status Register	—
0x8024_1034	RISR	Raw Interrupt Status Register	—
0x8024_1038	TXUICR	Transmit FIFO Underflow Interrupt Clear Register	—
0x8024_103C	RXOICR	Receive FIFO Overflow Interrupt Clear Register	—
0x8024_1040	SPIMECR	SPI Master Interrupt Clear Register	—
0x8024_1044	AHBECR	AHB Error Clear Register	—
0x8024_1048	ICR	Interrupt Clear Register	—
0x8129_4000	SHCFG	SHOSTIF Configuration Register	PRCRS.PRC3

Note 1. Address is internal memory address view.

Table 46.4 SHOSTIF related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	MRCTLI.MRCTLI01 MRCTLI.MRCTLI02 MRCTLI.MRCTLI03	MSTPCRI.MSTPCRI01	SLVACCCTL7.SHOSTIF_SL

46.3 Register Descriptions

The register base address is different between external SPI master view and internal memory address view except for SHCFG register.

- External SPI master view (8-bit addressing mode) : SHOSTIF_E08 = 0x20_1000

Note: When 32-bit addressing mode is used, please shift 8-bit address (base address + offset address) by 2 bits in the right direction.

- Internal memory address view: SHOSTIF = 0x8024_1000
- Internal memory address view for SHCFG: SHOSTIF_CFG = 0x8129_4000

46.3.1 CTRLR0 : Control Register 0

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SCPOL	SCPH	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	SCPH	Serial Clock Phase Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data is captured on the first edge of the serial clock. When SCPH = 1, the serial clock toggles for 1 cycle after the slave select line is activated, and data is captured on the second edge of the serial clock. 0: Serial clock toggles in the middle of the first bit 1: Serial clock toggles at the start of the first bit	R/W
9	SCPOL	Serial Clock Polarity Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the SPI master is not actively transferring data on the serial bus. 0: Inactive state of the serial clock is low 1: Inactive state of the serial clock is high	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The CTRLR0 register controls the serial data transfer. It is not possible to write to this register when SHOSTIF is enabled (ENR.ENABLE = 1).

46.3.2 ENR : Enable Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	SHOSTIF Enable Enables and disables all SHOSTIF operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is not possible to program some of the SHOSTIF control registers when enabled. When disabled, the SLEEP bit in the SHCFG register is set (after a delay) to inform the system that it is safe to remove the internal clock, thus saving power consumption in the system. 0: Disables SHOSTIF 1: Enables SHOSTIF	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The ENR register enables or disables the serial host interface.

46.3.3 RXFBTR : Receive FIFO Burst Threshold Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	RXFBTL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bit	Symbol	Function	R/W
5:0	RXFBTL[5:0]	Receive FIFO Burst Threshold For write operations on SPI slave interface, this register is used to build a burst on AHB master interface. The data is sent in bursts equal to the level specified in this register. If the RXFBTL value is 3 then a burst length of 4 is used for every write transfer. If SHOSTIF receives a burst smaller than the programmed value, then SHOSTIF issues a smaller burst on the AHB interface and completes the transaction. In case of 8-bit addressing mode, do not change from the initial value. In case of 32-bit addressing mode, recommendation value is 0x0F.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The RXFBTR register controls the burst threshold value for AHB interface from RX FIFO.

46.3.4 TXFTLR : Transmit FIFO Threshold Level Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TFT[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	TFT[5:0]	Transmit FIFO Threshold Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is 64. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The TXFTLR register controls the threshold value for the transmit FIFO memory.

46.3.5 RXFTLR : Receive FIFO Threshold Level Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x01C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	RFT[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	RFT[5:0]	Receive FIFO Threshold Controls the level of entries or above at which the receive FIFO controller triggers an interrupt. The FIFO depth is 64. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The RXFTLR register controls the threshold value for the received FIFO memory.

46.3.6 SR : Status Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSY	Busy Flag When set, indicates that a serial transfer is in progress. When cleared, indicates that the SHOSTIF is idle or disabled. 0: SHOSTIF is idle or disabled 1: SHOSTIF is actively transferring data	R
31:1	—	These bits are read as 0.	R

The SR register is a read-only register that indicates the current transfer status, FIFO status, and any transmission/reception errors that may have occurred. The Status register may be read at any time. None of the bits in this register request an interrupt.

46.3.7 IMR : Interrupt Mask Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x02C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPIMEM	AHBEM	TXUIM	—	—	RXFIM	RXOIM	—	—	TXEIM
Value after reset:	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	1

Bit	Symbol	Function	R/W
0	TXEIM	Transmit FIFO Empty Interrupt Mask 0: txe_intr interrupt is masked 1: txe_intr interrupt is not masked	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	RXOIM	Receive FIFO Overflow Interrupt Mask 0: rxo_intr interrupt is masked 1: rxo_intr interrupt is not masked	R/W
4	RXFIM	Receive FIFO Full Interrupt Mask 0: rxf_intr interrupt is masked 1: rxf_intr interrupt is not masked	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TXUIM	Transmit FIFO Underflow Mask 0: txu_intr interrupt is masked 1: txu_intr interrupt is not masked	R/W
8	AHBEM	AHB Error Interrupt Mask 0: ahbe_intr interrupt is masked 1: ahbe_intr interrupt is not masked	R/W
9	SPIMEM	SPI Master Error Interrupt Mask 0: spime_intr interrupt is masked 1: spime_intr interrupt is not masked	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The IMR register masks or enables all interrupts generated by the SHOSTIF. Actual interrupt mask is determined by combination with the SHCFG register.

46.3.8 ISR : Interrupt Status Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPIMES	AHBES	TXUIS	—	—	RXFIS	RXOIS	—	—	TXEIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1

Bit	Symbol	Function	R/W
0	TXEIS	Transmit FIFO Empty Interrupt Status 0: txe_intr interrupt is not active after masking 1: txe_intr interrupt is active after masking	R
2:1	—	These bits are read as 0.	R
3	RXOIS	Receive FIFO Overflow Interrupt Status 0: rxo_intr interrupt is not active after masking 1: rxo_intr interrupt is active after masking	R
4	RXFIS	Receive FIFO Full Interrupt Status 0: rxf_intr interrupt is not active after masking 1: rxf_intr interrupt is active after masking	R
6:5	—	These bits are read as 0.	R
7	TXUIS	Transmit FIFO Underflow Status 0: txu_intr interrupt is not active after masking 1: txu_intr interrupt is active after masking	R
8	AHBES	AHB Error Interrupt Status 0: ahbe_intr interrupt is not active after masking 1: ahbe_intr interrupt is active after masking	R
9	SPIMES	SPI Master Error Interrupt Status 0: spime_intr interrupt is not active after masking 1: spime_intr interrupt is active after masking	R
31:10	—	These bits are read as 0.	R

Note 1. Initial value depends on the setting of ENR register. The value is 0 when ENR.ENABLE = 0 and 1 when ENR.ENABLE = 1.

The ISR register reports the status of the SHOSTIF interrupts after they are masked.

46.3.9 RISR : Raw Interrupt Status Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPIMER	AHBER	TXUIR	—	—	RXFIR	RXOIR	—	—	TXEIR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1

Bit	Symbol	Function	R/W
0	TXEIR	Transmit FIFO Empty Raw Interrupt Status 0: txe_intr interrupt is not active prior to masking 1: txe_intr interrupt is active prior to masking	R
2:1	—	These bits are read as 0.	R
3	RXOIR	Receive FIFO Overflow Raw Interrupt Status 0: rxo_intr interrupt is not active prior to masking 1: rxo_intr interrupt is active prior to masking	R
4	RXFIR	Receive FIFO Full Raw Interrupt Status 0: rxf_intr interrupt is not active prior to masking 1: rxf_intr interrupt is active prior to masking	R
6:5	—	These bits are read as 0.	R
7	TXUIR	Transmit FIFO Underflow Raw Interrupt Status 0: txu_intr interrupt is not active prior to masking 1: txu_intr interrupt is active prior to masking	R
8	AHBER	AHB Error Raw Interrupt Status This bit indicates whether AHB master interface received an error response. This bit is cleared when the status register is read. 0: ahbe_intr interrupt is not active prior to masking 1: ahbe_intr interrupt is active prior to masking	R
9	SPIMER	SPI Master Error Raw Interrupt Status This bit is set to 1 under the following conditions: • If SPI slave receives a command which does not fall into the command set • If SPI slave receives an out-of-order read command (for example read data before read request in polling mode) • If SPI master writes more data than specified in the instruction phase This bit is cleared when the status register is read. 0: spime_intr interrupt is not active prior to masking 1: spime_intr interrupt is active prior to masking	R
31:10	—	These bits are read as 0.	R

Note 1. Initial value depends on the setting of the ENR register. The value is 0 when ENR.ENABLE = 0 and 1 when ENR.ENABLE = 1.

The RISR register reports the status of the SHOSTIF interrupts before they are masked.

46.3.10 TXUICR : Transmit FIFO Underflow Interrupt Clear Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x038

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXUICR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXUICR	Clear Transmit FIFO Underflow Interrupt This register reflects the status of the interrupt. A read from this register clears the txu_intr interrupt. Writing to this register has no effect. Read value is the same as TXUIS bit in the ISR register.	R
31:1	—	These bits are read as 0.	R

The TXUICR register is used to clear the txu_intr interrupts.

46.3.11 RXOICR : Receive FIFO Overflow Interrupt Clear Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x03C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXOICR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RXOICR	Clear Receive FIFO Overflow Interrupt This register reflects the status of the interrupt. A read from this register clears the rxo_intr interrupt. Writing has no effect. Read value is the same as RXOIS bit in the ISR register.	R
31:1	—	These bits are read as 0.	R

The RXOICR register is used to clear the rxo_intr interrupts.

46.3.12 SPIMECR : SPI Master Interrupt Clear Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPIMECR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPIMECR	Clear SPI Master Error interrupt This register reflects the status of the interrupt. A read from this register clears the spime_intr interrupt. Writing to this register has no effect. Read value is the same as the SPIMES bit in the ISR register.	R
31:1	—	These bits are read as 0.	R

The SPIMECR register is used to clear the spime_intr interrupts.

46.3.13 AHBECR : AHB Error Clear Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x044

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AHBECR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AHBECR	Clear AHB Error Interrupt This register reflects the status of the interrupt. A read from this register clears the ahbe_intr interrupt. Writing to this register has no effect. Read value is the same as AHBES bit in the ISR register.	R
31:1	—	These bits are read as 0.	R

The AHBECR register is used to clear the ahbe_intr interrupts.

46.3.14 ICR : Interrupt Clear Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x20_1000

Offset address: 0x048

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ICR	Clear Interrupts This register is set if any of the specified interrupts are active. A read clears the txo_intr, rxu_intr, rxo_intr, ahbe_intr, and the mst_intr interrupts. Writing to this register has no effect. Read value is the same as logical OR of the RXOIS, TXUIS, AHBES, and SPIMES bits in the ISR register.	R
31:1	—	These bits are read as 0.	R

The ICR register is used to clear the txo_intr, rxu_intr, rxo_intr, ahbe_intr, and the mst_intr interrupts.

46.3.15 SHCFG : SHOSTIF Configuration Register

Base address: SHOSTIF_CFG = 0x8129_4000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	INTMASKE[5:0]						—	—	INTMASKI[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	SLEEP	ADDRESSING	BYTESWAP	SPIMODE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SPIMODE[1:0]	SPI Frame Format Select Indicates the mode in which the SPI transfer occurs. 0 0: Standard SPI mode 0 1: SPI Dual mode 1 0: SPI Quad mode 1 1: SPI Octal mode	R/W
2	BYTESWAP	Byte Swap Mode Swap data by byte order on AHB bus. 0: No byte swap. Use this setting if byte order of data input by SPI master is [31:24], [23:16], [15:8], or [7:0]. 1: Byte swap. Use this setting if byte order of data input by SPI master is [7:0], [15:8], [23:16], or [31:24].	R/W
3	ADDRESSING	Addressing Mode Specifies unit of the address input by SPI master. 0: 32-bit 1: 8-bit	R/W
4	SLEEP	SHOSTIF Enable Flag Monitor Indicates whether the internal clock can be stopped. Note that when SHOSTIF is reset, this flag does not indicate correct state. 0: Not safe to stop the internal clock 1: Safe to stop the internal clock	R
15:5	—	These bits are read as 0. The write value should be 0.	R/W
21:16	INTMASKI[5:0]	Interrupt Mask Enable for Internal Interrupt (SHOST_INT) Bits correspond to the interrupts: • INTMASKI[0]: Transmit FIFO Empty Interrupt • INTMASKI[1]: Receive FIFO Overflow Interrupt • INTMASKI[2]: Receive FIFO Full Interrupt • INTMASKI[3]: Transmit FIFO underflow Interrupt • INTMASKI[4]: AHB Master interface error Interrupt • INTMASKI[5]: SPI Master interface error Interrupt 0: No masked 1: Masked	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
29:24	INTMASKE[5:0]	Interrupt Mask Enable for External Interrupt (HSPI_INT# signal) Bits correspond to the interrupts: • INTMASKE[0]: Transmit FIFO Empty Interrupt • INTMASKE[1]: Receive FIFO Overflow Interrupt • INTMASKE[2]: Receive FIFO Full Interrupt • INTMASKE[3]: Transmit FIFO Underflow Interrupt • INTMASKE[4]: AHB Master Interface Error Interrupt • INTMASKE[5]: SPI Master Interface Error Interrupt 0: No masked 1: Masked	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

The SHCFG register sets the operation mode and monitors the status of SHOSTIF. This register is not accessible from the external SPI master.

46.4 Operations

The SHOSTIF converts all transfers from SPI to AHB transactions. The external SPI master sends address to SHOSTIF. SHOSTIF takes the address and converts it into an AHB transfer. This enables the external master to access the internal memory directly without any CPU intervention. SHOSTIF can be enabled through boot mode on the reset and load boot program from external master.

All SPI transfers have the following phases:

- Instruction
- Address
- Data

46.4.1 Instruction Phase

Each transfer from the SPI master starts with an 8-bit instruction, which determines the transfer type and process for the rest of the transfer. Table 46.5 shows the instruction field decode.

Table 46.5 Instruction field decode

Bit	Bit name	Description
7	Read/Write	Defines whether the transfer is read or write. 0: Read 1: Write
6:5	Transfer Type	Defines the transfer type. 00: Read/write data 01: Read request 10: Read/write status 11: Read data with dummy
4:3	Wait Cycles	Defines the wait cycles for read transactions. Selects the number of wait cycles for a read transfer. This field is only applicable for read data transfers. When the transfer type is 00, the field decode is as follows: 00: 0 Wait cycles 01: 1 Wait cycle 10: Reserved 11: Reserved When the transfer type is 11, the field decode is as follows: 00: 8 Wait cycles 01: 16 Wait cycles 10: 24 Wait cycles 11: 32 Wait cycles Note: Wait cycles are not applicable for write data operation.
2:0	Data Length	Defines the data to be transmitted in the current transfer. Unit is 32 bits (4 bytes). 000: 1 001: 4 010: 8 011: 16 100: 32 101: 64 110: Prohibited 111: Prohibited

46.4.2 Address Phase

The address phase determines the address from which the data should be read or written on the AHB master interface. Address could be received in the standard, dual, quad, or octal frame format, which is defined by the SPIMODE[1:0] bits in SHCFG register. AHB interface address width is fixed to 32 bits. SPI interface expects to receive 24-bit address, which is sent to the AHB interface. Because the data is always read/written in frame of 32 bits, the address is always word-aligned. The 32-bit output address on the SPI to AHB bridge is dependent on the setting of addressing mode as shown in Table 46.6. Addressing mode is determined by the ADDRESSING bit in SHCFG register.

Table 46.6 Addressing mode

Addressing mode	Data length	Permitted SPI_ADDR*1	Permitted address granularity	SPI to AHB bridge output address
32-bit*2	1/4/8/16/32/64	0x0, 0x1, 0x2, …, 0xFF_FFFF	0x1	{000000b, {SPI_ADDR}, 00b}
8-bit	1	0x0, 0x4, 0x8, …, 0xFF_FFFC	0x4	{00000000b, {SPI_ADDR}}
	4	0x0, 0x10, 0x20, …, 0xFF_FFF0	0x10	
	8	0x0, 0x20, 0x40, …, 0xFF_FFE0	0x20	
	16	0x0, 0x40, 0x80, …, 0xFF_FFC0	0x40	
	32	0x0, 0x80, 0x100, …, 0xFF_FF80	0x80	
	64	0x0, 0x100, 0x200, …, 0xFF_FF00	0x100	

Note 1. SPI_ADDR = 24 bits address received on SPI Interface.
Note 2. Burst transfer across 1 Kbyte boundary is prohibited.

46.4.2.1 Address Conversion

The SPI master cannot access all the memory area in this LSI because the address size is 24-bit address on SPI frame. In this LSI, System SRAM, Mailbox and Semaphore, and SHOSTIF registers are accessible from the external SPI master. Address conversion circuit converts the SPI to AHB bridge output address to the internal address of System SRAM, Mailbox and Semaphore, and SHOSTIF registers. Table 46.7 shows address mapping between SPI address on SPI frame and internal address. Access to reserved area causes an AHB error interrupt.

Table 46.7 Address mapping

Module	SPI address*1	Internal address	Size
System SRAM0 (SYSRAM0)	0x00_0000 to 0x01_FFFF (32-bit) 0x00_0000 to 0x07_FFFC (8-bit)	0x1000_0000 to 0x1007_FFFC	512 KB
System SRAM1 (SYSRAM1)	0x02_0000 to 0x03_FFFF (32-bit) 0x08_0000 to 0x0F_FFFC (8-bit)	0x1008_0000 to 0x100F_FFFC	512 KB
System SRAM2 (SYSRAM2)	0x04_0000 to 0x05_FFFF (32-bit) 0x10_0000 to 0x17_FFFC (8-bit)	0x1010_0000 to 0x1017_FFFC	512 KB
System SRAM3 (SYSRAM3)	0x06_0000 to 0x07_FFFF (32-bit) 0x18_0000 to 0x1F_FFFC (8-bit)	0x1018_0000 to 0x101F_FFFC	512 KB
Mailbox and Semaphore (MBXSEM)	0x08_0000 to 0x08_03FF (32-bit) 0x20_0000 to 0x20_0FFC (8-bit)	0x8024_0000 to 0x8024_0FFC	4 KB
SHOSTIF registers	0x08_0400 to 0x08_07FF (32-bit) 0x20_1000 to 0x20_1FFC (8-bit)	0x8024_1000 to 0x8024_1FFC	4 KB
Reserved	0x08_0800 to 0xFF_FFFC (32-bit) 0x20_2000 to 0xFF_FFFF (8-bit)	Error response	—

Note 1. Depends on the addressing mode.

46.4.3 Data Phase

Data is received or transmitted in frames of 32 bits. The amount of data to be transmitted or received is defined by the Data Length field of Instruction phase.

Data frame is assumed MSB first, which means the byte order is [31:24], [23:16], [15:8], or [7:0]. When the BYTESWAP bit in the SHCFG register is set to 1, the byte order of Data frame can be swapped to [7:0], [15:8], [23:16], or [31:24].

46.4.4 Bridge Operation

The SHOSTIF performs as a bridge between the SPI master and AHB subsystem. The transfers from SPI master are directly converted in AHB read/write operations. The 8-bit instruction defines the characteristics of the transfer and direction of data flow, and SPIMODE[1:0] bits in the SHCFG register define the SPI operating mode. The following sections describe the read and write operations.

46.4.5 Write Operation

46.4.5.1 Write Instruction

The Instruction phase determines the amount of data to be transmitted during write operation. The SPI frame format is derived from SPIMODE[1:0] bits in the SHCFG register. SPI slave supports the instruction opcodes as shown in Table 46.8.

Table 46.8 Instruction set for write data operations

Instruction	Instruction name	Description
0x8i*1 i = 0 to 5	WRITEn n = 1, 4, 8, 16, 32, 64	Write operation
0xC0	WRITESTATUS0	Write Data Status instruction
0xC8	WRITESTATUS1	Write Data Status instruction with 1 wait cycle

Note 1. The lower 3 bits of the instruction determine the amount of data to be fetched by AHB master. For field decode, see Table 46.5.

46.4.5.2 Write Operation

Figure 46.2 shows how SHOSTIF performs write operation.

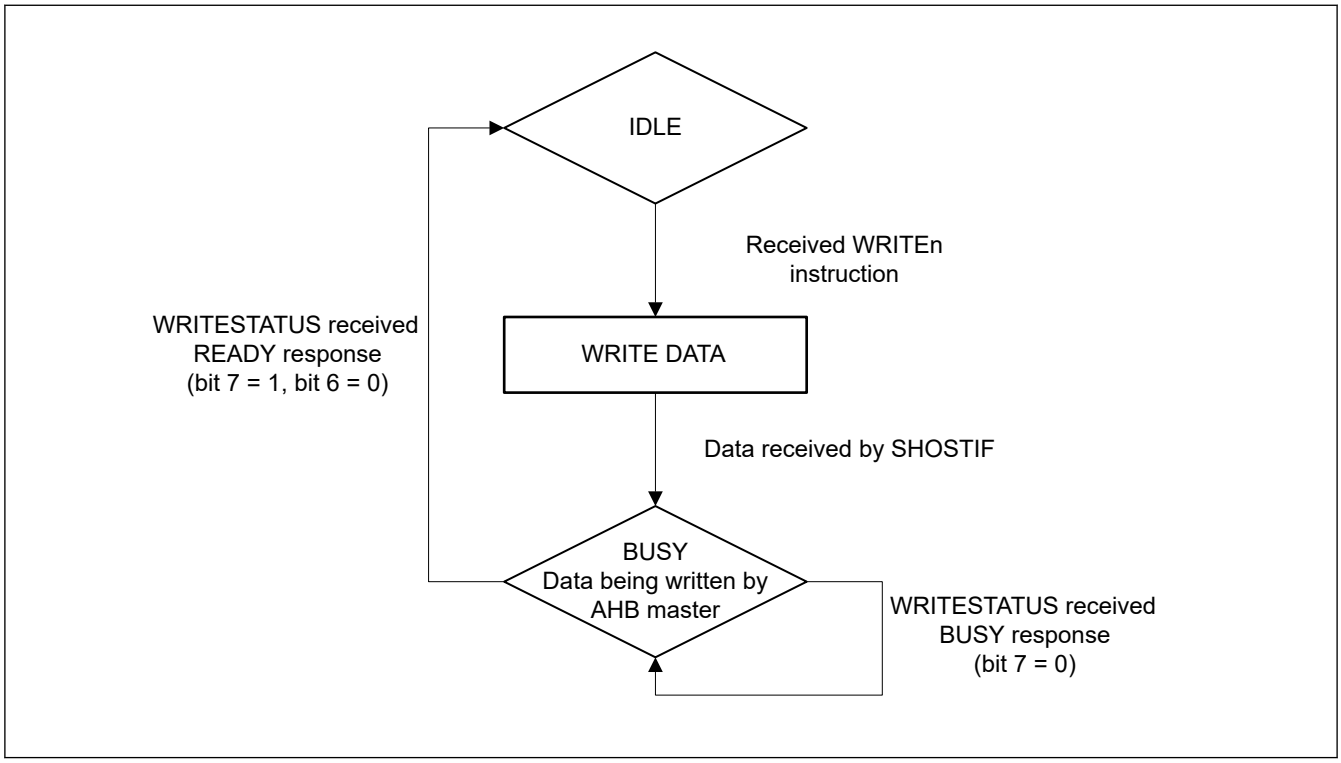


Figure 46.2 Write operation flow

(1) WRITEn (Write Data Instruction)

Data write transaction is always performed in the format specified by SPIMODE[1:0] bits in the SHCFG register. The instruction, data, and address phases are sent in single, dual, quad, or octal format. Write operation from the AHB master interface to the internal slave module starts after the data of length specified in the WRITEn instruction is received in the receive FIFO.

Figure 46.3 shows the write data operation using the WRITEn instruction.

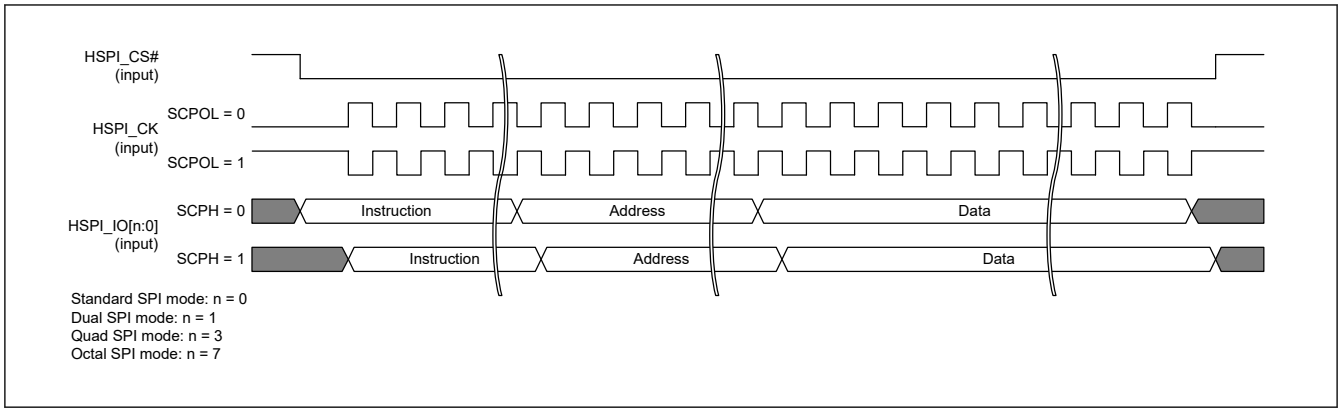


Figure 46.3 WRITEn instruction

(2) WRITESTATUSm (Write Status Instruction)

After the write operation is completed by the master, it should poll for status of the write operation using instruction opcode 0xC0 or 0xC8. Only when the master receives a READY response, it should start a new transfer.

Figure 46.4 shows the write status operation using the WRITESTATUSm instruction.

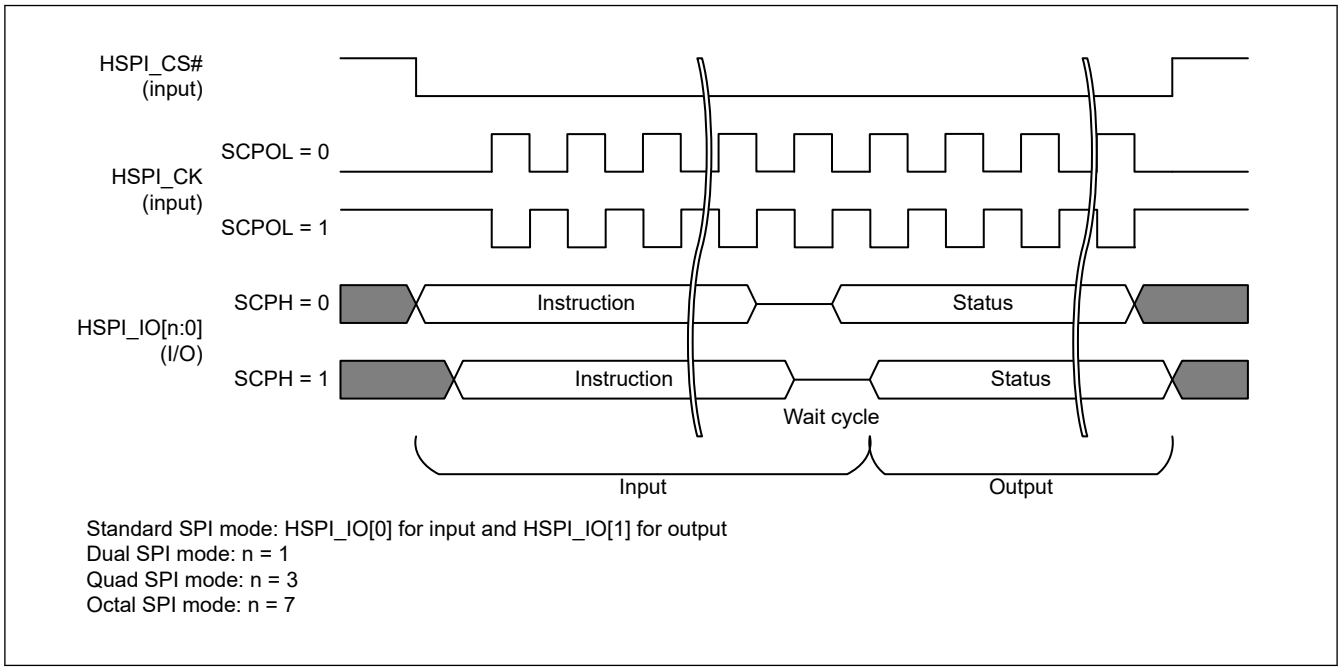


Figure 46.4 WRITESTATUSm instruction (in case of m = 0)

After the SPI master receives a READY response from SHOSTIF, it can move on to the next operation. The AHB interface may get an ERROR response for the write transfer. If the AHB interface gets an ERROR response, the WRITESTATUSm instruction indicates the same by sending a response of 0xC0. The SPI master can then issue a new write transfer or retry the same operation.

Table 46.9 Response field decode

Bit	Name	Description
7	READY	Indicates whether next operation is ready to accept or not. 0: Not ready 1: Ready
6	ERROR	Indicates whether AHB transaction ended with an ERROR response. 0: No error 1: Error
5:0	Reserved	Reserved

46.4.6 Read Operation

46.4.6.1 Read Instruction

In enhanced SPI (Dual/Quad/Octal) modes, all signals are bidirectional. Therefore, the master and slave know when to sample or drive the data on the line. The read instruction set defines the protocol under which the read transactions occur. There are two ways in which the SPI master can read data through SHOSTIF.

(1) Direct Read

In the Direct Read Instruction method, SPI Master initiates a read command and expects the data to be returned in the same command. This way the data is read in the same instruction. Be sure that the data is read on the AHB master interface within the specified cycles (DUMMY cycles).

(2) Read Data in Three Stages (this requires polling)

In some cases, the delay involved during the read operation is such that the read data availability cannot be guaranteed in terms of HSPI_CK cycles due to bus access conflict. To make sure that the data is read in the expected way during the read operation, SHOSTIF provides an alternate approach to divide the read operation into three parts:

- Read request phase
- Wait state
- Read data phase

SPI master requests the read data from SPI slave by sending only the instruction and address during the read request phase. The length of data (1, 4, 8, 16, 32, or 64) to be fetched is also encoded into the instruction. This is converted into an AHB transfer, and data is fetched and stored in the transmit FIFO. After the Read request phase, SPI master checks for availability of data by polling through the READSTATUS instruction. If SPI master receives a positive response for that instruction, then it reads the data using the READn instruction. SHOSTIF then sends all the data received from transmit FIFO to SPI master. Table 46.10 shows all the instructions applicable for read operation.

Table 46.10 Instruction set for read operations

Method	Instruction	Instruction name	Description
Direct Read	0x6i*1 i = 0 to 5	READ8n n = 1, 4, 8, 16, 32, 64	Read Data instruction with 8 wait cycles.
	0x6i*1 i = 8, 9, A, B, C, D	READ16n n = 1, 4, 8, 16, 32, 64	Read Data instruction with 16 wait cycles.
	0x7i*1 i = 0 to 5	READ24n n = 1, 4, 8, 16, 32, 64	Read Data instruction with 24 wait cycles.
	0x7i*1 i = 8, 9, A, B, C, D	READ32n n = 1, 4, 8, 16, 32, 64	Read Data instruction with 32 wait cycles.
3-Stage Read	0x2i*1 i = 0 to 5	READREQn n = 1, 4, 8, 16, 32, 64	Read request.
	0x40	READSTATUS0	Read Data ready instruction without wait cycles.
	0x48	READSTATUS1	Read Data ready instruction with 1 wait cycle.
	0x0i*1 i = 0 to 5	READ0n n = 1, 4, 8, 16, 32, 64	Read Data instruction without wait cycles.
	0x0i*1 i = 8, 9, A, B, C, D	READ1n n = 1, 4, 8, 16, 32, 64	Read Data instruction with 1 wait cycle.

Note 1. The lower 3 bits of the instruction determine the amount of data to be fetched by AHB master. For field decode, see Table 46.5.

46.4.6.2 Direct Read Operation

(1) READmn (Direct Read Data Instruction, m = 8, 16, 24, or 32)

In the direct read approach, only a single instruction is used to fetch the data using AHB interface. After SHOSTIF receives a command with more than or equal to 8 wait cycles (see Table 46.10 for details), SHOSTIF initiates a command on

AHB interface to fetch the required amount of data. After data is received from AHB interface, it is transmitted to the SPI interface after the required amount of wait cycles. Figure 46.5 shows the waveform of the direct read transfer.

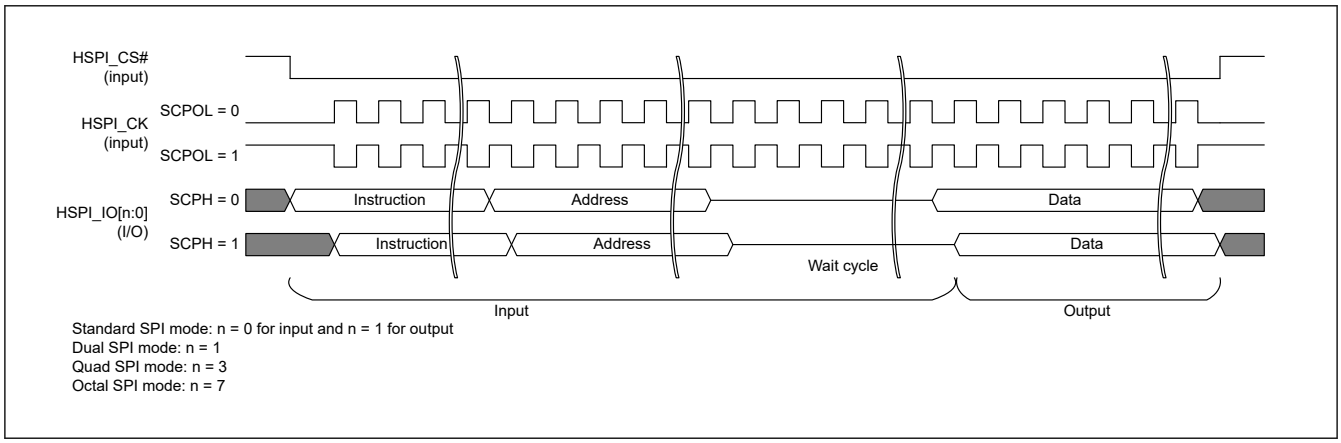


Figure 46.5 READmn instruction (m = 8, 16, 24, or 32)

If SHOSTIF is unable to fetch the data during specific wait cycles, then SHOSTIF indicates this by setting TXE bit in the Status Register (SR) and the Transmit FIFO Underflow Interrupt is asserted.

46.4.6.3 Three Stage Read Operation

When SHOSTIF is idle (the receive/transmit FIFO is empty), and a READREQn instruction is received, it is treated as a read request, and it starts fetching the data from AHB master interface. After the data is available, the READmn instruction returns the read data to SPI master. The SPI master ensures that the data is available before issuing another READmn instruction by polling for data availability. The AHB interface may receive an ERROR response for the read transfer and the READSTATUS instruction indicates this in the response while the SPI master performs the required actions.

Figure 46.6 shows the flow for read operation.

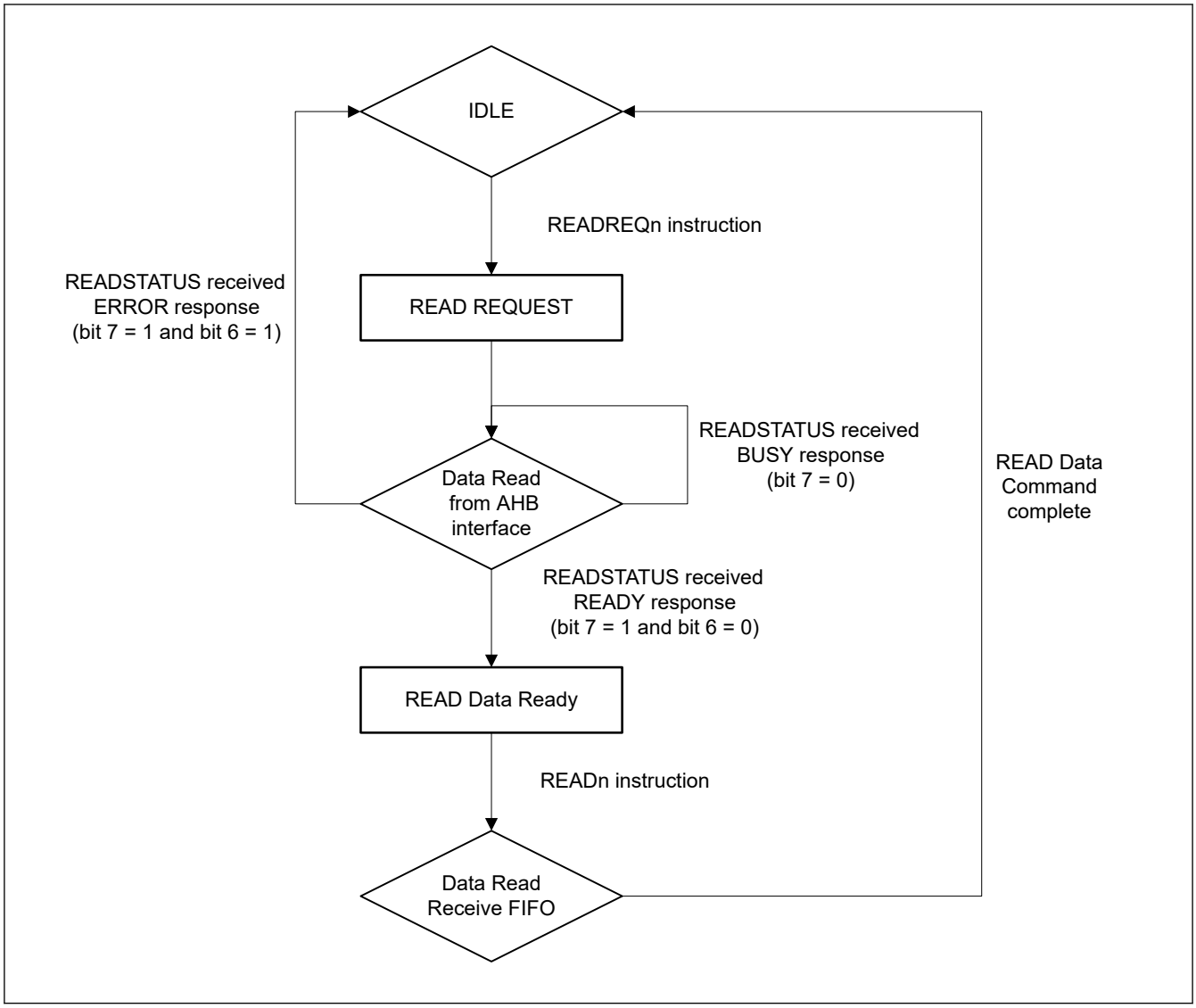


Figure 46.6 Read operation flow for 3-stage operation

(1) READREQn (Read Request Instruction)

Read transaction is always performed in the format specified by SPIMODE[1:0] bits in SHCFG register. This instruction does not require any data transfer and SHOSTIF expects only instruction and address.

Figure 46.7 shows the read request operation using READREQn instruction.

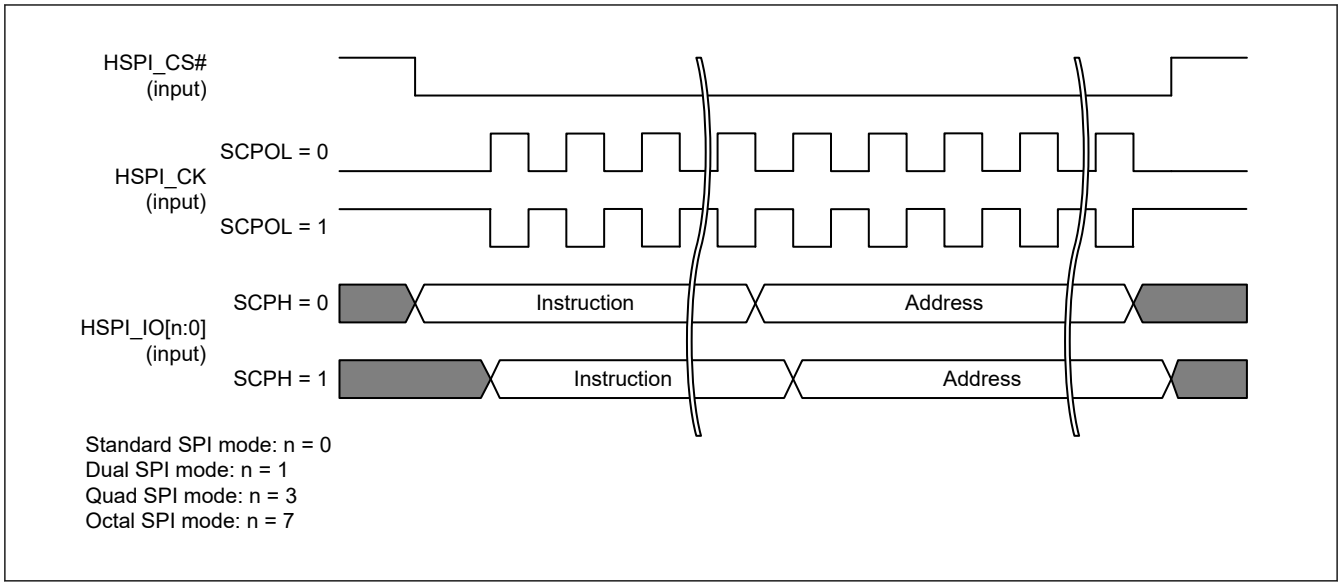


Figure 46.7 READREQn instruction

(2) READSTATUSm (Read Status Instruction, m = 0 or 1)

Read status instruction is used while SHOSTIF fetches the data from AHB interface. The SPI master always polls for data before performing a read data operation, to ensure that data is available for the read operation. SHOSTIF responds to the READSTATUSm command with 8-bit status. The MSB of the status bit indicates whether data is ready or not.

- Response bit 7 = 0 indicates that SHOSTIF is not ready with the data
- Response bit 7 = 1 and bit 6 = 0 indicates that SHOSTIF is ready with the data

Figure 46.8 shows the read status operation using READSTATUSm instruction.

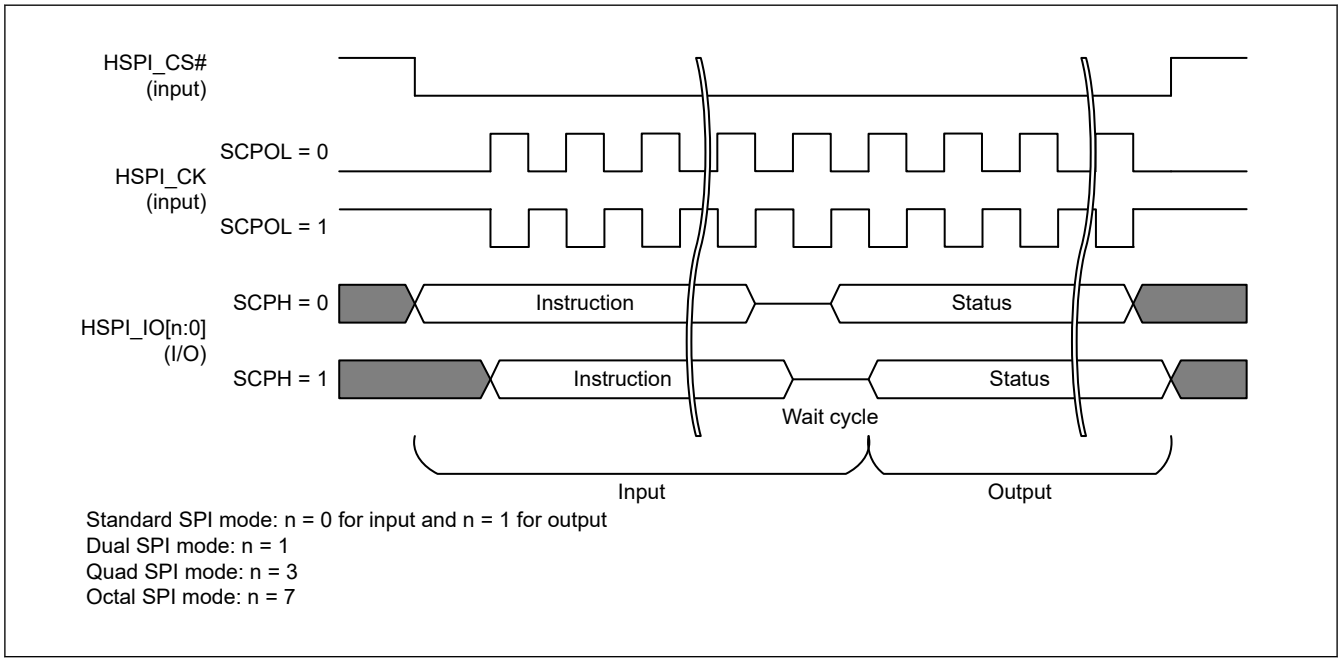


Figure 46.8 READSTATUSm instruction (in case of m = 1)

After the SPI master receives a READY response from SHOSTIF, it performs the read data operation to read the data from SHOSTIF. The AHB interface may get an ERROR response for the read transfer. In that situation, the READSTATUS instruction indicates the same status by sending a response of bit 7 = 1 and bit 6 = 1. SPI master can then issue a new read transfer or retry the same operation.

Table 46.11 Response field decode

Bit	Name	Description
7	READY	Indicates whether data is ready to be fetched from SHOSTIF. 0: Not ready 1: Ready
6	ERROR	Indicates whether AHB transaction ended with an ERROR response or not. 0: No error 1: Error
5:0	Reserved	Reserved

(3) READmn (Read Data Instruction, m = 0 or 1)

After the data is available, READmn instruction is used to read back data from SHOSTIF. This time SHOSTIF does not expect any address to be transmitted by the SPI master. After the instruction is received, the inputs/outputs change direction and revert the data after some wait cycles. Wait cycles can be sent by SHOSTIF depending on the instruction (Table 46.5).

Figure 46.9 shows the read data operation using READmn instruction.

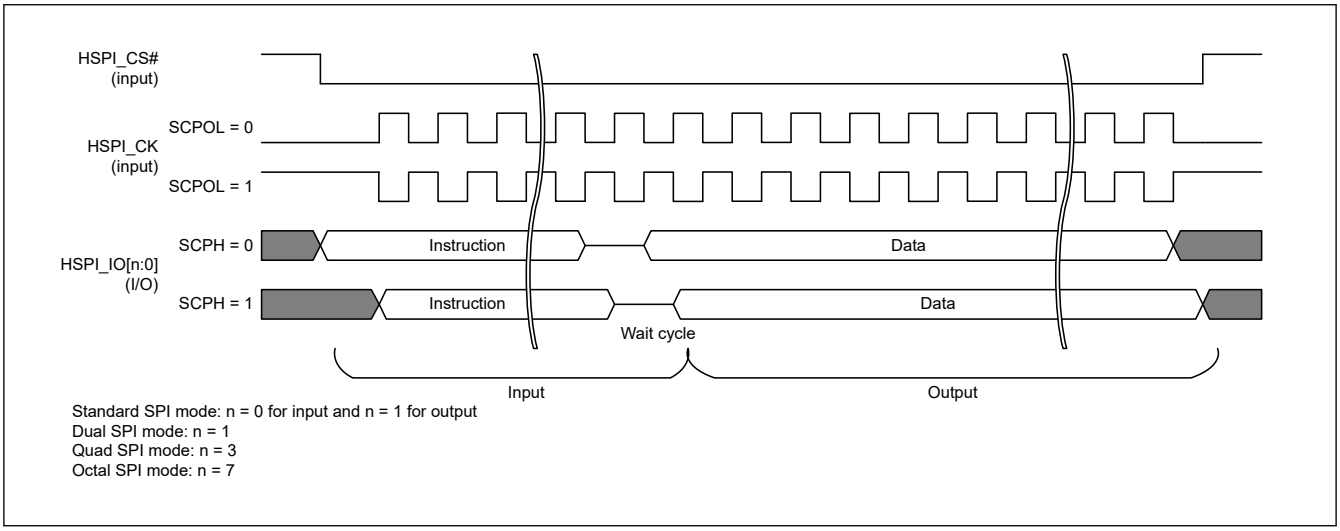


Figure 46.9 READmn instruction (in case of m = 1)

46.4.7 External SPI Master Expectations

The external SPI master must adhere to the flow specified for write and read operations. Any out-of-order operations by SPI master can move SHOSTIF into an undefined state. If SPI master behaves abnormally, then interrupt is sent by SHOSTIF. In such situations, reset SHOSTIF and restart the new transfer.

The following conditions may lead to an error state and the SPIMER bit in RISR register is set for notification.

- SPI master requests for read/write transfer when SHOSTIF is busy with AHB transfer. For example:
  - A read is requested by the SPI master and is followed by a write data transfer while read transfer is still ongoing on the AHB interface.
  - A write transfer is pending on the AHB interface but another read/write data transfer is received by SHOSTIF.
- SPI master reads less amount of data then requested during the Read request phase.
- If the SPI master requested for 16 data frames but it reads only 12 data frames, then the SPIMER bit in RISR register is set and the rest of the data is flushed from the FIFO by performing a soft reset.
- SPI master writes less amount of data than specified in the transfer length.
- If the SPI master requested for 16 data frames but it writes only 12 data frames and deselects the slave early, then the SPIMER bit in the RISR register is set and any of the data left in the FIFO is flushed by performing a soft reset.
- SPI master sends more data for a write, or SPI master read more data then requested.

- If AHB read transfer receives an error response from the slave, then no further transfer is performed for the current request, and data in the FIFO is flushed immediately. If read request transfer receives an error response on AHB interface, SHOSTIF does not expect read data transfer after that because there is no read data available in FIFO. If AHB master still performs a read data transfer, this results in an SPI master error.
- Successful read request transfer is followed by another read request from SPI master.
- A successful read request transfer (which does not receive an error response on the AHB interface) should be followed by read data or read status transfer. If a successful read request transfer is followed by another read request from SPI master, this results in an SPI master error.

46.4.8 AHB Master Interface

The AHB master interface is used as a secondary interface to read or write data for every SPI transfer. The AHB interface address and data width is fixed to 32. The following sections describe how SPI transfers are converted to AHB transactions by SHOSTIF.

46.4.8.1 Write Operation

When SHOSTIF receives a write request (WRITEn command) from SPI master, SHOSTIF saves the address per the rules specified in section 46.4.2. Address Phase. After data is received, it is transmitted as a burst transfer defined by RXFBTR register on AHB interface. This continues until the AHB slave is deselected by the AHB master. If AHB write transfer receives and ERROR response, it is indicated by ahbe_intr interrupt of the ISR and RISR registers. As soon as the error is received on the AHB interface, the rest of the data from SPI interface is discarded and no further transfers are performed for the current operation.

46.4.8.2 Read Operation

When SHOSTIF receives a read request (READmn command) from SPI master, SHOSTIF saves the address per the rules specified in section 46.4.2. Address Phase. SHOSTIF then waits for a burst to complete, and the data to be fully received. The data length is specified in the Data length field of the instruction. For data length of 1, 4, 8, or 16, the corresponding AHB burst type is used to fetch the data. For the burst length of 32 or more, INCR burst type is used to fetch the data.

If any read transfer receives an ERROR response, SHOSTIF aborts the rest of the transfer and indicates the same status in the ISR and RISR registers. When the SPI interface performs a READSTATUS instruction for data ready status, this is indicated to the SPI master with an error response of bit 6 = 1.

46.4.9 Interrupt

Interrupt signal is available in the internal CPU (Cortex-A55 and Cortex-R52) and external host CPU (SPI master). Multiple individual interrupt factors are combined to one signal for internal CPU and external host CPU. Individual interrupt factor can be masked commonly and separately. Figure 46.10 shows a diagram of the interrupt.

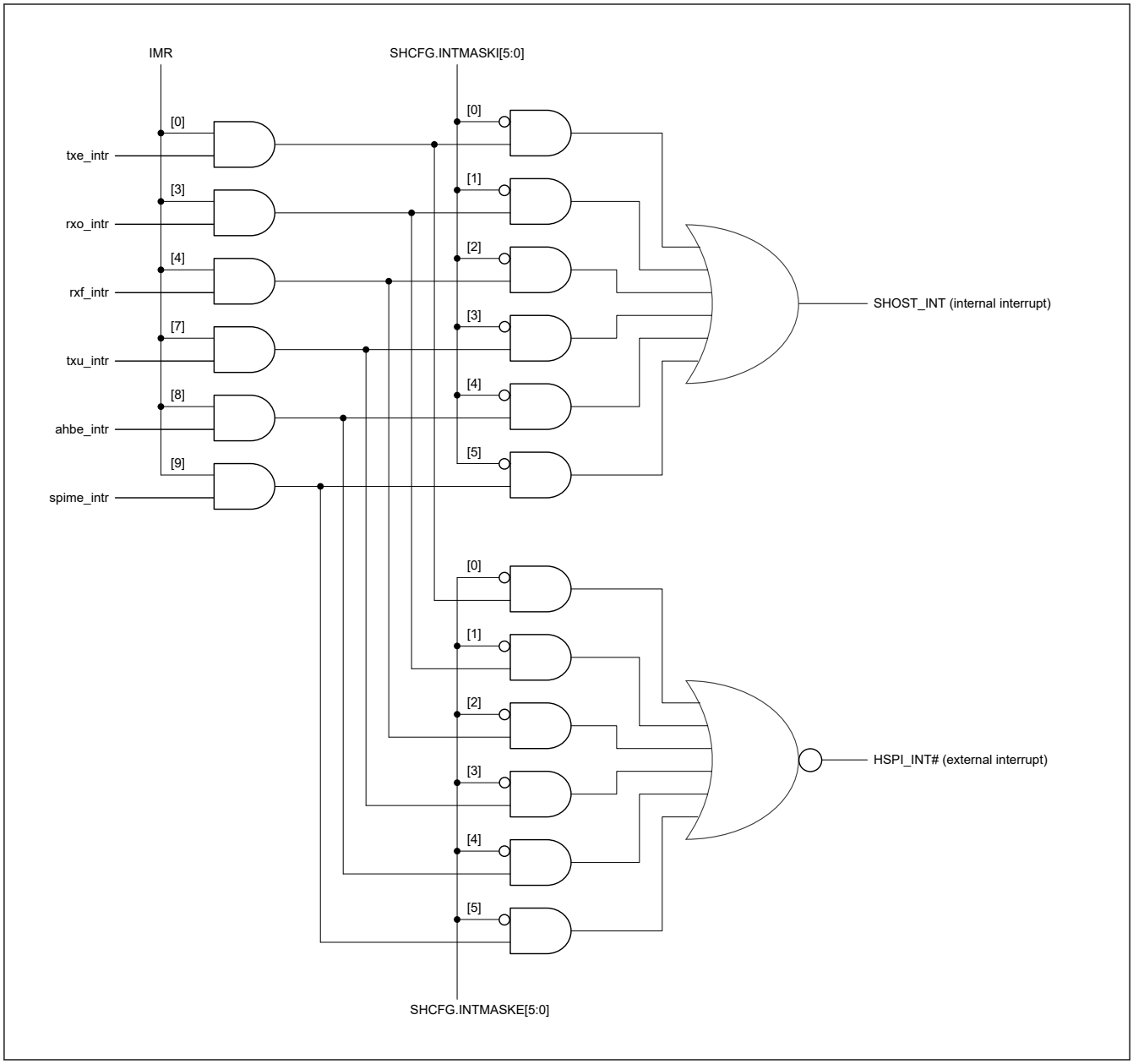


Figure 46.10 Diagram of interrupt signal

Table 46.12 SHOSTIF interrupt source (internal)

Name	Interrupt sources
SHOST_INT	SHOSTIF interrupt (logical OR of all individual interrupts)

Table 46.13 SHOSTIF individual interrupt factor (1 of 2)

Factor name	Description	Interrupt clear	Interrupt mask setting		
			Common	Internal CPU	External host CPU
			IMR register	SHCFG register	
				INTMASKI[5:0] bits	INTMASKE[5:0] bits
txe_intr	Transmit FIFO (read data FIFO) Empty interrupt. When the number of transmit FIFO entries is less than or equal to transmit FIFO threshold level, the interrupt is triggered.	Transmit FIFO entries is greater than threshold	b0	b0	b0

Table 46.13 SHOSTIF individual interrupt factor (2 of 2)

Factor name	Description	Interrupt clear	Interrupt mask setting		
			Common	Internal CPU	External host CPU
			IMR register	SHCFG register	
				INTMASKI[5:0] bits	INTMASKE[5:0] bits
rxo_intr	Receive FIFO (write data FIFO) Overflow interrupt. This interrupt is not triggered if module operates normally.	RXOICR or ICR register read	b3	b1	b1
rxf_intr	Receive FIFO (write data FIFO) Full interrupt. When the number of receive FIFO entries is greater than or equal to receive FIFO threshold level + 1, the interrupt is triggered.	Receive FIFO entries is less than threshold	b4	b2	b2
txu_intr	Transmit FIFO (read data FIFO) Underflow interrupt. The interrupt is triggered in any of the following conditions: ● Data length in READmn is greater than that in READREQn in the 3-stage approach ● Transmission of AHB master interface is not completed within wait cycles specified in READmn in direct read approach	TXUICR or ICR register read	b7	b3	b3
ahbe_intr	AHB Master Interface Error interrupt. When AHB master interface receives an error response, the interrupt is triggered.	AHBECR or ICR register read	b8	b4	b4
spime_intr	SPI Master Interface Error interrupt. The interrupt is triggered in any of the following conditions: ● Illegal instruction code ● Next read or write transaction is issued before the previous transfer is completed	SPEMECR or ICR register read	b9	b5	b5

Note: If Transmit FIFO Underflow interrupt (txu_intr) happens, read access from external bus master may be impossible. Set ENR.ENABLE to 0 once.

46.5 Usage Notes

- When changing the SPI data width, disable SHOSTIF first by setting ENR.ENABLE = 0 during no transmission. Then, change SHCFG.SPIMODE. After changing the SPI data width, enable SHOSTIF again and start transmission.
- During ENR.ENABLE = 0, register access from internal CPU is available but access from external SPI master is ignored. No interrupts occur and read data is undefined.
- Do not change Receive FIFO Burst Threshold Register (RXFBTR) from 0x3F (initial value) in case of 8-bit addressing mode. In case of 32-bit addressing mode, recommendation value is 0x0F.

47. Mailbox and Semaphore (MBXSEM)

47.1 Overview

Mailbox and semaphore are available for controlling exclusive access between internal CPU (Cortex-A55/Cortex-R52) and external host CPU. Table 47.1 shows specification of mailbox and semaphore registers. These registers are accessible from external host CPU via SHOSTIF as well as internal CPU.

Additionally, mailbox and semaphore are available for controlling exclusive access among internal CPUs (Cortex-A55, Cortex-R52 CPU0 and CPU1). Table 1.2 shows specification of mailbox and semaphore registers among internal CPUs.

Table 47.1 Specification of MBXSEM for Host Interface

Semaphore/ Mailbox	Function	Bit size	Number of register	Interrupt generation		Access property*5		
				Cortex-A55/ Cortex-R52	Host CPU	Cortex-A55/ Cortex-R52	Host CPU	Access size
Semaphore	Semaphore	1	8	No	No	R(C)/W	R(C)/W	8, 16, 32
	Read clear enable*1	8	1	No	No	R/W	R/W	8, 16, 32
Mailbox (External Host CPU to Cortex-A55/Cortex-R52)	Mailbox	32	4	No	No	R	R/W	8, 16, 32
	Interrupt Set*2 *3	4	1	Yes (each bit)	No	R	R/W	8, 16, 32
	Interrupt Clear*2 *4	4	1	Cleared (each bit)	No	R/W	R	8, 16, 32
Mailbox (Cortex-A55/ Cortex-R52 to External Host CPU)	Mailbox	32	4	No	No	R/W	R	8, 16, 32
	Interrupt Set*2 *3	4	1	No	Yes (combined)	R/W	R	8, 16, 32
	Interrupt Clear*2 *4	4	1	No	Cleared (when all bits become 0)	R	R/W	8, 16, 32

Note 1. Each bit corresponds to each semaphore register.
Note 2. Each bit represents the interrupt status. When read, same interrupt status can be read through interrupt set and clear registers.
Note 3. When 1 is written to a bit, the corresponding bit is set.
Note 4. When 1 is written to a bit, the corresponding bit is cleared.
Note 5. CoreSight AXI-AP can also access these registers. Access property is same as Cortex-A55/Cortex-R52.

Table 47.2 Specification of MBXSEM among internal CPUs (1 of 2)

Semaphore/ Mailbox	Function	Bit size	Number of register	Interrupt generation		Access property*8 *9		
				Target	Source	Target	Source	Access size
Semaphore	semaphore	1	8	No	No	R(C)/W	R(C)/W	8, 16, 32
	Read clear enable*1	8	1	No	No	R/W	R/W	8, 16, 32
Mailbox Source: Cortex-R52 CPUx Target: Cortex-A55 Corey (x = 0, 1, y = 0 to 3)	Mailbox	32	4 (total 32)	No	No	R	R/W	8, 16, 32
	Interrupt Set*2 *3	4	1 (total 8)	Yes (each reg*5)	No	R	R/W	8, 16, 32
	Interrupt Clear*2 *4	4	1 (total 8)	Cleared (each reg*6)	No	R/W	R	8, 16, 32
Mailbox Source: Cortex-A55 Corey Target: Cortex-R52 CPUx (x = 0, 1, y = 0 to 3)	Mailbox	32	4 (total 32)	No	No	R	R/W	8, 16, 32
	Interrupt Set*2 *3	4	1 (total 8)	Yes (each 4 regs*7)	No	R	R/W	8, 16, 32
	Interrupt Clear*2 *4	16	1 (total 2)	Cleared (each reg*6)	No	R/W	R	8, 16, 32

Table 47.2 Specification of MBXSEM among internal CPUs (2 of 2)

Semaphore/ Mailbox	Function	Bit size	Number of register	Interrupt generation		Access property*8 *9		
				Target	Source	Target	Source	Access size
Mailbox Source: Cortex-R52 CPUx Target: Cortex-R52 CPU|x-1| (x = 0, 1)	Mailbox	32	4 (total 8)	No	No	R	R/W	8, 16, 32
	Interrupt Set*2 *3	4	1 (total 2)	Yes (each reg*5)	No	R	R/W	8, 16, 32
	Interrupt Clear*2 *4	4	1 (total 2)	Cleared (each reg*6)	No	R/W	R	8, 16, 32
Mailbox Source: Cortex-A55 Corex Target: Cortex-A55 Corey (x = 0 to 3, y = 0 to 3 and y ≠ x)	Mailbox	32	4 (total 48)	No	No	R	R/W	8, 16, 32
	Interrupt Set*2 *3	4	1 (total 12)	Yes (each reg*5)	No	R	R/W	8, 16, 32
	Interrupt Clear*2 *4	4	1 (total 12)	Cleared (each reg*6)	No	R/W	R	8, 16, 32

Note 1. Each bit corresponds to each semaphore register.
Note 2. Each bit represents the interrupt status. When read, same interrupt status can be read through send and clear interrupt registers.
Note 3. When 1 is written to a bit, the corresponding bit is set.
Note 4. When 1 is written to a bit, the corresponding bit is cleared.
Note 5. Interrupt is generated when setting more than one bit each corresponding register.
Note 6. Interrupt is cleared when all bits become 0 each corresponding register.
Note 7. Interrupt is generated when setting more than one bit each corresponding 4 registers for Cortex-A55 Core0 to Core3.
Note 8. Access from other CPUs than source and target is always read-only.
Note 9. CoreSight AXI-AP can also access these registers. Access property is same as Cortex-A55 and Cortex-R52 for semaphore and read-only for mailbox.

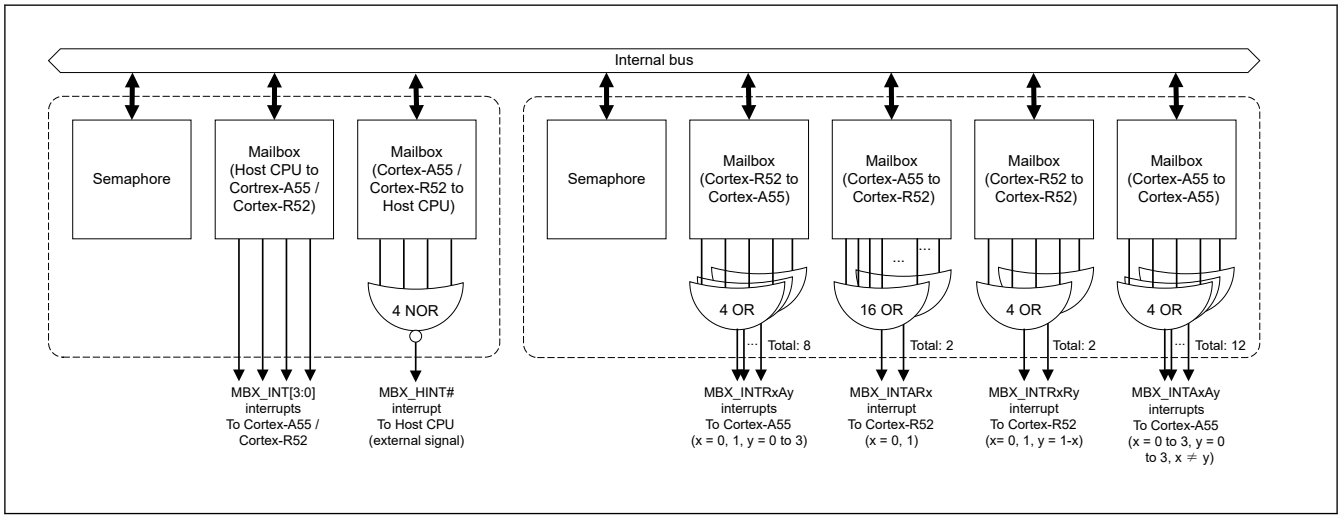


Figure 47.1 MBXSEM Block Diagram

Table 47.3 lists the output pin of the MBXSEM.

Table 47.3 MBXSEM I/O pin

Name	I/O	Function
MBX_HINT#	Output	Mailbox (Cortex-A55/Cortex-R52 to Host CPU) interrupt output pin

47.2 Register Map

Table 47.4 MBXSEM register map (1 of 4)

Address	Register symbol	Register name	Write protection
0x8024_0000 + 0x04 × n	SEMn	Semaphore Register n (n = 0 to 7)	—

Table 47.4 MBXSEM register map (2 of 4)

Address	Register symbol	Register name	Write protection
0x8024_0020	SEMRCEN	Semaphore Read Clear Enable Register	—
0x8024_0080 + 0x04 × n	MBXH2Cn	Host to CA55/CR52 Mailbox Register n (n = 0 to 3)	—
0x8024_0090	MBXISETH2C	Host to CA55/CR52 Mailbox Interrupt Set Register	—
0x8024_0094	MBXICLRH2C	Host to CA55/CR52 Mailbox Interrupt Clear Register	—
0x8024_0100 + 0x04 × n	MBXC2Hn	CA55/CR52 to Host Mailbox Register n (n = 0 to 3)	—
0x8024_0110	MBXISETC2H	CA55/CR52 to Host Mailbox Interrupt Set Register	—
0x8024_0114	MBXICLRC2H	CA55/CR52 to Host Mailbox Interrupt Clear Register	—
0x8024_0180 + 0x04 × n	SEMARn	Semaphore Register n Among Internal CPUs (n = 0 to 7)	—
0x8024_01A0	SEMRCENAR	Semaphore Read Clear Enable Register Among Internal CPUs	—
0x8024_0200 + 0x04 × n	MBXR0A0n	CR52 CPU0 to CA55 Core0 Mailbox Register n (n = 0 to 3)	—
0x8024_0210	MBXISETR0A0	CR52 CPU0 to CA55 Core0 Mailbox Interrupt Set Register	—
0x8024_0214	MBXICLRR0A0	CR52 CPU0 to CA55 Core0 Mailbox Interrupt Clear Register	—
0x8024_0280 + 0x04 × n	MBXR0A1n	CR52 CPU0 to CA55 Core1 Mailbox Register n (n = 0 to 3)	—
0x8024_0290	MBXISETR0A1	CR52 CPU0 to CA55 Core1 Mailbox Interrupt Set Register	—
0x8024_0294	MBXICLRR0A1	CR52 CPU0 to CA55 Core1 Mailbox Interrupt Clear Register	—
0x8024_0300 + 0x04 × n	MBXR0A2n	CR52 CPU0 to CA55 Core2 Mailbox Register n (n = 0 to 3)	—
0x8024_0310	MBXISETR0A2	CR52 CPU0 to CA55 Core2 Mailbox Interrupt Set Register	—
0x8024_0314	MBXICLRR0A2	CR52 CPU0 to CA55 Core2 Mailbox Interrupt Clear Register	—
0x8024_0380 + 0x04 × n	MBXR0A3n	CR52 CPU0 to CA55 Core3 Mailbox Register n (n = 0 to 3)	—
0x8024_0390	MBXISETR0A3	CR52 CPU0 to CA55 Core3 Mailbox Interrupt Set Register	—
0x8024_0394	MBXICLRR0A3	CR52 CPU0 to CA55 Core3 Mailbox Interrupt Clear Register	—
0x8024_0400 + 0x04 × n	MBXR1A0n	CR52 CPU1 to CA55 Core0 Mailbox Register n (n = 0 to 3)	—
0x8024_0410	MBXISETR1A0	CR52 CPU1 to CA55 Core0 Mailbox Interrupt Set Register	—
0x8024_0414	MBXICLRR1A0	CR52 CPU1 to CA55 Core0 Mailbox Interrupt Clear Register	—
0x8024_0480 + 0x04 × n	MBXR1A1n	CR52 CPU1 to CA55 Core1 Mailbox Register n (n = 0 to 3)	—
0x8024_0490	MBXISETR1A1	CR52 CPU1 to CA55 Core1 Mailbox Interrupt Set Register	—
0x8024_0494	MBXICLRR1A1	CR52 CPU1 to CA55 Core1 Mailbox Interrupt Clear Register	—
0x8024_0500 + 0x04 × n	MBXR1A2n	CR52 CPU1 to CA55 Core2 Mailbox Register n (n = 0 to 3)	—
0x8024_0510	MBXISETR1A2	CR52 CPU1 to CA55 Core2 Mailbox Interrupt Set Register	—
0x8024_0514	MBXICLRR1A2	CR52 CPU1 to CA55 Core2 Mailbox Interrupt Clear Register	—
0x8024_0580 + 0x04 × n	MBXR1A3n	CR52 CPU1 to CA55 Core3 Mailbox Register n (n = 0 to 3)	—
0x8024_0590	MBXISETR1A3	CR52 CPU1 to CA55 Core3 Mailbox Interrupt Set Register	—
0x8024_0594	MBXICLRR1A3	CR52 CPU1 to CA55 Core3 Mailbox Interrupt Clear Register	—
0x8024_0600 + 0x04 × n	MBXA0R0n	CA55 Core0 to CR52 CPU0 Mailbox Register n (n = 0 to 3)	—
0x8024_0610 + 0x04 × n	MBXA1R0n	CA55 Core1 to CR52 CPU0 Mailbox Register n (n = 0 to 3)	—
0x8024_0620 + 0x04 × n	MBXA2R0n	CA55 Core2 to CR52 CPU0 Mailbox Register n (n = 0 to 3)	—
0x8024_0630 + 0x04 × n	MBXA3R0n	CA55 Core3 to CR52 CPU0 Mailbox Register n (n = 0 to 3)	—
0x8024_0640	MBXISETA0R0	CA55 Core0 to CR52 CPU0 Mailbox Interrupt Set Register	—
0x8024_0644	MBXISETA1R0	CA55 Core1 to CR52 CPU0 Mailbox Interrupt Set Register	—
0x8024_0648	MBXISETA2R0	CA55 Core2 to CR52 CPU0 Mailbox Interrupt Set Register	—
0x8024_064C	MBXISETA3R0	CA55 Core3 to CR52 CPU0 Mailbox Interrupt Set Register	—

Table 47.4 MBXSEM register map (3 of 4)

Address	Register symbol	Register name	Write protection
0x8024_0650	MBXICLRAR0	CA55 to CR52 CPU0 Mailbox Interrupt Clear Register	—
0x8024_0680 + 0x04 × n	MBXA0R1n	CA55 Core0 to CR52 CPU1 Mailbox Register n (n = 0 to 3)	—
0x8024_0690 + 0x04 × n	MBXA1R1n	CA55 Core1 to CR52 CPU1 Mailbox Register n (n = 0 to 3)	—
0x8024_06A0 + 0x04 × n	MBXA2R1n	CA55 Core2 to CR52 CPU1 Mailbox Register n (n = 0 to 3)	—
0x8024_06B0 + 0x04 × n	MBXA3R1n	CA55 Core3 to CR52 CPU1 Mailbox Register n (n = 0 to 3)	—
0x8024_06C0	MBXISETA0R1	CA55 Core0 to CR52 CPU1 Mailbox Interrupt Set Register	—
0x8024_06C4	MBXISETA1R1	CA55 Core1 to CR52 CPU1 Mailbox Interrupt Set Register	—
0x8024_06C8	MBXISETA2R1	CA55 Core2 to CR52 CPU1 Mailbox Interrupt Set Register	—
0x8024_06CC	MBXISETA3R1	CA55 Core3 to CR52 CPU1 Mailbox Interrupt Set Register	—
0x8024_06D0	MBXICLRAR1	CA55 to CR52 CPU1 Mailbox Interrupt Clear Register	—
0x8024_0700 + 0x04 × n	MBXR0R1n	CR52 CPU0 to CR52 CPU1 Mailbox Register n (n = 0 to 3)	—
0x8024_0710	MBXISETR0R1	CR52 CPU0 to CR52 CPU1 Mailbox Interrupt Set Register	—
0x8024_0714	MBXICLRR0R1	CR52 CPU0 to CR52 CPU1 Mailbox Interrupt Clear Register	—
0x8024_0780 + 0x04 × n	MBXR1R0n	CR52 CPU1 to CR52 CPU0 Mailbox Register n (n = 0 to 3)	—
0x8024_0790	MBXISETR1R0	CR52 CPU1 to CR52 CPU0 Mailbox Interrupt Set Register	—
0x8024_0794	MBXICLRR1R0	CR52 CPU1 to CR52 CPU0 Mailbox Interrupt Clear Register	—
0x8024_0800 + 0x04 × n	MBXA0A1n	CA55 Core0 to CA55 Core1 Mailbox Register n (n = 0 to 3)	—
0x8024_0810	MBXISETA0A1	CA55 Core0 to CA55 Core1 Mailbox Interrupt Set Register	—
0x8024_0814	MBXICLRA0A1	CA55 Core0 to CA55 Core1 Mailbox Interrupt Clear Register	—
0x8024_0880 + 0x04 × n	MBXA0A2n	CA55 Core0 to CA55 Core2 Mailbox Register n (n = 0 to 3)	—
0x8024_0890	MBXISETA0A2	CA55 Core0 to CA55 Core2 Mailbox Interrupt Set Register	—
0x8024_0894	MBXICLRA0A2	CA55 Core0 to CA55 Core2 Mailbox Interrupt Clear Register	—
0x8024_0900 + 0x04 × n	MBXA0A3n	CA55 Core0 to CA55 Core3 Mailbox Register n (n = 0 to 3)	—
0x8024_0910	MBXISETA0A3	CA55 Core0 to CA55 Core3 Mailbox Interrupt Set Register	—
0x8024_0914	MBXICLRA0A3	CA55 Core0 to CA55 Core3 Mailbox Interrupt Clear Register	—
0x8024_0980 + 0x04 × n	MBXA1A0n	CA55 Core1 to CA55 Core0 Mailbox Register n (n = 0 to 3)	—
0x8024_0990	MBXISETA1A0	CA55 Core1 to CA55 Core0 Mailbox Interrupt Set Register	—
0x8024_0994	MBXICLRA1A0	CA55 Core1 to CA55 Core0 Mailbox Interrupt Clear Register	—
0x8024_0A00 + 0x04 × n	MBXA1A2n	CA55 Core1 to CA55 Core2 Mailbox Register n (n = 0 to 3)	—
0x8024_0A10	MBXISETA1A2	CA55 Core1 to CA55 Core2 Mailbox Interrupt Set Register	—
0x8024_0A14	MBXICLRA1A2	CA55 Core1 to CA55 Core2 Mailbox Interrupt Clear Register	—
0x8024_0A80 + 0x04 × n	MBXA1A3n	CA55 Core1 to CA55 Core3 Mailbox Register n (n = 0 to 3)	—
0x8024_0A90	MBXISETA1A3	CA55 Core1 to CA55 Core3 Mailbox Interrupt Set Register	—
0x8024_0A94	MBXICLRA1A3	CA55 Core1 to CA55 Core3 Mailbox Interrupt Clear Register	—
0x8024_0B00 + 0x04 × n	MBXA2A0n	CA55 Core2 to CA55 Core0 Mailbox Register n (n = 0 to 3)	—
0x8024_0B10	MBXISETA2A0	CA55 Core2 to CA55 Core0 Mailbox Interrupt Set Register	—
0x8024_0B14	MBXICLRA2A0	CA55 Core2 to CA55 Core0 Mailbox Interrupt Clear Register	—
0x8024_0B80 + 0x04 × n	MBXA2A1n	CA55 Core2 to CA55 Core1 Mailbox Register n (n = 0 to 3)	—
0x8024_0B90	MBXISETA2A1	CA55 Core2 to CA55 Core1 Mailbox Interrupt Set Register	—
0x8024_0B94	MBXICLRA2A1	CA55 Core2 to CA55 Core1 Mailbox Interrupt Clear Register	—
0x8024_0C00 + 0x04 × n	MBXA2A3n	CA55 Core2 to CA55 Core3 Mailbox Register n (n = 0 to 3)	—

Table 47.4 MBXSEM register map (4 of 4)

Address	Register symbol	Register name	Write protection
0x8024_0C10	MBXISETA2A3	CA55 Core2 to CA55 Core3 Mailbox Interrupt Set Register	—
0x8024_0C14	MBXICLRA2A3	CA55 Core2 to CA55 Core3 Mailbox Interrupt Clear Register	—
0x8024_0C80 + 0x04 × n	MBXA3A0n	CA55 Core3 to CA55 Core0 Mailbox Register n (n = 0 to 3)	—
0x8024_0C90	MBXISETA3A0	CA55 Core3 to CA55 Core0 Mailbox Interrupt Set Register	—
0x8024_0C94	MBXICLRA3A0	CA55 Core3 to CA55 Core0 Mailbox Interrupt Clear Register	—
0x8024_0D00 + 0x04 × n	MBXA3A1n	CA55 Core3 to CA55 Core1 Mailbox Register n (n = 0 to 3)	—
0x8024_0D10	MBXISETA3A1	CA55 Core3 to CA55 Core1 Mailbox Interrupt Set Register	—
0x8024_0D14	MBXICLRA3A1	CA55 Core3 to CA55 Core1 Mailbox Interrupt Clear Register	—
0x8024_0D80 + 0x04 × n	MBXA3A2n	CA55 Core3 to CA55 Core2 Mailbox Register n (n = 0 to 3)	—
0x8024_0D90	MBXISETA3A2	CA55 Core3 to CA55 Core2 Mailbox Interrupt Set Register	—
0x8024_0D94	MBXICLRA3A2	CA55 Core3 to CA55 Core2 Mailbox Interrupt Clear Register	—

Table 47.5 MBXSEM related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	—	—	SLVACCCTL7.MBXSEM_SL

47.3 Register Descriptions

Register base address is different between external Host CPU view and internal memory address view except.

- External host CPU view : BASE = 0x20_0000

  Note: When 32-bit addressing mode in SHOSTIF is used, please shift the address (base address + offset address) by 2 bits in the right direction.

- Internal CPU view: BASE = 0x8024_0000

Register offset address is described in byte address. Convert it to word address by dividing by 4 if 32-bit addressing mode is used in SHOSTIF.

47.3.1 Semaphore and Mailbox Registers between External Host and Internal CPUs

47.3.1.1 SEMn : Semaphore Register n (n = 0 to 7)

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x00 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SEM	Semaphore bit 0: Resource being used 1: Resource not being used	R/W*1
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When read clear function is disabled (SEMRCEN.SEMRCENn = 0), register value is retained after reading the register. When read clear function is enabled (SEMRCEN.SEMRCENn = 1), register is cleared to 0 after reading the register.

SEMn register is used to perform semaphore for synchronization and exclusive control between internal CPU (Cortex-A55/ Cortex-R52/CoreSight) and external host CPU. Resource status can be controlled when the resource is shared between CPUs. Semaphore bit can be cleared when reading this register if SEMRCEN register is enabled.

47.3.1.2 SEMRCEN : Semaphore Read Clear Enable Register

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SEMRCEN7	SEMRCEN6	SEMRCEN5	SEMRCEN4	SEMRCEN3	SEMRCEN2	SEMRCEN1	SEMRCEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SEMRCEN0	Enables or disables SEM0 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
1	SEMRCEN1	Enables or disables SEM1 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
2	SEMRCEN2	Enables or disables SEM2 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
3	SEMRCEN3	Enables or disables SEM3 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
4	SEMRCEN4	Enables or disables SEM4 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
5	SEMRCEN5	Enables or disables SEM5 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
6	SEMRCEN6	Enables or disables SEM6 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
7	SEMRCEN7	Enables or disables SEM7 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

SEMRCEN register enables and disables read clear function of SEMn register.

47.3.1.3 MBXH2Cn : Host to CA55/CR52 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x080 + 0x04 × n

Bit position:	31																												0
Bit field:	MBX[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from external host CPU to internal Cortex-A55/Cortex-R52/CoreSight	R

Note: Only read is available from internal Cortex-A55/Cortex-R52/CoreSight, and read and write are available from external host CPU.

MBXH2Cn register is used as mailbox from external host CPU to internal Cortex-A55/Cortex-R52/CoreSight.

47.3.1.4 MBXISETH2C : Host to CA55/CR52 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INT3S	MBX_INT2S	MBX_INT1S	MBX_INT0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INT0S	Generates or indicates MBX_INT0 interrupt of mailbox from external host CPU to internal Cortex-A55/Cortex-R52/CoreSight. Write from external host CPU: 0: No operation. MBX_INT0 interrupt status is not changed. 1: MBX_INT0 interrupt is asserted. Read: Indicates MBX_INT0 interrupt status	R
1	MBX_INT1S	Generates or indicates MBX_INT1 interrupt of mailbox from external host CPU to internal Cortex-A55/Cortex-R52/CoreSight. Write from external host CPU: 0: No operation. MBX_INT1 interrupt status is not changed. 1: MBX_INT1 interrupt is asserted. Read: Indicates MBX_INT1 interrupt status	R
2	MBX_INT2S	Generates or indicates MBX_INT2 interrupt of mailbox from external host CPU to internal Cortex-A55/Cortex-R52/CoreSight. Write from external host CPU: 0: No operation. MBX_INT2 interrupt status is not changed. 1: MBX_INT2 interrupt is asserted. Read: Indicates MBX_INT2 interrupt status	R

Bit	Symbol	Function	R/W
3	MBX_INT3S	Generates or indicates MBX_INT3 interrupt of mailbox from external host CPU to internal Cortex-A55/Cortex-R52/CoreSight. Write from external host CPU: 0: No operation. MBX_INT3 interrupt status is not changed. 1: MBX_INT3 interrupt is asserted. Read: Indicates MBX_INT3 interrupt status	R
31:4	—	These bits are read as 0.	R

Note: Only read is available from internal Cortex-A55/Cortex-R52/CoreSight, and read and write are available from external host CPU.

MBXISETH2C register is used to generate mailbox interrupt from external host CPU to internal Cortex-A55/Cortex-R52/CoreSight. Maximum 4 interrupts can be generated.

47.3.1.5 MBXICLRH2C : Host to CA55/CR52 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x094

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INT3C	MBX_INT2C	MBX_INT1C	MBX_INT0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INT0C	Clears or indicates MBX_INT0 interrupt of mailbox from external host CPU to internal Cortex-A55/Cortex-R52/CoreSight. Write from internal Cortex-A55/Cortex-R52/CoreSight: 0: No operation. MBX_INT0 interrupt status is not changed. 1: MBX_INT0 interrupt is cleared. Read: Indicates MBX_INT0 interrupt status	R/W
1	MBX_INT1C	Clears or indicates MBX_INT1 interrupt of mailbox from external host CPU to internal Cortex-A55/Cortex-R52/CoreSight. Write from internal Cortex-A55/Cortex-R52/CoreSight: 0: No operation. MBX_INT1 interrupt status is not changed. 1: MBX_INT1 interrupt is cleared. Read: Indicates MBX_INT1 interrupt status	R/W
2	MBX_INT2C	Clears or indicates MBX_INT2 interrupt of mailbox from external host CPU to internal Cortex-A55/Cortex-R52/CoreSight. Write from internal Cortex-A55/Cortex-R52/CoreSight: 0: No operation. MBX_INT2 interrupt status is not changed. 1: MBX_INT2 interrupt is cleared. Read: Indicates MBX_INT2 interrupt status	R/W
3	MBX_INT3C	Clears or indicates MBX_INT3 interrupt of mailbox from external host CPU to internal Cortex-A55/Cortex-R52/CoreSight. Write from internal Cortex-A55/Cortex-R52/CoreSight: 0: No operation. MBX_INT3 interrupt status is not changed. 1: MBX_INT3 interrupt is cleared. Read: Indicates MBX_INT3 interrupt status	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: Read and write are available from internal Cortex-A55/Cortex-R52/CoreSight, and only read is available from external host CPU.

MBXICLRH2C register is used to clear mailbox interrupt from external host CPU to internal Cortex-A55/Cortex-R52/ CoreSight.

47.3.1.6 MBXC2Hn : CA55/CR52 to Host Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x100 + 0x04 × n

Bit position:	31																													0
Bit field:	MBX[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU	R/W

Note: Read and write are available from internal Cortex-A55/Cortex-R52/CoreSight, and only read is available from external host CPU.

MBXC2Hn register is used as mailbox from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU.

47.3.1.7 MBXISETC2H : CA55/CR52 to Host Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x110

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_HINT3S	MBX_HINT2S	MBX_HINT1S	MBX_HINT0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_HINT0S	Generates or indicates MBX_HINT0 interrupt of mailbox from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU. Write from internal Cortex-A55/Cortex-R52/CoreSight: 0: No operation. MBX_INT0 interrupt status is not changed. 1: MBX_HINT0 interrupt is asserted. Read: Indicates MBX_HINT0 interrupt status	R/W
1	MBX_HINT1S	Generates or indicates MBX_HINT1 interrupt of mailbox from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU. Write from internal Cortex-A55/Cortex-R52/CoreSight: 0: No operation. MBX_INT1 interrupt status is not changed. 1: MBX_HINT1 interrupt is asserted. Read: Indicates MBX_HINT1 interrupt status	R/W
2	MBX_HINT2S	Generates or indicates MBX_HINT2 interrupt of mailbox from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU. Write from internal Cortex-A55/Cortex-R52/CoreSight: 0: No operation. MBX_INT2 interrupt status is not changed. 1: MBX_HINT2 interrupt is asserted. Read: Indicates MBX_HINT2 interrupt status	R/W

Bit	Symbol	Function	R/W
3	MBX_HINT3S	Generates or indicates MBX_HINT3 interrupt of mailbox from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU. Write from internal Cortex-A55/Cortex-R52/CoreSight: 0: No operation. MBX_INT3 interrupt status is not changed. 1: MBX_HINT3 interrupt is asserted. Read: Indicates MBX_HINT3 interrupt status	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: Read and write are available from internal Cortex-A55/Cortex-R52/CoreSight, and only read is available from external host CPU.

MBXISETC2H register is used to generate mailbox interrupt from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU. Maximum 4 interrupts can be generated but all the interrupts are combined to one output interrupt signal (MBX_HINT#) by logical NOR.

47.3.1.8 MBXICLRC2H : CA55/CR52 to Host Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x114

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_HINT3C	MBX_HINT2C	MBX_HINT1C	MBX_HINT0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_HINT0C	Clears or indicates MBX_HINT0 interrupt of mailbox from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU. Write from external host CPU: 0: No operation. MBX_HINT0 interrupt status is not changed. 1: MBX_HINT0 interrupt is cleared. Read: Indicates MBX_HINT0 interrupt status	R
1	MBX_HINT1C	Clears or indicates MBX_HINT1 interrupt of mailbox from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU. Write from external host CPU: 0: No operation. MBX_HINT1 interrupt status is not changed. 1: MBX_HINT1 interrupt is cleared. Read: Indicates MBX_HINT1 interrupt status	R
2	MBX_HINT2C	Clears or indicates MBX_HINT2 interrupt of mailbox from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU. Write from external host CPU: 0: No operation. MBX_HINT2 interrupt status is not changed. 1: MBX_HINT2 interrupt is cleared. Read: Indicates MBX_HINT2 interrupt status	R
3	MBX_HINT3C	Clears or indicates MBX_HINT3 interrupt of mailbox from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU. Write from external host CPU: 0: No operation. MBX_HINT3 interrupt status is not changed. 1: MBX_HINT3 interrupt is cleared. Read: Indicates MBX_HINT3 interrupt status	R
31:4	—	These bits are read as 0.	R

Note: Only read is available from internal Cortex-A55/Cortex-R52/CoreSight, and read and write are available from external host CPU.

MBXICLRC2H register is used to clear mailbox interrupt from internal Cortex-A55/Cortex-R52/CoreSight to external host CPU.

47.3.2 Semaphore Registers among Internal CPUs

47.3.2.1 SEMARn : Semaphore Register n among Internal CPUs (n = 0 to 7)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x180 + 0x04 × n

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SEM	Semaphore bit 0: Resource being used 1: Resource not being used	R/W*1
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When read clear function is disabled (SEMRCENAR.SEMRCENn = 0), register value is retained after reading the register. When read clear function is enabled (SEMRCENAR.SEMRCENn = 1), register is cleared to 0 after reading the register.

SEMARn register is used to perform semaphore for synchronization and exclusive control among Cortex-A55, Cortex-R52 CPU0 and CPU1. Resource status can be controlled when the resource is shared between CPUs. Semaphore bit can be cleared when reading this register if SEMRCEN register is enabled.

47.3.2.2 SEMRCENAR : Semaphore Read Clear Enable Register among Internal CPUs

Base address: MBXSEM = 0x8024_0000

Offset address: 0x1A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SEMR CEN7	SEMR CEN6	SEMR CEN5	SEMR CEN4	SEMR CEN3	SEMR CEN2	SEMR CEN1	SEMR CEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SEMRCEN0	Enables or disables SEMAR0 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
1	SEMRCEN1	Enables or disables SEMAR1 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
2	SEMRCEN2	Enables or disables SEMAR2 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
3	SEMRCEN3	Enables or disables SEMAR3 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W

Bit	Symbol	Function	R/W
4	SEMRCEN4	Enables or disables SEMAR4 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
5	SEMRCEN5	Enables or disables SEMAR5 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
6	SEMRCEN6	Enables or disables SEMAR6 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
7	SEMRCEN7	Enables or disables SEMAR7 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

SEMRCENAR register enables and disables read clear function of SEMARn register.

47.3.3 Mailbox Registers from Cortex-R52 to Cortex-A55

47.3.3.1 MBXR0A0n : CR52 CPU0 to CA55 Core0 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x200 + 0x04 × n

Bit position:	31																														0
Bit field:	MBX[31:0]																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-R52 CPU0 to Cortex-A55 Core0	R/W*1

Note 1. Read and write is available from Cortex-R52 CPU0 (Source). Only read is available from Cortex-A55 Core0 (Target). Access from other CPUs except source and target is also read only.

MBXR0A0n register is used as mailbox from Cortex-R52 CPU0 to Cortex-A55 Core0.

47.3.3.2 MBXISETR0A0 : CR52 CPU0 to CA55 Core0 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x210

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR0A0_3S	MBX_INTR0A0_2S	MBX_INTR0A0_1S	MBX_INTR0A0_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR0A0_0S	Sets or indicates interrupt status flag (MBX_INTR0A00) of mailbox 0 from Cortex-R52 CPU0 to Cortex-A55 Core0. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR0A0_1S	Sets or indicates interrupt status flag (MBX_INTR0A01) of mailbox 1 from Cortex-R52 CPU0 to Cortex-A55 Core0. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR0A0_2S	Sets or indicates interrupt status flag (MBX_INTR0A02) of mailbox 2 from Cortex-R52 CPU0 to Cortex-A55 Core0. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR0A0_3S	Sets or indicates interrupt status flag (MBX_INTR0A03) of mailbox 3 from Cortex-R52 CPU0 to Cortex-A55 Core0. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU0 (Source). Only read is available from Cortex-A55 Core0 (Target). Access from other CPUs except source and target is also read only.

MBXISETR0A0 register is used to generate mailbox interrupt from Cortex-R52 CPU0 to Cortex-A55 Core0. MBX_INTR0A0 interrupt is generated as logical OR of MBX_INTR0A00, MBX_INTR0A01, MBX_INTR0A02, or MBX_INTR0A03 status flags.

47.3.3.3 MBXICLRR0A0 : CR52 CPU0 to CA55 Core0 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x214

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR0A0_3C	MBX_INTR0A0_2C	MBX_INTR0A0_1C	MBX_INTR0A0_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR0A0_0C	Clears or indicates interrupt status flag (MBX_INTR0A00) of mailbox 0 from Cortex-R52 CPU0 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
1	MBX_INTR0A0_1C	Clears or indicates interrupt status flag (MBX_INTR0A01) of mailbox 1 from Cortex-R52 CPU0 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR0A0_2C	Clears or indicates interrupt status flag (MBX_INTR0A02) of mailbox 2 from Cortex-R52 CPU0 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR0A0_3C	Clears or indicates interrupt status flag (MBX_INTR0A03) of mailbox 3 from Cortex-R52 CPU0 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core0 (Target). Only read is available from Cortex-R52 CPU0 (Source). Access from other CPUs except source and target is also read only.

MBXICLRR0A0 register is used to clear mailbox interrupt from Cortex-R52 CPU0 to Cortex-A55 Core0. MBX_INTR0A0 interrupt is cleared when all the status flags are cleared.

47.3.3.4 MBXR0A1n : CR52 CPU0 to CA55 Core1 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x280 + 0x04 × n

Bit position:	31																												0
Bit field:	MBX[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-R52 CPU0 to Cortex-A55 Core1	R/W*1

Note 1. Read and write is available from Cortex-R52 CPU0 (Source). Only read is available from Cortex-A55 Core1 (Target). Access from other CPUs except source and target is also read only.

MBXR0A1n register is used as mailbox from Cortex-R52 CPU0 to Cortex-A55 Core1.

47.3.3.5 MBXISETR0A1 : CR52 CPU0 to CA55 Core1 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x290

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR0A1_3S	MBX_INTR0A1_2S	MBX_INTR0A1_1S	MBX_INTR0A1_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR0A1_0S	Sets or indicates interrupt status flag (MBX_INTR0A10) of mailbox 0 from Cortex-R52 CPU0 to Cortex-A55 Core1. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR0A1_1S	Sets or indicates interrupt status flag (MBX_INTR0A11) of mailbox 1 from Cortex-R52 CPU0 to Cortex-A55 Core1. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR0A1_2S	Sets or indicates interrupt status flag (MBX_INTR0A12) of mailbox 2 from Cortex-R52 CPU0 to Cortex-A55 Core1. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR0A1_3S	Sets or indicates interrupt status flag (MBX_INTR0A13) of mailbox 3 from Cortex-R52 CPU0 to Cortex-A55 Core1. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU0 (Source). Only read is available from Cortex-A55 Core1 (Target). Access from other CPUs except source and target is also read only.

MBXISETR0A1 register is used to generate mailbox interrupt from Cortex-R52 CPU0 to Cortex-A55 Core1. MBX_INTR0A1 interrupt is generated as logical OR of MBX_INTR0A10, MBX_INTR0A11, MBX_INTR0A12, or MBX_INTR0A13 status flags.

47.3.3.6 MBXICLRR0A1 : CR52 CPU0 to CA55 Core1 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x294

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR0A1_3C	MBX_INTR0A1_2C	MBX_INTR0A1_1C	MBX_INTR0A1_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR0A1_0C	Clears or indicates interrupt status flag (MBX_INTR0A10) of mailbox 0 from Cortex-R52 CPU0 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR0A1_1C	Clears or indicates interrupt status flag (MBX_INTR0A11) of mailbox 1 from Cortex-R52 CPU0 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR0A1_2C	Clears or indicates interrupt status flag (MBX_INTR0A12) of mailbox 2 from Cortex-R52 CPU0 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR0A1_3C	Clears or indicates interrupt status flag (MBX_INTR0A13) of mailbox 3 from Cortex-R52 CPU0 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core1 (Target). Only read is available from Cortex-R52 CPU0 (Source). Access from other CPUs except source and target is also read only.

MBXICLRR0A1 register is used to clear mailbox interrupt from Cortex-R52 CPU0 to Cortex-A55 Core1. MBX_INTR0A1 interrupt is cleared when all the status flags are cleared.

47.3.3.7 MBXR0A2n : CR52 CPU0 to CA55 Core2 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x300 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-R52 CPU0 to Cortex-A55 Core2	R/W*1

Note 1. Read and write is available from Cortex-R52 CPU0 (Source). Only read is available from Cortex-A55 Core2 (Target). Access from other CPUs except source and target is also read only.

MBXR0A2n register is used as mailbox from Cortex-R52 CPU0 to Cortex-A55 Core2.

47.3.3.8 MBXISETR0A2 : CR52 CPU0 to CA55 Core2 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x310

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR0A2_3S	MBX_INTR0A2_2S	MBX_INTR0A2_1S	MBX_INTR0A2_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR0A2_0S	Sets or indicates interrupt status flag (MBX_INTR0A20) of mailbox 0 from Cortex-R52 CPU0 to Cortex-A55 Core2. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR0A2_1S	Sets or indicates interrupt status flag (MBX_INTR0A21) of mailbox 1 from Cortex-R52 CPU0 to Cortex-A55 Core2. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR0A2_2S	Sets or indicates interrupt status flag (MBX_INTR0A22) of mailbox 2 from Cortex-R52 CPU0 to Cortex-A55 Core2. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR0A2_3S	Sets or indicates interrupt status flag (MBX_INTR0A23) of mailbox 3 from Cortex-R52 CPU0 to Cortex-A55 Core2. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU0 (Source). Only read is available from Cortex-A55 Core2 (Target). Access from other CPUs except source and target is also read only.

MBXISETR0A2 register is used to generate mailbox interrupt from Cortex-R52 CPU0 to Cortex-A55 Core2. MBX_INTR0A2 interrupt is generated as logical OR of MBX_INTR0A20, MBX_INTR0A21, MBX_INTR0A22, or MBX_INTR0A23 status flags.

47.3.3.9 MBXICLRR0A2 : CR52 CPU0 to CA55 Core2 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x314

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR0A2_3C	MBX_INTR0A2_2C	MBX_INTR0A2_1C	MBX_INTR0A2_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR0A2_0C	Clears or indicates interrupt status flag (MBX_INTR0A20) of mailbox 0 from Cortex-R52 CPU0 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR0A2_1C	Clears or indicates interrupt status flag (MBX_INTR0A21) of mailbox 1 from Cortex-R52 CPU0 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR0A2_2C	Clears or indicates interrupt status flag (MBX_INTR0A22) of mailbox 2 from Cortex-R52 CPU0 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR0A2_3C	Clears or indicates interrupt status flag (MBX_INTR0A23) of mailbox 3 from Cortex-R52 CPU0 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core2 (Target). Only read is available from Cortex-R52 CPU0 (Source). Access from other CPUs except source and target is also read only.

MBXICLRR0A2 register is used to clear mailbox interrupt from Cortex-R52 CPU0 to Cortex-A55 Core2. MBX_INTR0A2 interrupt is cleared when all the status flags are cleared.

47.3.3.10 MBXR0A3n : CR52 CPU0 to CA55 Core3 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x380 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-R52 CPU0 to Cortex-A55 Core3	R/W*1

Note 1. Read and write is available from Cortex-R52 CPU0 (Source). Only read is available from Cortex-A55 Core3 (Target). Access from other CPUs except source and target is also read only.

MBXR0A3n register is used as mailbox from Cortex-R52 CPU0 to Cortex-A55 Core3.

47.3.3.11 MBXISETR0A3 : CR52 CPU0 to CA55 Core3 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x390

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR0A3_3S	MBX_INTR0A3_2S	MBX_INTR0A3_1S	MBX_INTR0A3_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR0A3_0S	Sets or indicates interrupt status flag (MBX_INTR0A30) of mailbox 0 from Cortex-R52 CPU0 to Cortex-A55 Core3. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR0A3_1S	Sets or indicates interrupt status flag (MBX_INTR0A31) of mailbox 1 from Cortex-R52 CPU0 to Cortex-A55 Core3. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR0A3_2S	Sets or indicates interrupt status flag (MBX_INTR0A32) of mailbox 2 from Cortex-R52 CPU0 to Cortex-A55 Core3. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR0A3_3S	Sets or indicates interrupt status flag (MBX_INTR0A33) of mailbox 3 from Cortex-R52 CPU0 to Cortex-A55 Core3. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU0 (Source). Only read is available from Cortex-A55 Core3 (Target). Access from other CPUs except source and target is also read only.

MBXISETR0A3 register is used to generate mailbox interrupt from Cortex-R52 CPU0 to Cortex-A55 Core3. MBX_INTR0A3 interrupt is generated as logical OR of MBX_INTR0A30, MBX_INTR0A31, MBX_INTR0A32, or MBX_INTR0A33 status flags.

47.3.3.12 MBXICLRR0A3 : CR52 CPU0 to CA55 Core3 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x394

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR0A3_3C	MBX_INTR0A3_2C	MBX_INTR0A3_1C	MBX_INTR0A3_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR0A3_0C	Clears or indicates interrupt status flag (MBX_INTR0A30) of mailbox 0 from Cortex-R52 CPU0 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR0A3_1C	Clears or indicates interrupt status flag (MBX_INTR0A31) of mailbox 1 from Cortex-R52 CPU0 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR0A3_2C	Clears or indicates interrupt status flag (MBX_INTR0A32) of mailbox 2 from Cortex-R52 CPU0 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR0A3_3C	Clears or indicates interrupt status flag (MBX_INTR0A33) of mailbox 3 from Cortex-R52 CPU0 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core3 (Target). Only read is available from Cortex-R52 CPU0 (Source). Access from other CPUs except source and target is also read only.

MBXICLRR0A3 register is used to clear mailbox interrupt from Cortex-R52 CPU0 to Cortex-A55 Core3. MBX_INTR0A3 interrupt is cleared when all the status flags are cleared.

47.3.3.13 MBXR1A0n : CR52 CPU1 to CA55 Core0 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x400 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-R52 CPU1 to Cortex-A55 Core0	R/W*1

Note 1. Read and write is available from Cortex-R52 CPU1 (Source). Only read is available from Cortex-A55 Core0 (Target). Access from other CPUs except source and target is also read only.

MBXR1A0n register is used as mailbox from Cortex-R52 CPU1 to Cortex-A55 Core0.

47.3.3.14 MBXISETR1A0 : CR52 CPU1 to CA55 Core0 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x410

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR1A0_3S	MBX_INTR1A0_2S	MBX_INTR1A0_1S	MBX_INTR1A0_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR1A0_0S	Sets or indicates interrupt status flag (MBX_INTR1A00) of mailbox 0 from Cortex-R52 CPU1 to Cortex-A55 Core0. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR1A0_1S	Sets or indicates interrupt status flag (MBX_INTR1A01) of mailbox 1 from Cortex-R52 CPU1 to Cortex-A55 Core0. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR1A0_2S	Sets or indicates interrupt status flag (MBX_INTR1A02) of mailbox 2 from Cortex-R52 CPU1 to Cortex-A55 Core0. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR1A0_3S	Sets or indicates interrupt status flag (MBX_INTR1A03) of mailbox 3 from Cortex-R52 CPU1 to Cortex-A55 Core0. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU1 (Source). Only read is available from Cortex-A55 Core0 (Target). Access from other CPUs except source and target is also read only.

MBXISETR1A0 register is used to generate mailbox interrupt from Cortex-R52 CPU1 to Cortex-A55 Core0. MBX_INTR1A0 interrupt is generated as logical OR of MBX_INTR1A00, MBX_INTR1A01, MBX_INTR1A02, or MBX_INTR1A03 status flags.

47.3.3.15 MBXICLRR1A0 : CR52 CPU1 to CA55 Core0 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x414

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR1A0_3C	MBX_INTR1A0_2C	MBX_INTR1A0_1C	MBX_INTR1A0_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR1A0_0C	Clears or indicates interrupt status flag (MBX_INTR1A00) of mailbox 0 from Cortex-R52 CPU1 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR1A0_1C	Clears or indicates interrupt status flag (MBX_INTR1A01) of mailbox 1 from Cortex-R52 CPU1 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR1A0_2C	Clears or indicates interrupt status flag (MBX_INTR1A02) of mailbox 2 from Cortex-R52 CPU1 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR1A0_3C	Clears or indicates interrupt status flag (MBX_INTR1A03) of mailbox 3 from Cortex-R52 CPU1 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core0 (Target). Only read is available from Cortex-R52 CPU1 (Source). Access from other CPUs except source and target is also read only.

MBXICLRR1A0 register is used to clear mailbox interrupt from Cortex-R52 CPU1 to Cortex-A55 Core0. MBX_INTR1A0 interrupt is cleared when all the status flags are cleared.

47.3.3.16 MBXR1A1n : CR52 CPU1 to CA55 Core1 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x480 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-R52 CPU1 to Cortex-A55 Core1	R/W*1

Note 1. Read and write is available from Cortex-R52 CPU1 (Source). Only read is available from Cortex-A55 Core1 (Target). Access from other CPUs except source and target is also read only.

MBXR1A1n register is used as mailbox from Cortex-R52 CPU1 to Cortex-A55 Core1.

47.3.3.17 MBXISETR1A1 : CR52 CPU1 to CA55 Core1 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x490

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR1A1_3S	MBX_INTR1A1_2S	MBX_INTR1A1_1S	MBX_INTR1A1_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR1A1_0S	Sets or indicates interrupt status flag (MBX_INTR1A10) of mailbox 0 from Cortex-R52 CPU1 to Cortex-A55 Core1. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR1A1_1S	Sets or indicates interrupt status flag (MBX_INTR1A11) of mailbox 1 from Cortex-R52 CPU1 to Cortex-A55 Core1. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR1A1_2S	Sets or indicates interrupt status flag (MBX_INTR1A12) of mailbox 2 from Cortex-R52 CPU1 to Cortex-A55 Core1. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR1A1_3S	Sets or indicates interrupt status flag (MBX_INTR1A13) of mailbox 3 from Cortex-R52 CPU1 to Cortex-A55 Core1. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU1 (Source). Only read is available from Cortex-A55 Core1 (Target). Access from other CPUs except source and target is also read only.

MBXISETR1A1 register is used to generate mailbox interrupt from Cortex-R52 CPU1 to Cortex-A55 Core1. MBX_INTR1A1 interrupt is generated as logical OR of MBX_INTR1A10, MBX_INTR1A11, MBX_INTR1A12, or MBX_INTR1A13 status flags.

47.3.3.18 MBXICLRR1A1 : CR52 CPU1 to CA55 Core1 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x494

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR1A1_3C	MBX_INTR1A1_2C	MBX_INTR1A1_1C	MBX_INTR1A1_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR1A1_0C	Clears or indicates interrupt status flag (MBX_INTR1A10) of mailbox 0 from Cortex-R52 CPU1 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR1A1_1C	Clears or indicates interrupt status flag (MBX_INTR1A11) of mailbox 1 from Cortex-R52 CPU1 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR1A1_2C	Clears or indicates interrupt status flag (MBX_INTR1A12) of mailbox 2 from Cortex-R52 CPU1 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR1A1_3C	Clears or indicates interrupt status flag (MBX_INTR1A13) of mailbox 3 from Cortex-R52 CPU1 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core1 (Target). Only read is available from Cortex-R52 CPU1 (Source). Access from other CPUs except source and target is also read only.

MBXICLRR1A1 register is used to clear mailbox interrupt from Cortex-R52 CPU1 to Cortex-A55 Core1. MBX_INTR1A1 interrupt is cleared when all the status flags are cleared.

47.3.3.19 MBXR1A2n : CR52 CPU1 to CA55 Core2 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x500 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-R52 CPU1 to Cortex-A55 Core2	R/W*1

Note 1. Read and write is available from Cortex-R52 CPU1 (Source). Only read is available from Cortex-A55 Core2 (Target). Access from other CPUs except source and target is also read only.

MBXR1A2n register is used as mailbox from Cortex-R52 CPU1 to Cortex-A55 Core2.

47.3.3.20 MBXISETR1A2 : CR52 CPU1 to CA55 Core2 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x510

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR1A2_3S	MBX_INTR1A2_2S	MBX_INTR1A2_1S	MBX_INTR1A2_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR1A2_0S	Sets or indicates interrupt status flag (MBX_INTR1A20) of mailbox 0 from Cortex-R52 CPU1 to Cortex-A55 Core2. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR1A2_1S	Sets or indicates interrupt status flag (MBX_INTR1A21) of mailbox 1 from Cortex-R52 CPU1 to Cortex-A55 Core2. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR1A2_2S	Sets or indicates interrupt status flag (MBX_INTR1A22) of mailbox 2 from Cortex-R52 CPU1 to Cortex-A55 Core2. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR1A2_3S	Sets or indicates interrupt status flag (MBX_INTR1A23) of mailbox 3 from Cortex-R52 CPU1 to Cortex-A55 Core2. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU1 (Source). Only read is available from Cortex-A55 Core2 (Target). Access from other CPUs except source and target is also read only.

MBXISETR1A2 register is used to generate mailbox interrupt from Cortex-R52 CPU1 to Cortex-A55 Core2. MBX_INTR1A2 interrupt is generated as logical OR of MBX_INTR1A20, MBX_INTR1A21, MBX_INTR1A22, or MBX_INTR1A23 status flags.

47.3.3.21 MBXICLRR1A2 : CR52 CPU1 to CA55 Core2 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x514

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR1A2_3C	MBX_INTR1A2_2C	MBX_INTR1A2_1C	MBX_INTR1A2_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR1A2_0C	Clears or indicates interrupt status flag (MBX_INTR1A20) of mailbox 0 from Cortex-R52 CPU1 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR1A2_1C	Clears or indicates interrupt status flag (MBX_INTR1A21) of mailbox 1 from Cortex-R52 CPU1 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR1A2_2C	Clears or indicates interrupt status flag (MBX_INTR1A22) of mailbox 2 from Cortex-R52 CPU1 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR1A2_3C	Clears or indicates interrupt status flag (MBX_INTR1A23) of mailbox 3 from Cortex-R52 CPU1 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core2 (Target). Only read is available from Cortex-R52 CPU1 (Source). Access from other CPUs except source and target is also read only.

MBXICLRR1A2 register is used to clear mailbox interrupt from Cortex-R52 CPU1 to Cortex-A55 Core2. MBX_INTR1A2 interrupt is cleared when all the status flags are cleared.

47.3.3.22 MBXR1A3n : CR52 CPU1 to CA55 Core3 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x580 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-R52 CPU1 to Cortex-A55 Core3	R/W*1

Note 1. Read and write is available from Cortex-R52 CPU1 (Source). Only read is available from Cortex-A55 Core3 (Target). Access from other CPUs except source and target is also read only.

MBXR1A3n register is used as mailbox from Cortex-R52 CPU1 to Cortex-A55 Core3.

47.3.3.23 MBXISETR1A3 : CR52 CPU1 to CA55 Core3 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x590

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR1A3_3S	MBX_INTR1A3_2S	MBX_INTR1A3_1S	MBX_INTR1A3_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR1A3_0S	Sets or indicates interrupt status flag (MBX_INTR1A30) of mailbox 0 from Cortex-R52 CPU1 to Cortex-A55 Core3. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR1A3_1S	Sets or indicates interrupt status flag (MBX_INTR1A31) of mailbox 1 from Cortex-R52 CPU1 to Cortex-A55 Core3. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR1A3_2S	Sets or indicates interrupt status flag (MBX_INTR1A32) of mailbox 2 from Cortex-R52 CPU1 to Cortex-A55 Core3. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR1A3_3S	Sets or indicates interrupt status flag (MBX_INTR1A33) of mailbox 3 from Cortex-R52 CPU1 to Cortex-A55 Core3. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU1 (Source). Only read is available from Cortex-A55 Core3 (Target). Access from other CPUs except source and target is also read only.

MBXISETR1A3 register is used to generate mailbox interrupt from Cortex-R52 CPU1 to Cortex-A55 Core3. MBX_INTR1A3 interrupt is generated as logical OR of MBX_INTR1A30, MBX_INTR1A31, MBX_INTR1A32, or MBX_INTR1A33 status flags.

47.3.3.24 MBXICLRR1A3 : CR52 CPU1 to CA55 Core3 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x594

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR1A3_3C	MBX_INTR1A3_2C	MBX_INTR1A3_1C	MBX_INTR1A3_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR1A3_0C	Clears or indicates interrupt status flag (MBX_INTR1A30) of mailbox 0 from Cortex-R52 CPU1 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR1A3_1C	Clears or indicates interrupt status flag (MBX_INTR1A31) of mailbox 1 from Cortex-R52 CPU1 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR1A3_2C	Clears or indicates interrupt status flag (MBX_INTR1A32) of mailbox 2 from Cortex-R52 CPU1 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR1A3_3C	Clears or indicates interrupt status flag (MBX_INTR1A33) of mailbox 3 from Cortex-R52 CPU1 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core3 (Target). Only read is available from Cortex-R52 CPU1 (Source). Access from other CPUs except source and target is also read only.

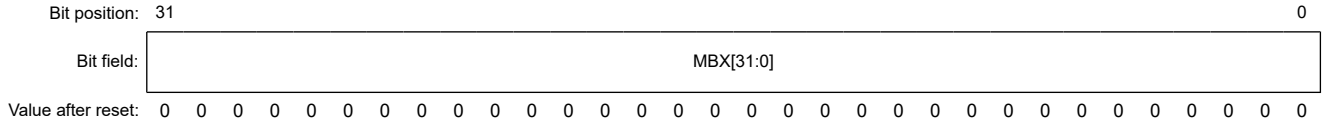
MBXICLRR1A3 register is used to clear mailbox interrupt from Cortex-R52 CPU1 to Cortex-A55 Core3. MBX_INTR1A3 interrupt is cleared when all the status flags are cleared.

47.3.4 Mailbox Registers from Cortex-A55 to Cortex-R52

47.3.4.1 MBXA0R0n : CA55 Core0 to CR52 CPU0 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x600 + 0x04 × n



Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core0 to Cortex-R52 CPU0	R/W*1

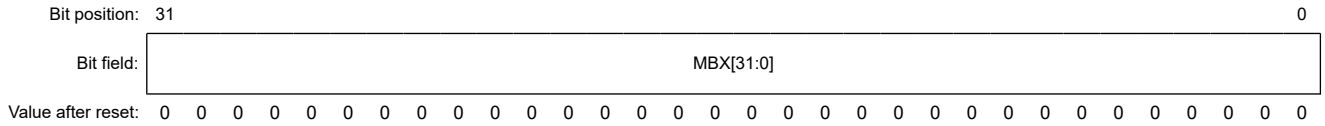
Note 1. Read and write is available from Cortex-A55 Core0 (Source). Only read is available from Cortex-R52 CPU0 (Target). Access from other CPUs except source and target is also read only.

MBXA0R0n register is used as mailbox from Cortex-A55 Core0 to Cortex-R52 CPU0.

47.3.4.2 MBXA1R0n : CA55 Core1 to CR52 CPU0 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x610 + 0x04 × n



Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core1 to Cortex-R52 CPU0	R/W*1

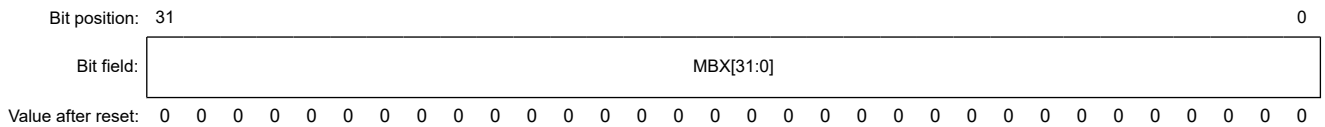
Note 1. Read and write is available from Cortex-A55 Core1 (Source). Only read is available from Cortex-R52 CPU0 (Target). Access from other CPUs except source and target is also read only.

MBXA1R0n register is used as mailbox from Cortex-A55 Core1 to Cortex-R52 CPU0.

47.3.4.3 MBXA2R0n : CA55 Core2 to CR52 CPU0 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x620 + 0x04 × n



Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core2 to Cortex-R52 CPU0	R/W*1

Note 1. Read and write is available from Cortex-A55 Core2 (Source). Only read is available from Cortex-R52 CPU0 (Target). Access from other CPUs except source and target is also read only.

MBXA2R0n register is used as mailbox from Cortex-A55 Core2 to Cortex-R52 CPU0.

47.3.4.4 MBXA3R0n : CA55 Core3 to CR52 CPU0 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x630 + 0x04 × n

Bit position:	31																												0
Bit field:	MBX[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core3 to Cortex-R52 CPU0	R/W*1

Note 1. Read and write is available from Cortex-A55 Core3 (Source). Only read is available from Cortex-R52 CPU0 (Target). Access from other CPUs except source and target is also read only.

MBXA3R0n register is used as mailbox from Cortex-A55 Core3 to Cortex-R52 CPU0.

47.3.4.5 MBXISETA0R0 : CA55 Core0 to CR52 CPU0 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x640

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA0R0_3S	MBX_INTA0R0_2S	MBX_INTA0R0_1S	MBX_INTA0R0_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA0R0_0S	Sets or indicates interrupt status flag (MBX_INTA0R00) of mailbox 0 from Cortex-A55 Core0 to Cortex-R52 CPU0. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA0R0_1S	Sets or indicates interrupt status flag (MBX_INTA0R01) of mailbox 1 from Cortex-A55 Core0 to Cortex-R52 CPU0. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA0R0_2S	Sets or indicates interrupt status flag (MBX_INTA0R02) of mailbox 2 from Cortex-A55 Core0 to Cortex-R52 CPU0. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
3	MBX_INTA0R0_3S	Sets or indicates interrupt status flag (MBX_INTA0R03) of mailbox 3 from Cortex-A55 Core0 to Cortex-R52 CPU0. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core0 (Source). Only read is available from Cortex-R52 CPU0 (Target). Access from other CPUs except source and target is also read only.

MBXISETA0R0 register is used to generate mailbox interrupt from Cortex-A55 Core0 to Cortex-R52 CPU0. MBX_INTAR0 interrupt is generated as logical OR of the following 16 status flags.

MBX_INTA0R00, MBX_INTA0R01, MBX_INTA0R02, MBX_INTA0R03 in MBXISETA0R0

MBX_INTA1R00, MBX_INTA1R01, MBX_INTA1R02, MBX_INTA1R03 in MBXISETA1R0

MBX_INTA2R00, MBX_INTA2R01, MBX_INTA2R02, MBX_INTA2R03 in MBXISETA2R0

MBX_INTA3R00, MBX_INTA3R01, MBX_INTA3R02, MBX_INTA3R03 in MBXISETA3R0

47.3.4.6 MBXISETA1R0 : CA55 Core1 to CR52 CPU0 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x644

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA1R0_3S	MBX_INTA1R0_2S	MBX_INTA1R0_1S	MBX_INTA1R0_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA1R0_0S	Sets or indicates interrupt status flag (MBX_INTA1R00) of mailbox 0 from Cortex-A55 Core1 to Cortex-R52 CPU0. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA1R0_1S	Sets or indicates interrupt status flag (MBX_INTA1R01) of mailbox 1 from Cortex-A55 Core1 to Cortex-R52 CPU0. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA1R0_2S	Sets or indicates interrupt status flag (MBX_INTA1R02) of mailbox 2 from Cortex-A55 Core1 to Cortex-R52 CPU0. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
3	MBX_INTA1R0_3S	Sets or indicates interrupt status flag (MBX_INTA1R03) of mailbox 3 from Cortex-A55 Core1 to Cortex-R52 CPU0. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core1 (Source). Only read is available from Cortex-R52 CPU0 (Target). Access from other CPUs except source and target is also read only.

MBXISETA1R0 register is used to generate mailbox interrupt from Cortex-A55 Core1 to Cortex-R52 CPU0. MBX_INTAR0 interrupt is generated as logical OR of the following 16 status flags.

MBX_INTA0R00, MBX_INTA0R01, MBX_INTA0R02, MBX_INTA0R03 in MBXISETA0R0

MBX_INTA1R00, MBX_INTA1R01, MBX_INTA1R02, MBX_INTA1R03 in MBXISETA1R0

MBX_INTA2R00, MBX_INTA2R01, MBX_INTA2R02, MBX_INTA2R03 in MBXISETA2R0

MBX_INTA3R00, MBX_INTA3R01, MBX_INTA3R02, MBX_INTA3R03 in MBXISETA3R0

47.3.4.7 MBXISETA2R0 : CA55 Core2 to CR52 CPU0 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x648

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA2R0_3S	MBX_INTA2R0_2S	MBX_INTA2R0_1S	MBX_INTA2R0_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA2R0_0S	Sets or indicates interrupt status flag (MBX_INTA2R00) of mailbox 0 from Cortex-A55 Core2 to Cortex-R52 CPU0. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA2R0_1S	Sets or indicates interrupt status flag (MBX_INTA2R01) of mailbox 1 from Cortex-A55 Core2 to Cortex-R52 CPU0. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA2R0_2S	Sets or indicates interrupt status flag (MBX_INTA2R02) of mailbox 2 from Cortex-A55 Core2 to Cortex-R52 CPU0. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
3	MBX_INTA2R0_3S	Sets or indicates interrupt status flag (MBX_INTA2R03) of mailbox 3 from Cortex-A55 Core2 to Cortex-R52 CPU0. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core2 (Source). Only read is available from Cortex-R52 CPU0 (Target). Access from other CPUs except source and target is also read only.

MBXISETA2R0 register is used to generate mailbox interrupt from Cortex-A55 Core2 to Cortex-R52 CPU0. MBX_INTAR0 interrupt is generated as logical OR of the following 16 status flags.

MBX_INTA0R00, MBX_INTA0R01, MBX_INTA0R02, MBX_INTA0R03 in MBXISETA0R0

MBX_INTA1R00, MBX_INTA1R01, MBX_INTA1R02, MBX_INTA1R03 in MBXISETA1R0

MBX_INTA2R00, MBX_INTA2R01, MBX_INTA2R02, MBX_INTA2R03 in MBXISETA2R0

MBX_INTA3R00, MBX_INTA3R01, MBX_INTA3R02, MBX_INTA3R03 in MBXISETA3R0

47.3.4.8 MBXISETA3R0 : CA55 Core3 to CR52 CPU0 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x64C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA3R0_3S	MBX_INTA3R0_2S	MBX_INTA3R0_1S	MBX_INTA3R0_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA3R0_0S	Sets or indicates interrupt status flag (MBX_INTA3R00) of mailbox 0 from Cortex-A55 Core3 to Cortex-R52 CPU0. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA3R0_1S	Sets or indicates interrupt status flag (MBX_INTA3R01) of mailbox 1 from Cortex-A55 Core3 to Cortex-R52 CPU0. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA3R0_2S	Sets or indicates interrupt status flag (MBX_INTA3R02) of mailbox 2 from Cortex-A55 Core3 to Cortex-R52 CPU0. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
3	MBX_INTA3R0_3S	Sets or indicates interrupt status flag (MBX_INTA3R03) of mailbox 3 from Cortex-A55 Core3 to Cortex-R52 CPU0. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core3 (Source). Only read is available from Cortex-R52 CPU0 (Target). Access from other CPUs except source and target is also read only.

MBXISETA3R0 register is used to generate mailbox interrupt from Cortex-A55 Core3 to Cortex-R52 CPU0. MBX_INTAR0 interrupt is generated as logical OR of the following 16 status flags.

MBX_INTA0R00, MBX_INTA0R01, MBX_INTA0R02, MBX_INTA0R03 in MBXISETA0R0

MBX_INTA1R00, MBX_INTA1R01, MBX_INTA1R02, MBX_INTA1R03 in MBXISETA1R0

MBX_INTA2R00, MBX_INTA2R01, MBX_INTA2R02, MBX_INTA2R03 in MBXISETA2R0

MBX_INTA3R00, MBX_INTA3R01, MBX_INTA3R02, MBX_INTA3R03 in MBXISETA3R0

47.3.4.9 MBXICLRAR0 : CA55 to CR52 CPU0 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x650

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MBX_INTA3R0_3C	MBX_INTA3R0_2C	MBX_INTA3R0_1C	MBX_INTA3R0_0C	MBX_INTA2R0_3C	MBX_INTA2R0_2C	MBX_INTA2R0_1C	MBX_INTA2R0_0C	MBX_INTA1R0_3C	MBX_INTA1R0_2C	MBX_INTA1R0_1C	MBX_INTA1R0_0C	MBX_INTA0R0_3C	MBX_INTA0R0_2C	MBX_INTA0R0_1C	MBX_INTA0R0_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA0R0_0C	Clears or indicates interrupt status flag (MBX_INTA0R00) of mailbox 0 from Cortex-A55 Core0 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA0R0_1C	Clears or indicates interrupt status flag (MBX_INTA0R01) of mailbox 1 from Cortex-A55 Core0 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA0R0_2C	Clears or indicates interrupt status flag (MBX_INTA0R02) of mailbox 2 from Cortex-A55 Core0 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
3	MBX_INTA0R0_3C	Clears or indicates interrupt status flag (MBX_INTA0R03) of mailbox 3 from Cortex-A55 Core0 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
4	MBX_INTA1R0_0C	Clears or indicates interrupt status flag (MBX_INTA1R00) of mailbox 0 from Cortex-A55 Core1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
5	MBX_INTA1R0_1C	Clears or indicates interrupt status flag (MBX_INTA1R01) of mailbox 1 from Cortex-A55 Core1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
6	MBX_INTA1R0_2C	Clears or indicates interrupt status flag (MBX_INTA1R02) of mailbox 2 from Cortex-A55 Core1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
7	MBX_INTA1R0_3C	Clears or indicates interrupt status flag (MBX_INTA1R03) of mailbox 3 from Cortex-A55 Core1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
8	MBX_INTA2R0_0C	Clears or indicates interrupt status flag (MBX_INTA2R00) of mailbox 0 from Cortex-A55 Core2 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
9	MBX_INTA2R0_1C	Clears or indicates interrupt status flag (MBX_INTA2R01) of mailbox 1 from Cortex-A55 Core2 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
10	MBX_INTA2R0_2C	Clears or indicates interrupt status flag (MBX_INTA2R02) of mailbox 2 from Cortex-A55 Core2 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
11	MBX_INTA2R0_3C	Clears or indicates interrupt status flag (MBX_INTA2R03) of mailbox 3 from Cortex-A55 Core2 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
12	MBX_INTA3R0_0C	Clears or indicates interrupt status flag (MBX_INTA3R00) of mailbox 0 from Cortex-A55 Core3 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
13	MBX_INTA3R0_1C	Clears or indicates interrupt status flag (MBX_INTA3R01) of mailbox 1 from Cortex-A55 Core3 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
14	MBX_INTA3R0_2C	Clears or indicates interrupt status flag (MBX_INTA3R02) of mailbox 2 from Cortex-A55 Core3 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
15	MBX_INTA3R0_3C	Clears or indicates interrupt status flag (MBX_INTA3R03) of mailbox 3 from Cortex-A55 Core3 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:16	—	These bits are read as 0. The write value should be 0.	R/W

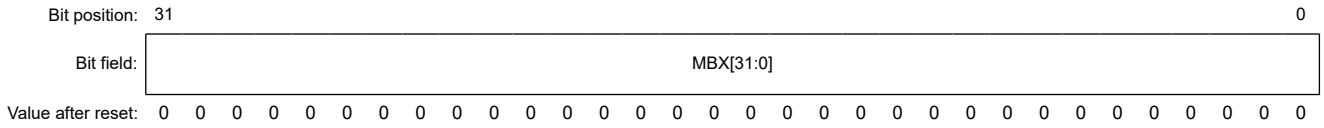
Note 1. Read and write is available from Cortex-R52 CPU0 (Target). Only read is available from Cortex-A55 Core0/1/2/3 (Source). Access from other CPUs except source and target is also read only.

MBXICLRAR0 register is used to clear mailbox interrupt from Cortex-A55 Core0/1/2/3 to Cortex-R52 CPU0. MBX_INTAR0 interrupt is cleared when all the status flags are cleared.

47.3.4.10 MBXA0R1n : CA55 Core0 to CR52 CPU1 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x680 + 0x04 × n



Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core0 to Cortex-R52 CPU1	R/W*1

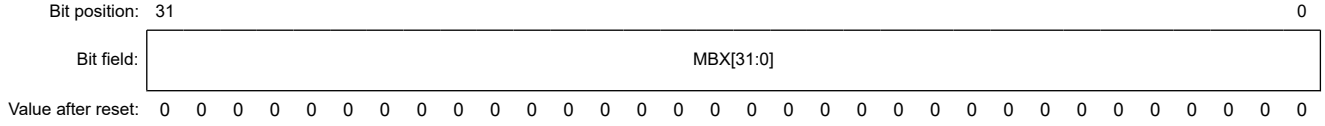
Note 1. Read and write is available from Cortex-A55 Core0 (Source). Only read is available from Cortex-R52 CPU1 (Target). Access from other CPUs except source and target is also read only.

MBXA0R1n register is used as mailbox from Cortex-A55 Core0 to Cortex-R52 CPU1.

47.3.4.11 MBXA1R1n : CA55 Core1 to CR52 CPU1 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x690 + 0x04 × n



Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core1 to Cortex-R52 CPU1	R/W*1

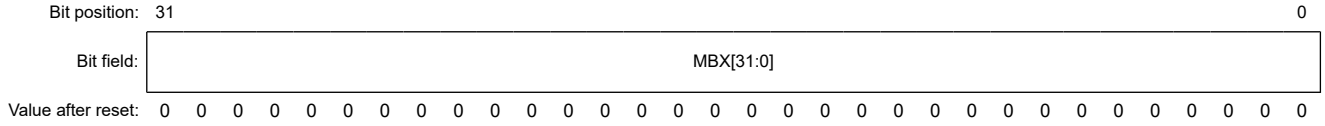
Note 1. Read and write is available from Cortex-A55 Core1 (Source). Only read is available from Cortex-R52 CPU1 (Target). Access from other CPUs except source and target is also read only.

MBXA1R1n register is used as mailbox from Cortex-A55 Core1 to Cortex-R52 CPU1.

47.3.4.12 MBXA2R1n : CA55 Core2 to CR52 CPU1 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x6A0 + 0x04 × n



Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core2 to Cortex-R52 CPU1	R/W*1

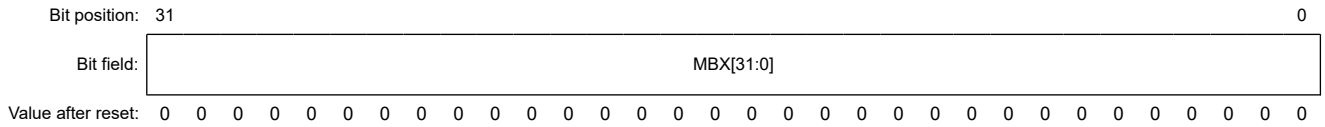
Note 1. Read and write is available from Cortex-A55 Core2 (Source). Only read is available from Cortex-R52 CPU1 (Target). Access from other CPUs except source and target is also read only.

MBXA2R1n register is used as mailbox from Cortex-A55 Core2 to Cortex-R52 CPU1.

47.3.4.13 MBXA3R1n : CA55 Core3 to CR52 CPU1 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x6B0 + 0x04 × n



Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core3 to Cortex-R52 CPU1	R/W*1

Note 1. Read and write is available from Cortex-A55 Core3 (Source). Only read is available from Cortex-R52 CPU1 (Target). Access from other CPUs except source and target is also read only.

MBXA3R1n register is used as mailbox from Cortex-A55 Core3 to Cortex-R52 CPU1.

47.3.4.14 MBXISETA0R1 : CA55 Core0 to CR52 CPU1 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x6C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA0R1_3S	MBX_INTA0R1_2S	MBX_INTA0R1_1S	MBX_INTA0R1_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA0R1_0S	Sets or indicates interrupt status flag (MBX_INTA0R10) of mailbox 0 from Cortex-A55 Core0 to Cortex-R52 CPU1. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA0R1_1S	Sets or indicates interrupt status flag (MBX_INTA0R11) of mailbox 1 from Cortex-A55 Core0 to Cortex-R52 CPU1. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA0R1_2S	Sets or indicates interrupt status flag (MBX_INTA0R12) of mailbox 2 from Cortex-A55 Core0 to Cortex-R52 CPU1. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA0R1_3S	Sets or indicates interrupt status flag (MBX_INTA0R13) of mailbox 3 from Cortex-A55 Core0 to Cortex-R52 CPU1. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core0 (Source). Only read is available from Cortex-R52 CPU1 (Target). Access from other CPUs except source and target is also read only.

MBXISETA0R1 register is used to generate mailbox interrupt from Cortex-A55 Core0 to Cortex-R52 CPU1.
MBX_INTAR1 interrupt is generated as logical OR of the following 16 status flags.

MBX_INTA0R10, MBX_INTA0R11, MBX_INTA0R12, MBX_INTA0R13 in MBXISETA0R1

MBX_INTA1R10, MBX_INTA1R11, MBX_INTA1R12, MBX_INTA1R13 in MBXISETA1R1

MBX_INTA2R10, MBX_INTA2R11, MBX_INTA2R12, MBX_INTA2R13 in MBXISETA2R1

MBX_INTA3R10, MBX_INTA3R11, MBX_INTA3R12, MBX_INTA3R13 in MBXISETA3R1

47.3.4.15 MBXISETA1R1 : CA55 Core1 to CR52 CPU1 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x6C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA1R1_3S	MBX_INTA1R1_2S	MBX_INTA1R1_1S	MBX_INTA1R1_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA1R1_0S	Sets or indicates interrupt status flag (MBX_INTA1R10) of mailbox 0 from Cortex-A55 Core1 to Cortex-R52 CPU1. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA1R1_1S	Sets or indicates interrupt status flag (MBX_INTA1R11) of mailbox 1 from Cortex-A55 Core1 to Cortex-R52 CPU1. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA1R1_2S	Sets or indicates interrupt status flag (MBX_INTA1R12) of mailbox 2 from Cortex-A55 Core1 to Cortex-R52 CPU1. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA1R1_3S	Sets or indicates interrupt status flag (MBX_INTA1R13) of mailbox 3 from Cortex-A55 Core1 to Cortex-R52 CPU1. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core1 (Source). Only read is available from Cortex-R52 CPU1 (Target). Access from other CPUs except source and target is also read only.

MBXISETA1R1 register is used to generate mailbox interrupt from Cortex-A55 Core1 to Cortex-R52 CPU1.
MBX_INTAR1 interrupt is generated as logical OR of the following 16 status flags.

MBX_INTA0R10, MBX_INTA0R11, MBX_INTA0R12, MBX_INTA0R13 in MBXISETA0R1

MBX_INTA1R10, MBX_INTA1R11, MBX_INTA1R12, MBX_INTA1R13 in MBXISETA1R1

MBX_INTA2R10, MBX_INTA2R11, MBX_INTA2R12, MBX_INTA2R13 in MBXISETA2R1

MBX_INTA3R10, MBX_INTA3R11, MBX_INTA3R12, MBX_INTA3R13 in MBXISETA3R1

47.3.4.16 MBXISETA2R1 : CA55 Core2 to CR52 CPU1 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x6C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA2R1_3S	MBX_INTA2R1_2S	MBX_INTA2R1_1S	MBX_INTA2R1_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA2R1_0S	Sets or indicates interrupt status flag (MBX_INTA2R10) of mailbox 0 from Cortex-A55 Core2 to Cortex-R52 CPU1. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA2R1_1S	Sets or indicates interrupt status flag (MBX_INTA2R11) of mailbox 1 from Cortex-A55 Core2 to Cortex-R52 CPU1. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA2R1_2S	Sets or indicates interrupt status flag (MBX_INTA2R12) of mailbox 2 from Cortex-A55 Core2 to Cortex-R52 CPU1. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA2R1_3S	Sets or indicates interrupt status flag (MBX_INTA2R13) of mailbox 3 from Cortex-A55 Core2 to Cortex-R52 CPU1. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core2 (Source). Only read is available from Cortex-R52 CPU1 (Target). Access from other CPUs except source and target is also read only.

MBXISETA2R1 register is used to generate mailbox interrupt from Cortex-A55 Core2 to Cortex-R52 CPU1.
MBX_INTAR1 interrupt is generated as logical OR of the following 16 status flags.

MBX_INTA0R10, MBX_INTA0R11, MBX_INTA0R12, MBX_INTA0R13 in MBXISETA0R1

MBX_INTA1R10, MBX_INTA1R11, MBX_INTA1R12, MBX_INTA1R13 in MBXISETA1R1

MBX_INTA2R10, MBX_INTA2R11, MBX_INTA2R12, MBX_INTA2R13 in MBXISETA2R1

MBX_INTA3R10, MBX_INTA3R11, MBX_INTA3R12, MBX_INTA3R13 in MBXISETA3R1

47.3.4.17 MBXISETA3R1 : CA55 Core3 to CR52 CPU1 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x6CC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA3R1_3S	MBX_INTA3R1_2S	MBX_INTA3R1_1S	MBX_INTA3R1_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA3R1_0S	Sets or indicates interrupt status flag (MBX_INTA3R10) of mailbox 0 from Cortex-A55 Core3 to Cortex-R52 CPU1. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA3R1_1S	Sets or indicates interrupt status flag (MBX_INTA3R11) of mailbox 1 from Cortex-A55 Core3 to Cortex-R52 CPU1. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA3R1_2S	Sets or indicates interrupt status flag (MBX_INTA3R12) of mailbox 2 from Cortex-A55 Core3 to Cortex-R52 CPU1. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA3R1_3S	Sets or indicates interrupt status flag (MBX_INTA3R13) of mailbox 3 from Cortex-A55 Core3 to Cortex-R52 CPU1. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core3 (Source). Only read is available from Cortex-R52 CPU1 (Target). Access from other CPUs except source and target is also read only.

MBXISETA3R1 register is used to generate mailbox interrupt from Cortex-A55 Core3 to Cortex-R52 CPU1.
MBX_INTAR1 interrupt is generated as logical OR of the following 16 status flags.

MBX_INTA0R10, MBX_INTA0R11, MBX_INTA0R12, MBX_INTA0R13 in MBXISETA0R1

MBX_INTA1R10, MBX_INTA1R11, MBX_INTA1R12, MBX_INTA1R13 in MBXISETA1R1

MBX_INTA2R10, MBX_INTA2R11, MBX_INTA2R12, MBX_INTA2R13 in MBXISETA2R1

MBX_INTA3R10, MBX_INTA3R11, MBX_INTA3R12, MBX_INTA3R13 in MBXISETA3R1

47.3.4.18 MBXICLRAR1 : CA55 to CR52 CPU1 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x6D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MBX_INTA3R1_3C	MBX_INTA3R1_2C	MBX_INTA3R1_1C	MBX_INTA3R1_0C	MBX_INTA2R1_3C	MBX_INTA2R1_2C	MBX_INTA2R1_1C	MBX_INTA2R1_0C	MBX_INTA1R1_3C	MBX_INTA1R1_2C	MBX_INTA1R1_1C	MBX_INTA1R1_0C	MBX_INTA0R1_3C	MBX_INTA0R1_2C	MBX_INTA0R1_1C	MBX_INTA0R1_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA0R1_0C	Clears or indicates interrupt status flag (MBX_INTA0R10) of mailbox 0 from Cortex-A55 Core0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA0R1_1C	Clears or indicates interrupt status flag (MBX_INTA0R11) of mailbox 1 from Cortex-A55 Core0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA0R1_2C	Clears or indicates interrupt status flag (MBX_INTA0R12) of mailbox 2 from Cortex-A55 Core0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA0R1_3C	Clears or indicates interrupt status flag (MBX_INTA0R13) of mailbox 3 from Cortex-A55 Core0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
4	MBX_INTA1R1_0C	Clears or indicates interrupt status flag (MBX_INTA1R10) of mailbox 0 from Cortex-A55 Core1 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
5	MBX_INTA1R1_1C	Clears or indicates interrupt status flag (MBX_INTA1R11) of mailbox 1 from Cortex-A55 Core1 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
6	MBX_INTA1R1_2C	Clears or indicates interrupt status flag (MBX_INTA1R12) of mailbox 2 from Cortex-A55 Core1 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
7	MBX_INTA1R1_3C	Clears or indicates interrupt status flag (MBX_INTA1R13) of mailbox 3 from Cortex-A55 Core1 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
8	MBX_INTA2R1_0C	Clears or indicates interrupt status flag (MBX_INTA2R10) of mailbox 0 from Cortex-A55 Core2 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
9	MBX_INTA2R1_1C	Clears or indicates interrupt status flag (MBX_INTA2R11) of mailbox 1 from Cortex-A55 Core2 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
10	MBX_INTA2R1_2C	Clears or indicates interrupt status flag (MBX_INTA2R12) of mailbox 2 from Cortex-A55 Core2 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
11	MBX_INTA2R1_3C	Clears or indicates interrupt status flag (MBX_INTA2R13) of mailbox 3 from Cortex-A55 Core2 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
12	MBX_INTA3R1_0C	Clears or indicates interrupt status flag (MBX_INTA3R10) of mailbox 0 from Cortex-A55 Core3 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
13	MBX_INTA3R1_1C	Clears or indicates interrupt status flag (MBX_INTA3R11) of mailbox 1 from Cortex-A55 Core3 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
14	MBX_INTA3R1_2C	Clears or indicates interrupt status flag (MBX_INTA3R12) of mailbox 2 from Cortex-A55 Core3 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
15	MBX_INTA3R1_3C	Clears or indicates interrupt status flag (MBX_INTA3R13) of mailbox 3 from Cortex-A55 Core3 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU1 (Target). Only read is available from Cortex-A55 Core0/1/2/3 (Source). Access from other CPUs except source and target is also read only.

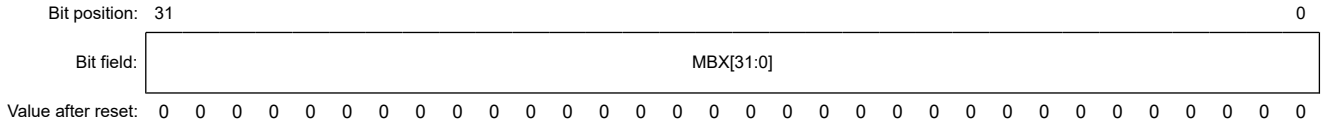
MBXICLRAR1 register is used to clear mailbox interrupt from Cortex-A55 Core0/1/2/3 to Cortex-R52 CPU1. MBX_INTAR1 interrupt is cleared when all the status flags are cleared.

47.3.5 Mailbox Registers between Cortex-R52 CPU0 and CPU1

47.3.5.1 MBXR0R1n : CR52 CPU0 to CR52 CPU1 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x700 + 0x04 × n



Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-R52 CPU0 to Cortex-R52 CPU1	R/W*1

Note 1. Read and write is available from Cortex-R52 CPU0 (Source). Only read is available from Cortex-R52 CPU1 (Target). Access from other CPUs except source and target is also read only.

MBXR0R1n register is used as mailbox from Cortex-R52 CPU0 to Cortex-R52 CPU1.

47.3.5.2 MBXISETR0R1 : CR52 CPU0 to CR52 CPU1 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x710

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR0R1_3S	MBX_INTR0R1_2S	MBX_INTR0R1_1S	MBX_INTR0R1_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR0R1_0S	Sets or indicates interrupt status flag (MBX_INTR0R10) of mailbox 0 from Cortex-R52 CPU0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
1	MBX_INTR0R1_1S	Sets or indicates interrupt status flag (MBX_INTR0R11) of mailbox 1 from Cortex-R52 CPU0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR0R1_2S	Sets or indicates interrupt status flag (MBX_INTR0R12) of mailbox 2 from Cortex-R52 CPU0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR0R1_3S	Sets or indicates interrupt status flag (MBX_INTR0R13) of mailbox 3 from Cortex-R52 CPU0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU0 (Source). Only read is available from Cortex-R52 CPU1 (Target). Access from other CPUs except source and target is also read only.

MBXISETR0R1 register is used to generate mailbox interrupt from Cortex-R52 CPU0 to Cortex-R52 CPU1. MBX_INTR0R1 interrupt is generated as logical OR of MBX_INTR0R10, MBX_INTR0R11, MBX_INTR0R12, or MBX_INTR0R13 status flags.

47.3.5.3 MBXICLRR0R1 : CR52 CPU0 to CR52 CPU1 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x714

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR0R1_3C	MBX_INTR0R1_2C	MBX_INTR0R1_1C	MBX_INTR0R1_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR0R1_0C	Clears or indicates interrupt status flag (MBX_INTR0R10) of mailbox 0 from Cortex-R52 CPU0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR0R1_1C	Clears or indicates interrupt status flag (MBX_INTR0R11) of mailbox 1 from Cortex-R52 CPU0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
2	MBX_INTR0R1_2C	Clears or indicates interrupt status flag (MBX_INTR0R12) of mailbox 2 from Cortex-R52 CPU0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR0R1_3C	Clears or indicates interrupt status flag (MBX_INTR0R13) of mailbox 3 from Cortex-R52 CPU0 to Cortex-R52 CPU1. Write from Cortex-R52 CPU1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU1 (Target). Only read is available from Cortex-R52 CPU0 (Source). Access from other CPUs except source and target is also read only.

MBXICLRR0R1 register is used to clear mailbox interrupt from Cortex-R52 CPU0 to Cortex-R52 CPU1. MBX_INTR0R1 interrupt is cleared when all the status flags are cleared.

47.3.5.4 MBXR1R0n : CR52 CPU1 to CR52 CPU0 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x780 + 0x04 × n

Bit position:	31																													0
Bit field:	MBX[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-R52 CPU1 to Cortex-R52 CPU0	R/W*1

Note 1. Read and write is available from Cortex-R52 CPU1 (Source). Only read is available from Cortex-R52 CPU0 (Target). Access from other CPUs except source and target is also read only.

MBXR1R0n register is used as mailbox from Cortex-R52 CPU1 to Cortex-R52 CPU0.

47.3.5.5 MBXISETR1R0 : CR52 CPU1 to CR52 CPU0 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x790

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR1R0_3S	MBX_INTR1R0_2S	MBX_INTR1R0_1S	MBX_INTR1R0_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR1R0_0S	Sets or indicates interrupt status flag (MBX_INTR1R00) of mailbox 0 from Cortex-R52 CPU1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTR1R0_1S	Sets or indicates interrupt status flag (MBX_INTR1R01) of mailbox 1 from Cortex-R52 CPU1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR1R0_2S	Sets or indicates interrupt status flag (MBX_INTR1R02) of mailbox 2 from Cortex-R52 CPU1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR1R0_3S	Sets or indicates interrupt status flag (MBX_INTR1R03) of mailbox 3 from Cortex-R52 CPU1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU1 (Source). Only read is available from Cortex-R52 CPU0 (Target). Access from other CPUs except source and target is also read only.

MBXISETR1R0 register is used to generate mailbox interrupt from Cortex-R52 CPU1 to Cortex-R52 CPU0. MBX_INTR1R0 interrupt is generated as logical OR of MBX_INTR1R00, MBX_INTR1R01, MBX_INTR1R02, or MBX_INTR1R03 status flags.

47.3.5.6 MBXICLRR1R0 : CR52 CPU1 to CR52 CPU0 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x794

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTR1R0_3C	MBX_INTR1R0_2C	MBX_INTR1R0_1C	MBX_INTR1R0_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTR1R0_0C	Clears or indicates interrupt status flag (MBX_INTR1R00) of mailbox 0 from Cortex-R52 CPU1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1

Bit	Symbol	Function	R/W
1	MBX_INTR1R0_1C	Clears or indicates interrupt status flag (MBX_INTR1R01) of mailbox 1 from Cortex-R52 CPU1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTR1R0_2C	Clears or indicates interrupt status flag (MBX_INTR1R02) of mailbox 2 from Cortex-R52 CPU1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTR1R0_3C	Clears or indicates interrupt status flag (MBX_INTR1R03) of mailbox 3 from Cortex-R52 CPU1 to Cortex-R52 CPU0. Write from Cortex-R52 CPU0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-R52 CPU0 (Target). Only read is available from Cortex-R52 CPU1 (Source). Access from other CPUs except source and target is also read only.

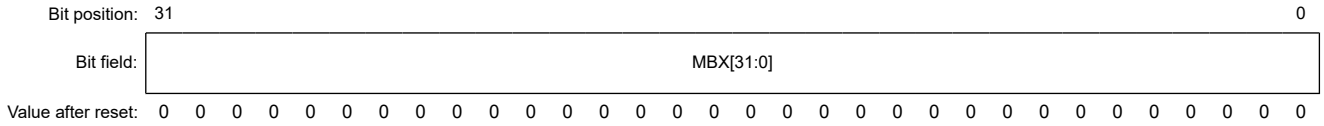
MBXICLRR1R0 register is used to clear mailbox interrupt from Cortex-R52 CPU1 to Cortex-R52 CPU0. MBX_INTR1R0 interrupt is cleared when all the status flags are cleared.

47.3.6 Mailbox Registers among Cortex-A55 Cores

47.3.6.1 MBXA0A1n : CA55 Core0 to CA55 Core1 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x800 + 0x04 × n



Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core0 to Cortex-A55 Core1	R/W*1

Note 1. Read and write is available from Cortex-A55 Core0 (Source). Only read is available from Cortex-A55 Core1 (Target). Access from other CPUs except source and target is also read only.

MBXA0A1n register is used as mailbox from Cortex-A55 Core0 to Cortex-A55 Core1.

47.3.6.2 MBXISETA0A1 : CA55 Core0 to CA55 Core1 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x810

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA0A1_3S	MBX_INTA0A1_2S	MBX_INTA0A1_1S	MBX_INTA0A1_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA0A1_0S	Sets or indicates interrupt status flag (MBX_INTA0A10) of mailbox 0 from Cortex-A55 Core0 to Cortex-A55 Core1. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA0A1_1S	Sets or indicates interrupt status flag (MBX_INTA0A11) of mailbox 1 from Cortex-A55 Core0 to Cortex-A55 Core1. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA0A1_2S	Sets or indicates interrupt status flag (MBX_INTA0A12) of mailbox 2 from Cortex-A55 Core0 to Cortex-A55 Core1. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA0A1_3S	Sets or indicates interrupt status flag (MBX_INTA0A13) of mailbox 3 from Cortex-A55 Core0 to Cortex-A55 Core1. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core0 (Source). Only read is available from Cortex-A55 Core1 (Target). Access from other CPUs except source and target is also read only.

MBXISETA0A1 register is used to generate mailbox interrupt from Cortex-A55 Core0 to Cortex-A55 Core1. MBX_INTA0A1 interrupt is generated as logical OR of MBX_INTA0A10, MBX_INTA0A11, MBX_INTA0A12, or MBX_INTA0A13 status flags.

47.3.6.3 MBXICLRA0A1 : CA55 Core0 to CA55 Core1 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x814

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA0A1_3C	MBX_INTA0A1_2C	MBX_INTA0A1_1C	MBX_INTA0A1_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA0A1_0C	Clears or indicates interrupt status flag (MBX_INTA0A10) of mailbox 0 from Cortex-A55 Core0 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA0A1_1C	Clears or indicates interrupt status flag (MBX_INTA0A11) of mailbox 1 from Cortex-A55 Core0 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA0A1_2C	Clears or indicates interrupt status flag (MBX_INTA0A12) of mailbox 2 from Cortex-A55 Core0 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA0A1_3C	Clears or indicates interrupt status flag (MBX_INTA0A13) of mailbox 3 from Cortex-A55 Core0 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core1 (Target). Only read is available from Cortex-A55 Core0 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA0A1 register is used to clear mailbox interrupt from Cortex-A55 Core0 to Cortex-A55 Core1. MBX_INTA0A1 interrupt is cleared when all the status flags are cleared.

47.3.6.4 MBXA0A2n : CA55 Core0 to CA55 Core2 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x880 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core0 to Cortex-A55 Core2	R/W*1

Note 1. Read and write is available from Cortex-A55 Core0 (Source). Only read is available from Cortex-A55 Core2 (Target). Access from other CPUs except source and target is also read only.

MBXA0A2n register is used as mailbox from Cortex-A55 Core0 to Cortex-A55 Core2.

47.3.6.5 MBXISETA0A2 : CA55 Core0 to CA55 Core2 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x890

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA0A2_3S	MBX_INTA0A2_2S	MBX_INTA0A2_1S	MBX_INTA0A2_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA0A2_0S	Sets or indicates interrupt status flag (MBX_INTA0A20) of mailbox 0 from Cortex-A55 Core0 to Cortex-A55 Core2. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA0A2_1S	Sets or indicates interrupt status flag (MBX_INTA0A21) of mailbox 1 from Cortex-A55 Core0 to Cortex-A55 Core2. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA0A2_2S	Sets or indicates interrupt status flag (MBX_INTA0A22) of mailbox 2 from Cortex-A55 Core0 to Cortex-A55 Core2. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA0A2_3S	Sets or indicates interrupt status flag (MBX_INTA0A23) of mailbox 3 from Cortex-A55 Core0 to Cortex-A55 Core2. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core0 (Source). Only read is available from Cortex-A55 Core2 (Target). Access from other CPUs except source and target is also read only.

MBXISETA0A2 register is used to generate mailbox interrupt from Cortex-A55 Core0 to Cortex-A55 Core2. MBX_INTA0A2 interrupt is generated as logical OR of MBX_INTA0A20, MBX_INTA0A21, MBX_INTA0A22, or MBX_INTA0A23 status flags.

47.3.6.6 MBXICLRA0A2 : CA55 Core0 to CA55 Core2 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x894

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA0A2_3C	MBX_INTA0A2_2C	MBX_INTA0A2_1C	MBX_INTA0A2_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA0A2_0C	Clears or indicates interrupt status flag (MBX_INTA0A20) of mailbox 0 from Cortex-A55 Core0 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA0A2_1C	Clears or indicates interrupt status flag (MBX_INTA0A21) of mailbox 1 from Cortex-A55 Core0 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA0A2_2C	Clears or indicates interrupt status flag (MBX_INTA0A22) of mailbox 2 from Cortex-A55 Core0 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA0A2_3C	Clears or indicates interrupt status flag (MBX_INTA0A23) of mailbox 3 from Cortex-A55 Core0 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core2 (Target). Only read is available from Cortex-A55 Core0 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA0A2 register is used to clear mailbox interrupt from Cortex-A55 Core0 to Cortex-A55 Core2. MBX_INTA0A2 interrupt is cleared when all the status flags are cleared.

47.3.6.7 MBXA0A3n : CA55 Core0 to CA55 Core3 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x900 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core0 to Cortex-A55 Core3	R/W*1

Note 1. Read and write is available from Cortex-A55 Core0 (Source). Only read is available from Cortex-A55 Core3 (Target). Access from other CPUs except source and target is also read only.

MBXA0A3n register is used as mailbox from Cortex-A55 Core0 to Cortex-A55 Core3.

47.3.6.8 MBXISETA0A3 : CA55 Core0 to CA55 Core3 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x910

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA0A3_3S	MBX_INTA0A3_2S	MBX_INTA0A3_1S	MBX_INTA0A3_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA0A3_0S	Sets or indicates interrupt status flag (MBX_INTA0A30) of mailbox 0 from Cortex-A55 Core0 to Cortex-A55 Core3. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA0A3_1S	Sets or indicates interrupt status flag (MBX_INTA0A31) of mailbox 1 from Cortex-A55 Core0 to Cortex-A55 Core3. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA0A3_2S	Sets or indicates interrupt status flag (MBX_INTA0A32) of mailbox 2 from Cortex-A55 Core0 to Cortex-A55 Core3. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA0A3_3S	Sets or indicates interrupt status flag (MBX_INTA0A33) of mailbox 3 from Cortex-A55 Core0 to Cortex-A55 Core3. Write from Cortex-A55 Core0 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core0 (Source). Only read is available from Cortex-A55 Core3 (Target). Access from other CPUs except source and target is also read only.

MBXISETA0A3 register is used to generate mailbox interrupt from Cortex-A55 Core0 to Cortex-A55 Core3. MBX_INTA0A3 interrupt is generated as logical OR of MBX_INTA0A30, MBX_INTA0A31, MBX_INTA0A32, or MBX_INTA0A33 status flags.

47.3.6.9 MBXICLRA0A3 : CA55 Core0 to CA55 Core3 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x914

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA0A3_3C	MBX_INTA0A3_2C	MBX_INTA0A3_1C	MBX_INTA0A3_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA0A3_0C	Clears or indicates interrupt status flag (MBX_INTA0A30) of mailbox 0 from Cortex-A55 Core0 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA0A3_1C	Clears or indicates interrupt status flag (MBX_INTA0A31) of mailbox 1 from Cortex-A55 Core0 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA0A3_2C	Clears or indicates interrupt status flag (MBX_INTA0A32) of mailbox 2 from Cortex-A55 Core0 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA0A3_3C	Clears or indicates interrupt status flag (MBX_INTA0A33) of mailbox 3 from Cortex-A55 Core0 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core3 (Target). Only read is available from Cortex-A55 Core0 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA0A3 register is used to clear mailbox interrupt from Cortex-A55 Core0 to Cortex-A55 Core3. MBX_INTA0A3 interrupt is cleared when all the status flags are cleared.

47.3.6.10 MBXA1A0n : CA55 Core1 to CA55 Core0 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0x980 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core1 to Cortex-A55 Core0	R/W*1

Note 1. Read and write is available from Cortex-A55 Core1 (Source). Only read is available from Cortex-A55 Core0 (Target). Access from other CPUs except source and target is also read only.

MBXA1A0n register is used as mailbox from Cortex-A55 Core1 to Cortex-A55 Core0.

47.3.6.11 MBXISETA1A0 : CA55 Core1 to CA55 Core0 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x990

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA1A0_3S	MBX_INTA1A0_2S	MBX_INTA1A0_1S	MBX_INTA1A0_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA1A0_0S	Sets or indicates interrupt status flag (MBX_INTA1A00) of mailbox 0 from Cortex-A55 Core1 to Cortex-A55 Core0. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA1A0_1S	Sets or indicates interrupt status flag (MBX_INTA1A01) of mailbox 1 from Cortex-A55 Core1 to Cortex-A55 Core0. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA1A0_2S	Sets or indicates interrupt status flag (MBX_INTA1A02) of mailbox 2 from Cortex-A55 Core1 to Cortex-A55 Core0. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA1A0_3S	Sets or indicates interrupt status flag (MBX_INTA1A03) of mailbox 3 from Cortex-A55 Core1 to Cortex-A55 Core0. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core1 (Source). Only read is available from Cortex-A55 Core0 (Target). Access from other CPUs except source and target is also read only.

MBXISETA1A0 register is used to generate mailbox interrupt from Cortex-A55 Core1 to Cortex-A55 Core0. MBX_INTA1A0 interrupt is generated as logical OR of MBX_INTA1A00, MBX_INTA1A01, MBX_INTA1A02, or MBX_INTA1A03 status flags.

47.3.6.12 MBXICLRA1A0 : CA55 Core1 to CA55 Core0 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0x994

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA1A0_3C	MBX_INTA1A0_2C	MBX_INTA1A0_1C	MBX_INTA1A0_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA1A0_0C	Clears or indicates interrupt status flag (MBX_INTA1A00) of mailbox 0 from Cortex-A55 Core1 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA1A0_1C	Clears or indicates interrupt status flag (MBX_INTA1A01) of mailbox 1 from Cortex-A55 Core1 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA1A0_2C	Clears or indicates interrupt status flag (MBX_INTA1A02) of mailbox 2 from Cortex-A55 Core1 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA1A0_3C	Clears or indicates interrupt status flag (MBX_INTA1A03) of mailbox 3 from Cortex-A55 Core1 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core0 (Target). Only read is available from Cortex-A55 Core1 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA1A0 register is used to clear mailbox interrupt from Cortex-A55 Core1 to Cortex-A55 Core0. MBX_INTA1A0 interrupt is cleared when all the status flags are cleared.

47.3.6.13 MBXA1A2n : CA55 Core1 to CA55 Core2 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0xA00 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core1 to Cortex-A55 Core2	R/W*1

Note 1. Read and write is available from Cortex-A55 Core1 (Source). Only read is available from Cortex-A55 Core2 (Target). Access from other CPUs except source and target is also read only.

MBXA1A2n register is used as mailbox from Cortex-A55 Core1 to Cortex-A55 Core2.

47.3.6.14 MBXISETA1A2 : CA55 Core1 to CA55 Core2 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xA10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA1A2_3S	MBX_INTA1A2_2S	MBX_INTA1A2_1S	MBX_INTA1A2_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA1A2_0S	Sets or indicates interrupt status flag (MBX_INTA1A20) of mailbox 0 from Cortex-A55 Core1 to Cortex-A55 Core2. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA1A2_1S	Sets or indicates interrupt status flag (MBX_INTA1A21) of mailbox 1 from Cortex-A55 Core1 to Cortex-A55 Core2. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA1A2_2S	Sets or indicates interrupt status flag (MBX_INTA1A22) of mailbox 2 from Cortex-A55 Core1 to Cortex-A55 Core2. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA1A2_3S	Sets or indicates interrupt status flag (MBX_INTA1A23) of mailbox 3 from Cortex-A55 Core1 to Cortex-A55 Core2. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core1 (Source). Only read is available from Cortex-A55 Core2 (Target). Access from other CPUs except source and target is also read only.

MBXISETA1A2 register is used to generate mailbox interrupt from Cortex-A55 Core1 to Cortex-A55 Core2. MBX_INTA1A2 interrupt is generated as logical OR of MBX_INTA1A20, MBX_INTA1A21, MBX_INTA1A22, or MBX_INTA1A23 status flags.

47.3.6.15 MBXICLRA1A2 : CA55 Core1 to CA55 Core2 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xA14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA1A2_3C	MBX_INTA1A2_2C	MBX_INTA1A2_1C	MBX_INTA1A2_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA1A2_0C	Clears or indicates interrupt status flag (MBX_INTA1A20) of mailbox 0 from Cortex-A55 Core1 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA1A2_1C	Clears or indicates interrupt status flag (MBX_INTA1A21) of mailbox 1 from Cortex-A55 Core1 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA1A2_2C	Clears or indicates interrupt status flag (MBX_INTA1A22) of mailbox 2 from Cortex-A55 Core1 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA1A2_3C	Clears or indicates interrupt status flag (MBX_INTA1A23) of mailbox 3 from Cortex-A55 Core1 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core2 (Target). Only read is available from Cortex-A55 Core1 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA1A2 register is used to clear mailbox interrupt from Cortex-A55 Core1 to Cortex-A55 Core2. MBX_INTA1A2 interrupt is cleared when all the status flags are cleared.

47.3.6.16 MBXA1A3n : CA55 Core1 to CA55 Core3 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0xA80 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core1 to Cortex-A55 Core3	R/W*1

Note 1. Read and write is available from Cortex-A55 Core1 (Source). Only read is available from Cortex-A55 Core3 (Target). Access from other CPUs except source and target is also read only.

MBXA1A3n register is used as mailbox from Cortex-A55 Core1 to Cortex-A55 Core3.

47.3.6.17 MBXISETA1A3 : CA55 Core1 to CA55 Core3 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xA90

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA1A3_3S	MBX_INTA1A3_2S	MBX_INTA1A3_1S	MBX_INTA1A3_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA1A3_0S	Sets or indicates interrupt status flag (MBX_INTA1A30) of mailbox 0 from Cortex-A55 Core1 to Cortex-A55 Core3. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA1A3_1S	Sets or indicates interrupt status flag (MBX_INTA1A31) of mailbox 1 from Cortex-A55 Core1 to Cortex-A55 Core3. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA1A3_2S	Sets or indicates interrupt status flag (MBX_INTA1A32) of mailbox 2 from Cortex-A55 Core1 to Cortex-A55 Core3. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA1A3_3S	Sets or indicates interrupt status flag (MBX_INTA1A33) of mailbox 3 from Cortex-A55 Core1 to Cortex-A55 Core3. Write from Cortex-A55 Core1 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core1 (Source). Only read is available from Cortex-A55 Core3 (Target). Access from other CPUs except source and target is also read only.

MBXISETA1A3 register is used to generate mailbox interrupt from Cortex-A55 Core1 to Cortex-A55 Core3. MBX_INTA1A3 interrupt is generated as logical OR of MBX_INTA1A30, MBX_INTA1A31, MBX_INTA1A32, or MBX_INTA1A33 status flags.

47.3.6.18 MBXICLRA1A3 : CA55 Core1 to CA55 Core3 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xA94

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA1A3_3C	MBX_INTA1A3_2C	MBX_INTA1A3_1C	MBX_INTA1A3_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA1A3_0C	Clears or indicates interrupt status flag (MBX_INTA1A30) of mailbox 0 from Cortex-A55 Core1 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA1A3_1C	Clears or indicates interrupt status flag (MBX_INTA1A31) of mailbox 1 from Cortex-A55 Core1 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA1A3_2C	Clears or indicates interrupt status flag (MBX_INTA1A32) of mailbox 2 from Cortex-A55 Core1 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA1A3_3C	Clears or indicates interrupt status flag (MBX_INTA1A33) of mailbox 3 from Cortex-A55 Core1 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core3 (Target). Only read is available from Cortex-A55 Core1 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA1A3 register is used to clear mailbox interrupt from Cortex-A55 Core1 to Cortex-A55 Core3. MBX_INTA1A3 interrupt is cleared when all the status flags are cleared.

47.3.6.19 MBXA2A0n : CA55 Core2 to CA55 Core0 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0xB00 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core2 to Cortex-A55 Core0	R/W*1

Note 1. Read and write is available from Cortex-A55 Core2 (Source). Only read is available from Cortex-A55 Core0 (Target). Access from other CPUs except source and target is also read only.

MBXA2A0n register is used as mailbox from Cortex-A55 Core2 to Cortex-A55 Core0.

47.3.6.20 MBXISETA2A0 : CA55 Core2 to CA55 Core0 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xB10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA2A0_3S	MBX_INTA2A0_2S	MBX_INTA2A0_1S	MBX_INTA2A0_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA2A0_0S	Sets or indicates interrupt status flag (MBX_INTA2A00) of mailbox 0 from Cortex-A55 Core2 to Cortex-A55 Core0. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA2A0_1S	Sets or indicates interrupt status flag (MBX_INTA2A01) of mailbox 1 from Cortex-A55 Core2 to Cortex-A55 Core0. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA2A0_2S	Sets or indicates interrupt status flag (MBX_INTA2A02) of mailbox 2 from Cortex-A55 Core2 to Cortex-A55 Core0. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA2A0_3S	Sets or indicates interrupt status flag (MBX_INTA2A03) of mailbox 3 from Cortex-A55 Core2 to Cortex-A55 Core0. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core2 (Source). Only read is available from Cortex-A55 Core0 (Target). Access from other CPUs except source and target is also read only.

MBXISETA2A0 register is used to generate mailbox interrupt from Cortex-A55 Core2 to Cortex-A55 Core0. MBX_INTA2A0 interrupt is generated as logical OR of MBX_INTA2A00, MBX_INTA2A01, MBX_INTA2A02, or MBX_INTA2A03 status flags.

47.3.6.21 MBXICLRA2A0 : CA55 Core2 to CA55 Core0 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xB14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA2A0_3C	MBX_INTA2A0_2C	MBX_INTA2A0_1C	MBX_INTA2A0_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA2A0_0C	Clears or indicates interrupt status flag (MBX_INTA2A00) of mailbox 0 from Cortex-A55 Core2 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA2A0_1C	Clears or indicates interrupt status flag (MBX_INTA2A01) of mailbox 1 from Cortex-A55 Core2 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA2A0_2C	Clears or indicates interrupt status flag (MBX_INTA2A02) of mailbox 2 from Cortex-A55 Core2 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA2A0_3C	Clears or indicates interrupt status flag (MBX_INTA2A03) of mailbox 3 from Cortex-A55 Core2 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core0 (Target). Only read is available from Cortex-A55 Core2 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA2A0 register is used to clear mailbox interrupt from Cortex-A55 Core2 to Cortex-A55 Core0. MBX_INTA2A0 interrupt is cleared when all the status flags are cleared.

47.3.6.22 MBXA2A1n : CA55 Core2 to CA55 Core1 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0xB80 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core2 to Cortex-A55 Core1	R/W*1

Note 1. Read and write is available from Cortex-A55 Core2 (Source). Only read is available from Cortex-A55 Core1 (Target). Access from other CPUs except source and target is also read only.

MBXA2A1n register is used as mailbox from Cortex-A55 Core2 to Cortex-A55 Core1.

47.3.6.23 MBXISETA2A1 : CA55 Core2 to CA55 Core1 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xB90

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA2A1_3S	MBX_INTA2A1_2S	MBX_INTA2A1_1S	MBX_INTA2A1_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA2A1_0S	Sets or indicates interrupt status flag (MBX_INTA2A10) of mailbox 0 from Cortex-A55 Core2 to Cortex-A55 Core1. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA2A1_1S	Sets or indicates interrupt status flag (MBX_INTA2A11) of mailbox 1 from Cortex-A55 Core2 to Cortex-A55 Core1. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA2A1_2S	Sets or indicates interrupt status flag (MBX_INTA2A12) of mailbox 2 from Cortex-A55 Core2 to Cortex-A55 Core1. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA2A1_3S	Sets or indicates interrupt status flag (MBX_INTA2A13) of mailbox 3 from Cortex-A55 Core2 to Cortex-A55 Core1. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core2 (Source). Only read is available from Cortex-A55 Core1 (Target). Access from other CPUs except source and target is also read only.

MBXISETA2A1 register is used to generate mailbox interrupt from Cortex-A55 Core2 to Cortex-A55 Core1. MBX_INTA2A1 interrupt is generated as logical OR of MBX_INTA2A10, MBX_INTA2A11, MBX_INTA2A12, or MBX_INTA2A13 status flags.

47.3.6.24 MBXICLRA2A1 : CA55 Core2 to CA55 Core1 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xB94

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA2A1_3C	MBX_INTA2A1_2C	MBX_INTA2A1_1C	MBX_INTA2A1_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA2A1_0C	Clears or indicates interrupt status flag (MBX_INTA2A10) of mailbox 0 from Cortex-A55 Core2 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA2A1_1C	Clears or indicates interrupt status flag (MBX_INTA2A11) of mailbox 1 from Cortex-A55 Core2 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA2A1_2C	Clears or indicates interrupt status flag (MBX_INTA2A12) of mailbox 2 from Cortex-A55 Core2 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA2A1_3C	Clears or indicates interrupt status flag (MBX_INTA2A13) of mailbox 3 from Cortex-A55 Core2 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core1 (Target). Only read is available from Cortex-A55 Core2 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA2A1 register is used to clear mailbox interrupt from Cortex-A55 Core2 to Cortex-A55 Core1. MBX_INTA2A1 interrupt is cleared when all the status flags are cleared.

47.3.6.25 MBXA2A3n : CA55 Core2 to CA55 Core3 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0xC00 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core2 to Cortex-A55 Core3	R/W*1

Note 1. Read and write is available from Cortex-A55 Core2 (Source). Only read is available from Cortex-A55 Core3 (Target). Access from other CPUs except source and target is also read only.

MBXA2A3n register is used as mailbox from Cortex-A55 Core2 to Cortex-A55 Core3.

47.3.6.26 MBXISETA2A3 : CA55 Core2 to CA55 Core3 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xC10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA2A3_3S	MBX_INTA2A3_2S	MBX_INTA2A3_1S	MBX_INTA2A3_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA2A3_0S	Sets or indicates interrupt status flag (MBX_INTA2A30) of mailbox 0 from Cortex-A55 Core2 to Cortex-A55 Core3. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA2A3_1S	Sets or indicates interrupt status flag (MBX_INTA2A31) of mailbox 1 from Cortex-A55 Core2 to Cortex-A55 Core3. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA2A3_2S	Sets or indicates interrupt status flag (MBX_INTA2A32) of mailbox 2 from Cortex-A55 Core2 to Cortex-A55 Core3. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA2A3_3S	Sets or indicates interrupt status flag (MBX_INTA2A33) of mailbox 3 from Cortex-A55 Core2 to Cortex-A55 Core3. Write from Cortex-A55 Core2 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core2 (Source). Only read is available from Cortex-A55 Core3 (Target). Access from other CPUs except source and target is also read only.

MBXISETA2A3 register is used to generate mailbox interrupt from Cortex-A55 Core2 to Cortex-A55 Core3. MBX_INTA2A3 interrupt is generated as logical OR of MBX_INTA2A30, MBX_INTA2A31, MBX_INTA2A32, or MBX_INTA2A33 status flags.

47.3.6.27 MBXICLRA2A3 : CA55 Core2 to CA55 Core3 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xC14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA2A3_3C	MBX_INTA2A3_2C	MBX_INTA2A3_1C	MBX_INTA2A3_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA2A3_0C	Clears or indicates interrupt status flag (MBX_INTA2A30) of mailbox 0 from Cortex-A55 Core2 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA2A3_1C	Clears or indicates interrupt status flag (MBX_INTA2A31) of mailbox 1 from Cortex-A55 Core2 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA2A3_2C	Clears or indicates interrupt status flag (MBX_INTA2A32) of mailbox 2 from Cortex-A55 Core2 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA2A3_3C	Clears or indicates interrupt status flag (MBX_INTA2A33) of mailbox 3 from Cortex-A55 Core2 to Cortex-A55 Core3. Write from Cortex-A55 Core3 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core3 (Target). Only read is available from Cortex-A55 Core2 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA2A3 register is used to clear mailbox interrupt from Cortex-A55 Core2 to Cortex-A55 Core3. MBX_INTA2A3 interrupt is cleared when all the status flags are cleared.

47.3.6.28 MBXA3A0n : CA55 Core3 to CA55 Core0 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0xC80 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core3 to Cortex-A55 Core0	R/W*1

Note 1. Read and write is available from Cortex-A55 Core3 (Source). Only read is available from Cortex-A55 Core0 (Target). Access from other CPUs except source and target is also read only.

MBXA3A0n register is used as mailbox from Cortex-A55 Core3 to Cortex-A55 Core0.

47.3.6.29 MBXISETA3A0 : CA55 Core3 to CA55 Core0 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xC90

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA3A0_3S	MBX_INTA3A0_2S	MBX_INTA3A0_1S	MBX_INTA3A0_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA3A0_0S	Sets or indicates interrupt status flag (MBX_INTA3A00) of mailbox 0 from Cortex-A55 Core3 to Cortex-A55 Core0. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA3A0_1S	Sets or indicates interrupt status flag (MBX_INTA3A01) of mailbox 1 from Cortex-A55 Core3 to Cortex-A55 Core0. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA3A0_2S	Sets or indicates interrupt status flag (MBX_INTA3A02) of mailbox 2 from Cortex-A55 Core3 to Cortex-A55 Core0. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA3A0_3S	Sets or indicates interrupt status flag (MBX_INTA3A03) of mailbox 3 from Cortex-A55 Core3 to Cortex-A55 Core0. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core3 (Source). Only read is available from Cortex-A55 Core0 (Target). Access from other CPUs except source and target is also read only.

MBXISETA3A0 register is used to generate mailbox interrupt from Cortex-A55 Core3 to Cortex-A55 Core0. MBX_INTA3A0 interrupt is generated as logical OR of MBX_INTA3A00, MBX_INTA3A01, MBX_INTA3A02, or MBX_INTA3A03 status flags.

47.3.6.30 MBXICLRA3A0 : CA55 Core3 to CA55 Core0 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xC94

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA3A0_3C	MBX_INTA3A0_2C	MBX_INTA3A0_1C	MBX_INTA3A0_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA3A0_0C	Clears or indicates interrupt status flag (MBX_INTA3A00) of mailbox 0 from Cortex-A55 Core3 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA3A0_1C	Clears or indicates interrupt status flag (MBX_INTA3A01) of mailbox 1 from Cortex-A55 Core3 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA3A0_2C	Clears or indicates interrupt status flag (MBX_INTA3A02) of mailbox 2 from Cortex-A55 Core3 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA3A0_3C	Clears or indicates interrupt status flag (MBX_INTA3A03) of mailbox 3 from Cortex-A55 Core3 to Cortex-A55 Core0. Write from Cortex-A55 Core0 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core0 (Target). Only read is available from Cortex-A55 Core3 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA3A0 register is used to clear mailbox interrupt from Cortex-A55 Core3 to Cortex-A55 Core0. MBX_INTA3A0 interrupt is cleared when all the status flags are cleared.

47.3.6.31 MBXA3A1n : CA55 Core3 to CA55 Core1 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0xD00 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core3 to Cortex-A55 Core1	R/W*1

Note 1. Read and write is available from Cortex-A55 Core3 (Source). Only read is available from Cortex-A55 Core1 (Target). Access from other CPUs except source and target is also read only.

MBXA3A1n register is used as mailbox from Cortex-A55 Core3 to Cortex-A55 Core1.

47.3.6.32 MBXISETA3A1 : CA55 Core3 to CA55 Core1 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xD10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA3A1_3S	MBX_INTA3A1_2S	MBX_INTA3A1_1S	MBX_INTA3A1_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA3A1_0S	Sets or indicates interrupt status flag (MBX_INTA3A10) of mailbox 0 from Cortex-A55 Core3 to Cortex-A55 Core1. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA3A1_1S	Sets or indicates interrupt status flag (MBX_INTA3A11) of mailbox 1 from Cortex-A55 Core3 to Cortex-A55 Core1. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA3A1_2S	Sets or indicates interrupt status flag (MBX_INTA3A12) of mailbox 2 from Cortex-A55 Core3 to Cortex-A55 Core1. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA3A1_3S	Sets or indicates interrupt status flag (MBX_INTA3A13) of mailbox 3 from Cortex-A55 Core3 to Cortex-A55 Core1. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core3 (Source). Only read is available from Cortex-A55 Core1 (Target). Access from other CPUs except source and target is also read only.

MBXISETA3A1 register is used to generate mailbox interrupt from Cortex-A55 Core3 to Cortex-A55 Core1. MBX_INTA3A1 interrupt is generated as logical OR of MBX_INTA3A10, MBX_INTA3A11, MBX_INTA3A12, or MBX_INTA3A13 status flags.

47.3.6.33 MBXICLRA3A1 : CA55 Core3 to CA55 Core1 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xD14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA3A1_3C	MBX_INTA3A1_2C	MBX_INTA3A1_1C	MBX_INTA3A1_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA3A1_0C	Clears or indicates interrupt status flag (MBX_INTA3A10) of mailbox 0 from Cortex-A55 Core3 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA3A1_1C	Clears or indicates interrupt status flag (MBX_INTA3A11) of mailbox 1 from Cortex-A55 Core3 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA3A1_2C	Clears or indicates interrupt status flag (MBX_INTA3A12) of mailbox 2 from Cortex-A55 Core3 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA3A1_3C	Clears or indicates interrupt status flag (MBX_INTA3A13) of mailbox 3 from Cortex-A55 Core3 to Cortex-A55 Core1. Write from Cortex-A55 Core1 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core1 (Target). Only read is available from Cortex-A55 Core3 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA3A1 register is used to clear mailbox interrupt from Cortex-A55 Core3 to Cortex-A55 Core1. MBX_INTA3A1 interrupt is cleared when all the status flags are cleared.

47.3.6.34 MBXA3A2n : CA55 Core3 to CA55 Core2 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000

Offset address: 0xD80 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from Cortex-A55 Core3 to Cortex-A55 Core2	R/W*1

Note 1. Read and write is available from Cortex-A55 Core3 (Source). Only read is available from Cortex-A55 Core2 (Target). Access from other CPUs except source and target is also read only.

MBXA3A2n register is used as mailbox from Cortex-A55 Core3 to Cortex-A55 Core2.

47.3.6.35 MBXISETA3A2 : CA55 Core3 to CA55 Core2 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xD90

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA3A2_3S	MBX_INTA3A2_2S	MBX_INTA3A2_1S	MBX_INTA3A2_0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA3A2_0S	Sets or indicates interrupt status flag (MBX_INTA3A20) of mailbox 0 from Cortex-A55 Core3 to Cortex-A55 Core2. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA3A2_1S	Sets or indicates interrupt status flag (MBX_INTA3A21) of mailbox 1 from Cortex-A55 Core3 to Cortex-A55 Core2. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA3A2_2S	Sets or indicates interrupt status flag (MBX_INTA3A22) of mailbox 2 from Cortex-A55 Core3 to Cortex-A55 Core2. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA3A2_3S	Sets or indicates interrupt status flag (MBX_INTA3A23) of mailbox 3 from Cortex-A55 Core3 to Cortex-A55 Core2. Write from Cortex-A55 Core3 (Source): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is set. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core3 (Source). Only read is available from Cortex-A55 Core2 (Target). Access from other CPUs except source and target is also read only.

MBXISETA3A2 register is used to generate mailbox interrupt from Cortex-A55 Core3 to Cortex-A55 Core2. MBX_INTA3A2 interrupt is generated as logical OR of MBX_INTA3A20, MBX_INTA3A21, MBX_INTA3A22, or MBX_INTA3A23 status flags.

47.3.6.36 MBXICLRA3A2 : CA55 Core3 to CA55 Core2 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000

Offset address: 0xD94

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INTA3A2_3C	MBX_INTA3A2_2C	MBX_INTA3A2_1C	MBX_INTA3A2_0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INTA3A2_0C	Clears or indicates interrupt status flag (MBX_INTA3A20) of mailbox 0 from Cortex-A55 Core3 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
1	MBX_INTA3A2_1C	Clears or indicates interrupt status flag (MBX_INTA3A21) of mailbox 1 from Cortex-A55 Core3 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
2	MBX_INTA3A2_2C	Clears or indicates interrupt status flag (MBX_INTA3A22) of mailbox 2 from Cortex-A55 Core3 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
3	MBX_INTA3A2_3C	Clears or indicates interrupt status flag (MBX_INTA3A23) of mailbox 3 from Cortex-A55 Core3 to Cortex-A55 Core2. Write from Cortex-A55 Core2 (Target): 0: No operation. Interrupt status flag is not changed. 1: Interrupt status flag is cleared. Read: Indicates interrupt status flag	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read and write is available from Cortex-A55 Core2 (Target). Only read is available from Cortex-A55 Core3 (Source). Access from other CPUs except source and target is also read only.

MBXICLRA3A2 register is used to clear mailbox interrupt from Cortex-A55 Core3 to Cortex-A55 Core2. MBX_INTA3A2 interrupt is cleared when all the status flags are cleared.

47.4 Operation

47.4.1 Semaphore

47.4.1.1 Between external host and internal CPU

SEMn registers are used to perform up to 8 semaphores for synchronization and exclusive control between internal CPU (Cortex-A55/Cortex-R52) and external host CPU.

SEM bit in SEMn register is used to set the status of the resource shared between CPUs to indicate whether the resource is being used or not. Writing 1 to this bit maintains its value. When the read clear function is enabled (SEMRCEN.SEMRCENn bit = 1), reading SEM bit of corresponding SEM register clears its value to 0.

When the SEM0 register is used for semaphore control for a resource A, for example, the following procedure takes place, where CPU0 is Cortex-R52 and CPU1 is external host CPU or vice versa:

1. Use the initialization routine of CPU0 to set the SEMRCEN.SEMRCEN0 bit to 1 (this enables the read clear function for SEM0 register).
2. Set the SEM bit to 1 (resource not being used) in the SEM0 register.
3. For example, when CPU1 attempts to use the resource A, CPU1 reads the SEM0 register repeatedly until 1 is read.
4. When SEM0.SEM = 1 is read, CPU1 confirms that the resource is not being used.
   This operation clears the SEM0.SEM bit to 0.
5. Then CPU1 uses the resource A.
   If CPU0 attempts to use the resource A while CPU1 is using it, 0 is read from the SEM0.SEM bit (resource A being used by CPU1). CPU0 repeats read operation and waits until the resource is released.
6. When CPU1 completes its use of the resource A, 1 is written in the SEM0.SEM bit to indicate that the resource A is released.
7. If CPU0 attempts to use the resource A after the above step (6), it confirms whether the resource is used or not by reading the SEM0.SEM bit. When the SEM0.SEM bit is 1, CPU0 starts to use the resource A. This operation clears the SEM0.SEM bit to 0.

CoreSight AXI-AP can also access these registers. Access property is same as Cortex-A55/Cortex-R52.

47.4.1.2 Among internal CPUs

SEMARn registers are used to perform up to 8 semaphores for synchronization and exclusive control among internal CPUs (Cortex-A55, Cortex-R52 CPU0 and CPU1). Features and usage are same as SEMn registers. CoreSight AXI-AP can also access these registers. Access property is same as CPU.

47.4.2 Mailbox

47.4.2.1 Between external host and internal CPU

MBXH2Cn and MBXC2Hn register are used to perform up to 4 32-bit mailboxes for communication between internal CPU (Cortex-A55/Cortex-R52) and external host CPU. MBXH2Cn registers are mailboxes from external host CPU to Cortex-A55/Cortex-R52 and MBXC2Hn registers are vice versa. CoreSight AXI-AP can also access these registers. Access property is same as Cortex-A55/Cortex-R52.

47.4.2.2 Among internal CPUs

Mailboxes among internal CPUs are available each combination of source and target CPU and core. Table 47.6 lists the mailbox among internal CPUs. Up to 4 32-bit mailboxes are supported each combination. Read and write is available from source CPU. Only read is available from target CPU. Access from other CPUs except source and target is also read only. CoreSight AXI-AP can also access these registers by read only.

Table 47.6 Mailbox among internal CPUs (1 of 2)

Source CPU	Target CPU	Mailbox registers
Cortex-R52 CPU0	Cortex-A55 Core0	MBXR0A0n
	Cortex-A55 Core1	MBXR0A1n
	Cortex-A55 Core2	MBXR0A2n
	Cortex-A55 Core3	MBXR0A3n

Table 47.6 Mailbox among internal CPUs (2 of 2)

Source CPU	Target CPU	Mailbox registers
Cortex-R52 CPU1	Cortex-A55 Core0	MBXR1A0n
	Cortex-A55 Core1	MBXR1A1n
	Cortex-A55 Core2	MBXR1A2n
	Cortex-A55 Core3	MBXR1A3n
Cortex-A55 Core0	Cortex-R52 CPU0	MBXA0R0n
Cortex-A55 Core1		MBXA1R0n
Cortex-A55 Core2		MBXA2R0n
Cortex-A55 Core3		MBXA3R0n
Cortex-A55 Core0	Cortex-R52 CPU1	MBXA0R1n
Cortex-A55 Core1		MBXA1R1n
Cortex-A55 Core2		MBXA2R1n
Cortex-A55 Core3		MBXA3R1n
Cortex-R52 CPU0	Cortex-R52 CPU1	MBXR0R1n
Cortex-R52 CPU1	Cortex-R52 CPU0	MBXR1R0n
Cortex-A55 Core0	Cortex-A55 Core1	MBXA0A1n
	Cortex-A55 Core2	MBXA0A2n
	Cortex-A55 Core3	MBXA0A3n
Cortex-A55 Core1	Cortex-A55 Core0	MBXA1A0n
	Cortex-A55 Core2	MBXA1A2n
	Cortex-A55 Core3	MBXA1A3n
Cortex-A55 Core2	Cortex-A55 Core0	MBXA2A0n
	Cortex-A55 Core1	MBXA2A1n
	Cortex-A55 Core3	MBXA2A3n
Cortex-A55 Core3	Cortex-A55 Core0	MBXA3A0n
	Cortex-A55 Core1	MBXA3A1n
	Cortex-A55 Core2	MBXA3A2n

47.4.3 Interrupt

Interrupt signals are available to internal CPUs and external host CPU (target CPU). Up to 4 interrupt status flags are available related to 4 mailbox registers but can be set independent of use of mailbox. Mailbox interrupt set registers are used to set interrupt status flag by writing 1 to corresponding bit. Writing 0 to corresponding bit is to retain the current interrupt status flag of the bit. Generated interrupt is logical NOR of interrupt status flags for external host CPU and logical OR for among internal CPUs.

Mailbox interrupt clear registers are used to clear interrupt status flag by writing 1 to corresponding bit. Current interrupt status flag can be confirmed by reading these registers. Interrupt is cleared when all the status flags are cleared.

Table 47.7 MBXSEM Interrupt Source from external host to internal CPU

Name	Interrupt sources	Set register	Clear register
MBX_INT0	Mailbox (Host CPU to Cortex-A55/Cortex-R52) interrupt 0	MBXISETH2C	MBXICLRH2C
MBX_INT1	Mailbox (Host CPU to Cortex-A55/Cortex-R52) interrupt 1		
MBX_INT2	Mailbox (Host CPU to Cortex-A55/Cortex-R52) interrupt 2		
MBX_INT3	Mailbox (Host CPU to Cortex-A55/Cortex-R52) interrupt 3		

Table 47.8 MBXSEM Interrupt Source from internal CPU to external host CPU

Name	Interrupt sources	Set register	Clear register
MBX_HINT0	Mailbox (Cortex-A55/Cortex-R52 to Host CPU) interrupt 0	MBXISETC2H	MBXICLRC2H
MBX_HINT1	Mailbox (Cortex-A55/Cortex-R52 to Host CPU) interrupt 1		
MBX_HINT2	Mailbox (Cortex-A55/Cortex-R52 to Host CPU) interrupt 2		
MBX_HINT3	Mailbox (Cortex-A55/Cortex-R52 to Host CPU) interrupt 3		

Note: MBX_HINT[3:0] are combined to one output interrupt signal (MBX_HINT#) by logical NOR.

Table 47.9 MBXSEM Interrupt Source from Cortex-R52 to Cortex-A55

Name	Interrupt sources	Set register	Clear register
MBX_INTR0A0	Mailbox (Cortex-R52 CPU0 to Cortex-A55 Core0) interrupt	MBXISETR0A0	MBXICLRR0A0
MBX_INTR0A1	Mailbox (Cortex-R52 CPU0 to Cortex-A55 Core1) interrupt	MBXISETR0A1	MBXICLRR0A1
MBX_INTR0A2	Mailbox (Cortex-R52 CPU0 to Cortex-A55 Core2) interrupt	MBXISETR0A2	MBXICLRR0A2
MBX_INTR0A3	Mailbox (Cortex-R52 CPU0 to Cortex-A55 Core3) interrupt	MBXISETR0A3	MBXICLRR0A3
MBX_INTR1A0	Mailbox (Cortex-R52 CPU1 to Cortex-A55 Core0) interrupt	MBXISETR1A0	MBXICLRR1A0
MBX_INTR1A1	Mailbox (Cortex-R52 CPU1 to Cortex-A55 Core1) interrupt	MBXISETR1A1	MBXICLRR1A1
MBX_INTR1A2	Mailbox (Cortex-R52 CPU1 to Cortex-A55 Core2) interrupt	MBXISETR1A2	MBXICLRR1A2
MBX_INTR1A3	Mailbox (Cortex-R52 CPU1 to Cortex-A55 Core3) interrupt	MBXISETR1A3	MBXICLRR1A3

Table 47.10 MBXSEM Interrupt Source from Cortex-A55 to Cortex-R52

Name	Interrupt sources	Set register	Clear register
MBX_INTAR0	Mailbox (Cortex-A55 to Cortex-R52 CPU0) interrupt	MBXISETA0R0 MBXISETA1R0 MBXISETA2R0 MBXISETA3R0	MBXICLRAR0
MBX_INTAR1	Mailbox (Cortex-A55 to Cortex-R52 CPU1) interrupt	MBXISETA0R1 MBXISETA1R1 MBXISETA2R1 MBXISETA3R1	MBXICLRAR0

Table 47.11 MBXSEM Interrupt Source between Cortex-R52 CPU0 and CPU1

Name	Interrupt sources	Set register	Clear register
MBX_INTR0R1	Mailbox (Cortex-R52 CPU0 to Cortex-R52 CPU1) interrupt	MBXISETR0R1	MBXICLRR0R1
MBX_INTR1R0	Mailbox (Cortex-R52 CPU1 to Cortex-R52 CPU0) interrupt	MBXISETR1R0	MBXICLRR1R0

Table 47.12 MBXSEM Interrupt Source among Cortex-A55 Cores (1 of 2)

Name	Interrupt sources	Set register	Clear register
MBX_INTA0A1	Mailbox (CA55 core0 to CA55 core1) interrupt	MBXISETA0A1	MBXICLRA0A1
MBX_INTA0A2	Mailbox (CA55 core0 to CA55 core2) interrupt	MBXISETA0A2	MBXICLRA0A2
MBX_INTA0A3	Mailbox (CA55 core0 to CA55 core3) interrupt	MBXISETA0A3	MBXICLRA0A3
MBX_INTA1A0	Mailbox (CA55 core1 to CA55 core0) interrupt	MBXISETA1A0	MBXICLRA1A0
MBX_INTA1A2	Mailbox (CA55 core1 to CA55 core2) interrupt	MBXISETA1A2	MBXICLRA1A2
MBX_INTA1A3	Mailbox (CA55 core1 to CA55 core3) interrupt	MBXISETA1A3	MBXICLRA1A3
MBX_INTA2A0	Mailbox (CA55 core2 to CA55 core0) interrupt	MBXISETA2A0	MBXICLRA2A0
MBX_INTA2A1	Mailbox (CA55 core2 to CA55 core1) interrupt	MBXISETA2A1	MBXICLRA2A1
MBX_INTA2A3	Mailbox (CA55 core2 to CA55 core3) interrupt	MBXISETA2A3	MBXICLRA2A3
MBX_INTA3A0	Mailbox (CA55 core3 to CA55 core0) interrupt	MBXISETA3A0	MBXICLRA3A0
MBX_INTA3A1	Mailbox (CA55 core3 to CA55 core1) interrupt	MBXISETA3A1	MBXICLRA3A1

Table 47.12 MBXSEM Interrupt Source among Cortex-A55 Cores (2 of 2)

Name	Interrupt sources	Set register	Clear register
MBX_INTA3A2	Mailbox (CA55 core3 to CA55 core2) interrupt	MBXISETA3A2	MBXICLRA3A2

48. Encoder Interface Subsystem

48.1 Overview

Encoder interface subsystem consists of EnDat 2.2 (ENDAT), BiSS-C (BISS), HIPERFACE DSL (HDSL), and A-format (AFMT). The encoder interface I/O pins are shared with these modules, and one module is statically selected for each unit. Table 48.1 describes the encoder interface subsystem specifications and Figure 48.1 shows a block diagram of the encoder interface subsystem.

Serial communications interface for encoder interface (SCIE) is also used for general encoder interface. It is not included in encoder interface subsystem but its I/O pins are mapped to the same pins as encoder interface I/O.

Interrupts from all the modules including SCIE are shared to ENCIFn_INT[3:0] in interrupt controller (ICU), and interrupts from one module are statically selected for each unit. For details, see section 12.4.1. Event Handling.

Table 48.1 Encoder interface subsystem specifications

Item	Description
Number of units	16 units (12 in LLPP0 and 4 in LLPP1)
Supported encoder interface	• EnDat 2.2 (ENDAT) • BiSS-C (BISS) • HIPERFACE DSL (HDSL) • A-format (AFMT)
Event link	Trigger input
Module-stop function	Module-stop state can be set to reduce power consumption for each module

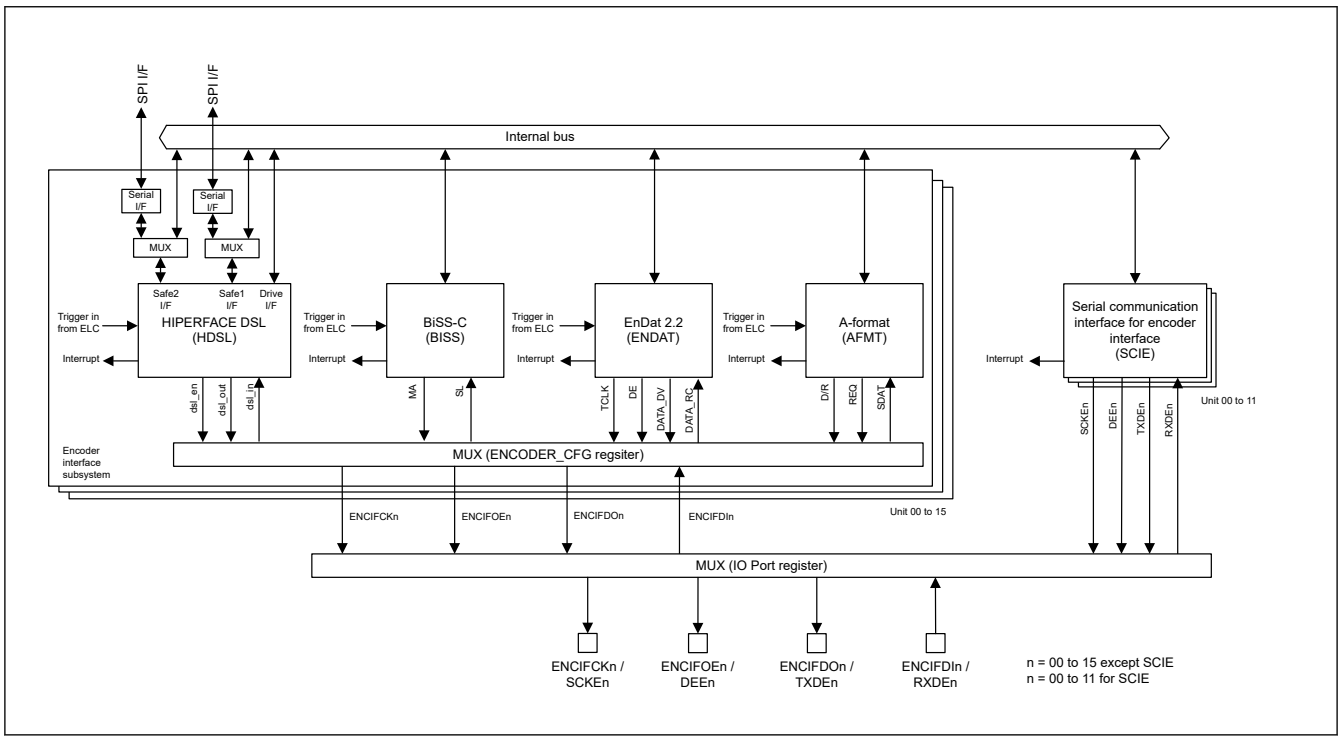


Figure 48.1 Block diagram of encoder interface subsystem

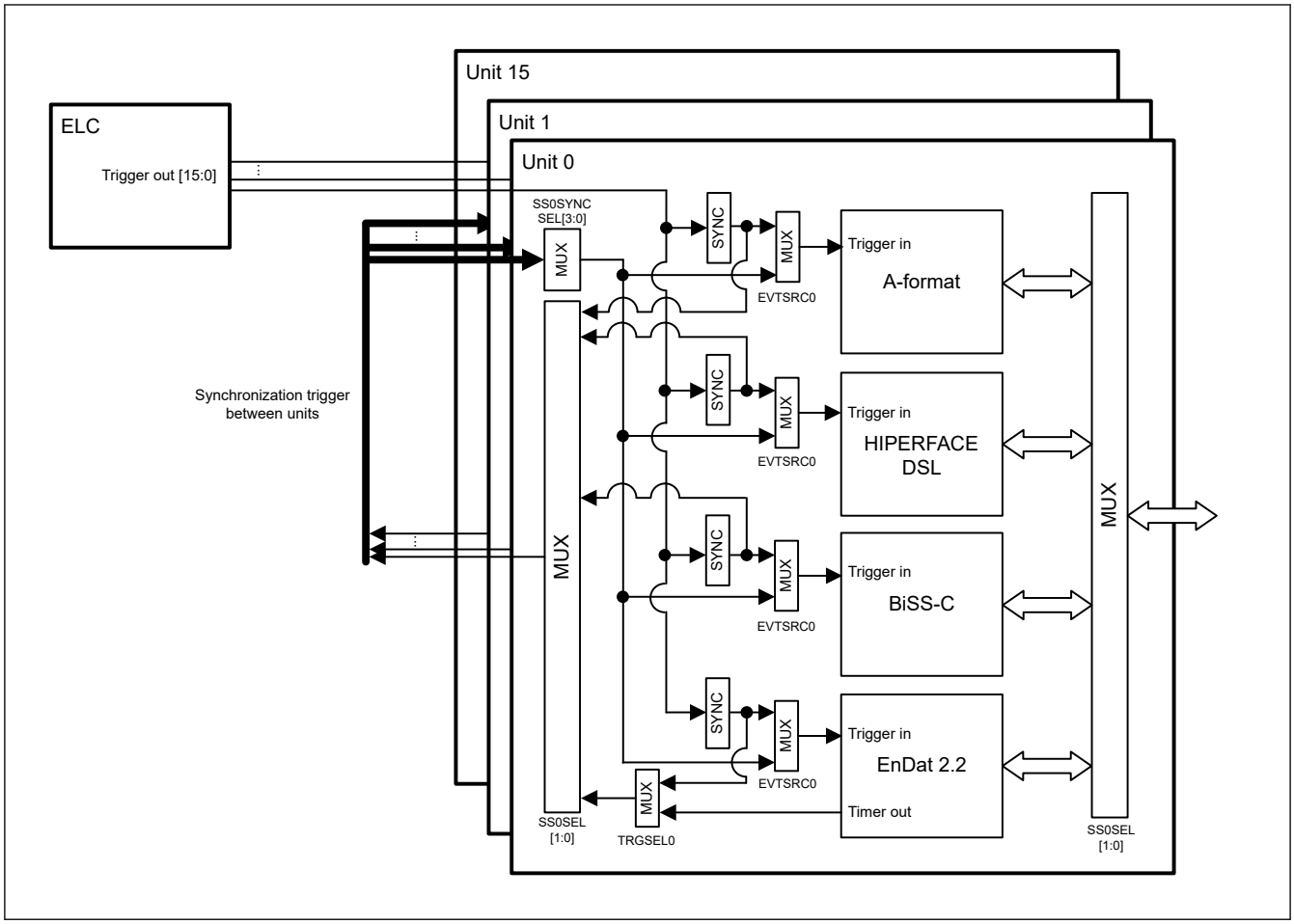


Figure 48.2 Trigger input

Table 48.2 lists the input/output pins of the encoder interface subsystem.

Table 48.2 Pin configuration of the encoder interface subsystem

Unit	Pin name	I/O	Function
Unit n	ENCIFCKn	Output	Encoder interface clock output pin
	ENCIFOEn	Output	Encoder interface data output enable pin
	ENCIFDOn	Output	Encoder interface data output pin
	ENCIFDIn	Input	Encoder interface data input pin

Note: n = 00 to 15

Table 48.3 Encoder interface pin function of each module

Pin name	EnDat 2.2	BiSS-C	HIPERFACE DSL	A-format
ENCIFCKn	TCLK	MA	(N/A)	(N/A)
ENCIFOEn	DE	(N/A)	dsl_en	D/R
ENCIFDOn	DATA_DV	(N/A)	dsl_out	REQ
ENCIFDIn	DATA_RC	SL	dsl_in	SDAT

Note: n = 00 to 15

48.2 Register Map

Table 48.4 ENCSS interface register map

Address	Register symbol	Register name	Write protection
0x8029_1000 + 0x04 × n	ENCODER_CFGn	Encoder IF Configuration Register n (n = 0 to 7)	PRCRN.PRC3
0x8029_1020 + 0x04 × n	BISS_STATn	BiSS-C Status Register n (n = 0 to 7)	—
0x8029_1040 + 0x04 × n	HDSL_STATn	HIPERFACE DSL Status Register n (n = 0 to 7)	—
0x8029_1060 + 0x04 × n	ENDAT_CFGn	EnDat Configuration Register n (n = 0 to 7)	PRCRN.PRC3
0x8029_1080 + 0x04 × n	AFMT_STATn	A-Format Status Register n (n = 0 to 7)	—
0x8029_10A0 + 0x04 × n	HDSLn_OS_D	HIPERFACE_DSL Online Status D Register n (n = 0 to 15)	—
0x8029_10E0 + 0x04 × n	HDSLn_OS_1	HIPERFACE_DSL Online Status 1 Register n (n = 0 to 15)	—
0x8029_1120 + 0x04 × n	HDSLn_OS_2	HIPERFACE_DSL Online Status 2 Register n (n = 0 to 15)	—
0x8029_1160 + 0x04 × n	HDSL_HOSTFn	HIPERFACE_DSL Host F Register n (n = 0, 1)	—

Table 48.5 ENCSS related system control register

Target	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
All the registers	—	—	SLVACCCTL7.LLPP_SL[*1]

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

48.3 Register Descriptions

48.3.1 ENCODER_CFGn : Encoder IF Configuration Register n (n = 0 to 7)

Base address: ENCSS = 0x8029_1000

Offset address: 0x000 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SS1SYNCSEL[3:0]				HF1SAFE2SEL	TRGSEL1	SS1SEL[1:0]		EVTSRC1	HF1SAFE1SEL	HF1ENDIAN
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SS0SYNCSEL[3:0]				HF0SAFE2SEL	TRGSEL0	SS0SEL[1:0]		EVTSRC0	HF0SAFE1SEL	HF0ENDIAN
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	HF0ENDIAN	HIPERFACE DSL Unit 2*n Endian Mode Byte sequence selection for register access 0: Little endian 1: Big endian	R/W
1	HF0SAFE1SEL	HIPERFACE DSL Unit 2*n Safe 1 Interface Select Select the interface mode for safe 1 interface 0: Internal bus interface 1: SPI interface	R/W
2	EVTSRC0	Event Source Select Select the event trigger source for encoder interface subsystem Unit 2*n 0: Trigger out 2*n from ELC 1: Trigger out from other units of encoder interface subsystem. Used unit is selected by SS0SYNCSEL[3:0] bits. The event source unit and triggered unit should be same protocol.	R/W

Bit	Symbol	Function	R/W
4:3	SS0SEL[1:0]	Encoder Interface Port Select Select the encoder interface for Unit 2*n 0 0: HIPERFACE DSL 0 1: BiSS-C 1 0: EnDat 2.2 1 1: A-format	R/W
5	TRGSEL0	EnDat 2.2 Unit 2*n Trigger Output Source Select Select the source of the ENDAT trigger output for other units of encoder interface subsystem 0: Synchronized Trigger out 2*n from ELC 1: Timer output from EnDat 2.2 Unit 2*n	R/W
6	HF0SAFE2SEL	HIPERFACE DSL Unit 2*n Safe 2 Interface Select Select the interface mode for safe 2 interface 0: Internal bus interface 1: SPI interface	R/W
10:7	SS0SYNCSEL[3:0]	Event Source Unit Number Select Specify the unit number of encoder interface subsystem of which trigger output is used as the event source When using own timer output for EnDat 2.2 as event source, specify the same unit number.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	HF1ENDIAN	HIPERFACE DSL Unit 2*n+1 Endian Mode Byte sequence selection for register access 0: Little endian 1: Big endian	R/W
17	HF1SAFE1SEL	HIPERFACE DSL Unit 2*n+1 Safe 1 Interface Select Select the interface mode for safe 1 interface 0: Internal bus interface 1: SPI interface	R/W
18	EVTSRC1	Event Source Select Select the event trigger source for encoder interface subsystem Unit 2*n+1 0: Trigger out 2*n+1 from ELC 1: Trigger out from other units of encoder interface subsystem. Used unit is selected by SS0SYNCSEL[3:0] bits. The event source unit and triggered unit should be same protocol.	R/W
20:19	SS1SEL[1:0]	Encoder Interface Port Select Select the encoder interface for Unit 2*n+1 0 0: HIPERFACE DSL 0 1: BiSS-C 1 0: EnDat 2.2 1 1: A-format	R/W
21	TRGSEL1	EnDat 2.2 Unit 2*n+1 Trigger Output Source Select Select the source of the ENDAT trigger output for other units of encoder interface subsystem 0: Synchronized Trigger out 2*n+1 from ELC 1: Timer output from EnDat 2.2 Unit 2*n+1	R/W
22	HF1SAFE2SEL	HIPERFACE DSL Unit 2*n+1 Safe 2 Interface Select Select the interface mode for safe 2 interface 0: Internal bus interface 1: SPI interface	R/W
26:23	SS1SYNCSEL[3:0]	Event Source Unit Number Select Specify the unit number of encoder interface subsystem of which trigger output is used as the event source When using own timer output for EnDat 2.2 as event source, specify the same unit number.	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

The ENCODER_CFGn register is used to set the encoder interface operating mode. This register should be set before releasing the module stop for each encoder interface module.

48.3.2 BISS_STATn : BiSS-C Status Register n (n = 0 to 7)

Base address: ENCSS = 0x8029_1000

Offset address: 0x020 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	BISS1_EOT	—	—	—	BISS1_ERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	BISS0_EOT	—	—	—	BISS0_ERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bit	Symbol	Function	R/W
0	BISS0_ERR	BiSS-C Unit 2*n Error 0: External or internal error occurred 1: No error occurred	R
3:1	—	These bits are read as 0.	R
4	BISS0_EOT	End-of-Transmission from BiSS-C Unit 2*n 0: Data transmission active 1: Data transmission inactive	R
15:5	—	These bits are read as 0.	R
16	BISS1_ERR	BiSS-C Unit 2*n+1 Error 0: External or internal error occurred 1: No error occurred	R
19:17	—	These bits are read as 0.	R
20	BISS1_EOT	End-of-Transmission from BiSS-C Unit 2*n+1 0: Data transmission active 1: Data transmission inactive	R
31:21	—	These bits are read as 0.	R

The BISS_STATn register indicates the status of BiSS-C. (n = 0 to 7)

48.3.3 HDSL_STATn : HIPERFACE DSL Status Register n (n = 0 to 7)

Base address: ENCSS = 0x8029_1000

Offset address: 0x040 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	HDSL1_ENCERR	HDSL1_ACTHERR	HDSL1_ACCERR	HDSL1_EDTERR	HDSL1_SPERR	HDSL1_SCHERR	HDSL1_ESTON	—	—	HDSL1_SYNCLKD	HDSL1_LINK
Value after reset:	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HDSL0_ENCERR	HDSL0_ACTHERR	HDSL0_ACCERR	HDSL0_EDTERR	HDSL0_SPERR	HDSL0_SCHERR	HDSL0_ESTON	—	—	HDSL0_SYNCLKD	HDSL0_LINK
Value after reset:	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
0	HDSL0_LINK	HIPERFACE-DSL Link Connection indication from HIPERFACE-DSL Unit 2*n 0: Disconnect 1: Connect	R
1	HDSL0_SYNCLCKD	HIPERFACE-DSL Sync Locked Position sampling resolution locked from HIPERFACE-DSL Unit 2*n 0: Unlock 1: Locked	R
3:2	—	These bits are read as 0.	R
4	HDSL0_ESTON	HIPERFACE-DSL Estimator ON Position estimator activated from HIPERFACE-DSL Unit 2*n 0: Inactive 1: Active	R
5	HDSL0_SCHERR	HIPERFACE-DSL Safe Channel Error Transmission error in safe channel 1 from HIPERFACE-DSL Unit 2*n 0: No error 1: Error	R
6	HDSL0_SPERR	HIPERFACE-DSL Safe Position Error Safe position not valid from HIPERFACE-DSL Unit 2*n 0: No error 1: Error	R
7	HDSL0_EDTERR	HIPERFACE-DSL Estimator Deviation Threshold Error Estimator deviation threshold reached from HIPERFACE-DSL Unit 2*n 0: No error 1: Error	R
8	HDSL0_ACCERR	HIPERFACE-DSL Acceleration Error Fast channel/position error from HIPERFACE-DSL Unit 2*n 0: No error 1: Error	R
9	HDSL0_ACTHERR	HIPERFACE-DSL Acceleration Threshold Error Fast channel/position threshold error from HIPERFACE-DSL Unit 2*n 0: No error 1: Error	R
10	HDSL0_ENCERR	HIPERFACE-DSL Encoding Error 8B/10B encoding error in DSL frame transmission from HIPERFACE-DSL Unit 2*n 0: No error 1: Error	R
15:11	—	These bits are read as 0.	R
16	HDSL1_LINK	HIPERFACE-DSL Link Connection indication from HIPERFACE-DSL Unit 2*n+1 0: Disconnect 1: Connect	R
17	HDSL1_SYNCLCKD	HIPERFACE-DSL Sync Locked Position sampling resolution locked from HIPERFACE-DSL Unit 2*n+1 0: Unlock 1: Locked	R
19:18	—	These bits are read as 0.	R
20	HDSL1_ESTON	HIPERFACE-DSL Estimator ON Position estimator activated from HIPERFACE-DSL Unit 2*n+1 0: Inactive 1: Active	R
21	HDSL1_SCHERR	HIPERFACE-DSL Safe Channel Error Transmission error in safe channel 1 from HIPERFACE-DSL Unit 2*n+1 0: No error 1: Error	R

Bit	Symbol	Function	R/W
22	HDSL1_SPERR	HIPERFACE-DSL Safe Position Error Safe position not valid from HIPERFACE-DSL Unit 2*n+1 0: No error 1: Error	R
23	HDSL1_EDTERR	HIPERFACE-DSL Estimator Deviation Threshold Error Estimator deviation threshold reached from HIPERFACE-DSL Unit 2*n+1 0: No error 1: Error	R
24	HDSL1_ACCERR	HIPERFACE-DSL Acceleration Error Fast channel/position error from HIPERFACE-DSL Unit 2*n+1 0: No error 1: Error	R
25	HDSL1_ACTHERR	HIPERFACE-DSL Acceleration Threshold Error Fast channel/position threshold error from HIPERFACE-DSL Unit 2*n+1 0: No error 1: Error	R
26	HDSL1_ENCERR	HIPERFACE-DSL Encoding Error 8B/10B encoding error in DSL frame transmission from HIPERFACE-DSL Unit 2*n+1 0: No error 1: Error	R
31:27	—	These bits are read as 0.	R

The HDSL_STATn register indicates the status of HIPERFACE-DSL. (n = 0 to 7)

48.3.4 ENDAT_CFGn : EnDat Configuration Register n (n = 0 to 7)

Base address: ENCSS = 0x8029_1000

Offset address: 0x060 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ENDAT1_AXADD[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ENDAT0_AXADD[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	ENDAT0_AXADD[4:0]	EnDat Unit 2*n Axis Address The component address for EnDat 2.2 Unit 2*n	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
20:16	ENDAT1_AXADD[4:0]	EnDat Unit 2*n+1 Axis Address The component address for EnDat 2.2 Unit 2*n+1	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The ENDAT_CFGn register is used to set the EnDat 2.2. (n = 0 to 7)

48.3.5 AFMT_STATn : A-Format Status Register n (n = 0 to 7)

Base address: ENCSS = 0x8029_1000

Offset address: 0x080 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	AFMT1_TMOUT	AFMT1_EOT	AFMT1_BUSY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	AFMT0_TMOUT	AFMT0_EOT	AFMT0_BUSY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	AFMT0_BUSY	A-format Busy Transmission busy from A-format Unit 2*n 0: Busy 1: Idle	R
1	AFMT0_EOT	A-format End-Of-Frame End-of-transmission from A-format Unit 2*n 0: Data transfer end (cleared by data read) 1: Idle or in-progress	R
2	AFMT0_TMOUT	A-format Timeout Timeout error from A-format Unit 2*n 0: No error 1: Time out error	R
15:3	—	These bits are read as 0.	R
16	AFMT1_BUSY	A-format Busy Transmission busy from A-format Unit 2*n+1 0: Busy 1: Idle	R
17	AFMT1_EOT	A-format End-of-Frame End-of-transmission from A-format Unit 2*n+1 0: Data transfer end (cleared by data read) 1: Idle or in-progress	R
18	AFMT1_TMOUT	A-format Timeout Timeout error from A-format Unit 2*n+1 0: No error 1: Time out error	R
31:19	—	These bits are read as 0.	R

The AFMT_STATn register indicates the status of A-format. (n = 0 to 7)

48.3.6 HDSLn_OS_D : HIPERFACE_DSL Online Status D Register n (n = 0 to 15)

Base address: ENCSS = 0x8029_1000

Offset address: 0x0A0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	INT	SUM	—	—	POS	—	DTE	PRST	POSTX[1:0]		MIN	ANS	—	QMLW	FREL	—
Value after reset:	0	0	0	1	1	0	1	1	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	FREL	Channel Status for long message 0: The channel for the long message is in use. 1: The channel for the long message is free.	R
2	QMLW	Quality monitoring at Low level 0: Current value of quality monitoring greater than or equal to 14 1: Current value of quality monitoring less than 14	R
3	—	This bit is read as 0.	R
4	ANS	Incorrect Answer Detected 0: No error detected in the last answer to a long message 1: The last answer to a long message was damaged	R
5	MIN	Acknowledgment of Message Initialization 0: Parameter Channel not functioning 1: The DSL encoder sends a figure by which the initialization of the Parameter Channel is acknowledged	R
7:6	POSTX[1:0]	Position Transmission Status 0 0: Position request is transmitted to the DSL encoder 0 1: Safe position 1 was received 1 0: Fast position was received or position newly updated by estimator 1 1: Safe position 2 was received	R
8	PRST	Protocol Reset 0: HDSLn running 1: HDSLn has restarted the protocol	R
9	DTE	Deviation Threshold Error 0: Current value of deviation smaller than the specified maximum 1: Current value of deviation greater than the specified maximum	R
10	—	This bit is read as 0.	R
11	POS	Estimator Turned On 0: No fast position error 1: A source of an error in the fast position was identified or an alignment procedure is currently being carried out. It is probable that the last fast position is invalid. Be aware that the fast position read through drive interface is provided by the estimator.	R
12	—	This bit is read as 1.	R
13	—	This bit is read as 0.	R
14	SUM	Summary Byte 0: The last valid value from MIR_SUM was zero. 1: The last valid value from MIR_SUM was not zero. The importance of this flag depends on the particular error source that leads to a set MIR_SUM.	R

Bit	Symbol	Function	R/W
15	INT	Status of the Interrupt Output 0: Interrupt output on the low level 1: Interrupt output on the high level	R
31:16	—	These bits are read as 0.	R

The HDSLn_OS_D register is used to indicate Online Status D of HIPERFACE_DSL. (n = 0 to 15)

48.3.7 HDSLn_OS_1 : HIPERFACE_DSL Online Status 1 Register n (n = 0 to 15)

Base address: ENCSS = 0x8029_1000

Offset address: 0x0E0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SINT	SSUM	SCE	—	—	VPOS	—	PRST	POSTX[1:0]		MIN	—	—	QMLW	—	FRES
Value after reset:	0	0	1	1	0	1	0	1	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
0	FRES	Channel Status for the short message 0: The channel for the short message is in use. 1: The channel for the short message is free.	R
1	—	This bit is read as 0.	R
2	QMLW	Quality Monitoring at Low level 0: Current value of quality monitoring greater than or equal to 14 1: Current value of quality monitoring less than 14	R
4:3	—	These bits are read as 0.	R
5	MIN	Acknowledgment of Message Initialization 0: Parameter channel not functioning 1: The DSL encoder sends a figure by which the initialization of the parameter channel is acknowledged.	R
7:6	POSTX[1:0]	Position Transmission Status 0 0: Position request is transmitted to the DSL encoder. 0 1: Safe position 1 was received. 1 0: Fast position was received or position newly updated by estimator. 1 1: Safe position 2 was received.	R
8	PRST	Protocol Reset 0: HDSL0 running 1: HDSL0 has restarted the protocol	R
9	—	This bit is read as 0.	R
10	VPOS	Safe Position Invalid 0: The last safe position received was correct. 1: An error in the safe position was identified. It is expected that the safe position transmitted from the encoder is invalid.	R
11	—	This bit is read as 0.	R
12	—	This bit is read as 1.	R
13	SCE	CRC Error on the Safe Channel 0: The last Safe Channel CRC received was correct. 1: The last Safe Channel CRC received was wrong. It is expected that the last safe position 2 transmitted is invalid.	R

Bit	Symbol	Function	R/W
14	SSUM	Safe Summary Bit 0: The last valid value from SAFE_SUM was zero. 1: The last valid value from SAFE_SUM was not zero. The importance of this flag depends on the particular error source that leads to a set SAFE_SUM.	R
15	SINT	Status of the Interrupt Output 0: Interrupt output on the low level 1: Interrupt output on the high level	R
31:16	—	These bits are read as 0.	R

The HDSLn_OS_1 register is used to indicate Online Status 1 of HIPERFACE_DSL. (n = 0 to 15)

48.3.8 HDSLn_OS_2 : HIPERFACE_DSL Online Status 2 Register n (n = 0 to 15)

Base address: ENCSS = 0x8029_1000

Offset address: 0x120 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SSUM2	SCE2	—	—	VPOS2	—	PRST	POSTX[1:0]		—	—	—	QMLW	—	—
Value after reset:	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
1:0	—	This bit is read as 0.	R
2	QMLW	Quality Monitoring at Low level 0: Current value of quality monitoring greater than or equal to 14 1: Current value of quality monitoring less than 14	R
5:3	—	These bits are read as 0.	R
7:6	POSTX[1:0]	Position Transmission Status 0 0: Position request is transmitted to the DSL encoder. 0 1: Safe position 1 was received. 1 0: Fast position was received or position newly updated by estimator. 1 1: Safe position 2 was received.	R
8	PRST	Protocol Reset 0: HDSL0 running 1: HDSL0 has restarted the protocol	R
9	—	This bit is read as 0.	R
10	VPOS2	Safe Position 2 Invalid 0: The last safe position received in channel 2 was correct. 1: A source of an error in the safe position in channel 2 was identified. It is probable that the safe position transmitted from Channel 2 is invalid. Suitable measures must be installed in the user application.	R
11	—	This bit is read as 0.	R
12	—	This bit is read as 1.	R
13	SCE2	Transmission Error Channel 2 0: The last data received in channel 2 was correct. 1: The last safe channel 2 CRC received was wrong. It is expected that the last safe position 2 transmitted is invalid. Suitable measures must be installed in the user application.	R

Bit	Symbol	Function	R/W
14	SSUM2	Summary Bit Channel 2 0: Neither TEST2 nor ERR2 is set. 1: One of the indications TEST2 or ERR2 is set. The error reaction to the flag depends on the meaning of the bit.	R
31:15	—	These bits are read as 0.	R

The HDSLn_OS_2 register is used to indicate Online Status 2 of HIPERFACE_DSL. (n = 0 to 15)

48.3.9 HDSL_HOSTFn : HIPERFACE DSL Host_f Control Register n (n = 0, 1)

Base address: ENCSS = 0x8029_1000

Offset address: 0x160 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	HOST 2_F7	HOST 1_F7	HOST D_F7	—	HOST 2_F6	HOST 1_F6	HOST D_F6	—	HOST 2_F5	HOST 1_F5	HOST D_F5	—	HOST 2_F4	HOST 1_F4	HOST D_F4
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	HOST 2_F3	HOST 1_F3	HOST D_F3	—	HOST 2_F2	HOST 1_F2	HOST D_F2	—	HOST 2_F1	HOST 1_F1	HOST D_F1	—	HOST 2_F0	HOST 1_F0	HOST D_F0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HOSTD_F0	Drive Interface Register Freeze (hostd_f) for HDSL Unit 8*n 0: Not freeze register update 1: Freeze register update	R/W
1	HOST1_F0	Safe 1 Interface Register Freeze (host1_f) for HDSL Unit 8*n 0: Not freeze register update 1: Freeze register update	R/W
2	HOST2_F0	Safe 2 Interface Register Freeze (host2_f) for HDSL Unit 8*n 0: Not freeze register update 1: Freeze register update	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	HOSTD_F1	Drive Interface Register Freeze (hostd_f) for HDSL Unit 8*n+1 0: Not freeze register update 1: Freeze register update	R/W
5	HOST1_F1	Safe 1 Interface Register Freeze (host1_f) for HDSL Unit 8*n+1 0: Not freeze register update 1: Freeze register update	R/W
6	HOST2_F1	Safe 2 Interface Register Freeze (host2_f) for HDSL Unit 8*n+1 0: Not freeze register update 1: Freeze register update	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	HOSTD_F2	Drive Interface Register Freeze (hostd_f) for HDSL Unit 8*n+2 0: Not freeze register update 1: Freeze register update	R/W
9	HOST1_F2	Safe 1 Interface Register Freeze (host1_f) for HDSL Unit 8*n+2 0: Not freeze register update 1: Freeze register update	R/W
10	HOST2_F2	Safe 2 Interface Register Freeze (host2_f) for HDSL Unit 8*n+2 0: Not freeze register update 1: Freeze register update	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	HOSTD_F3	Drive Interface Register Freeze (hostd_f) for HDSL Unit 8*n+3 0: Not freeze register update 1: Freeze register update	R/W
13	HOST1_F3	Safe 1 Interface Register Freeze (host1_f) for HDSL Unit 8*n+3 0: Not freeze register update 1: Freeze register update	R/W
14	HOST2_F3	Safe 2 Interface Register Freeze (host2_f) for HDSL Unit 8*n+3 0: Not freeze register update 1: Freeze register update	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	HOSTD_F4	Drive Interface Register Freeze (hostd_f) for HDSL Unit 8*n+4 0: Not freeze register update 1: Freeze register update	R/W
17	HOST1_F4	Safe 1 Interface Register Freeze (host1_f) for HDSL Unit 8*n+4 0: Not freeze register update 1: Freeze register update	R/W
18	HOST2_F4	Safe 2 Interface Register Freeze (host2_f) for HDSL Unit 8*n+4 0: Not freeze register update 1: Freeze register update	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	HOSTD_F5	Drive Interface Register Freeze (hostd_f) for HDSL Unit 8*n+5 0: Not freeze register update 1: Freeze register update	R/W
21	HOST1_F5	Safe 1 Interface Register Freeze (host1_f) for HDSL Unit 8*n+5 0: Not freeze register update 1: Freeze register update	R/W
22	HOST2_F5	Safe 2 Interface Register Freeze (host2_f) for HDSL Unit 8*n+5 0: Not freeze register update 1: Freeze register update	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
24	HOSTD_F6	Drive Interface Register Freeze (hostd_f) for HDSL Unit 8*n+6 0: Not freeze register update 1: Freeze register update	R/W
25	HOST1_F6	Safe 1 Interface Register Freeze (host1_f) for HDSL Unit 8*n+6 0: Not freeze register update 1: Freeze register update	R/W
26	HOST2_F6	Safe 2 Interface Register Freeze (host2_f) for HDSL Unit 8*n+6 0: Not freeze register update 1: Freeze register update	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	HOSTD_F7	Drive Interface Register Freeze (hostd_f) for HDSL Unit 8*n+7 0: Not freeze register update 1: Freeze register update	R/W
29	HOST1_F7	Safe 1 Interface Register Freeze (host1_f) for HDSL Unit 8*n+7 0: Not freeze register update 1: Freeze register update	R/W
30	HOST2_F7	Safe 2 Interface Register Freeze (host2_f) for HDSL Unit 8*n+7 0: Not freeze register update 1: Freeze register update	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The HDSL_HOSTFn register is used to control register freeze signals (hostd_f, host1_f, and host2_f). (n = 0, 1)

When it is set to high, no further updates are done for multi-byte registers.

48.4 Usage Notes

48.4.1 Release of Register Write Protection

ENCODER_CFG and ENDAT_CFG registers are write protected. To release write protection, use the following steps:

1. Write 0x0000A508 to PRCRN register (release the write protection by PRC3 bit).
2. Write value to ENCODER_CFG or ENDAT_CFG registers.
3. Write 0x0000A500 to PRCRN register (set the write protection by PRC3 bit).

49. EnDat 2.2 (ENDAT)

49.1 Overview

The MPU includes EnDat 2.2 (ENDAT) master to enable communication between the encoder and EnDat 2.2 interface. Table 49.1 describes the ENDAT specifications and Figure 49.1 shows a block diagram of the ENDAT.

Table 49.1 EnDat 2.2 (ENDAT) specifications

Item	Description
Number of units	16 units (12 in LLPP0 and 4 in LLPP1)
Protocol	EnDat 2.2
Functions	• EnDat 2.2 Basic • Host interface: 32 bits APB • Basic functions of EnDat data transfer: – Clock generator – Data word length – Continuous clock – Error handling – Encoder status – CRC – Cable preparation time – Recovery time II – Recovery time III • Advanced functions of EnDat data transfer – Watchdog – Reset with selected additional information • Strobe source: Hardware, timer and software • Digital filter for data input
Interrupt sources	• Basic interrupt: 1 – Watchdog, receive register updated
Event link	STR input for HW trigger provided from ELC
Module-stop function	Module-stop state can be set to reduce power consumption

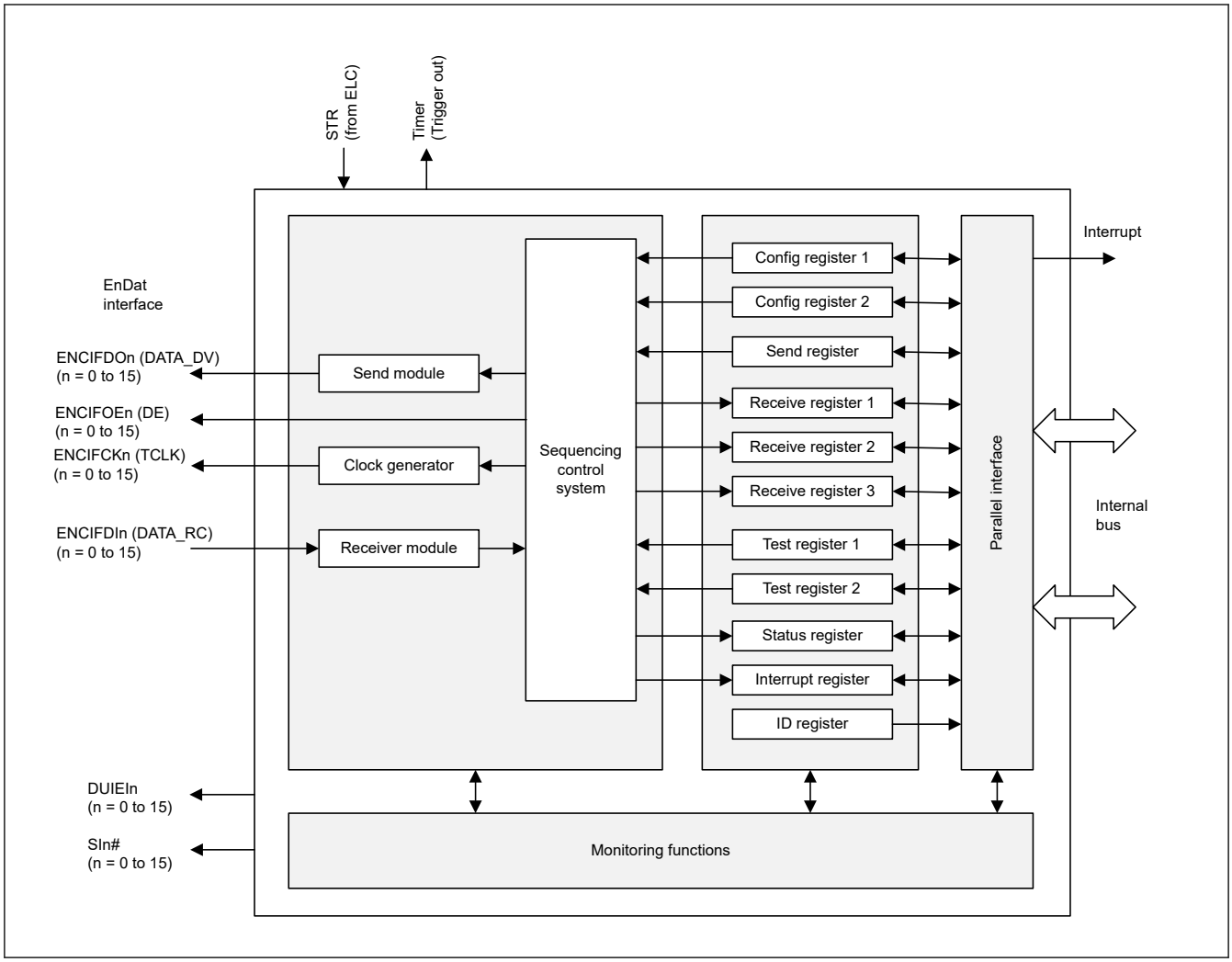


Figure 49.1 Block diagram of EnDat 2.2 (ENDAT)

Table 49.2 lists the input/output pins of the ENDAT.

Table 49.2 Pin configuration of the ENDAT

Unit	Pin name	I/O	Function
ENDATn	ENCIFOEn (DE)	Output	EnDat 2.2 unit n drive enable
	ENCIFDOn (DATA_DV)	Output	EnDat 2.2 unit n data output (drive)
	ENCIFCKn (TCLK)	Output	EnDat 2.2 unit n clock generator output
	ENCIFDIn (DATA_RC)	Input	EnDat 2.2 unit n data input (receive)
	DUEIn	Output	EnDat 2.2 unit n data transfer
	TST_OUTn	Output	EnDat 2.2 unit n data input after internal synchronization
	SIn#	Output	EnDat 2.2 unit n start pulse

Note: n = 0 to 15

Table 49.3 ENDAT interrupt sources

Name	Interrupt sources
ENDATn_INT1	Interrupt output 1

Note: n = 0 to 15

49.2 Register Map

Table 49.4 ENDAT register map

Address	Register symbol	Register name	Write protection
0x9020_4000 + 0x1_0000 × m (m = 0 to 11) 0x9030_4000 + 0x1_0000 × (m - 12) (m = 12 to 15)	SEND	Send Register	—
0x9020_4004 + 0x1_0000 × m (m = 0 to 11) 0x9030_4004 + 0x1_0000 × (m - 12) (m = 12 to 15)	EMPFR1_L	Receive Register 1, Low Double Word	—
0x9020_4008 + 0x1_0000 × m (m = 0 to 11) 0x9030_4008 + 0x1_0000 × (m - 12) (m = 12 to 15)	EMPFR1_H	Receive Register 1, High Double Word	—
0x9020_400C + 0x1_0000 × m (m = 0 to 11) 0x9030_400C + 0x1_0000 × (m - 12) (m = 12 to 15)	EMPFR2	Receive Register 2	—
0x9020_4010 + 0x1_0000 × m (m = 0 to 11) 0x9030_4010 + 0x1_0000 × (m - 12) (m = 12 to 15)	EMPFR3	Receive Register 3	—
0x9020_4014 + 0x1_0000 × m (m = 0 to 11) 0x9030_4014 + 0x1_0000 × (m - 12) (m = 12 to 15)	KONFR1	Configuration Register 1	—
0x9020_4018 + 0x1_0000 × m (m = 0 to 11) 0x9030_4018 + 0x1_0000 × (m - 12) (m = 12 to 15)	KONFR2	Configuration Register 2	—
0x9020_4020 + 0x1_0000 × m (m = 0 to 11) 0x9030_4020 + 0x1_0000 × (m - 12) (m = 12 to 15)	STATR	Status Register	—
0x9020_4024 + 0x1_0000 × m (m = 0 to 11) 0x9030_4024 + 0x1_0000 × (m - 12) (m = 12 to 15)	INTMR	Interrupt Mask Register	—
0x9020_4038 + 0x1_0000 × m (m = 0 to 11) 0x9030_4038 + 0x1_0000 × (m - 12) (m = 12 to 15)	SWSTRBR	SW Strobe Register	—
0x9020_403C + 0x1_0000 × m (m = 0 to 11) 0x9030_403C + 0x1_0000 × (m - 12) (m = 12 to 15)	IDR	ID Register	—

Table 49.5 ENDAT related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0 (LLPP0)	—	MSTPCRJ.MSTPCRJ03	SLVACCCTL7.LLPP_SL*1
1 (LLPP0)	—	MSTPCRJ.MSTPCRJ07	SLVACCCTL7.LLPP_SL*1
2 (LLPP0)	—	MSTPCRJ.MSTPCRJ11	SLVACCCTL7.LLPP_SL*1
3 (LLPP0)	—	MSTPCRJ.MSTPCRJ15	SLVACCCTL7.LLPP_SL*1
4 (LLPP0)	—	MSTPCRJ.MSTPCRJ19	SLVACCCTL7.LLPP_SL*1
5 (LLPP0)	—	MSTPCRJ.MSTPCRJ23	SLVACCCTL7.LLPP_SL*1
6 (LLPP0)	—	MSTPCRJ.MSTPCRJ27	SLVACCCTL7.LLPP_SL*1
7 (LLPP0)	—	MSTPCRJ.MSTPCRJ31	SLVACCCTL7.LLPP_SL*1
8 (LLPP0)	—	MSTPCRK.MSTPCRK03	SLVACCCTL7.LLPP_SL*1
9 (LLPP0)	—	MSTPCRK.MSTPCRK07	SLVACCCTL7.LLPP_SL*1
10 (LLPP0)	—	MSTPCRK.MSTPCRK11	SLVACCCTL7.LLPP_SL*1
11 (LLPP0)	—	MSTPCRK.MSTPCRK15	SLVACCCTL7.LLPP_SL*1
12 (LLPP1)	—	MSTPCRK.MSTPCRK19	SLVACCCTL7.LLPP_SL*1
13 (LLPP1)	—	MSTPCRK.MSTPCRK23	SLVACCCTL7.LLPP_SL*1
14 (LLPP1)	—	MSTPCRK.MSTPCRK27	SLVACCCTL7.LLPP_SL*1
15 (LLPP1)	—	MSTPCRK.MSTPCRK31	SLVACCCTL7.LLPP_SL*1

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

49.3 Register Descriptions

All register access operation should be executed with 32-bit access. It is not possible to write to or read from a single byte or word of a register. Unused bits are filled with zeros.

49.3.1 SEND : Send Register

Base address: ENDATm = 0x9020_4000 + 0x1_0000 × m (m = 0 to 11)
ENDATi = 0x9030_4000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	M[5:0]						A[7:0]								D[15:0]															
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	D[15:0]	Parameters, Instructions See Table 49.6.	R/W
23:16	A[7:0]	Memory Range Select (MRS) code, Address, Port address See Table 49.6.	R/W
29:24	M[5:0]	Mode command See Table 49.6.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

The SEND register contains data to be transmitted to the EnDat encoder.

Table 49.6 shows an overview of the supported mode commands with data to be entered into the send register.

Table 49.6 is list of the EnDat master side, however, some commands may not be supported depending on the EnDat encoder that is the EnDat slave side.

Table 49.6 Mode commands

Mode command	M[5:0]	A[7:0]	D[15:0]
Encoder transmit position values	0x07	(Reserved)	(Reserved)
Encoder transmit position values with additional information	0x38	(Reserved)	(Reserved)
Selection of memory area	0x0E	MRS code	(Reserved)
Encoder receive parameters	0x1C	Address	Parameters
Encoder transmit parameters	0x23	Address	(Reserved)
Encoder receive reset	0x2A	Address	(Reserved)
Encoder transmit test values	0x15	(Reserved)	(Reserved)
Encoder receive communication command	0x12	Address	Instruction
Encoder receive test command	0x31	Port address	(Reserved)
Encoder transmit position values with additional information and selection of the memory area	0x09	MRS code	(Reserved)
Encoder transmit position values with additional information and receive parameters	0x1B	Address	Parameters
Encoder transmit position values with additional information and parameters	0x24	Address	(Reserved)
Encoder transmit position values with additional information and receive error reset	0x2D	Address	(Reserved)
Encoder transmit position values with additional information and receive test command	0x36	Port address	(Reserved)

Table 49.7 MRS code

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Request
Additional information 1								
0	1	0	0	0	0	0	0	Send additional info 1 without data contents (NOP)
0	1	0	0	0	0	0	1	Send diagnostic value
0	1	0	0	0	0	1	0	Send position value 2 word 1 LSB
0	1	0	0	0	0	1	1	Send position value 2 word 2
0	1	0	0	0	1	0	0	Send position value 2 word 3 MSB
0	1	0	0	0	1	0	1	Acknowledge memory content LSB
0	1	0	0	0	1	1	0	Acknowledge memory content MSB
0	1	0	0	0	1	1	1	Acknowledge MRS code
0	1	0	0	1	0	0	0	Acknowledge test command
0	1	0	0	1	0	0	1	Send test values word 1 LSB
0	1	0	0	1	0	1	0	Send test values word 2
0	1	0	0	1	0	1	1	Send test values word 3 MSB
0	1	0	0	1	1	0	0	Send temperature 1
0	1	0	0	1	1	0	1	Send temperature 2
0	1	0	0	1	1	1	0	Additional sensors
0	1	0	0	1	1	1	1	Stop sending additional information 1
Additional information 2								
0	1	0	1	0	0	0	0	Send additional information 2 without data contents
0	1	0	1	0	0	0	1	Send commutation
0	1	0	1	0	0	1	0	Send acceleration
0	1	0	1	0	0	1	1	Send commutation and acceleration
0	1	0	1	0	1	0	0	Send limit position signals
0	1	0	1	0	1	0	1	Send limit position signals & acceleration
0	1	0	1	0	1	1	0	Asynchronous position value word 1 LSB
0	1	0	1	0	1	1	1	Asynchronous position value word 2
0	1	0	1	1	0	0	0	Asynchronous position value word 3 MSB
0	1	0	1	1	0	0	1	Operating status error sources
0	1	0	1	—	—	—	—	Not assigned
0	1	0	1	1	1	1	1	Stop sending additional information 2

Note: Data transfer from the send register to the send module occurs only after a completed write cycle and completed transmission cycle to/from encoder.

Only values for the mode commands defined above are entered into the send register. Only EnDat2.2 mode commands are entered in transmission mode 1 (safety mode). This ensures that the most recently entered valid mode command is active if an attempt is made to enter an invalid mode command into the send register.

49.3.2 Receive Registers

The transfer of data from the receive module to the corresponding receive registers occurs at the same time as related status information is updated, directly after the serial-to-parallel conversion.

49.3.2.1 EMPFR1_L : Receive Register 1, Low Double Word

Base address: ENDATm = 0x9020_4000 + 0x1_0000 × m (m = 0 to 11)
ENDATi = 0x9030_4000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x04

Bit position:	31																												0
Bit field:	RDATA_L[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RDATA_L[31:0]	Received data, bit 31 to bit 0 See Table 49.8.	R

49.3.2.2 EMPFR1_H : Receive Register 1, High Double Word

Base address: ENDATm = 0x9020_4000 + 0x1_0000 × m (m = 0 to 11)
ENDATi = 0x9030_4000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	POS1	IF2	F1	CRC[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RDATA_H[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RDATA_H[15:0]	Received data, bit 47 to bit 32 See Table 49.8.	R
20:16	CRC[4:0]	Received CRC	R
21	F1	Safety-relevant error bit F1 0: No error occurred 1: Error occurred or if forced dynamic sampling is active	R
22	IF2	Safety-relevant error bit F2 inversion 0: Error occurred or if forced dynamic sampling is active 1: No error occurred	R
23	POS1	Absolute position value 1 This bit marks Pos1, to which the following additional information with Pos2 belongs. 0: No Pos1 1: Pos1	R
31:24	—	These bits are read as 0.	R

Depending on the transmitted type 2.1 mode command, EMPFR1_L/EMPFR1_H registers contain different data. With EnDat type 2.2 mode commands, the position value is always entered into FMPFR1_L/EMPFR1_H.

Table 49.8 Received data (1 of 2)

Entry	RDATA_H[15:8]	RDATA_H[7:0]	RDATA_L[31:24]	RDATA_L[23:16]	RDATA_L[15:8]	RDATA_L[7:0]
EnDat position value	MSB*3	…	…	…	…	D[7:0]
MRS code*1	(Invalid)	(Invalid)	(Invalid)	MRS code	D[15:8]	D[7:0]

Table 49.8 Received data (2 of 2)

Entry	RDATA_H[15:8]	RDATA_H[7:0]	RDATA_L[31:24]	RDATA_L[23:16]	RDATA_L[15:8]	RDATA_L[7:0]
Address and parameter*1	(Invalid)	(Invalid)	(Invalid)	Address	D[15:8]	D[7:0]
Port address and data*1	(Invalid)	(Invalid)	(Invalid)	Port Address	D[15:8]	D[7:0]
Test values*1	(Invalid)	DV[39:32]	DV[31:24]	DV[23:16]	DV[15:8]	DV[7:0]
Address and instruction*2	(Invalid)	(Invalid)	(Invalid)	Address	D[15:8]	D[7:0]

Note 1. Only for type 2.1 EnDat mode commands
Note 2. Only for EnDat mode command 010010
Note 3. The position of the MSB shifts depending on the length of the position value. The bits to left of the MSB are not part of the position value.

49.3.2.3 EMPFR2 : Receive Register 2

Base address: ENDATm = 0x9020_4000 + 0x1_0000 × m (m = 0 to 11)
ENDATi = 0x9030_4000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x0C

Bit position:	31																													0
Bit field:	RDATA2[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RDATA2[31:0]	Received data 2 Additional information 2 See Table 49.9.	R

If a type 2.2 mode command was sent, EMPFR2 register will contain the contents of additional information 2 and its CRC. Please note that the Additional information 2 is supported is difference depending on the EnDat encoder.

Table 49.9 Received data 2

Entry	RDATA2[31:24]	RDATA2[23:16]	RDATA2[15:8]	RDATA2[7:0]
Additional information 2*1	CRC*2	MRS code*3	D[15:8]	D[7:0]

Note 1. Only with type 2.2 mode commands; otherwise the register will not be updated.
Note 2. CRC: bit 28 to bit 24
Note 3. If the EnDat encoder does not support the Additional information 2, the MRS code may be 0x5F.

49.3.2.4 EMPFR3 : Receive Register 3

Base address: ENDATm = 0x9020_4000 + 0x1_0000 × m (m = 0 to 11)
ENDATi = 0x9030_4000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x10

Bit position:	31			28				24	23							16	15													0
Bit field:	—	—	—	CRC[4:0]					C[7:0]								D[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	D[15:0]	Data in the additional information 1*1	R
23:16	C[7:0]	Memory Range Select (MRS) code in the additional information 1*1 *2	R

Bit	Symbol	Function	R/W
28:24	CRC[4:0]	CRC for additional information 1*1	R
31:29	—	These bits are read as 0.	R

Note 1. Only with type 2.2 mode commands; otherwise, the register will not be updated.
Note 2. If the EnDat encoder does not support the Additional information 1, the MRS code may be 0x4F.

If a type 2.2 mode command was sent, EMPFR3 register will contain the contents of additional information 1 and its CRC.

Please note that the Additional information 1 is supported is difference depending on the EnDat encoder.

49.3.3 Configuration Registers

Setting required for operation and for handling of the corresponding protocols must be made in the configuration registers.

All of the configuration registers can only be written to if no EnDat transmission is active. The bit 30 of STATR register can be evaluated for this purpose.

49.3.3.1 KONFR1 : Configuration Register 1

Base address: ENDATm = 0x9020_4000 + 0x1_0000 × m (m = 0 to 11)
ENDATi = 0x9030_4000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SET[1:0]		ICRST	FSYS[2:0]			—	DELAYCMP	CPTIME[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AUTORST	RSTWD	DWLG[5:0]						FTCLK[3:0]				—	DT	DU	HWSTRB
Value after reset:	0	0	0	0	1	1	0	1	1	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
0	HWSTRB	Hardware strobe Enable external / STR signal as strobe signal. 0: HW strobe not enabled 1: HW strobe enabled	R/W
1	DU	DU Defines the unconditional data transfer to receive registers 1, 2, 3, and 4 on completion of a data transmission process, despite a flag being set in the status register. For safety applications, DU = 1 must be set. 0: Data transfer according to flag set in the status register 1: Data transfer despite the flag in the status register	R/W
2	DT	DT Select the EnDat continuous clock mode. 0: Continuous clock off 1: Continuous clock on	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7:4	FTCLK[3:0]	f_{TCLK} Setting (4 bits) of transmission rate for EnDat from 100 kHz to 16 MHz (EnDat) 0x0: 16.67 MHz 0x1: 16.67 MHz 0x2: 16.67 MHz 0x3: 16.67 MHz 0x6: 8.33 MHz 0xB: 4.16 MHz 0xC: 2 MHz 0xD: 1 MHz 0xE: 200 kHz 0xF: 100 kHz Others: Not permitted	R/W
13:8	DWLG[5:0]	Data word length*1 Data word length is set binary with 6 bits for EnDat, is the sum of MT (multi-turn bits) and ST (single-turn bits). The permissible setting range for EnDat is from 8 bits to 48 bits. 0x00: Not permitted ⋮ 0x07: Not permitted 0x08: Data word length = 8 bits ⋮ 0x0D: Data word length = 13 bits ⋮ 0x30: Data word length = 48 bits 0x31: Not permitted ⋮ 0x3F: Not permitted	R/W
14	RSTWD	Reset window The set bit allows resetting of the status and error register only within a defined time period. For safety applications: Reset window = 1 0: Resetting of the registers mentioned above can be performed at any time. (i.e. without considering malfunctions). 1: Resetting of the registers mentioned above must be performed within a defined time period for acceptance by the protocol engine.	R/W
15	AUTORST	Automatic reset If this bit is set, resetting of the status register and error register is performed automatically. For safety applications: Automatic reset = 0 0: Resetting of the above-mentioned registers must be performed by the application. 1: Resetting of the above-mentioned registers is done automatically. However, this resetting only occurs in the next EnDat transmission with the start of data reception.	R/W
23:16	CPTIME[7:0]	Cable propagation time The cable propagation time determined is stored. The application may change this value. If that is the case the status register's propagation time measurement (LZM) bit will automatically be reset. The binary value has a step width of one system clock. At a system clock of 100 MHz, this corresponds to a setting range from 0 µs to 2.55 µs in steps of 10.0 ns.	R/W
24	DELAYCMP	Delay compensation This bit switches propagation delay compensation on. When this bit is set, propagation time measurement is performed with the next data transmission to the EnDat encoder. The ENDAT determines the cable propagation time and saves this in KONFR1 register. This value is used to determine propagation delay compensation. To measure the propagation time again, the delay compensation bit must be reset and set again. For 16-bit access, it must be considered that the measured cable propagation time value is overwritten with 0x00. 0: Delay compensation off 1: Delay compensation on	R/W
25	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
28:26	FSYS[2:0]	System clock frequency (f_{sys}) The system frequency actually used must be selected. 1 1 0: 100 MHz Others: Not permitted	R/W
29	ICRST	IC reset Setting of this bit has the effect that the entire ENDAT is reset to its initial state. 0: IC reset inactive 1: IC reset active	R/W
31:30	SET[1:0]	EnDat set These two bits set the EnDat transmission mode. 0 0: Not permitted 0 1: Not permitted 1 0: EnDat 1 1: Not permitted	R/W

Note 1. The data word length has to set to 40/d bit while using mode command "encoder transmit test values".

The KONFR1 register specifies the basic settings and EnDat functions.

49.3.3.2 KONFR2 : Configuration Register 2

Base address: ENDATm = 0x9020_4000 + 0x1_0000 × m (m = 0 to 11)
ENDATi = 0x9030_4000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	HWSTRBDLY[7:0]								—	—	FILTER[2:0]			RCVTIME[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WTDOG[7:0]								TMSMPLRATE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TMSMPLRATE[7:0]	Timer for sampling rate 256 different sampling rates can be set. For details, see section 49.4.4. Timer for Sampling Rate. In the default setting 0x00 or 0x80, the timer is off.	R/W
15:8	WTDOG[7:0]	Watchdog 256 different watchdog time values can be set. For details, see section 49.4.5. Watchdog Timer. In the default setting 0x00 or 0x80, the watchdog is off.	R/W
18:16	RCVTIME[2:0]	Recovery time III t_{ST} This time is to be set in accordance with EnDat specification. The set time has an accuracy of ±0.1 µs. 0 0 0: 1/2 TCLK 0 0 1: 0.5 µs 0 1 0: 1 µs 0 1 1: 1.5 µs 1 0 0: 2 µs 1 0 1: 4 µs 1 1 0: 8 µs 1 1 1: 10 µs	R/W

Bit	Symbol	Function	R/W
21:19	FILTER[2:0]	Filter This digital filter for the Data_RC data input can be adjusted in eight steps (3 bits) as below. The filter setting value corresponds to system clock cycles.*1 0 0 0: Off 0 0 1: 3 cycles 0 1 0: 4 cycles 0 1 1: 5 cycles 1 0 0: 6 cycles 1 0 1: 10 cycles 1 1 0: 20 cycles 1 1 1: 40 cycles	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	HWSTRBDLY[7:0]	Hardware strobe delay*2 The application can enter a value for the HW strobe delay. The binary value has a step width of one system clock. 0x00: Off 0x01: Not permitted 0x02: Not permitted 0x03: 3 cycles 0x04: 4 cycles 0x05: 5 cycles ⋮ 0xFF: 255 cycles	R/W

Note 1. The filter must be set according to the transmission rate of the serial interface to the encoder.
Example: f_{TCLK} = 1 MHz (corresponds to 100 system clock cycles with f_{CLK} = 100 MHz)
For the filter, 1/10 of the f_{CLK} must be set. That means 10 system clock cycles → setting: 101b.

Note 2. At a system clock of 100 MHz, this corresponds to a value range from 30 ns to 2.55 µs in steps of 10 ns.

The KONFR2 register specifies protocol-independent settings.

49.3.4 STATR : Status Register

Base address: ENDATm = 0x9020_4000 + 0x1_0000 × m (m = 0 to 11)
ENDATi = 0x9030_4000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDY	RDY4 STRB	SPDR DY	—	—	—	—	—	LZM	LZK	—	—	—	FTYPII I	WTDG	SPIKE
Value after reset:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WRN	RM	BUSY	CRCZI 2	CRCZI 1	IERR2	EMPF R3	EMPF R2	—	—	MRS_ ADR	FTYPII	FTYPI	CRC_ PRTY	ERR1	EMPF R1
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EMPFR1	EMPFR1 register update This status flag indicates that the data in EMPFR1 register has been updated. This flag is ignored if the DU bit in KONFR1 register is enabled. 0: EMPFR1 register not updated 1: EMPFR1 register updated	R/W
1	ERR1	Error 1 The status bit "error 1" from the EnDat protocol is entered. 0: Error1 did not occur 1: Error1 occurred	R/W

Bit	Symbol	Function	R/W
2	CRC_PRTY	CRC-PW / parity With EnDat transmission it represents the result of the CRC check of the received value (position value, parameter or test value). 0: CRC check okay 1: CRC check faulty	R/W
3	FTYPI	F Type I Shows type I error handling in accordance with the EnDat specification at Annex A2. 0: A type I error did not occur 1: A type I error occurred	R/W
4	FTYPII	F Type II Shows type II error handling in accordance with the EnDat specification at Annex A2. 0: A Type II error did not occur 1: A Type II error occurred	R/W
5	MRS_ADR	MRS/Adr The occurrence of an addressing or acknowledgment error is shown. The errors (F type I / II) are special cases of MRS/address errors, i.e. they are a sub-quality of these. Accordingly, whenever a type I or type II error is identified, the MRS/Adr bit is set. For example, if an MRS/address bit is recognized incorrectly due to a disturbance, only the MRS/Adr status bit will be set, not the F TYP I/II bits. 0: No acknowledgment or addressing error has occurred 1: An acknowledgment or addressing error has occurred	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	EMPFR2	EMPFR2 register update This status flag indicates that the data in EMPFR2 register has been updated. It must be cleared after EMPFR2 register has been read to allow the ENDAT to rewrite data. 0: EMPFR2 not updated 1: EMPFR2 updated	R/W
9	EMPFR3	EMPFR3 register update This status flag indicates that the data in EMPFR3 register has been updated. It must be cleared after EMPFR3 register has been read to allow the ENDAT to rewrite data. 0: EMPFR3 not updated 1: EMPFR3 updated	R/W
10	IERR2	/Error2 Contains the "Error 2" status bit from the EnDat protocol (only with EnDat2.2 commands). 0: /Error2 occurred 1: /Error2 did not occur	R/W
11	CRCZI1	CRC-ZI1 During EnDat transmissions, this bit indicates the result of the CRC checking of additional information (ZI) 1. 0: CRC check of ZI1 okay 1: CRC check of ZI1 faulty	R/W
12	CRCZI2	CRC-ZI2 During EnDat transmissions, this bit indicates the result of the CRC checking of additional information (ZI) 2. 0: CRC check of ZI2 okay 1: CRC check of ZI2 faulty	R/W
13	BUSY	Busy Contains the "Busy" status bit as transmitted in the EnDat protocol. 0: Busy = 0 1: Busy = 1	R/W
14	RM	RM Contains the "RM" status bit as transmitted in the EnDat protocol. 0: RM = 0 1: RM = 1	R/W
15	WRN	WRN Contains the "WRN" status bit as transmitted in the EnDat protocol. 0: WRN = 0 1: WRN = 1	R/W

Bit	Symbol	Function	R/W
16	SPIKE	Spike Reports that a "Spike" was detected at the data input port. Condition: filter in the KONFR1 register is set. 0: No spike 1: Spike occurred	R/W
17	WTDG	Watchdog Reports triggering of the watchdog. Condition: watchdog in KONFR2 register is set. 0: Watchdog not triggered 1: Watchdog triggered	R/W
18	FTYPIII	F Type III Type II error (transmission layer) triggers F type III. Error recognition occurs in the EnDat master. 0: The error did not occur 1: The error occurred	R/W
21:19	—	These bits are read as 0. The write value should be 0.	R/W
22	LZK*1	Delay compensation This bit reports if propagation delay compensation is active. If propagation delay compensation in KONFR1 register is switched off, this bit and propagation time measurement will automatically be reset. 0: LZK inactive 1: LZK active	R/W
23	LZM*1	Propagation time measurement This bit reports that propagation time measurement was successfully completed. Condition: propagation delay compensation (LZK) in KONFR1 register is set. 0: LZM incomplete 1: LZM complete	R/W
28:24	—	These bits are read as 0. The write value should be 0.	R/W
29	SPDRDY	Speed ready This bit reports that a new velocity value has been calculated. 0: No new velocity value calculated 1: New velocity value calculated	R/W
30	RDY4STRB	Ready for strobe This bit reports that data transmission has ended and that the EnDat protocol automation machine is ready for the next transmission. The time values "Recovery time 1" (t_m) and "Recovery time 2" (t_R) are completed. This bit cannot be reset by writing a "1" as this is a status display of the current conditions of internal automation engines. The bit cannot cause an interrupt. 0: No ready 1: Ready	R/W
31	RDY	Ready If the ready bit is set, the status register is completely updated. All checks have been performed. Data transmission is not yet completed, however, meaning that the EnDat protocol automation machine is not yet ready again. 0: No ready 1: Ready	R/W

Note 1. Neither the LZM nor the LZK bit can be reset by writing a "1" to the respective bit as this is a status display of the current conditions of internal automation engines. Neither of the two bits can cause an interrupt.

The status bits are created by the sequencing controller, as required. Status information remains set until it is reset by the application.

The application can selectively reset status information with a write command. This occurs by writing "1" to the selected bits. In the event of concurrent access, the internal sequencing controller has priority. This ensures that status information is not 'lost'.

The bit 15 to bit 11 in STATR register are only valid when additional information 1 or 2 has been received.

Note: The status register should be read after each data transmission. It provides information about validity of the data contained in the receive registers. The status bits must be reset in order that the internal sequencing controller can recognize a renewed setting of the status bit.

Note: Each bit (except for LZM, LZK, RDY4STRB) can trigger an interrupt. Masking is performed with the INTMR register. If a bit that has been set (and thus has triggered an interrupt) is reset and if no other bit has triggered an interrupt, the interrupt is deasserted.

49.3.5 INTMR : Interrupt Mask Register

Base address: ENDATm = 0x9020_4000 + 0x1_0000 × m (m = 0 to 11)
ENDATi = 0x9030_4000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDY	—	SPDRDY	—	—	—	—	—	—	—	—	—	—	FTYPIII	WTDG	SPIKE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WRN	RM	BUSY	CRCZI2	CRCZI1	IERR2	EMPFR3	EMPFR2	—	—	MRS_ADR	FTYPII	FTYPI	CRC_PRTY	ERR1	EMPFR1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EMPFR1	Interrupt mask for EMPFR1 register update 0: Masked 1: Not masked	R/W
1	ERR1	Interrupt mask for Error 1 0: Masked 1: Not masked	R/W
2	CRC_PRTY	Interrupt mask for CRC-PW / parity 0: Masked 1: Not masked	R/W
3	FTYPI	Interrupt mask for F Type I 0: Masked 1: Not masked	R/W
4	FTYPII	Interrupt mask for F Type II 0: Masked 1: Not masked	R/W
5	MRS_ADR	Interrupt mask for MRS/Adr 0: Masked 1: Not masked	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	EMPFR2	Interrupt mask for EMPFR2 register update 0: Masked 1: Not masked	R/W
9	EMPFR3	Interrupt mask for EMPFR3 register update 0: Masked 1: Not masked	R/W
10	IERR2	Interrupt mask for /Error2 0: Masked 1: Not masked	R/W
11	CRCZI1	Interrupt mask for CRC-ZI1 0: Masked 1: Not masked	R/W
12	CRCZI2	Interrupt mask for CRC-ZI2 0: Masked 1: Not masked	R/W

Bit	Symbol	Function	R/W
13	BUSY	Interrupt mask for Busy 0: Masked 1: Not masked	R/W
14	RM	Interrupt mask for RM 0: Masked 1: Not masked	R/W
15	WRN	Interrupt mask for WRN 0: Masked 1: Not masked	R/W
16	SPIKE	Interrupt mask for Spike 0: Masked 1: Not masked	R/W
17	WTDG	Interrupt mask for Watchdog 0: Masked 1: Not masked	R/W
18	FTYPIII	Interrupt mask for F Type III 0: Masked 1: Not masked	R/W
28:19	—	These bits are read as 0. The write value should be 0.	R/W
29	SPDRDY	Interrupt mask for Speed ready 0: Masked 1: Not masked	R/W
30	—	This bit is read as 0. The write value should be 0.	R/W
31	RDY	Interrupt mask for Ready 0: Masked 1: Not masked	R/W

The INTMR register is for the masking of the STATR register's interrupt sources. All bits shown in the status register (except for LZM, LZK, RDY4STRB) can generate an interrupt. The bit assignments of the interrupt mask register are identical to those of the status register. An interrupt is allowed by setting the corresponding bit to 1.

49.3.6 SWSTRBR : SW Strobe Register

Base address: ENDATm = 0x9020_4000 + 0x1_0000 × m (m = 0 to 11)
ENDATi = 0x9030_4000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x38

Bit position:	31																												0
Bit field:																													
Value after reset:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	SW strobe The application triggers the data transfer through a write command (set this register to 1). This option is used on applications that are not time-critical.	W

49.3.7 IDR : ID Register

Base address: ENDATm = 0x9020_4000 + 0x1_0000 × m (m = 0 to 11)
ENDATi = 0x9030_4000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x3C

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

Bit	Symbol	Function	R/W
31:0	n/a	EnDat specification ID	R

49.4 Operation

49.4.1 Control Sequence

Figure 49.2 shows the basic operating sequence of ENDAT. The flow charts are valid for EnDat transmission.

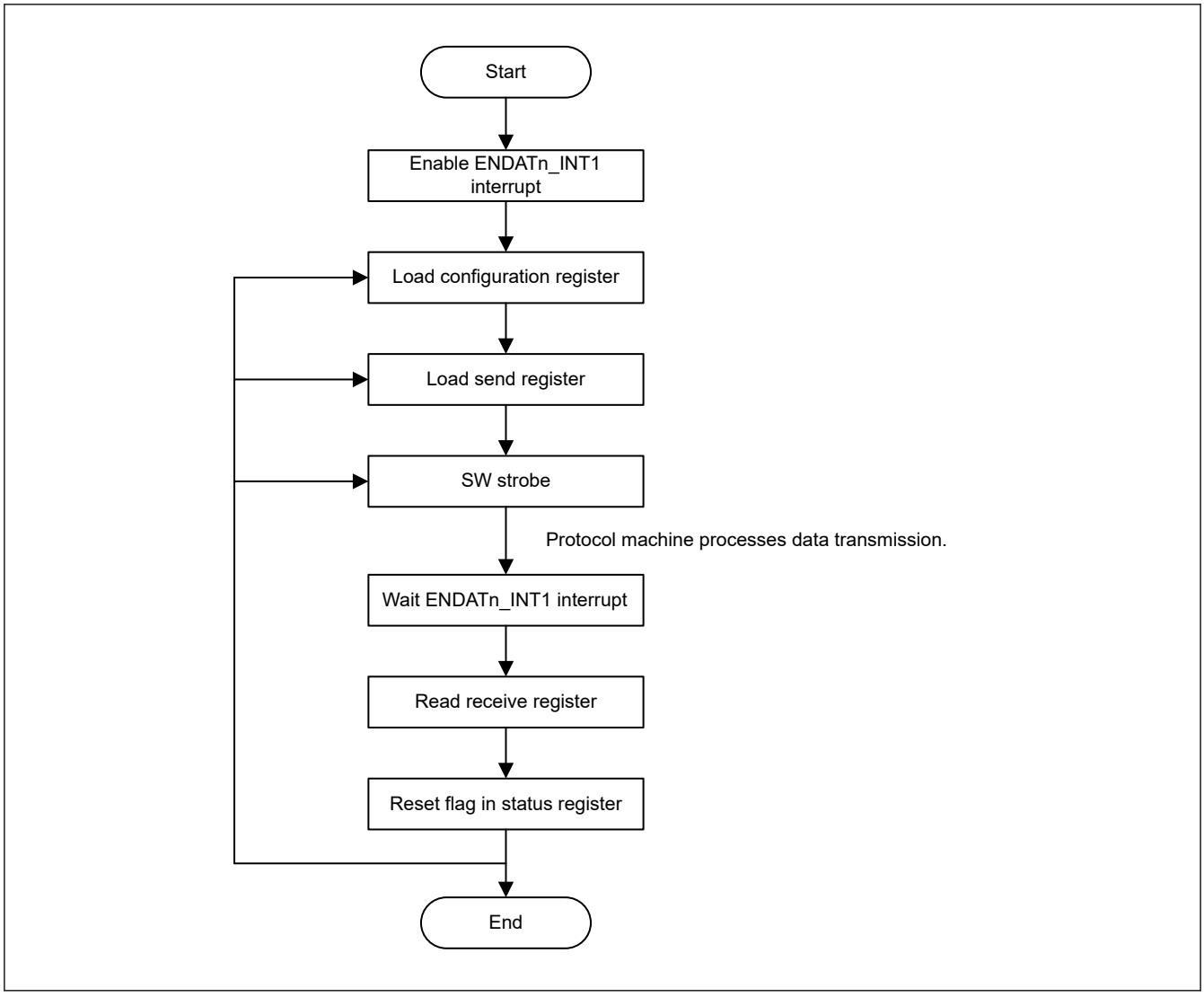


Figure 49.2 Control sequence

49.4.2 EnDat Data Transfer

EnDat is a bidirectional serial interface. The ENDAT starts transmission by sending a clock frequency with a defined length. Synchronously with the clock, a mode command (and other data, if required) is transmitted to the encoder. The encoder responds by sending the requested data to the subsequent electronics. The transmission direction can change multiple times during data transfer (half-duplex).

The EnDat protocol must be selected in the KONFR1 register.

49.4.2.1 Operating Sequence

Figure 49.3 shows the operating sequence of EnDat data transfer. The mode commands can be divided into three groups.

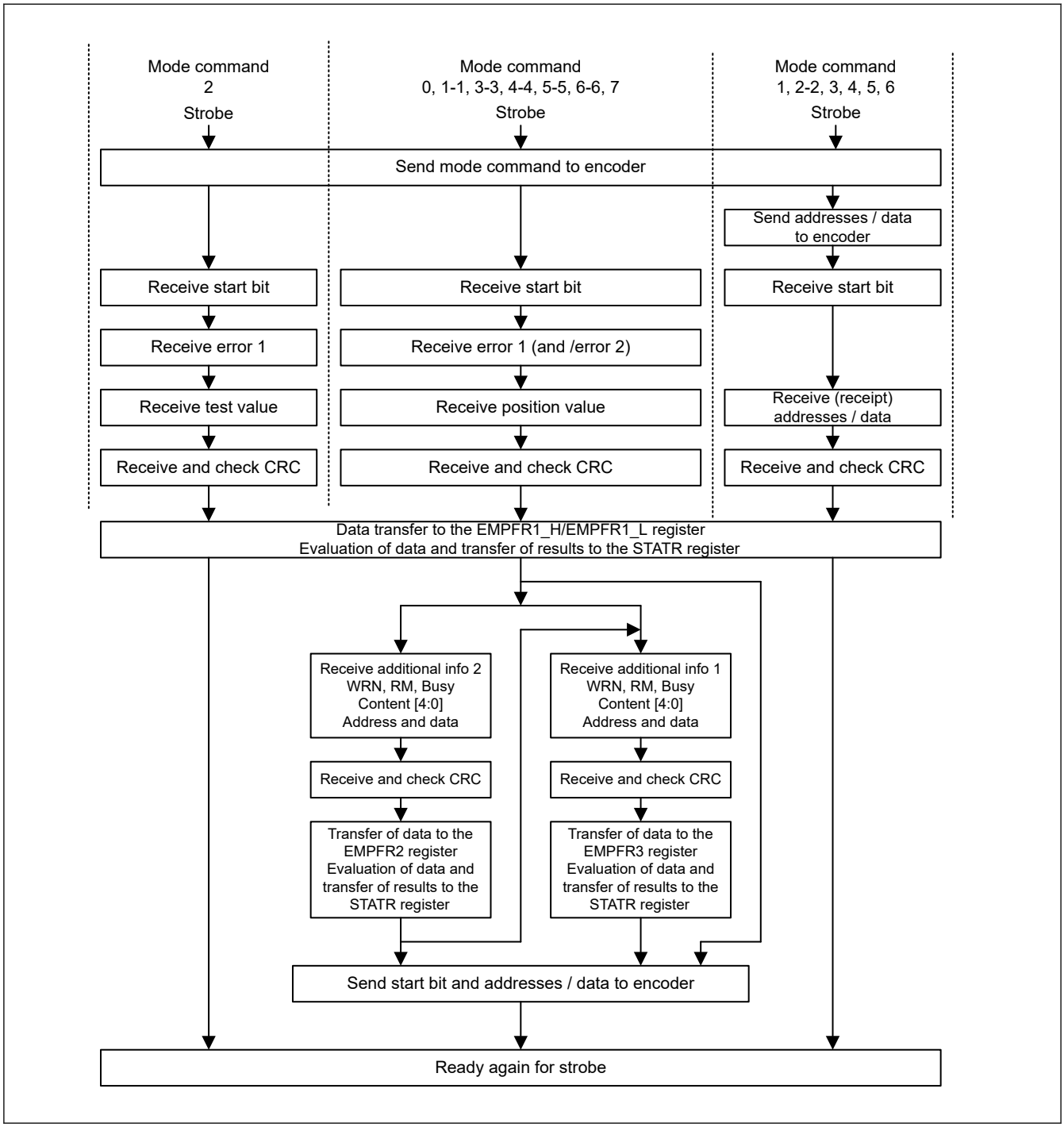


Figure 49.3 EnDat operating sequence

On mode commands for transfer of additional information (center block), the additional information previously selected by mode command 001001b and MRS code for selection (and deselection) of additional information determines the further progress of the protocol, i.e. whether and how many clocks will be sent to receive additional information.

49.4.2.2 Basic Functions

(1) Clock Generator

The clock generator creates the transmission clock required for the synchronous serial interface. The transmission frequency f_{TCLK} is set in the KONFR1 register. The pulse/break ratio specified for EnDat is complied with.

The clock generator also ensures that the number of clocks required for the respective mode commands is output. This is ensured during operation with and without delay compensation.

The number of clocks depends on the data word length and the mode command used as per EnDat specification.

(2) Data Word Length

The number of bits to be transferred for the position value must be set in the KONFR1 register.

(3) Continuous Clock

The ENDAT supports the possibility of executing a position value transfer with continuous clock using mode command 000111b. The settings are made in the KONFR1 register. (Default setting: Continuous clock = off)

Note: The updating rate can be very high. The EMPFR1_L/EMPFR1_H registers are not updated if the corresponding flag in the STATR register is not reset by the application. It is therefore recommended that the data transfer bit, DU, be set to 1 in the KONFR1 register.

(4) Error Handling

The ENDAT executes type I and type II error handling. Type III error handling is executed only for EnDat2.1 mode commands.

Note: This function is supported in the FR register. The results of type I and type II error handling are shown in the STATR register.

(5) Encoder Status

The status messaged from the encoder transported in the EnDat protocol are reflected to ERR1, IERR2, WRN, RM, and BUSY bits in the STATR register after every data transfer.

(6) Cyclical Redundance Check (CRC)

The CRC bits received are compared with the CRC bits generated from the received data flow.

The CRC result of the position value and the data and addresses (with EnDat 2.1 commands) is saved in the STATR register under CRC-PW.

The CRC result of additional information 1 (2) is shown in the STATR register under CRC-ZI1 (2).

(7) Cable Propagation Time

At high transmission rates and with long cable lengths, the cable propagation time must be compensated for technical reasons.

The cable propagation time is measured during the next EnDat transmission[*1] if a corresponding entry is made in the KONFR1 register (cable propagation time = on). The propagation time measurement is carried out at a reduced transmission frequency (independent of the selected setting).

The measured value "cable propagation time" is saved in the KONFR1 register and is taken into account for subsequent transmissions.

After completion of the propagation time measurement, the bit "propagation time measurement completed" (LZM) in the STATR register is set to 1 and the bit "Delay compensation" (LZK) in the STATR register is set to 1.

If the measurement is to be repeated, the delay compensation bit (DELAYCMP) must be reset and then set again in the KONFR1 register. (For 16-bit access, it must be considered that the measured cable propagation time value in the KONFR1 register is overwritten with 0x00.)

In additional to that, the application can correct the (measured) cable propagation time by overwriting the propagation value in the KONFR1 register. In this case the propagation time measurement (LZM) display in the STATR register disappears (LZM to 0).

This function is supported with certified cables only.

Note 1. Use mode command 0 "Encoder transmit position values"

(8) Recovery Time II (Max. 500 ns)

During recover time the EnDat master does not permit starting an EnDat transmission. If delay compensation is enabled and the propagation time value entered in the control register is higher than or equals 500 ns, the master does not implement it additionally. In this case, the master is ready for a new transmission at 5 to 6 system clocks after the H/L edge on DATA_RC. The bit 30, RDY4STRB, in the STATR register signals to the user that the EnDat system is ready for another transmission.

(9) Recovery Time III

At high transmission rates and with long cable lengths, recovery time III t_{ST} must, for technical reasons, be set in the KONFR2 register. (Default setting: 2 µs)

Note: The application can change the value for recovery time III t_{ST} if required. The time (t_{ST}) in the KONFR2 register should be greater than the measured propagation delay time plus 20 percent. (t_{ST} > 1.2 × propagation delay time)

49.4.2.3 Advanced Functions

(1) Watchdog

The watchdog allows to set a maximum value for the transmission duration. If this value is exceeded (e.g. due to an error), the transmission will be aborted and the EnDat sequencing controller is redefined to the outset. Register will not be reset, however. If a propagation time measurement or delay compensation was carried out before, it is retained and remains active. (This means that the delay compensation is not reset, either.)

The watchdog timer can be set in the KONFR2 register. If it is enabled, it starts when the EnDat transmission begins and stops when the EnDat transmission ends (EnDat engine idle). The status of the watchdog is shown in the STATR register.

The reaction of the watchdog on the interrupt output can be made visible by means of the interrupt mask. (Default setting = off)

The watchdog is ineffective with the following functions:

- Continuous clock
- Mode command 5 ("Encoder receive reset")
- Mode command 3 and 4 ("Encoder receive/transmit parameters")

(2) Reset with Selected Additional Information

If additional information was selected before a reset, it must be taken into consideration that this information is lost due to the reset. However, the measuring instrument keeps this information and still expects clock pulses for the additional information. For this reason, the EnDat2.1-Reset command must be given after the reset of the EnDat Master. (This EnDat command then deselects the additional information in the measuring instrument too.)

49.4.3 Interrupt

The ENDATn_INT1 interrupt output triggers status information at the output in accordance with INTMR register settings. (n = 0 to 15)

49.4.4 Timer for Sampling Rate

Timer values for the sampling rate are set in the KONFR2 register.

Table 49.10 shows the programming values of sampling rate.

Two ranges are available for selection (microseconds and milliseconds).

Note: If the timer and the HW strobe delay are active at the same time, KONFR2[31:24], the timer output (Trigger signal), / timer, stays active, but transmission is only triggered with an H/L edge at the /STR input from ELC.

Table 49.10 Programming values of sampling rate

µs: KONFR2[7] = 0		ms: KONFR2[7] = 1	
KONFR2[6:0]	**Sampling rate (µs)**	**KONFR2[6:0]**	**Sampling rate (ms)**
0x00	Stop	0x00	Stop
0x01	2	0x01	0.2
0x02	4	0x02	0.4
0x03	6	0x03	0.6
⋮	⋮	⋮	⋮
0x0A	20	0x0A	2.0
⋮	⋮	⋮	⋮
0x7F	254	0x7F	25.4

49.4.5 Watchdog Timer

Setting for the Watchdog is entered in the KONFR2 register.

Table 49.11 shows the programming values of Watchdog timer.

Two ranges are available for selection (microseconds and milliseconds).

Note: Accuracy: +2 µs or 0.2 ms. Depending on the range setting.

Table 49.11 Programming values of timer

µs: KONFR2[15] = 0		ms: KONFR2[15] = 1	
KONFR2[14:8]	**Timer (µs)**	**KONFR2[14:8]**	**Timer (ms)**
0x00	Stop	0x00	Stop
0x01	2	0x01	0.2
0x02	4	0x02	0.4
0x03	6	0x03	0.6
⋮	⋮	⋮	⋮
0x0A	20	0x0A	2.0
⋮	⋮	⋮	⋮
0x7F	254	0x7F	25.4

50. BiSS-C (BISS)

50.1 Overview

The MPU includes BiSS (BISS) master to enable communication between the encoder and BiSS-C interface. Table 50.1 describes the BiSS-C specifications and Figure 50.1 shows a block diagram of the BISS.

Table 50.1 BiSS-C (BISS) specifications

Item	Description
Number of units	16 units (12 in LLPP0 and 4 in LLPP1)
Protocol	BiSS-C, SSI
Functions	• Bidirectional BiSS communication with up to 8 slaves • Support SSI protocol for unidirectional data transmission • 1 physical BiSS channel each unit • Synchronous sensor data acquisition with cycle data transfers • Slave register operations during cyclic data transfers • Automatic compensation of line delays and conversion times • Data lengths of up to 64 bits for sensor data, configurable for each slave • Data verification by CRC polynomials of up to 16 bits per slave • Separate memory banks enable free controller access during BiSS sensor data transfers • 64 bytes memory for bidirectional register communication • Parallel interface with 32-bit data/address bus • Point-to-point connection (no bus connection: No MO pin) • System Clock (fCLK): 20 MHz (PCLKBISS/4)
Interrupt sources	• EOT: End-of-transmission • NER: Error output
Event link	GETSENS input for HW trigger provided from ELC
Module-stop function	Module-stop state can be set to reduce power consumption

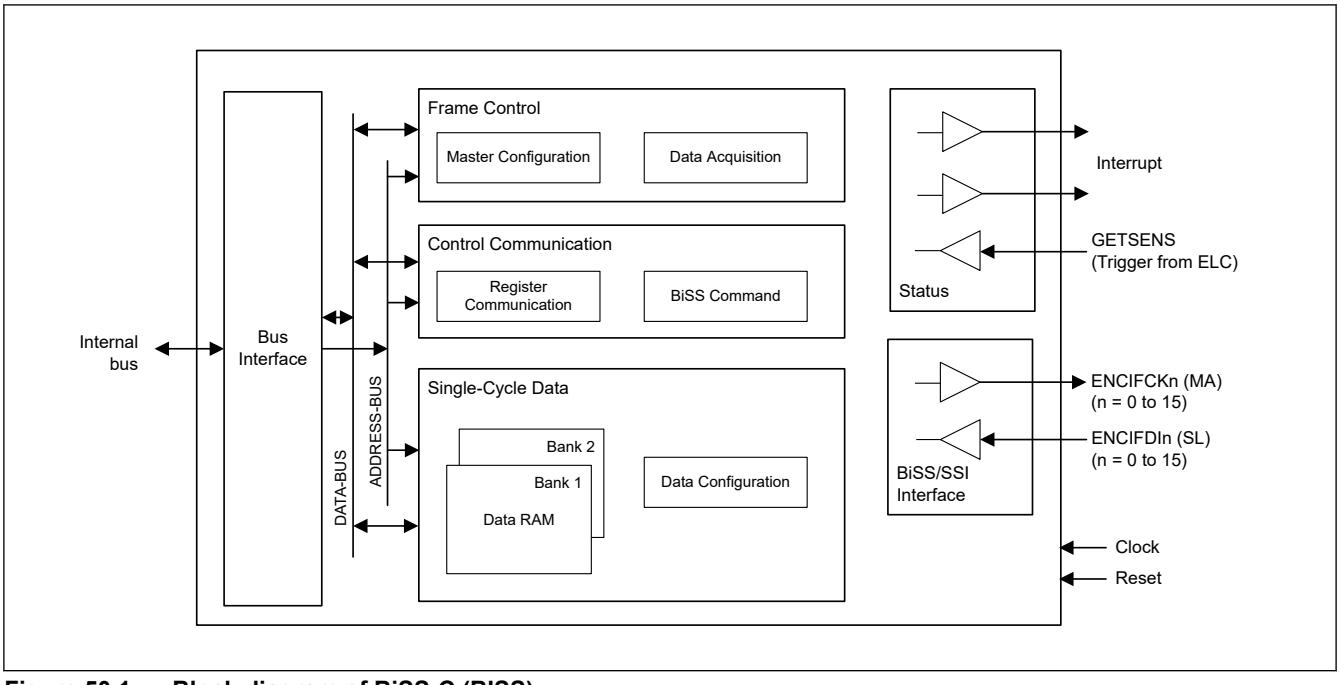


Figure 50.1 Block diagram of BiSS-C (BISS)

Table 50.2 PIN configuration of the BISS

Unit	Pin name	I/O	Function
BISSn	ENCIFCKn (MA)	Output	BiSS-C unit n clock output
	ENCIFDIn (SL)	Input	BiSS-C unit n data input

Note: n = 0 to 15

Table 50.3 BISS interrupt sources

Name	Interrupt sources
BISSn_EOT	End-of-transmission interrupt
BISSn_NER	Error interrupt

Note: n = 0 to 15

50.2 Register Map

Table 50.4 BISS register map

Address	Register symbol	Register name	Write protection
0x9020_3000 + 0x08 × (n - 1) + 0x1_0000 × m (m = 0 to 11) 0x9030_3000 + 0x08 × (n - 1) + 0x1_0000 × (m - 12) (m = 12 to 15)	SCDATALn	Sensor Data L (n = 1 to 8)	—
0x9020_3004 + 0x08 × (n - 1) + 0x1_0000 × m (m = 0 to 11) 0x9030_3004 + 0x08 × (n - 1) + 0x1_0000 × (m - 12) (m = 12 to 15)	SCDATAHn	Sensor Data H (n = 1 to 8)	—
0x9020_3080 + 0x04 × (n - 1) + 0x1_0000 × m (m = 0 to 11) 0x9030_3080 + 0x04 × (n - 1) + 0x1_0000 × (m - 12) (m = 12 to 15)	RDATAn	Register Data n (n = 1 to 16)	—
0x9020_30C0 + 0x04 × (n - 1) + 0x1_0000 × m (m = 0 to 11) 0x9030_30C0 + 0x04 × (n - 1) + 0x1_0000 × (m - 12) (m = 12 to 15)	CNFSLVn	Configuration Slave n (n = 1 to 8)	—
0x9020_30E0 + 0x1_0000 × m (m = 0 to 11) 0x9030_30E0 + 0x1_0000 × (m - 12) (m = 12 to 15)	REGACC	Register Access	—
0x9020_30E4 + 0x1_0000 × m (m = 0 to 11) 0x9030_30E4 + 0x1_0000 × (m - 12) (m = 12 to 15)	CTRLCOMM	Control Communication	—
0x9020_30E8 + 0x1_0000 × m (m = 0 to 11) 0x9030_30E8 + 0x1_0000 × (m - 12) (m = 12 to 15)	MSTRCNF	Master Configuration	—
0x9020_30EC + 0x1_0000 × m (m = 0 to 11) 0x9030_30EC + 0x1_0000 × (m - 12) (m = 12 to 15)	CHCNF	Channel Configuration	—
0x9020_30F0 + 0x1_0000 × m (m = 0 to 11) 0x9030_30F0 + 0x1_0000 × (m - 12) (m = 12 to 15)	STINF	Status Information	—
0x9020_30F4 + 0x1_0000 × m (m = 0 to 11) 0x9030_30F4 + 0x1_0000 × (m - 12) (m = 12 to 15)	INSTREG	Instruction Register	—
0x9020_30F8 + 0x1_0000 × m (m = 0 to 11) 0x9030_30F8 + 0x1_0000 × (m - 12) (m = 12 to 15)	STINF2	Status Information 2	—

Table 50.5 BISS related system control register (1 of 2)

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0 (LLPP0)	—	MSTPCRJ.MSTPCRJ02	SLVACCCTL7.LLPP_SL*1
1 (LLPP0)	—	MSTPCRJ.MSTPCRJ06	SLVACCCTL7.LLPP_SL*1
2 (LLPP0)	—	MSTPCRJ.MSTPCRJ10	SLVACCCTL7.LLPP_SL*1
3 (LLPP0)	—	MSTPCRJ.MSTPCRJ14	SLVACCCTL7.LLPP_SL*1
4 (LLPP0)	—	MSTPCRJ.MSTPCRJ18	SLVACCCTL7.LLPP_SL*1
5 (LLPP0)	—	MSTPCRJ.MSTPCRJ22	SLVACCCTL7.LLPP_SL*1
6 (LLPP0)	—	MSTPCRJ.MSTPCRJ26	SLVACCCTL7.LLPP_SL*1
7 (LLPP0)	—	MSTPCRJ.MSTPCRJ30	SLVACCCTL7.LLPP_SL*1
8 (LLPP0)	—	MSTPCRK.MSTPCRK02	SLVACCCTL7.LLPP_SL*1
9 (LLPP0)	—	MSTPCRK.MSTPCRK06	SLVACCCTL7.LLPP_SL*1
10 (LLPP0)	—	MSTPCRK.MSTPCRK10	SLVACCCTL7.LLPP_SL*1
11 (LLPP0)	—	MSTPCRK.MSTPCRK14	SLVACCCTL7.LLPP_SL*1
12 (LLPP1)	—	MSTPCRK.MSTPCRK18	SLVACCCTL7.LLPP_SL*1

Table 50.5 BISS related system control register (2 of 2)

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
13 (LLPP1)	—	MSTPCRK.MSTPCRK22	SLVACCCTL7.LLPP_SL*1
14 (LLPP1)	—	MSTPCRK.MSTPCRK26	SLVACCCTL7.LLPP_SL*1
15 (LLPP1)	—	MSTPCRK.MSTPCRK30	SLVACCCTL7.LLPP_SL*1

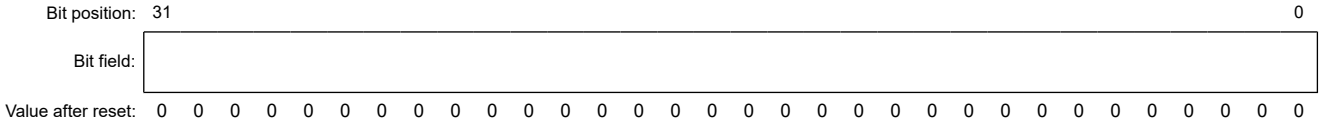
Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

50.3 Register Descriptions

50.3.1 SCDATALn : Sensor Data L (n = 1 to 8)

Base address: BISSm = 0x9020_3000 + 0x1_0000 × m (m = 0 to 11)
BISSi = 0x9030_3000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x00 + 0x08 × (n - 1)

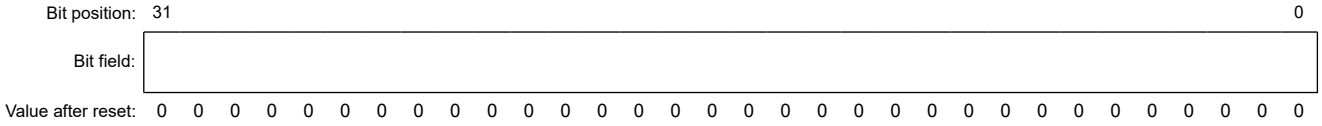


Bit	Symbol	Function	R/W
31:0	n/a	Sensor data lower for slave n This register stores the lower four bytes (SCDATAn[31:0]). For details, see section 50.4.3. Sensor Data.	R/W

50.3.2 SCDATAHn : Sensor Data H (n = 1 to 8)

Base address: BISSm = 0x9020_3000 + 0x1_0000 × m (m = 0 to 11)
BISSi = 0x9030_3000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x04 + 0x08 × (n - 1)

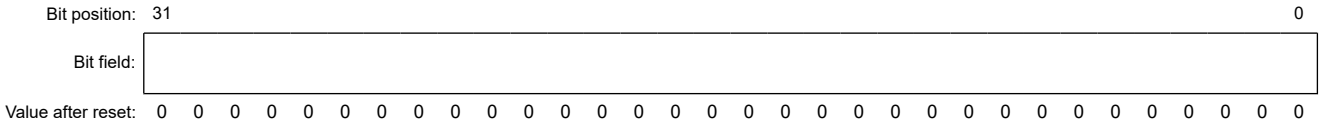


Bit	Symbol	Function	R/W
31:0	n/a	Sensor data upper for slave n This register stores the upper four bytes (SCDATAn[63:32]). For details, see section 50.4.3. Sensor Data.	R/W

50.3.3 RDATAn : Register Data n (n = 1 to 16)

Base address: BISSm = 0x9020_3000 + 0x1_0000 × m (m = 0 to 11)
BISSi = 0x9030_3000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0x80 + 0x04 × (n - 1)



Bit	Symbol	Function	R/W
31:0	n/a	● Register communication data n (RDATAn) Using register communication in control communication (CTRLCOMM.CTS = 1). There is an individual 64 bytes storage area used to store the data of automatic register communication. In case of a register write access to a BiSS slave, RDATAn must be written with register data before starting the BiSS communication with INSTREG.INSTR. In case of a register read access to a BiSS slave, the received register data is stored in RDATAn after successful BiSS communication. In contradiction to SCDATAn there is only one memory bank for RDATAn. Therefore, after performing register communication, RDATAn may only be accessed, if no control communication is currently active (STINF.REGEND = 1). ● Slave addressing for BiSS commands (IDS[7:0]) Using BiSS commands in control communication (CTRLCOMM.CTS = 0). When 0x00: All slaves addressed (broadcast). The bits correspond to N. (For details, see section 50.4.4.1. BiSS Commands.) 0: Slave with ID (7 – N) is not addressed 1: Slave with ID (7 – N) is addressed	R/W

50.3.4 CNFSLVn : Configuration Slave n (n = 1 to 8)

Base address: BISSm = 0x9020_3000 + 0x1_0000 × m (m = 0 to 11)
BISSi = 0x9030_3000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0xC0 + 0x04 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SCRCSTART[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SELCRCS	SCRCLENPOLY[6:0]							GRAYS	ENSCD	SCDLEN[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	SCDLEN[5:0]	Single-cycle data length of slave n For more information about the length, refer to section 50.5.1. The value to be set to SCDLEN in the CNFSVn register and section 50.5.2. The storage of received CRC when SCDLEN = 33 bits or more. 0x00: Single-cycle data length 1 0x01: Single-cycle data length 2 ⋮ 0x3F: Single-cycle data length 64	R/W
6	ENSCD	Enable single-cycle data for slave n 0: Single-cycle data not available 1: Single-cycle data available	R/W
7	GRAYS	Enable GRAY to binary conversion for SSI encoder*1 This bit is valid when CHCNF.CFGCH1[1:0] = 0x2. 0: SSI single-cycle data binary coded 1: SSI single-cycle data GRAY coded	R/W

Bit	Symbol	Function	R/W
14:8	SCRCLENPOLY[6:0]	CRC length with predefined CRC polynomial for slave n (SCRCLEN[6:0]) 0x00: CRC verification deactivated 0x03: 1011b = 0xB 0x04: 1_0011b = 0x13 0x05: 10_0101b = 0x25 0x06: 100_0011b = 0x43 0x07: 1000_1001b = 0x89 0x08: 1_0010_1111b = 0x12F 0x10: 1_1001_0000_1101_1001b = 0x190D9 Others: codes are not permitted CRC polynomial for slave n (SCRCPOLY[7:1]) Since the least significant bit of the CRC polynomial is always 1, there is no need to store this information in the CRC polynomial. Therefore, a maximum polynomial length of 8 bit is possible. If the maximum 8 bits polynomial length is not required, the used polynomial (without the LSB bit) is stored right aligned and leading bits are filled with 0. Example: The CRC polynomial 0x43 = 100 0011b is stored as 010 0001b (value 0x21). 0: CRC verification deactivated n (other than 0): n & 1	R/W
15	SELCRCS	CRC polynomial selection for slave n BiSS uses CRC polynomials depending on the data length predefined in the Electronic Data Sheet (EDS) and Profile Definitions. This bit chooses between predefined polynomials selected according to the CRC length with SCRCLEN and individual CRC polynomials SCRCPOLY. If SCRCLEN = 0, the CRC verification is deactivated. 0: Apply CRC polynomial as predefined in SCRCLEN 1: Apply CRC polynomial as individually defined in SCRCPOLY	R/W
31:16	SCRCSTART[15:0]	CRC calculation start value for slave n The CRC start values can be used to safely differentiate between single-cycle data (SCD) data words e.g., two position words provided by two different sensors. The same CRC start value that is used in the sensor needs to be applied in the master to validate the received single-cycle data. The CRC start value has a length of 16 bits. A typical default slave CRC start value is 0x0000.	R/W

Note 1. FREQR = 0x05 should be set, if CHCNF.CFGCH1 = 0x01.

50.3.5 REGACC : Register Access

Base address: BISSm = 0x9020_3000 + 0x1_0000 × m (m = 0 to 11)
BISSi = 0x9030_3000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0xE0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	REGNUM[5:0]						WNR	REGADR[6:0]							—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	The read values are undefined. The write value should be 0.	R/W
22:16	REGADR[6:0]	Register communication start address	R/W
23	WNR	Register communication direction 0: Read register data 1: Write register data	R/W
29:24	REGNUM[5:0]	Number of consecutive registers to access If REGNUM > 1, a sequential register access is performed. (See BiSS-C Protocol Description) 0x00: Register count 1 0x01: Register count 2 ⋮ 0x3F: Register count 64	R/W
31:30	—	The read values are undefined. The write value should be 0.	R/W

This register is used to configure the BiSS-C control communication for register communication. For details, see section 50.4.4.2. Register Access.

50.3.6 CTRLCOMM : Control Communication

Base address: BISSm = 0x9020_3000 + 0x1_0000 × m (m = 0 to 11)
BISSi = 0x9030_3000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0xE4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	NOCRC	SINGLEBANK	FREQR[2:0]			FREQS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CTS	REGVERS	SID_CMD_IDT[2:0]			—	—	HOLDCDM	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	HOLDCDM	Behavior of MA signal at the end of frame In general operation, the BiSS master is configured according to the data of the connected BiSS slave(s) as described in CNFSLVn registers. A timeout period follows after the data transmission. And after the timeout period expires, the end of the data transmission is signaled to the BiSS master by a rising edge of SL. As soon as the timeout is detected by the BiSS master, it can start a new BiSS frame. If SL signal is not digital low at the end of the frame, e.g., because the BiSS-C frame has not been clocked out completely, MA signal can be programmed with HOLDCDM = 1 to be constant until the next frame begins. Thereby it is ensured that the slave can recognize CDM and take part in control communication, even if the master is not configured according to the data of the connected BiSS slave(s). 0: MA changes to digital high after detecting the slave's timeout termination at SL 1: MA remains constant until the next frame begins	R/W
10:9	—	These bits are read as 0. The write value should be 0.	R/W
13:11	SID_CMD_IDT[2:0]	● Slave addressing for register communication (SLAVEID[2:0]) Using register communication in control communication (CTS = 1). ● BiSS Command of addressed slave (CMD[1:0]) (Corresponds to SID_CMD_IDT[2:1] bits) Using BiSS commands in control communication (CTS = 0). CMD determines which BiSS command is sent. Some BiSS commands are predefined in the BiSS-C Protocol Description, others can be used for device-specific functionality of the slave. ● BiSS command execution control (IDA_TEST) (Corresponds to SID_CMD_IDT[0] bit) Using BiSS commands in control communication (CTS = 0). In contrast to the broadcast BiSS commands, the reception of an addressed BiSS command is confirmed during IDA (ID Acknowledge) by the addressed slave(s). This bit defines if the master triggers the command execution immediately or after checking the IDA bits (see Figure 50.7). The BiSS command is only executed if the check is successful. 0: Immediate execution 1: The slave's feedback (IDA) is tested before execution. The BiSS command is not executed if test fails.	R/W
14	REGVERS	Type of protocol for register communication Supports autonomous register communication with BiSS-C. REGVERS is used to select the protocol type. 0: Not permitted 1: Register communication BiSS-C	R/W
15	CTS	Type of control communication 0: BiSS Commands 1: Register communication	R/W

Bit	Symbol	Function	R/W
20:16	FREQS[4:0]	Single-cycle data clock frequency at MA (fMA) The master clock at MA signal is generated for both BiSS and SSI from the system clock (fCLK) and depends on the frequency division by FREQS. With a system clock frequency of 20 MHz, the clock frequency at MA ranges from 10 MHz down to 62.5 kHz. Single-cycle data clock frequency (fMA): (N: Decimal number) 0x00: fCLK/2/(N + 1) ⋮ 0x0F: fCLK/2/(N + 1) 0x10: Not permitted 0x11: fCLK/20/(N - 15) ⋮ 0x1F: fCLK/20/(N - 15)	R/W
23:21	FREQR[2:0]	BiSS register data frequency*1 *2 BiSS slave device recognizes an idle bus at the end of a transmission frame via timeout detection. Thus, the choice of possible clock frequencies is limited as the duration of each logic level may not exceed the shortest timeout of all the connected slaves. 0 0 0: $fMA/2^{(N + 1)}$ ⋮ 1 1 1: $fMA/2^{(N + 1)}$	R/W
24	SINGLEBANK	Usage of one/two RAM bank(s) for SCDATA With SINGLEBANK = 0, two RAM banks are used for buffering SCDATA. Thus, it is ensured that one RAM bank can be written with single-cycle data received at the BiSS interface while the single-cycle data content of the other RAM bank is read. A downgrade is possible by setting SINGLEBANK = 1. 0: Two RAM banks are used for SCDATA 1: One RAM bank is used for SCDATA	R/W
25	NOCRC	Storage of received CRC in SCDATA When NOCRC = 1, the CRC is not updated to the SCDATA register, but it is not cleared to 0. Therefore, switching from NOCRC = 0 to NOCRC = 1 leaves the last data in the register. When clearing the CRC to 0, the SCDDATA register must be cleared according to the number of the RAM banks used. 0: CRC of SCD is stored in SCDATA 1: CRC of SCD is not stored in SCDATA	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. FREQR = 0x05 should be set, if CHCNF.CFGCH1 = 0x01.
Note 2. BiSS-C devices generally permit a lower clock frequency. BiSS-C devices do not use a MA clock duty cycle (PWM signal) and can be operated down to 80 kHz.

The control communication enables protected and confirmed reading and writing of the registers of a slave (register communication). It is also used for protected and confirmed sending of BiSS commands to specified slaves or to all slaves. The control data is compounded of 1 bit per single-cycle data frame in each direction. The control data bit sent as part of the single-cycle data frame from master to the slaves is called CDM (Control Data Master) and the control data bit sent from a slave to the master is called CDS (Control Data Slave). For more information about the control communication refer to the BiSS-C Protocol Description.

50.3.7 MSTRCNF : Master Configuration

Base address: BISSm = 0x9020_3000 + 0x1_0000 × m (m = 0 to 11)
BISSi = 0x9030_3000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0xE8

Bit position:	31							24	23							16								7							0
Bit field:	VERSION[7:0]								REVISION[7:0]								—	—	—	—	—	—	—	—	FREQAGS[7:0]							
Value after reset:	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	FREQAGS[7:0]	Frame repetition rate (AGS frequency divider)*1 *2 FREQAGS controls the automatic data transmission (INSTREG.AGS). With FREQAGS the frame repetition rate can be set to a dedicated ratio of the system clock frequency (fCLK). With a system clock frequency of 20 MHz frame repetition rates from 1 µs to 4 ms are possible. FREQAGS must be set in a way that the time in between two frames is greater than a complete cycle. One cycle consists of the transmission of a request, an acknowledge signal (including line delay), a start bit (including processing time), a control data bit (BiSS-C), the sensor data and CRC bits of each slave and the maximum BiSS timeout of all connected slaves. FREQAGS is activated with INSTREG.AGS. (N: Decimal number, fCLK: system clock frequency) 0x00: fCLK/20/(N + 1) ⋮ 0x7B: fCLK/20/(N + 1) 0x7C: AGSMIN 0x7D: AGSINFINITE ⋮ 0x7F: AGSINFINITE 0x80: fCLK/625/(N - 127) ⋮ 0xFF: fCLK/625/(N - 127)	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	REVISION[7:0]	Design ID	R
31:24	VERSION[7:0]	Version	R

Note 1. AGSMIN: The master automatically restarts the next frame after the previous frame is finished. The master automatically generates the highest frame rate possible. AGSMIN requires complete BiSS frames to ensure a low level at SL at the end of each frame. The rate depends on the configured sensor data clock frequency (CTRLCOMM.FREQS), the slave configurations (e.g., CNFSLVn.SCDLEN), the maximum processing time of the slaves and the total system line delay.

Note 2. AGSINFINITE: The master does not automatically restart the next cycle. AGSINFINITE requires a trigger event to start the next frame. The next frame is started by a digital pulse at GETSENS signal or by setting INSTREG.INSTR.

50.3.8 CHCNF : Channel Configuration

Base address: BISSm = 0x9020_3000 + 0x1_0000 × m (m = 0 to 11)
BISSi = 0x9030_3000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0xEC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFGCH1[1:0]		—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	—	This bit is read as 1. The write value should be reset value.	R/W
7:1	—	These bits are read as 0. The write value should be reset value.	R/W
9:8	CFGCH1[1:0]	Channel configuration 0 0: Not permitted 0 1: BiSS-C 1 0: SSI 1 1: Channel is not used (no device connected)	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

50.3.9 STINF : Status Information

Base address: BISSm = 0x9020_3000 + 0x1_0000 × m (m = 0 to 11)
BISSi = 0x9030_3000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0xF0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CDMTIMEOUT	CDSSEL	REGBYTES[5:0]						SVALID[15:8]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SVALID[7:0]								nERR	nAGSERR	nDELAYERR	nSCDERR	nREGERR	REGEND	—	EOT
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1

Bit	Symbol	Function	R/W
0	EOT	End of transmission EOT flag is connected to EOT signal and signals a running frame (see section 50.4.2. BiSS-C). 0: Data transmission active 1: Data transmission not active	R
1	—	This bit is read as 0. The write value should be 0.	R/W
2	REGEND	End of Control Communication REGEND indicates a finished control communication. It is also set if the control communication was not successful. REGEND flag is reset in the following three patterns: ● After power on ● Control communication has not started after initialization by INSTREG.INIT ● Control communication running 0: Control communication running or not started since power on/INIT 1: Control communication completed/not running	R
3	nREGERR	Control communication error nREGERR flag is set after power on and after executing INSTREG.INIT. If a register data error is detected, the number of bytes transmitted correctly before the error occurred is provided by the register message REGBYTES. In case of an error, the transmission of data is terminated. The following issues can cause error indication: ● Read register access: CRC error or inverted R-bit ● Write register access error in repeated CDS data ● BiSS Commands: error in IDA 0: Error in last control communication 1: No error in last control communication	R
4	nSCDERR	Single-cycle data (SCD) transmission error An error in the single-cycle data detected by checksum verification (CRC) is shown with nSCDERR. If nSCDERR = 0, the faulty sensor can be identified by reading SVALID bits. 0: Error in last single-cycle data transmission 1: No error in last single-cycle data transmission	R
5	nDELAYERR	Start bit in register communication*1 nDELAYERR flag is set after power on and after executing INSTREG.INIT. nDELAYERR notifies about a missing start bit in last register communication. The addressed slave register might not be implemented. 0: Missing start bit in last register data transmission. Register address is not implemented or register access processing time too long (> 20 ms, if CTRLCOMM.FREQR = 0x05). 1: No missing start bit in last register data transmission	R

Bit	Symbol	Function	R/W
6	nAGSERR	Automatic Get Sensor (AGS) error When automatic transmission of sensor data is enabled with instruction bit INSTREG.AGS, an AGS watchdog error nAGSERR is indicated, if no new cycle could be initiated. If the last BiSS frame has not been finished in time, the next BiSS frame will be omitted. The following BiSS frame will be executed if possible. nAGSERR flag is also indicated, if a write access to INSTREG register is performed (e.g., by writing INSTREG.BREAK). 0: At least one BiSS frame has been omitted 1: No missing BiSS frames	R
7	nERR	State of the error interrupt (NER) nERR flag indicates the state of NER signal. Any of the errors is indicated. 0: Internal error occurred 1: No error occurred	R
23:8	SVALID[15:0]	SCDATAn validity indication (n: slave, n = 1 to 8)*2 Two bits are assigned to each slave, the upper bit is valid (SVALID[1]: SVALID1 for Slave 1, SVALID[3]: SVALID2 for Slave 2, SVALID[5]: SVALID3 for Slave 3, and so on). The lower bit is invalid, and the write value should be 0. The CRC verification result of the received single-cycle sensor data of every BiSS frame is written to the validity SVALID for each slave separately. If the CRC is disabled in the slave configuration, the correspondent SVALID flag is set after reading of sensor data has been completed. The validity flags are reset by writing to SVALID. 0: Single-cycle data invalid 1: Single-cycle data valid	R/W
29:24	REGBYTES[5:0]	Number of valid register bytes In case of an error in a sequential register communication (nREGERR = 0), the number of faultless transmitted register values are stored in REGBYTES. These bits are reset if an initialization sequence (INSTREG.INIT) is transmitted or a new control communication started.	R
30	CDSSEL	State of Control Data Slave (CDS)*3 0: CDS value 0 1: CDS value 1	R
31	CDMTIMEOUT	Control Data Master (CDM) timeout reached For BiSS-C control communication, a minimum number of SCD cycles with exclusively CDM = 0 must be sent before starting a new control communication frame. For a manual control communication by software, CDMTIMEOUT bit indicates that ≥ 14 SCD cycles with exclusively CDM = 0 have already been sent. 0: Timeout of BiSS-C control communication not reached 1: Timeout of BiSS-C control communication reached	R

Note 1. This controller can abort register communication after tbusy_r (as defined in BiSS-C Protocol Description).
Note 2. After reading sensor data, it is recommended to reset SVALID flags in order to recognize updated sensor data.
Note 3. There are two RAM banks for SVALID, CDSSEL, and STINF2.CDS1 that switch simultaneously to SCDATA RAM banks.

The status information is combined in a set of registers. It indicates all device and communication states of the master.

50.3.10 INSTREG : Instruction Register

Base address: BISSm = 0x9020_3000 + 0x1_0000 × m (m = 0 to 11)
BISSi = 0x9030_3000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0xF4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	MAVS	MAFS	—	—	—	—	BREAK	HOLD BANK	SWBANK	INIT	INSTR[2:0]			AGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AGS	Automatic Get Sensor Data (AGS) The automatic data transmission is enabled with AGS, and the frame rate is set with MSTRCNF.FREQAGS bits. For details, see section 50.4.5. Data Acquisition. FREQAGS = AGSMIN: Start of data transmission after slave's timeout FREQAGS = AGSINFINITE: Start of data transmission triggered by GETSENS FREQAGS = Other: Start of data transmission equally spaced at a constant rate 0: No automatic data transmission 1: Automatic data transmission	R/W
3:1	INSTR[2:0]	Single-Cycle Data (SCD) Control Instruction INSTR can be used to trigger data transmission by program. For details, see section 50.4.5. Data Acquisition. When AGS = 0: 0 0 0: Reset state. No INSTR performed. 0 0 1: Start one frame with CDM = 1. INSTR automatically resets. 0 1 0: Start one frame with CDM = 0. INSTR automatically resets. 0 1 1: Start one frame with CDM = not(CDSSEL). INSTR automatically resets. 1 0 0: Start one frame with control communication. INSTR automatically resets. 1 1 1: Start one reduced frame with control communication. INSTR automatically resets. When AGS = 1: 0 0 1: Upcoming frames with CDM = 1. 0 1 0: Upcoming frames with CDM = 0. 0 1 1: Upcoming frames with CDM = not(CDSSEL). 1 0 0: Upcoming frames with control communication. INSTR will be reset at the end of the control communication. 1 1 1: Upcoming reduced frames with control communication. INSTR will be reset at the end of the control communication.	R/W
4	INIT	Start initialization sequence*1 Typically, the BiSS data line SL is set to digital high by the last slave after timeout. After power on or after a failure, the last slave may not be defined, and the SL line is digital low. An initialization sequence started by INIT can be used to initialize the slave chain. Additionally, an INIT sets the status bits nSCDERR, nREGERR, nDELAYERR and nAGSERR in STINF register to 1 and resets REGEND to 0. CTRLCOMM.REGVERS is used to select the protocol type. 0: No operation 1: Initialize (REGVERS = 1: BiSS-C)	R/W
5	SWBANK	RAM Bank Switching By inverting SWBANK value a switch of the RAM banks is forced. This can be used, for instance, to initialize the content of both RAM banks. 0: No operation when Previous value = 0. RAM banks are switched when Previous value = 1. 1: RAM banks are switched when Previous value = 0. No operation when Previous value = 1.	R/W
6	HOLDBANK	RAM Bank Control When reading more than one Data RAM registers by the controller, the RAM banks in the master could be switched once a sensor data transmission via BiSS is completed. To avoid unexpected bank switching, the controller can set HOLDBANK before reading Data RAM registers and releases it afterwards. If bank switching was prevented with HOLDBANK = 1, the bank is switched after resetting HOLDBANK = 0. If a new BiSS frame is scheduled before HOLDBANK is reset (e.g., when using INSTREG.AGS with a high MSTRCNF.FREQAGS), no bank switching is performed but STINF.nAGSERR will be set. 0: Automatic bank switching is permitted 1: Automatic bank switching is prevented	R/W
7	BREAK	Start BREAK Sequence*2 All current actions can be aborted using the BREAK command so that the controller enters a defined state if one of the sensors does not respond correctly. 0: No operation 1: Abort data transmission	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	MAFS	Control of the MA line 0: MA line is not controlled 1: MA line is forced with MAVS	R/W

Bit	Symbol	Function	R/W
13	MAVS	MA line force level 0: Low 1: High	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. With REGVERS = 1 (BiSS-C) and an INIT, the master stores the measured line delay in SCDATA1[7:0] register. The unit of these values is 1/4 of the configured single-cycle data clock frequency configured with CTRLCOMM.FREQS.
tLine Delay = SCDATA1[7:0]/(4 × fMA)
For the INIT sequence, the maximum line delay is 255. If this limit is exceeded, the INIT sequence is aborted and STINF.nAGSERR is activated (nAGSERR = 0). The INIT sequence does not test the SL state. A constant SL = 0 state is ignored and does not set any error state.

Note 2. An INIT should be executed after BREAK to ensure resetting status bits.

50.3.11 STINF2 : Status Information 2

Base address: BISSm = 0x9020_3000 + 0x1_0000 × m (m = 0 to 11)
BISSi = 0x9030_3000 + 0x1_0000 × (i - 12) (i = 12 to 15)

Offset address: 0xF8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	SWBANKFAIL	—	—	—	—	IDL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CDS1	SL1
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
0	SL1	SL input line state The actual state of the BiSS data SL input. 0: SL line level low 1: SL line level high	R
1	CDS1	Control Data Slave (CDS) bit value This bit has the same contents as STINF.CDSSEL. 0: CDS value 0 1: CDS value 1	R
15:2	—	The read values are undefined.	R
19:16	IDL[3:0]	Number of set ID lock bits during control communication*1 At the beginning of each control communication, slave IDs are automatically assigned to the connected slaves by considering the physical order within the daisy chain. IDL indicates the total number of IDL bits that have been set by the BiSS slaves. It corresponds to the number of slaves that take part in control communication. IDL = 9 indicates that there are more than eight BiSS slaves connected that can take part in control communication. Since only 8 BiSS slaves are addressable via CTRLCOMM.SLAVEID, the first slave(s) cannot be addressed unless other slaves have been disabled via predefined BiSS commands as defined in BiSS-C Protocol Description. 0: No IDL bits received 1 to 8: 1 to 8 IDL bits (corresponds to 1…8 slaves) 9: There are more than 8 slaves available. Note, only the first 8 slave IDs are addressable with CTRLCOMM.SLAVEID.	R
23:20	—	The read values are undefined.	R
24	SWBANKFAIL	Bank switching status Switching SCDATA RAM bank—which is executed at the end of a BiSS frame—also switches the corresponding STINF.SVALID flags. It fails if it disables bank switching via INSTREG.HOLDBANK. 0: Bank switching (SCDATA) successful 1: Bank switching (SCDATA) not successful	R

Bit	Symbol	Function	R/W
31:25	—	These bits are read as 0.	R

Note 1. A control communication must be performed before the correct number of slaves is indicated with IDL.

50.4 Operation

50.4.1 BiSS Function

In point-to-point configuration, BiSS devices including one or several sensors are connected to the clock line MA and the data return line SL. RS422 transceivers are available on-chip for robust differential communication in industrial environments. Two types of BiSS communication are supported:

- Single-Cycle Data (SCD):

  Sensor data with up to 64 bits for each slave (e.g., measurement data, error, and warning flag, Sign of Life counter, …).

- Control Communication:

  E.g., register communication for reading/writing the sensor's registers (e.g., for reading an Electronic Datasheet or calibrating sensors).

This controller provides two RAM memory banks for each slave enabling simultaneous access by CPU while new sensor data is being transmitted via BiSS interface. A 64-byte memory supports register transfers. Sensor data acquisition is started by a software instruction, via GETSENS signal or automatically at a configurable frame repetition rate. The end of the sensor data transmission is indicated at BISSn_EOT (EOT: n = 0 to 15) interrupt. An error during transmission is flagged by BISSn_NER (NER: n = 0 to 15) interrupt. Error details can be accessed in the status registers. This controller generates a clock signal for sensor communication using 20 MHz system clock.

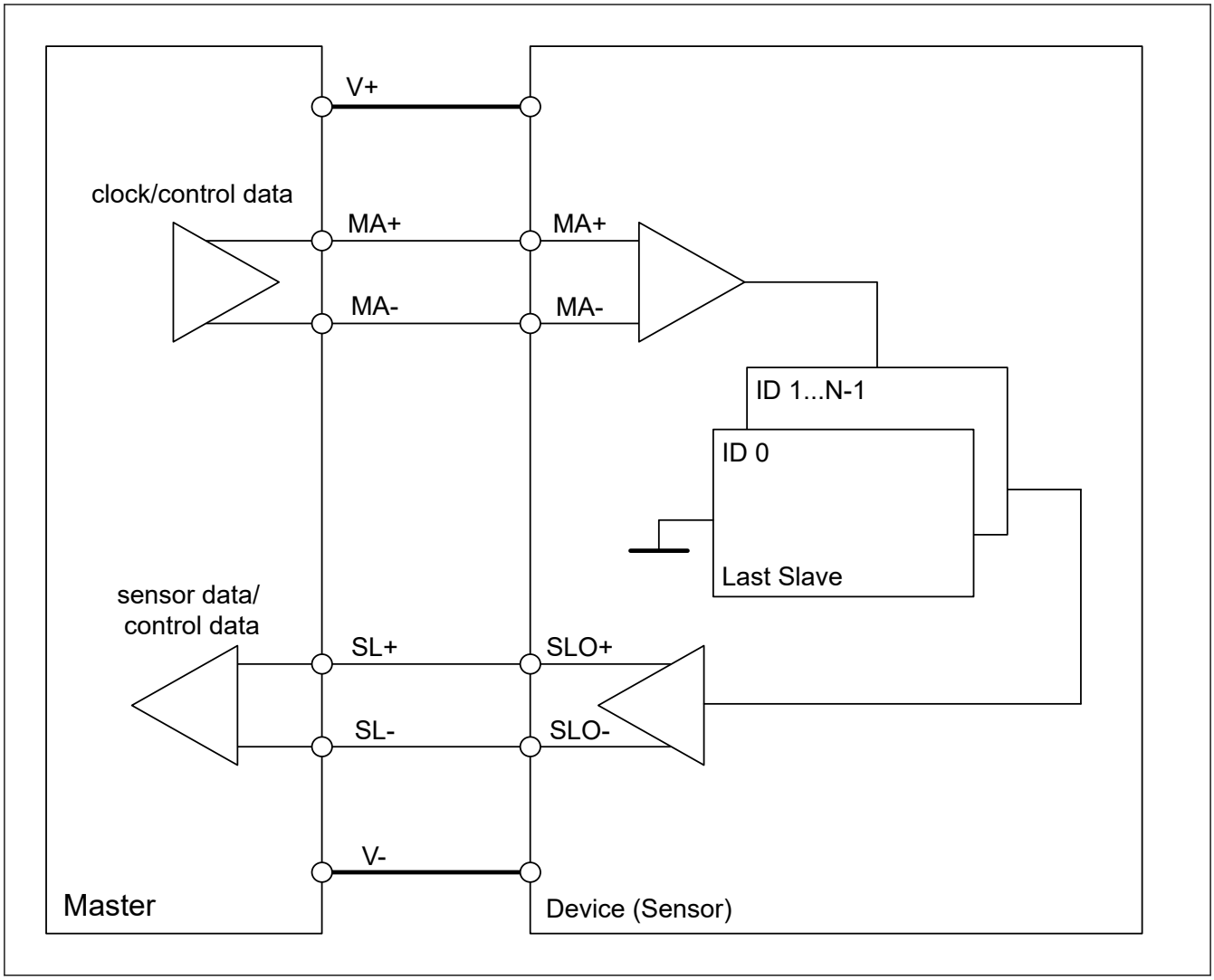


Figure 50.2 Point-to-point connection to device with several slaves

The open-source BiSS Interface Protocol implements a real-time interface for digital, serial, and secure communication between drive, sensor and actuator. In the point-to-point configuration, BiSS uses one clock line MA from the master to the sensor and one data line SL from the sensor to the master. A device may contain multiple slaves. The data input signal SL of the last slave is set to digital low. The slaves are daisy-chained and the data output signal SLO of the first slave is directly connected to the master. The BiSS protocol describes cyclic data frames and differentiates between process data and control data. Process data are completely transmitted in each frame (SCD) used as sensor data, and control data are transmitted one bit per frame (CD) used for commands and register access.

50.4.2 BiSS-C

(1) BiSS-C Frame

The BiSS-C Frame starts with a digital high on the clock line MA and is notified with a falling edge at EOT signal. On the first falling edge of MA, all slaves check the SL signal for a digital low determining the last slave. With the first rising edge at MA, all sensors start calculating their sensor data. The second rising edge of MA forces all slaves to acknowledge the BiSS-C frame with a falling edge at SL. The master uses the acknowledge to measure the line delay. When the sensor data calculation is finished, the last slave in the chain generates the start bit which will be passed synchronously through all slaves to the master. Subsequent to the start bit follows one control data bit (CDS) for all slaves which is set according to the rules of the control frame. After the CDS bit, the process data including sensor data is sent with the most significant bit (MSB) first. At the end, the master sends its control data bit (CDM) inverted on the clock line MA to conclude the BiSS-C Frame.

During processing the frame, all slaves observe the MA clock line and change into the timeout state if MA is stable for a specific time defined within each slave. In the timeout state, only the last slave forces its SL signal to digital high. The other

slaves in the chain indicate to the master that all slaves are in the timeout state. After detecting the slave's timeout with SL = 1, the master may change the MA clock line to digital high or keeps the clock line constant until the next frame begins.

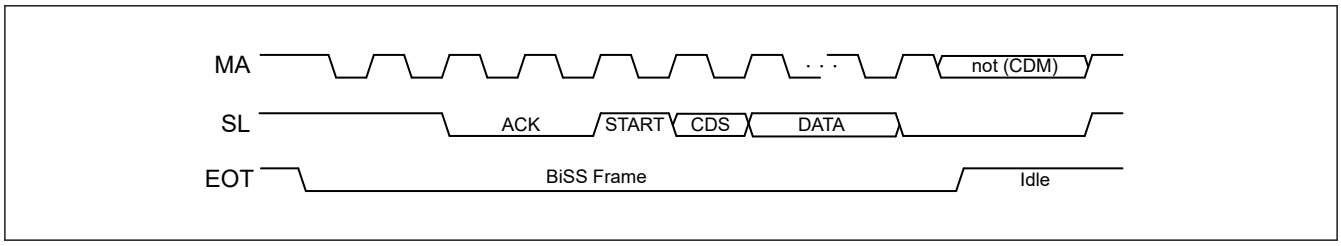


Figure 50.3 BiSS-C frame in point-to-point configuration

(2) BiSS-C Register Communication

The master is able to transmit a BiSS-C register write access to the slave without the slave's CDS feedback. The master cannot verify that the BiSS-C register write access to the slave did succeed. At the end of the SCD frame, the master sends the CDM bit inverted on the MA clock line.

(3) BiSS-C Initialization Sequence

An initialization (INIT) sequence is necessary if the last slave in the point-to-point configuration is not defined and the data line SL is digital low. In the initialization sequence, two digital low pulses are generated at MA. The slave(s) should start communication at the second falling edge of the MA pulse and answer the end of communications by generating a rising edge on SL after the BiSS timeout detection. The time between the second rising-edge at MA and the falling-edge at SL is measured as line delay and stored in the single-cycle data RAM, see INSTREG.INIT bit function.

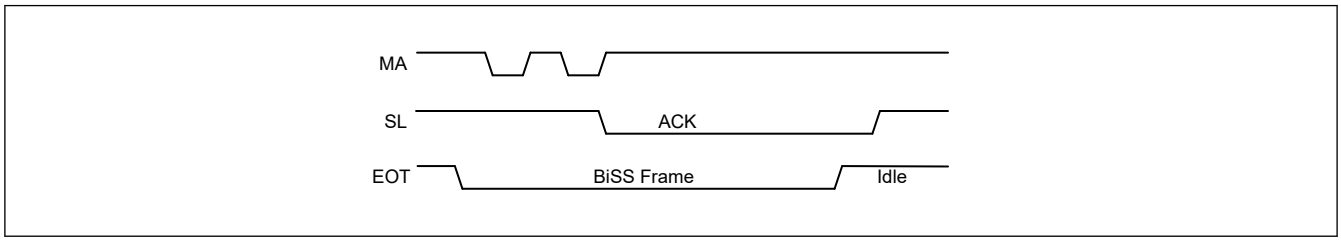


Figure 50.4 BiSS-C initialization sequence

50.4.3 Sensor Data

For sensor data, an 8x8 byte single-cycle data memory SCDATA is provided. Eight slaves with up to 64 bits of single-cycle data each are supported.

Table 50.6 Address mapping of single-cycle data (SCDATA)

Offset address	Register	Content
0x00 + 0x08 × (n - 1)	SCDATALn[31:0]	SCDATAn[31:0]
0x04 + 0x08 × (n - 1)	SCDATAHn[31:0]	SCDATAn[63:32]

Note: Slave: n = 1 to 8

The sensor data is arranged in the memory area with the least significant bit (LSB) at the lower address at bit position 0. The memory is written byte-by-byte and unused bits are set to zero. Unused bytes remain unchanged. In BiSS the CRC bits are inverted before transmission. The received CRC bits are reinverted and then stored at the most significant SCDATA byte(s). This can be disabled with CTRLCOMM.NOCRC = 1.

- Example: BiSS Sensor Bus with 3 Slaves
  The following example shows the address mapping of the sensor data and CRC bits generated by three slaves.
  - Slave 1: 19 + 2 bits of sensor data, 6 bits of CRC: Total length of 27 bits
  - Slave 2: 12 + 2 bits of sensor data, 5 bits of CRC: Total length of 19 bits
  - Slave 3: 24 bits of sensor data, 16 bits of CRC: Total length of 40 bits

Table 50.7 Example address mapping of sensor data

Offset address	SCDATA[31:0]
SCDATAL1[31:0]	Not changed [7:0], 000b, Sensor data 1 [20:0]
SCDATAH1[31:0]	00b, CRC 1 [5:0], Not changed [23:0]
SCDATAL2[31:0]	Not changed [15:0], 00b, Sensor data 2 [13:0]
SCDATAH2[31:0]	000b, CRC 2 [4:0], Not changed [23:0]
SCDATAL3[31:0]	Not changed [7:0], Sensor data 3 [23:0]
SCDATAH3[31:0]	CRC 3 [15:0], Not changed [15:0]
SCDATALn[31:0] (n = 4 to 8)	Not changed [31:0]
SCDATAHn[31:0] (n = 4 to 8)	Not changed [31:0]

In order to import new sensor data during access, this controller has two memory banks for sensor data. While sensor data is being read and written into the first RAM bank during processing of the BiSS Frame, the second RAM bank section containing sensor data of the previous frame can be read. The relevant sensor data memory banks are swapped at the end of the BiSS Frame. This can be prevented by setting INSTREG.HOLDBANK. Simultaneously the validity (STINF.SVALID) that corresponds to the RAM bank is also swapped.

As an example, the basic procedure of sensor data is below:

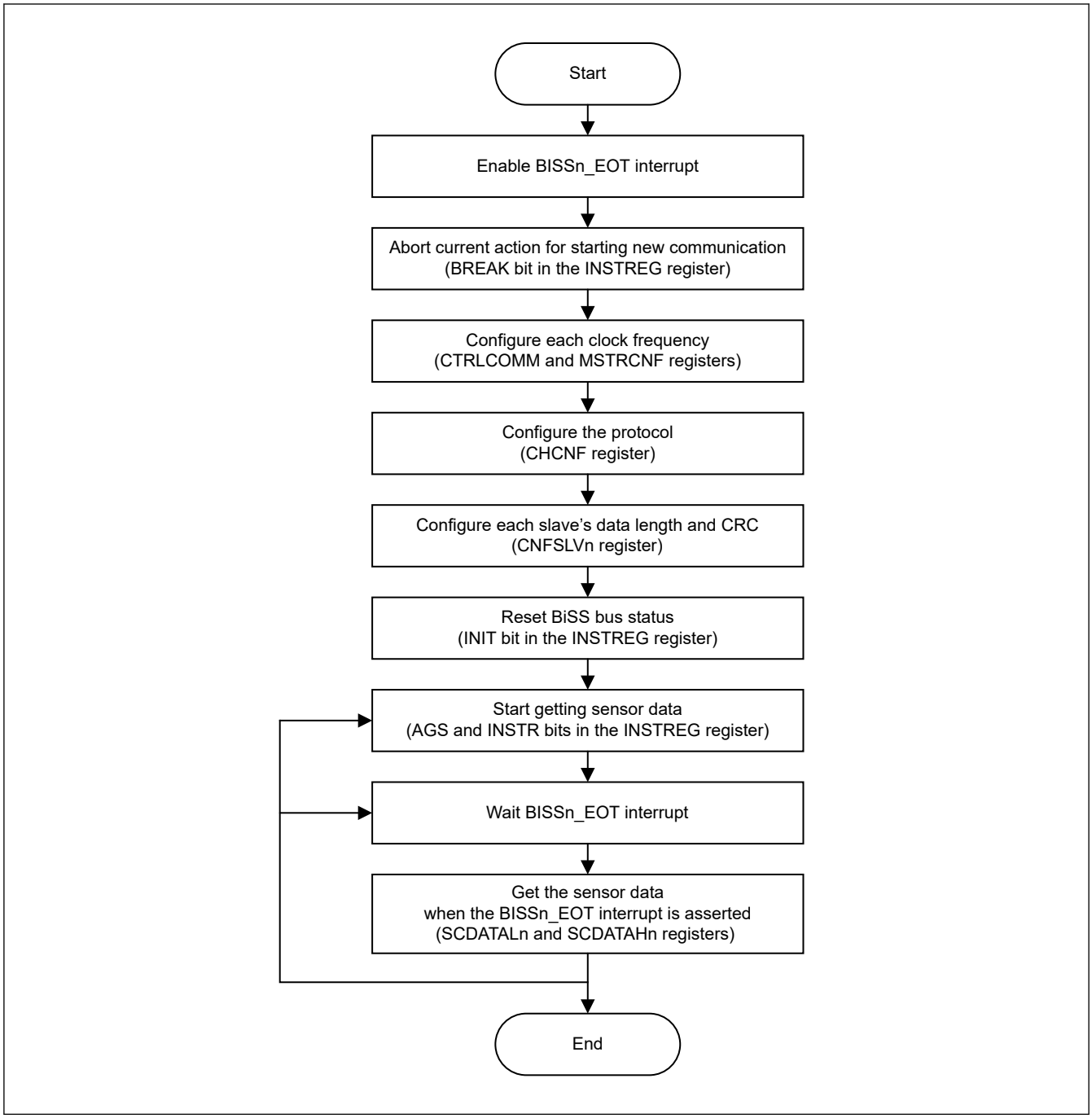


Figure 50.5 Basic operation of sensor data

50.4.4 Control Frame

50.4.4.1 BiSS Commands

The BiSS-C Protocol Description defines BiSS commands as a subset of the control communication. A BiSS command can address one or several slaves by setting the appropriate bits in IDS. To address all slaves via broadcast command, IDS has to be reset to zero. IDS[7:0] bits correspond to the lower 8 bits of Register Data 1 (RDATA1) register.

Slave addressing for BiSS commands:

- IDS[7:0] = 0x00: All slaves addressed (broadcast)
- IDS[N]:
  0: Slave with ID (7 – N) is not addressed (e.g., IDS[0]: ID 7, IDS[1]: ID 6, IDS[7]: ID 0)
  1: Slave with ID (7 – N) is addressed

CMD (corresponding to CTRLCOMM.SID_CMD_IDT[2:1] bits) determines which BiSS command is sent.

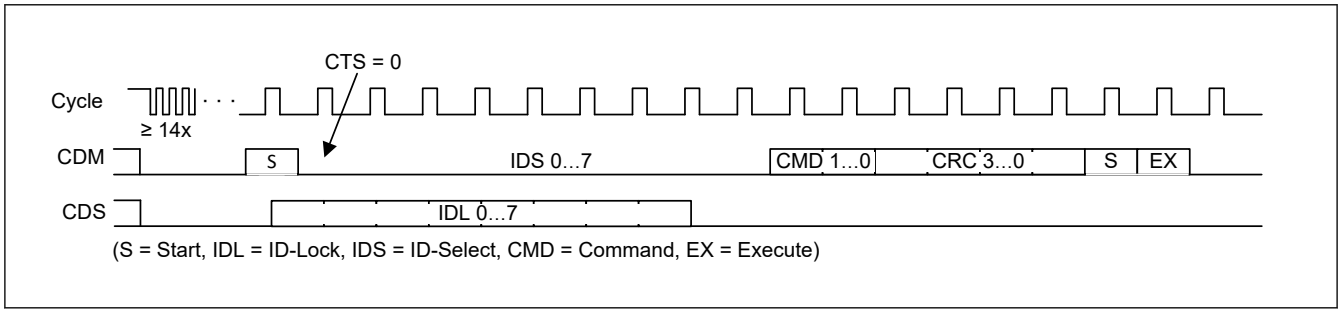


Figure 50.6 Control frame for broadcast BiSS command

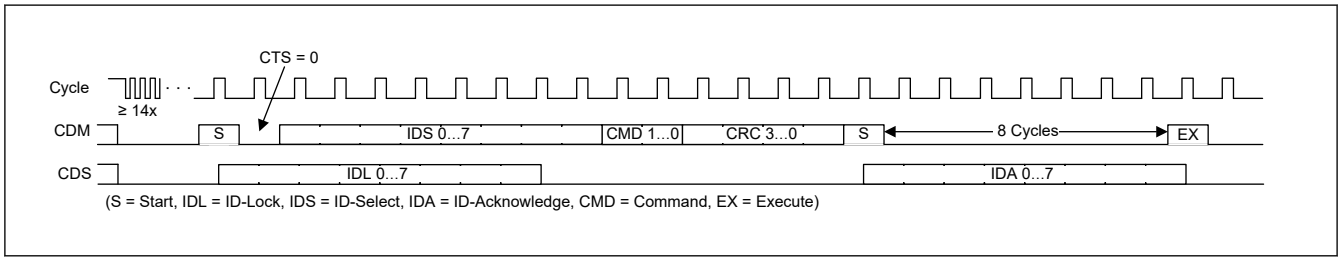


Figure 50.7 Control frame for addressed BiSS command

50.4.4.2 Register Access

The register start address (REGADR), register access direction (WNR), number of bytes (REGNUM) in REGACC register, and slave address (SLAVEID) in CTRLCOMM register are used to configure the BiSS-C control communication for register communication. If a byte count of 0 is entered for REGNUM, it means the transmission of a single register value.

Note: All configuration parameters should be stable during the control communication frame.

The following figure exemplarily shows a control frame including a register read access.

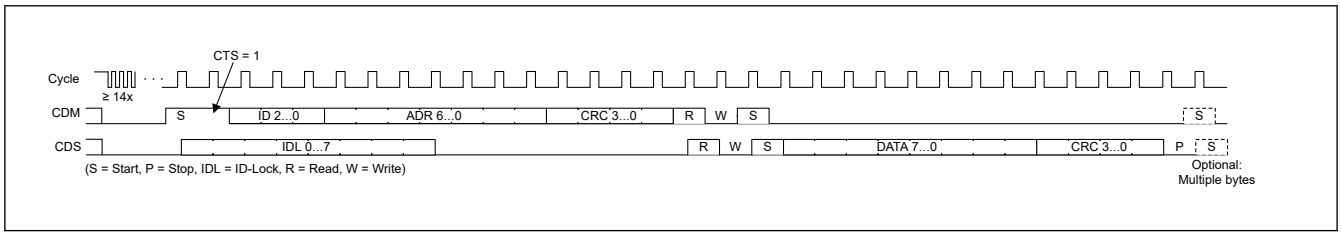


Figure 50.8 Control frame for register communication (register read access)

50.4.5 Data Acquisition

Trigger data transmissions:

- Automatically (MSTRCNF.FREQAGS ≠ AGSINFINITE)
- By GETSENS signal (MSTRCNF.FREQAGS = AGSINFINITE)
- By selecting a program with INSTREG.INSTR

(1) Automatic Sensor Data Request

With INSTREG.AGS = 1 and MSTRCNF.FREQAGS = AGSMIN, data transmissions are started automatically as soon as the slave's timeout is detected. With FREQAGS configured to a constant frame repetition rate, data transmissions are started automatically at the configured frame repetition rate.

(2) Sensor Data Request by GETSENS Signal

GETSENS signal can be used to start a new BiSS frame if enabled by INSTREG.AGS = 1 and MSTRCNF.FREQAGS = AGSINFINITE. The delay between the trigger event at GETSENS and the falling edge at MA is defined by the system clock fCLK and the single-cycle clock frequency fMA (configured with CTRLCOMM.FREQS).

tGETSENS to falling edge at MA = 2/fCLK + 1/(2 × fMA)

(3) Program Selection with Instructions

INSTREG.INSTR is used to trigger data transmissions by program. With INSTREG.AGS = 0, the master starts the data transmission after finishing writing INSTREG register and resets INSTR after executing the frame automatically. STINF.nAGSERR error will be generated if the SL line is low at the start of the frame. The error can be suppressed by setting MSTRCNF.FREQAGS = AGSMIN due to waiting for a high state at SL. A BiSS-C control communication (refer to CTRLCOMM.REGVERS) can be triggered via INSTR as well. It is also possible to enable reduced BiSS frames for control communication with BiSS commands. Within a running control communication, the CDM bit is generated automatically. Otherwise, the generation of CDM depends on INSTR.

50.5 Usage Notes

50.5.1 The value to be set to SCDLEN in the CNFSVn register

For the data length set in SCDLEN, the following sum is required.

MT (multi-turn) bits + ST (single-turn) bits + 2 bits (Error bit and Warning bit between data and CRC in SCD frame)

50.5.2 The storage of received CRC when SCDLEN = 33 bits or more

The CRC of SCD is normally stored at the end of SCDATA, but if SCDLEN is set to 33 bits or more, it is not stored in SCDATA. However, CRC check is executed every time by HW, the result is informed by the status flag (nSCDERR).

When need to read the CRC data, it is required to set the data length including CRC to SCDLEN (ex: set 49 (43 + 6) bits when data = 43 bits and CRC = 6 bits). In this case, the CRC check is required to be executed by SW.

51. HIPERFACE DSL (HDSL)

51.1 Overview

The MPU includes HIPERFACE DSL (HDSL) master to enable communication between the encoder and HIPERFACE DSL interface. Table 51.1 describes the HDSL specifications and Figure 51.1 shows a block diagram of the HDSL.

Table 51.1 HIPERFACE DSL (HDSL) specifications

Item	Description
Number of units	16 units (12 in LLPP0 and 4 in LLPP1)
Protocol	HIPERFACE DSL
Transfer rate	Up to 9.375 Mbaud
Functions	• HIPERFACE DSL with safety • Safe channel 1 interface is selectable from internal bus access or external MCU access through SPI. This safe channel 1 interface enables safety application and it can be implemented in this MPU together with normal application. • Safe channel 2 interface is selectable from internal bus access or external MCU access through SPI • SensorHub SPI PIPE interface is not supported • System diagnostics function • Fast position function – Estimator – Free running mode – SYNC mode • Status and error messages – Event register – Online status – Status summary of the motor feedback system – Motor feedback system error messages • Safety implementation
Interrupt sources	• Configurable interrupt: 4 – Normal related – Safety related – Fast position received – Safety position received • Error related interrupt: 7 – Position estimator activated – Transmission error in safe channel 1 – Safe position not valid – Fast channel/position error – Fast channel/position threshold error – DSL message encoding error – Estimator deviation threshold reached
Event link	SYNC input for HW trigger provided from ELC
Module-stop function	Module-stop state can be set to reduce power consumption

This HIPERFACE DSL® MASTER is licensed from SICK AG. For details, see the following SICK document.

- HIPERFACE DSL® MASTER Safety Integration Manual

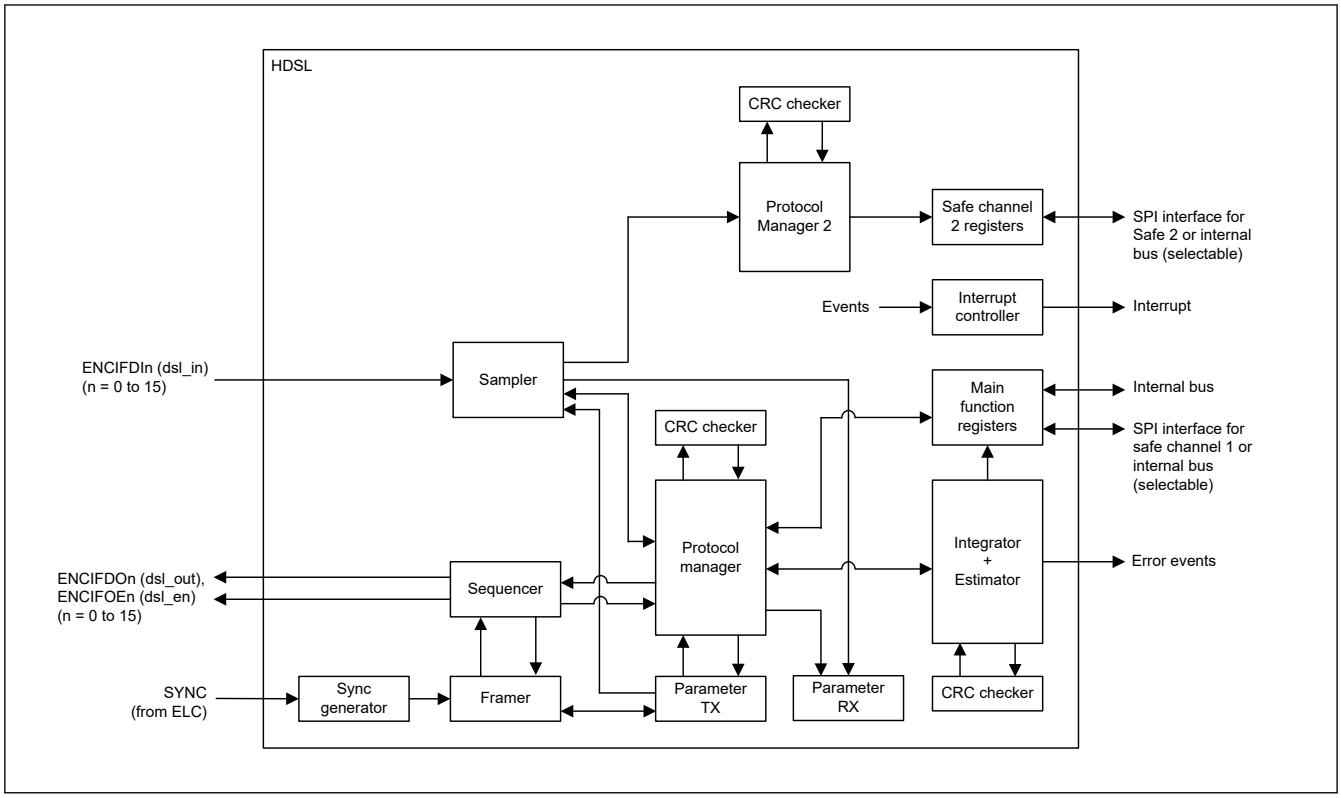


Figure 51.1 Block diagram of HIPERFACE DSL (HDSL)

Table 51.2 lists the input/output pins of the HIPERFACE DSL.

Table 51.2 Pin configuration of the HDSL

Unit	Pin name	I/O	Function
HDSLn	ENCIFOEn (dsl_en)	Output	HDSL unit n DSL cable transceiver, activation
	ENCIFDOn (dsl_out)	Output	HDSL unit n DSL cable, output data
	ENCIFDIn (dsl_in)	Input	HDSL unit n DSL cable, input data
	HDSLn_LINK	Output	HDSL unit n LINK
	HDSLn_SMPL	Output	HDSL unit n Test signal line sampler
	HDSLn_CLK1	Input	HDSL unit n SPI clock safe channel 1
	HDSLn_SEL1	Input	HDSL unit n SPI selection safe channel 1
	HDSLn_MISO1	Output	HDSL unit n data output safe channel 1
	HDSLn_MOSI1	Input	HDSL unit n data input safe channel 1
	HDSLn_CLK2	Input	HDSL unit n SPI clock safe channel 2
	HDSLn_SEL2	Input	HDSL unit n SPI selection safe channel 2
	HDSLn_MISO2	Output	HDSL unit n data output safe channel 2
	HDSLn_MOSI2	Input	HDSL unit n data input safe channel 2

Note: n = 0 to 15

Table 51.3 HDSL interrupt sources (1 of 2)

Name	Interrupt sources
HDSLn INT	Interrupt output
HDSLn INTS	Interrupt output (safety related)
HDSLn_FP	Fast position received interrupt
HDSLn_SP	Safety position received interrupt

Table 51.3 HDSL interrupt sources (2 of 2)

Name	Interrupt sources
HDSLn estimator on	Position estimator activated
HDSLn safe channel error	Transmission error in safe channel 1
HDSLn safe position error	Safe position not valid
HDSLn acceleration error	Fast channel/position error
HDSLn acceleration threshold error	Fast channel/position threshold
HDSLn encoding error	DSL message encoding error
HDSLn deviation threshold error	Estimator deviation threshold reached

Note: n = 0 to 15

51.2 Register Map

Table 51.4 HDSL Drive interface register map (1 of 3)

Address	Register symbol	Register name	Write protection
0x9020_0000 + 0x1_0000 × m (m = 0 to 11) 0x9030_0000 + 0x1_0000 × (m - 12) (m = 12 to 15)	SYS_CTRL	System Control Register	—
0x9020_0001 + 0x1_0000 × m (m = 0 to 11) 0x9030_0001 + 0x1_0000 × (m - 12) (m = 12 to 15)	SYNC_CTRL	Synchronization Control Register	—
0x9020_0003 + 0x1_0000 × m (m = 0 to 11) 0x9030_0003 + 0x1_0000 × (m - 12) (m = 12 to 15)	MASTER_QM	Quality Monitoring Register	—
0x9020_0004 + 0x1_0000 × m (m = 0 to 11) 0x9030_0004 + 0x1_0000 × (m - 12) (m = 12 to 15)	EVENT_H	High Byte Event Register	—
0x9020_0005 + 0x1_0000 × m (m = 0 to 11) 0x9030_0005 + 0x1_0000 × (m - 12) (m = 12 to 15)	EVENT_L	Low Byte Event Register	—
0x9020_0006 + 0x1_0000 × m (m = 0 to 11) 0x9030_0006 + 0x1_0000 × (m - 12) (m = 12 to 15)	MASK_H	High Byte Event Mask Register	—
0x9020_0007 + 0x1_0000 × m (m = 0 to 11) 0x9030_0007 + 0x1_0000 × (m - 12) (m = 12 to 15)	MASK_L	Low Byte Event Mask Register	—
0x9020_0008 + 0x1_0000 × m (m = 0 to 11) 0x9030_0008 + 0x1_0000 × (m - 12) (m = 12 to 15)	MASK_SUM	Summary Mask Register	—
0x9020_0009 + 0x1_0000 × m (m = 0 to 11) 0x9030_0009 + 0x1_0000 × (m - 12) (m = 12 to 15)	EDGES	Edge Register	—
0x9020_000A + 0x1_0000 × m (m = 0 to 11) 0x9030_000A + 0x1_0000 × (m - 12) (m = 12 to 15)	DELAY	Run Time Delay Register	—
0x9020_000B + 0x1_0000 × m (m = 0 to 11) 0x9030_000B + 0x1_0000 × (m - 12) (m = 12 to 15)	VERSION	Version Register	—
0x9020_000D + 0x1_0000 × m (m = 0 to 11) 0x9030_000D + 0x1_0000 × (m - 12) (m = 12 to 15)	ENC_ID2	Encoder ID 2 Register	—
0x9020_000E + 0x1_0000 × m (m = 0 to 11) 0x9030_000E + 0x1_0000 × (m - 12) (m = 12 to 15)	ENC_ID1	Encoder ID 1 Register	—
0x9020_000F + 0x1_0000 × m (m = 0 to 11) 0x9030_000F + 0x1_0000 × (m - 12) (m = 12 to 15)	ENC_ID0	Encoder ID 0 Register	—
0x9020_0010 + 0x1_0000 × m (m = 0 to 11) 0x9030_0010 + 0x1_0000 × (m - 12) (m = 12 to 15)	POS4	Fast Position Byte 4 Register	—
0x9020_0011 + 0x1_0000 × m (m = 0 to 11) 0x9030_0011 + 0x1_0000 × (m - 12) (m = 12 to 15)	POS3	Fast Position Byte 3 Register	—
0x9020_0012 + 0x1_0000 × m (m = 0 to 11) 0x9030_0012 + 0x1_0000 × (m - 12) (m = 12 to 15)	POS2	Fast Position Byte 2 Register	—
0x9020_0013 + 0x1_0000 × m (m = 0 to 11) 0x9030_0013 + 0x1_0000 × (m - 12) (m = 12 to 15)	POS1	Fast Position Byte 1 Register	—

Table 51.4 HDSL Drive interface register map (2 of 3)

Address	Register symbol	Register name	Write protection
0x9020_0014 + 0x1_0000 × m (m = 0 to 11) 0x9030_0014 + 0x1_0000 × (m - 12) (m = 12 to 15)	POS0	Fast Position Byte 0 Register	—
0x9020_0015 + 0x1_0000 × m (m = 0 to 11) 0x9030_0015 + 0x1_0000 × (m - 12) (m = 12 to 15)	VEL2	Speed Byte 2 Register	—
0x9020_0016 + 0x1_0000 × m (m = 0 to 11) 0x9030_0016 + 0x1_0000 × (m - 12) (m = 12 to 15)	VEL1	Speed Byte 1 Register	—
0x9020_0017 + 0x1_0000 × m (m = 0 to 11) 0x9030_0017 + 0x1_0000 × (m - 12) (m = 12 to 15)	VEL0	Speed Byte 0 Register	—
0x9020_0018 + 0x1_0000 × m (m = 0 to 11) 0x9030_0018 + 0x1_0000 × (m - 12) (m = 12 to 15)	SUMMARY	Mirror Status Summary Register	—
0x9020_0020 + 0x1_0000 × m (m = 0 to 11) 0x9030_0020 + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_BUFFER0	Parameter Channel Buffer 0 Register	—
0x9020_0021 + 0x1_0000 × m (m = 0 to 11) 0x9030_0021 + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_BUFFER1	Parameter Channel Buffer 1 Register	—
0x9020_0022 + 0x1_0000 × m (m = 0 to 11) 0x9030_0022 + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_BUFFER2	Parameter Channel Buffer 2 Register	—
0x9020_0023 + 0x1_0000 × m (m = 0 to 11) 0x9030_0023 + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_BUFFER3	Parameter Channel Buffer 3 Register	—
0x9020_0024 + 0x1_0000 × m (m = 0 to 11) 0x9030_0024 + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_BUFFER4	Parameter Channel Buffer 4 Register	—
0x9020_0025 + 0x1_0000 × m (m = 0 to 11) 0x9030_0025 + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_BUFFER5	Parameter Channel Buffer 5 Register	—
0x9020_0026 + 0x1_0000 × m (m = 0 to 11) 0x9030_0026 + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_BUFFER6	Parameter Channel Buffer 6 Register	—
0x9020_0027 + 0x1_0000 × m (m = 0 to 11) 0x9030_0027 + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_BUFFER7	Parameter Channel Buffer 7 Register	—
0x9020_0028 + 0x1_0000 × m (m = 0 to 11) 0x9030_0028 + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_ADD_H	High Byte Long Message Address Register	—
0x9020_0029 + 0x1_0000 × m (m = 0 to 11) 0x9030_0029 + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_ADD_L	Low Byte Long Message Address Register	—
0x9020_002A + 0x1_0000 × m (m = 0 to 11) 0x9030_002A + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_OFF_H	High Byte Long Message Address Offset Register	—
0x9020_002B + 0x1_0000 × m (m = 0 to 11) 0x9030_002B + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_OFF_L	Low Byte Long Message Address Offset Register	—
0x9020_002C + 0x1_0000 × m (m = 0 to 11) 0x9030_002C + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_CTRL	Parameter Channel Control Register	—
0x9020_002D + 0x1_0000 × m (m = 0 to 11) 0x9030_002D + 0x1_0000 × (m - 12) (m = 12 to 15)	PIPE_S	SensorHub Channel Status Register	—
0x9020_002E + 0x1_0000 × m (m = 0 to 11) 0x9030_002E + 0x1_0000 × (m - 12) (m = 12 to 15)	PIPE_D	SensorHub Channel Data Register	—
0x9020_002F + 0x1_0000 × m (m = 0 to 11) 0x9030_002F + 0x1_0000 × (m - 12) (m = 12 to 15)	PC_DATA	Parameter Channel Short Message Mirror Register	—
0x9020_0038 + 0x1_0000 × m (m = 0 to 11) 0x9030_0038 + 0x1_0000 × (m - 12) (m = 12 to 15)	ACC_EE_CNT	Fast Position Error Counter Register	—
0x9020_0039 + 0x1_0000 × m (m = 0 to 11) 0x9030_0039 + 0x1_0000 × (m - 12) (m = 12 to 15)	MAXACC	Fast Position Acceleration Boundary Register	—
0x9020_003A + 0x1_0000 × m (m = 0 to 11) 0x9030_003A + 0x1_0000 × (m - 12) (m = 12 to 15)	MAXDEV_H	Fast Position Estimator Deviation High Byte Register	—
0x9020_003B + 0x1_0000 × m (m = 0 to 11) 0x9030_003B + 0x1_0000 × (m - 12) (m = 12 to 15)	MAXDEV_L	Fast Position Estimator Deviation Low Byte Register	—

Table 51.4 HDSL Drive interface register map (3 of 3)

Address	Register symbol	Register name	Write protection
0x9020_003F + 0x1_0000 × m (m = 0 to 11) 0x9030_003F + 0x1_0000 × (m - 12) (m = 12 to 15)	DUMMY	Dummy Register	—
0x9020_0060 + 0x1 × n + 0x1_0000 × m (m = 0 to 11) 0x9030_0060 + 0x1 × n + 0x1_0000 × (m - 12) (m = 12 to 15)	MIR_STn	Mirror Status n Register (n = 0 to 7)	—

Table 51.5 HDSL Safe 1 interface register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x9020_1009 + 0x1_0000 × m (m = 0 to 11) 0x9030_1009 + 0x1_0000 × (m - 12) (m = 12 to 15)	EDGES*1	Edge Register	—
0x9020_100A + 0x1_0000 × m (m = 0 to 11) 0x9030_100A + 0x1_0000 × (m - 12) (m = 12 to 15)	DELAY*1	Run Time Delay Register	—
0x9020_100B + 0x1_0000 × m (m = 0 to 11) 0x9030_100B + 0x1_0000 × (m - 12) (m = 12 to 15)	VERSION*1	Version Register	—
0x9020_100D + 0x1_0000 × m (m = 0 to 11) 0x9030_100D + 0x1_0000 × (m - 12) (m = 12 to 15)	ENC_ID2*1	Encoder ID 2 Register	—
0x9020_100E + 0x1_0000 × m (m = 0 to 11) 0x9030_100E + 0x1_0000 × (m - 12) (m = 12 to 15)	ENC_ID1*1	Encoder ID 1 Register	—
0x9020_100F + 0x1_0000 × m (m = 0 to 11) 0x9030_100F + 0x1_0000 × (m - 12) (m = 12 to 15)	ENC_ID0*1	Encoder ID 0 Register	—
0x9020_1019 + 0x1_0000 × m (m = 0 to 11) 0x9030_1019 + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOS4	Safe Position 4 Register	—
0x9020_101A + 0x1_0000 × m (m = 0 to 11) 0x9030_101A + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOS3	Safe Position 3 Register	—
0x9020_101B + 0x1_0000 × m (m = 0 to 11) 0x9030_101B + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOS2	Safe Position 2 Register	—
0x9020_101C + 0x1_0000 × m (m = 0 to 11) 0x9030_101C + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOS1	Safe Position 1 Register	—
0x9020_101D + 0x1_0000 × m (m = 0 to 11) 0x9030_101D + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOS0	Safe Position 0 Register	—
0x9020_101E + 0x1_0000 × m (m = 0 to 11) 0x9030_101E + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOSCRC_H	Safe Position CRC High Byte Register	—
0x9020_101F + 0x1_0000 × m (m = 0 to 11) 0x9030_101F + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOSCRC_L	Safe Position CRC Low Byte Register	—
0x9020_1035 + 0x1_0000 × m (m = 0 to 11) 0x9030_1035 + 0x1_0000 × (m - 12) (m = 12 to 15)	SAFE_CTRL	Safe System Control Register	—
0x9020_1036 + 0x1_0000 × m (m = 0 to 11) 0x9030_1036 + 0x1_0000 × (m - 12) (m = 12 to 15)	SAFE_SUM	Safe Status Summary Register	—
0x9020_1037+ 0x1_0000 × m (m = 0 to 11) 0x9030_1037 + 0x1_0000 × (m - 12) (m = 12 to 15)	S_PC_DATA	Parameter Channel Short Message Register	—
0x9020_103D + 0x1_0000 × m (m = 0 to 11) 0x9030_103D + 0x1_0000 × (m - 12) (m = 12 to 15)	EVENT_S	Safe Event Register	—
0x9020_103E + 0x1_0000 × m (m = 0 to 11) 0x9030_103E + 0x1_0000 × (m - 12) (m = 12 to 15)	MASK_S	Safe Event Mask Register	—
0x9020_103F + 0x1_0000 × m (m = 0 to 11) 0x9030_103F + 0x1_0000 × (m - 12) (m = 12 to 15)	DUMMY*1	Dummy Register	—
0x9020_1040 + 0x1 × n + 0x1_0000 × m (m = 0 to 11) 0x9030_1040 + 0x1 × n + 0x1_0000 × (m - 12) (m = 12 to 15)	ENC_STn	Encoder Status n Register (n = 0 to 7)	—
0x9020_107C + 0x1_0000 × m (m = 0 to 11) 0x9030_107C + 0x1_0000 × (m - 12) (m = 12 to 15)	SRSSI	Slave RSSI Register	—

Table 51.5 HDSL Safe 1 interface register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x9020_107E + 0x1_0000 × m (m = 0 to 11) 0x9030_107E + 0x1_0000 × (m - 12) (m = 12 to 15)	MAIL	Slave Mail Register	—
0x9020_107F + 0x1_0000 × m (m = 0 to 11) 0x9030_107F + 0x1_0000 × (m - 12) (m = 12 to 15)	PING	Slave Ping Register	—

Note 1. These registers are shared with Drive Interface.

Table 51.6 HDSL Safe 2 interface register map

Address	Register symbol	Register name	Write protection
0x9020_200B + 0x1_0000 × m (m = 0 to 11) 0x9030_200B + 0x1_0000 × (m - 12) (m = 12 to 15)	VERSION2	Version in Safe Channel2 Register	—
0x9020_200F + 0x1_0000 × m (m = 0 to 11) 0x9030_200F + 0x1_0000 × (m - 12) (m = 12 to 15)	ENC2_ID	Encoder ID in Safe Channel2 Register	—
0x9020_2018 + 0x1_0000 × m (m = 0 to 11) 0x9030_2018 + 0x1_0000 × (m - 12) (m = 12 to 15)	STATUS2	Safe Channel 2 Status Register	—
0x9020_2019 + 0x1_0000 × m (m = 0 to 11) 0x9030_2019 + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOS24	Safe Position Chanel 2 Byte 4 Register	—
0x9020_201A + 0x1_0000 × m (m = 0 to 11) 0x9030_201A + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOS23	Safe Position Chanel 2 Byte 3 Register	—
0x9020_201B + 0x1_0000 × m (m = 0 to 11) 0x9030_201B + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOS22	Safe Position Chanel 2 Byte 2 Register	—
0x9020_201C + 0x1_0000 × m (m = 0 to 11) 0x9030_201C + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOS21	Safe Position Chanel 2 Byte 1 Register	—
0x9020_201D + 0x1_0000 × m (m = 0 to 11) 0x9030_201D + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOS20	Safe Position Chanel 2 Byte 0 Register	—
0x9020_201E + 0x1_0000 × m (m = 0 to 11) 0x9030_201E + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOSCRC2_H	Safe Position CRC 2 High Byte Register	—
0x9020_201F + 0x1_0000 × m (m = 0 to 11) 0x9030_201F + 0x1_0000 × (m - 12) (m = 12 to 15)	VPOSCRC2_L	Safe Position CRC 2 Low Byte Register	—
0x9020_203F + 0x1_0000 × m (m = 0 to 11) 0x9030_203F + 0x1_0000 × (m - 12) (m = 12 to 15)	DUMMY2	Dummy in Safe Channel2 Register	—

Table 51.7 HDSL related system control register (1 of 2)

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0 (LLPP0)	—	MSTPCRJ.MSTPCRJ01	SLVACCCTL7.LLPP_SL*1
1 (LLPP0)	—	MSTPCRJ.MSTPCRJ05	SLVACCCTL7.LLPP_SL*1
2 (LLPP0)	—	MSTPCRJ.MSTPCRJ09	SLVACCCTL7.LLPP_SL*1
3 (LLPP0)	—	MSTPCRJ.MSTPCRJ13	SLVACCCTL7.LLPP_SL*1
4 (LLPP0)	—	MSTPCRJ.MSTPCRJ17	SLVACCCTL7.LLPP_SL*1
5 (LLPP0)	—	MSTPCRJ.MSTPCRJ21	SLVACCCTL7.LLPP_SL*1
6 (LLPP0)	—	MSTPCRJ.MSTPCRJ25	SLVACCCTL7.LLPP_SL*1
7 (LLPP0)	—	MSTPCRJ.MSTPCRJ29	SLVACCCTL7.LLPP_SL*1
8 (LLPP0)	—	MSTPCRK.MSTPCRK01	SLVACCCTL7.LLPP_SL*1
9 (LLPP0)	—	MSTPCRK.MSTPCRK05	SLVACCCTL7.LLPP_SL*1
10 (LLPP0)	—	MSTPCRK.MSTPCRK09	SLVACCCTL7.LLPP_SL*1
11 (LLPP0)	—	MSTPCRK.MSTPCRK13	SLVACCCTL7.LLPP_SL*1
12 (LLPP1)	—	MSTPCRK.MSTPCRK17	SLVACCCTL7.LLPP_SL*1

Table 51.7 HDSL related system control register (2 of 2)

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
13 (LLPP1)	—	MSTPCRK.MSTPCRK21	SLVACCCTL7.LLPP_SL*1
14 (LLPP1)	—	MSTPCRK.MSTPCRK25	SLVACCCTL7.LLPP_SL*1
15 (LLPP1)	—	MSTPCRK.MSTPCRK29	SLVACCCTL7.LLPP_SL*1

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

51.3 Register Descriptions

HDSL has main function registers for drive interface and safe 1 interface, and secondary registers for safe 2 interface. The drive interface registers, safe 1 interface registers, and safe 2 interface registers can be accessed through the internal bus. The safe 1 interface register and safe 2 interface register can be accessed through each SPI interface. In this case, register address should be referred to offset address only.

And register access operation can be executed with 8-bit/16-bit/32-bit access. However, when 16-bit/32-bit access with unappropriate R/W attribution for the registers is executed, HDSL operation cannot be guaranteed.

51.3.1 Drive Interface Register

51.3.1.1 SYS_CTRL : System Control Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PRST	MRST	FRST	LOOP	—	—	SPOL	OEN
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OEN	Activation of the output 0: The impedance of the DSL cable is high. 1: The DSL cables are activated for output to the DSL Slave.	R/W
1	SPOL	Polarity of the synchronization pulse 0: The SYNC trailing edge is used. 1: The SYNC leading edge is used.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	LOOP	Test drive interface Value for the read-back test for drive interface. This value has no other purpose.	R/W
5	FRST	Pipeline FIFO, reset 0: Normal FIFO access 1: The FIFO is reset. Data is not stored and cannot be read.	R/W
6	MRST	Messages reset 0: Normal parameters channel action 1: The parameters channel is reset. Current short and long messages are discarded.	R/W
7	PRST	Protocol reset 0: Normal protocol action 1: A forced reset of the protocol status will be initiated. If the bit is cleared, a restart of the connection is triggered.	R/W

The SYS_CTRL register contains the main control bits of the DSL Master.

51.3.1.2 SYNC_CTRL : Synchronization Control Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ES[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ES[7:0]	External synchronization 0x00: Position sampling during free running at the shortest cycle time. Others: Position sampling with the SYNC input synchronized. The value from ES determines the number of position samplings carried out in one SYNC cycle. The user must match the number of samplings per cycle to the shortest frame length.	W

The synchronization control register for control of the synchronization contains the bit with which the synchronization source for position sampling is controlled.

51.3.1.3 MASTER_QM : Quality Monitoring Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LINK	—	—	—	QM[3:0]			
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
3:0	QM[3:0]	Quality monitoring bits Quality monitoring value. Higher values indicate a better connection. If the quality monitoring reaches the value 0x0, a forced reset of the protocol is carried out.	R
6:4	—	These bits are read as 0.	R
7	LINK	DSL protocol connection status It should be noted that LINK is also represented at the HDSLn_LINK output. (n = 0 to 15). 0: No connection present or connection error due to a communications error. 1: Protocol connection between DSL Master and Slave was established.	R

The MASTER_QM register contains the quality monitoring value for the data connection.

51.3.1.4 EVENT_H : High Byte Event Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	INT	SUM	—	—	POS	—	DTE	PRST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRST	Protocol reset warning 0: Normal protocol action 1: The forced protocol reset was triggered.	R/W*1
1	DTE	Estimator deviation threshold error 0: Current value of deviation smaller than the specified maximum. 1: Current value of deviation greater than the specified maximum.	R/W*1
2	—	This bit is read as 0.	R
3	POS	Estimator turned on This error usually indicates a transmission error on the DSL connection. If this error occurs frequently, the wiring of the DSL connection should be checked. If this error occurs continuously, there is probably an error in the motor feedback system. 0: The data for the fast position was correctly transmitted. 1: Fast position data consistency error. The fast position read through drive interface is supplied by the estimator.	R/W*1
5:4	—	These bits are read as 0.	R
6	SUM	Remote event monitoring When the SUM bit is set, an error or a warning has been transmitted from the DSL Slave. The frequency inverter application must check the SUMMARY register to obtain a detailed description. This bit is level sensitive. 0: All DSL Slave events are cleared. 1: The DSL Slave has signaled an event and the summary mask is set accordingly.	R/W*1
7	INT	Interrupt status 0: No interrupt in DSL Master 1: DSL Master interrupt occurred	R

Note 1. Write 0 only to the bit to be cleared.

51.3.1.5 EVENT_L : Low Byte Event Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MIN	ANS	—	QMLW	FREL	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	FREL	Channel free for "long message" If the bit is set, the frequency inverter application can trigger a "long message". Provided no answer has been received from the DSL Slave, this bit remains deleted. As the processing duration of a "long message" in the motor feedback system is not specified, a user time limit condition should be installed via the frequency inverter application. When a time limit is exceeded, the MRST bit in the SYS_CTRL register is set, which causes the Parameters Channel to be reset. 0: No "long message" can be sent. 1: A "long message" can be sent on the Parameters Channel.	R/W*1
2	QMLW	Quality monitoring low-value warning This warning indicates that a transmission error occurred at bit level for one of the CRC values. If this error occurs frequently, the wiring of the DSL connection should be checked. 0: Quality monitoring value greater than or equal to "14" 1: Quality monitoring value below "14"	R/W*1
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	ANS	Erroneous answer to "long message" This error indicates that the transmission of an answer from the DSL Slave to the last "long message" failed. The frequency inverter application must send the "long message" again. 0: The last answers to "long messages" were error-free. 1: An error occurred during the answer to a long message. The effectiveness of the previous transaction is not known.	R/W*1
5	MIN	Message initialization When this warning is displayed, the Parameters Channel is still in the initialization status and no "short message" or "long message" can be triggered. 0: No acknowledgment for the initialization received. 1: An acknowledgment was received from the Slave for the initialization of a message.	R/W*1
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write 0 only to the bit to be cleared.

The EVENT_H/EVENT_L register contains the messaging bits for all warning and error modes of the DSL system.

All messaging bits are set by the DSL Master if a corresponding status is determined.

An event bit that has been set is not reset by the DSL Master. The frequency inverter application must clear bits that have been set.

51.3.1.6 MASK_H : High Byte Event Mask Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MSUM	—	—	MPOS	—	MDTE	MPRST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MPRST	Mask for protocol reset warning 0: A protocol reset does not set the interrupt signal. 1: A protocol reset sets the interrupt signal.	W
1	MDTE	Mask for estimator deviation threshold error warning 0: A high estimator deviation threshold error value does not set the interrupt signal. 1: A high estimator deviation threshold error value sets the interrupt signal.	W
2	—	The write value should be 0.	W
3	MPOS	Mask for fast position error 0: An error in the fast position does not set the interrupt signal. 1: An error in the fast position sets the interrupt signal.	W
5:4	—	The write value should be 0.	W
6	MSUM	Mask for remote event monitoring 0: DSL Slave events that are masked in the SUMMARY register do not set the interrupt signal. 1: DSL Slave events that are masked in the SUMMARY register set the interrupt signal.	W
7	—	The write value should be 0.	W

51.3.1.7 MASK_L : Low Byte Event Mask Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MMIN	MANS	—	MQMLW	MFREL	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	The write value should be 0.	W
1	MFREL	Mask for "channel free for "long message"" 0: If a "long message" can be sent on the Parameters Channel, the interrupt signal is not set. 1: If a "long message" can be sent on the Parameters Channel, the interrupt signal is set.	W
2	MQMLW	Mask for low-quality monitoring value warning 0: A low-quality monitoring value does not set the interrupt signal. 1: A low-quality monitoring value sets the interrupt signal.	W
3	—	The write value should be 0.	W
4	MANS	Mask for erroneous answer to "long message" 0: A transmission error during the answer to a long message does not set the interrupt signal. 1: A transmission error during the answer to a long message sets the interrupt signal.	W
5	MMIN	Mask for message initialization confirmation 0: The acknowledgment for the initialization of a DSL Slave message does not set the interrupt signal. 1: The acknowledgment for the initialization of a DSL Slave message sets the interrupt signal.	W
7:6	—	The write value should be 0.	W

In the MASK_H/MASK_L registers, the events are set with which the interrupt signal is set.

51.3.1.8 MASK_SUM : Summary Mask Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MSUM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MSUM[7:0]	Mask for status summary bits 0: In the set status, the corresponding status summary bit does not set the SUM event monitoring and the interrupt signal. 1: In the set status, the corresponding status summary bit sets the SUM event monitoring and the interrupt signal.	W

In the MASK_SUM register, the DSL Slave collective events are determined with which the SUM event monitoring in the EVENT_H register, as well as the interrupt signal, are set.

Several events can be masked to trigger an interrupt. In addition, events from the DSL Master can be combined with these events.

51.3.1.9 EDGES : Edge Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)
HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	EDGES							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	EDGES	Identification of edges in the cable signal 0: No edge was detected in the time period of the corresponding bit. 1: An edge was detected in the time period of the corresponding bit.	R

The EDGES register contains the time control for the DSL cable bit sampling and can be used to monitor the connection quality.

Each individual EDGES register bit is set if, at system start-up, an edge of the test signal is detected during the time period of the corresponding bit. An edge is defined as a change in cable value between successive detections. The sampling is carried out eight times as fast as the cable bit rate.

The register is write-protected. The contents of this register do not change after the start-up phase. A new bit sampling pattern is only generated after a forced reset of the protocol.

51.3.1.10 DELAY : Run Time Delay Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)
HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RSSI[3:0]				CBLDLY[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CBLDLY[3:0]	4-bit value for cable delay This value gives the cable signal round trip delay of cable and transceivers in bits. This value enables a rough estimate of cable length to be made.	R
7:4	RSSI[3:0]	Indication of the received signal strength 4-bit value for the cable signal strength, from "0" to "12". Higher values indicate better connection quality. Low connection (< 2) quality affects QM. RSSI is continuously updated during operation and used for signal monitoring during run time.	R

The DELAY register stores information about the run-time delay of the system cable and the signal strength. The register can be used to monitor the connection quality.

The register is write-protected.

The value for Line Delay does not change after the start-up phase. A fresh value for Line Delay is only measured after a forced reset of the protocol.

Table 51.8 shows the relationship between the value in Line Delay and the cable length of the DSL connection.

Table 51.8 Cable delay

Cable delay (CBLDLY[3:0])	DSL connection cable delay	Cable length DSL connection
0	< 100 ns	< 10 m
1	100 to 200 ns	10 to 20 m
2	200 to 300 ns	20 to 30 m
3	300 to 400 ns	30 to 40 m
4	400 to 500 ns	40 to 50 m
5	500 to 600 ns	50 to 60 m
6	600 to 700 ns	60 to 70 m
7	700 to 800 ns	70 to 80 m
8	800 to 900 ns	80 to 90 m
9	900 to 1000 ns	90 to 100 m

51.3.1.11 VERSION : Version Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)
HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CODE[1:0]		MAJOR[1:0]		MINOR[3:0]			
Value after reset:	0	1	0	1	0	1	1	1

Bit	Symbol	Function	R/W
3:0	MINOR[3:0]	Minor release number The current version is 7.	R
5:4	MAJOR[1:0]	Major release number The current version is 1.	R
7:6	CODE[1:0]	Type of module 0 1: DSL Master Others: Reserved	R

The VERSION register contains the release version of the DSL Master module. The register is write-protected.

51.3.1.12 ENC_ID2 : Encoder ID 2 Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)
HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SCI[2:0]			ENCID[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ENCID[3:0]	Encoder designation code bit 19 to bit 16	R

Bit	Symbol	Function	R/W
6:4	SCI[2:0]	Indication of special characters 3-bit special character for later enhancements of the encoder designation code. Not allocated.	R
7	—	This bit is read as 0.	R

The ENC_ID2/1/0 registers contain the designation code of the motor feedback system connected to the DSL Master. In the current protocol specification, the designation code is 20 bits long. With later enhancements, the free bits in the encoder ID registers are used to indicate special characters.

These registers are write-protected.

The individual ENC_ID2/1/0 register bits are allocated as follows:

Table 51.9 Encoder designation code

Bit	Symbol	Function
19	ENC_ID2.ENCID[3]	Continue In High status, Continue indicates that ENDID is longer than 20 bits (for future use).
18:16	ENC_ID2.ENCID[2:0]	Reserved These bits are read as 0.
15:12	ENC_ID1.ENCID[7:4]	User-defined encoder index 4-bit value (0 to 15) for user-defined encoder index
11	ENC_ID1.ENCID[3]	Reserved This bit is read as 0.
10	ENC_ID1.ENCID[2]	Sign In High status, Sign indicates that the position value is signed, in Low status, Sign indicates that the position value is not signed.
9:8	ENC_ID1.ENCID[1:0]	#Pos – #Acc Length of position information (standard value: 40 bits) minus length of the acceleration value transmitted (standard value: 11 bits).
7:4	ENC_ID0.ENCID[7:4]	
3:0	ENC_ID0.ENCID[3:0]	#Acc – 8 Length of the acceleration value transmitted (standard value: 11 bits) minus 8.

51.3.1.13 ENC_IDn : Encoder ID n Register (n = 0, 1)

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)
HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x0E (ENC_ID1)
0x0F (ENC_ID0)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENCID[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ENCID[7:0]	Encoder designation code, byte n	R

51.3.1.14 POSn : Fast Position Byte n Register (n = 0 to 4)

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x10 (POS4)
0x11 (POS3)
0x12 (POS2)
0x13 (POS1)
0x14 (POS0)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	POS[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	POS[7:0]	Fast position, byte n Position value of the motor feedback system (length: 40 bits), incrementally generated.	R

The POSn registers for the fast position contain the value of the motor feedback system connected. This position is generated incrementally from the safe position at start-up and is updated with every protocol frame.

After every eight protocol frames, the fast position is checked against the safe position (VPOSn register).

The position sampling point is determined by the ES bits in SYNC_CTRL register.

Only those POS bits are activated that lie within the range that the motor feedback system has actually measured. All other higher value bits are read as "0". The number of measurable bits can be taken from ENCID bits 9 to 0 in the ENC_ID0 to 2 registers.

If Sign is set in the ENC_IDn register, the value of the fast position is given signed in the two's complement.

The units of the position value are (steps). These registers are write-protected.

51.3.1.15 VELn : Speed Byte n Register (n = 0 to 2)

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x15 (VEL2)
0x16 (VEL1)
0x17 (VEL0)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VEL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VEL[7:0]	Speed, byte n Speed of the motor feedback system (length: 24 bits)	R

The VELn registers contain the speed values of the connected motor feedback system. This value is calculated as a Δ position from the acceleration value (ΔΔposition) transmitted on the process data channel and the currently updated protocol frame.

The speed sampling point is determined by the ES bits of the SYNC_CTRL register. The units of the speed value are steps/frame cycle time.

These registers are write-protected.

51.3.1.16 MIR_SUM : Mirror Status Summary Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SUM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SUM[7:0]	Status summary bit bit [0] (interface): 0: The DSL Slave protocol has not triggered an error, warning, or event. 1: An error, a warning, or an event associated with the DSL Slave protocol interface was triggered. bit [7:1] (external resource): 0: The corresponding error, warning, or event is not active. 1: An error, a warning, or an event associated with the DSL Slave external resources was triggered.	R

The MIR_SUM register contains the same information for the drive application. The MIR_SUM register is based on the encoder mirror registers MIR_ST7 to MIR_ST0 in the DSL Slave and therefore delayed when compared to the SAFE_SUM register. The bits in the MIR_SUM register can be read by the drive application only.

51.3.1.17 PC_BUFn : Parameter Channel Buffer n Register (n = 0 to 7)

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x20 + 0x1 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PCBUF[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PCBUF[7:0]	Parameter Channel, byte n Eight bytes for the answer to a long message (read operation) or for a "long message" write operation. PCBUF[0]: bit 7 to bit 0 for Parameters Channel PCBUF[1]: bit 15 to bit 8 for Parameters Channel PCBUF[2]: bit 23 to bit 16 for Parameters Channel PCBUF[3]: bit 31 to bit 24 for Parameters Channel PCBUF[4]: bit 39 to bit 32 for Parameters Channel PCBUF[5]: bit 47 to bit 40 for Parameters Channel PCBUF[6]: bit 55 to bit 48 for Parameters Channel PCBUF[7]: bit 63 to bit 56 for Parameters Channel	R/W

The eight PC_BUFn registers of the Parameters Channel buffer contain the answer to the last "long message" request or the data for a "long message" write operation.

Depending on the length of the "long message" answer, the registers are used as follows:

Table 51.10 Data length of the "long message" (1 of 2)

Length of the "long message"	Register used (Offset address)
8 bytes	0x20 to 0x27
4 bytes	0x20 to 0x23

Table 51.10 Data length of the "long message" (2 of 2)

Length of the "long message"	Register used (Offset address)
2 bytes	0x20 to 0x21
0 bytes	None

These registers are also for the reporting of error conditions arising from a "long message" operation. If, when accessing a resource, an error due to a "long message" arises (e.g. invalid data, error in the A/D conversion), after the answering message has been received the LOFF bit in the PC_ADD_H register is set. In this case, the Parameters Channel buffer bytes 0 and 1 contain an error code.

The meaning of the error code depends on the particular HIPERFACE DSL® encoder.

51.3.1.18 PC_ADD_H : High Byte Long Message Address Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	LRW	LOFF	LIND	LLEN[1:0]		LADDH[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	LADDH[1:0]	Long message address, bit 9 to bit 8 Database entry with 10-bit address for a "long message" operation.	W
3:2	LLEN[1:0]	Data length of the "long message" 0 0: No data bytes 0 1: 2 data bytes 1 0: 4 data bytes 1 1: 8 data bytes	W
4	LIND	Indirect addressing of long messages 0: Direct addressing of "long messages". The operation affects the database entry given in the current address. 1: Indirect addressing of "long messages". During this operation, the stored address content in the given database entry is evaluated.	W
5	LOFF	Long message addressing mode/long message error 0: **Write access:** Addressing of "long message" without offset. The offset value from the PCOFFn registers is not used. **Read access:** The last "long message" was correctly processed. 1: **Write access:** Offset addressing of "long message". The offset value from the PCOFFn registers is used in the resource of the selected database entry as a sub-address. **Read access:** The last "long message" caused an error.	R/W
6	LRW	Long message, read/write mode 0: "long message" write operation 1: "long message" read operation	W
7	—	This bit is read as 0. The write value should be 0.	R/W

51.3.1.19 PC_ADD_L : Low Byte Long Message Address Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x29

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LADDL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	LADDL[7:0]	Long message address, bit 7 to bit 0 Database entry with 10-bit address for a "long message" operation	W

The addresses and the addressing mode for "long messages" sent over the Parameters Channel are determined in the PC_ADD_H/PC_ADD_L registers.

In addition, the PCA_DD_H register contains indications of errors arising from "long message" operations. For this sort of error, the Parameters Channel buffer contains the error code in bytes 0 and 1 associated with this status.

51.3.1.20 PC_OFF_H : High Byte Long Message Address Offset Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x2A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LID	LOFFADDH[6:0]						
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	LOFFADDH[6:0]	Long message offset value, bit 14 to bit 8 The 15-bit offset value of the "long message" address offset is stored in these bits.	W
7	LID	Long message identification	R

51.3.1.21 PC_OFF_L : Low Byte Long Message Address Offset Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x2B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LOFFADDL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	LOFFADDL[7:0]	Long message offset value, bit 7 to bit 0 The 15-bit offset value of the "long message" address offset is stored in these bits.	W

The PC_OFF_H/PC_OFF_L registers for long messages are used in "long message" operations if LOFF is set in the PC_ADD_H register. In this case, the LOFFADD value from these registers is used to communicate with the sub-address of a multiple-byte encoder resource.

Only write access is possible for these registers.

51.3.1.22 PC_CTRL : Parameter Channel Control Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x2C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LSTA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LSTA	Control of the long message start 0: No effect 1: A "long message" transaction is started with the values currently stored in the "long message" registers.	W
7:1	—	The write value should be 0.	W

The PC_CTRL register for the Parameters Channel handles the start of "long message" transactions. After setting all "long message" registers (registers PC_BUF0 to 7, PC_ADD_H/PC_ADD_L, and PC_OFF_H/PC_OFF_L), the "long message" is transmitted to the DSL Slave by setting the LSTA bit.

51.3.1.23 PIPE_S : SensorHub Channel Status Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x2D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	POVR	PEMP	PERR	PSCI
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSCI	Indication for special characters in the SensorHub Channel This bit is stored together with the pipeline data byte in question in the FIFO buffer. 0: Indication for "no special character". 1: A special character was received in the SensorHub Channel.	R
1	PERR	Coding error of the bits in the SensorHub Channel This bit is stored together with the pipeline data byte in question in the FIFO buffer. 0: No error in bit coding. 1: The bit level coding of the data currently in the SensorHub Channel is erroneous.	R
2	PEMP	The SensorHub channel buffer is empty This bit is updated after every access to the FIFO buffer. 0: No "buffer empty" error. 1: A read request was issued, but the FIFO buffer for SensorHub Channel data is empty. In this case, PIPE_D contains the value 0x00.	R

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>3</td><td>POVR</td><td>SensorHub Channel overflow
This bit is deleted after the read process.
Special characters are normally used as data separators or to indicate special events. To obtain information about which special character was received, the PIPE_D register must be read. All 8b10b special characters can be used on the SensorHub channel. An exception is the "K30.7" symbol that is used in HIPERFACE DSL® to indicate "no data" and is not stored in the FIFO buffer. Below table contains the supported 8b10b special characters.
<table>
<tr><th>Special characters</th><th>Coding in register PIPE_D</th></tr>
<tr><td>K28.0</td><td>0x1C</td></tr>
<tr><td>K28.1</td><td>0x3C</td></tr>
<tr><td>K28.2</td><td>0x5C</td></tr>
<tr><td>K28.3</td><td>0x7C</td></tr>
<tr><td>K28.4</td><td>0x9C</td></tr>
<tr><td>K28.5</td><td>0xBC</td></tr>
<tr><td>K28.6</td><td>0xDC</td></tr>
<tr><td>K28.7</td><td>0xFC</td></tr>
<tr><td>K23.7</td><td>0xF7</td></tr>
<tr><td>K27.7</td><td>0xFB</td></tr>
<tr><td>K29.7</td><td>0xFD</td></tr>
</table>
0: The capacity of the FIFO buffer for SensorHub Channel data is not yet exhausted.
1: The capacity of the 8-byte FIFO buffer for SensorHub Channel data was exhausted and since the last read process, values have been discarded.</td><td>R</td></tr>
<tr><td>7:4</td><td>—</td><td>These bits are read as 0.</td><td>R</td></tr>
</table>

The PIPE_S register provides information about the current status of the SensorHub Channel. When this register is read, the current data from the FIFO buffer is read and stored in an intermediate register so that a subsequent read process in the PIPE_D register can be considered to be completed at the same time as the PIPE_S register is read. Using this mechanism, any deviation between status and data information in this instance will prevent new data entering the SensorHub Channel during access to the FIFO buffer.

51.3.1.24 PIPE_D : SensorHub Channel Data Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x2E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SCDATA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SCDATA[7:0]	SensorHub Channel data 8-bit value of the FIFO buffer for SensorHub Channel data	R

The PIPE_D register contains the SensorHub Channel data that is stored in an 8 byte FIFO buffer. If new data arrives at the buffer when it is full, before PIPE_D is read, the oldest value is discarded and the POVR bit in PIPE_S is set. If a read request is issued when the buffer is empty, the PEMP bit in PIPE_S is set and the value 0x00 is transmitted. At the moment that the PIPE_S register is accessed, the corresponding PIPE_D value is frozen to guarantee synchronization between status and data information.

51.3.1.25 PC_DATA : Parameter Channel Short Message Mirror Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x2F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PCDATA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PCDATA[7:0]	"Short message" Mirror register data 8-bit value of the requested mirror register	R

When reading the mirror registers MIR_ST7 to MIR_ST0 instead of the real remote registers, the answer is available immediately in the PC_DATA register.

51.3.1.26 ACC_ERR_CNT : Fast Position Error Counter Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x38

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CNT[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	CNT[4:0]	Position error count/threshold **Read:** 5-bit value of count of transmitted fast position values with consecutive transmission errors **Write:** 5-bit value for threshold	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The ACC_ERR_CNT register returns the count of transmitted fast position values with consecutive transmission errors. The value is clamped to a maximum of 31 (0x1F).

With a writing access the error threshold can be set. This value is also clamped to a maximum of 31 (0x1F). If the count of transmitted fast position values with consecutive transmission errors exceeds this threshold, the error threshold will be set to 1.

51.3.1.27 MAXACC : Fast Position Acceleration Boundary Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x39

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RES[1:0]		MNT[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	MNT[5:0]	Mantissa of fast position acceleration boundary	W
7:6	RES[1:0]	Resolution of fast position acceleration boundary	W

The MAXACC register allows setting an acceleration threshold for a given application. This threshold is used by the fast position estimator to clamp the acceleration of the estimated position during communication or sensor failures of the fast position channel.

51.3.1.28 MAXDEV_H : Fast Position Estimator Deviation High Byte Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x3A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DEVH[7:0]							
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	DEVH[7:0]	Fast position estimator deviation high byte **Read:** 16-bit value of position deviation **Write:** 16-bit value for deviation threshold	R/W

51.3.1.29 MAXDEV_L : Fast Position Estimator Deviation Low Byte Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x3B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DEVL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	DEVL[7:0]	Fast position estimator deviation low byte **Read:** 16-bit value of position deviation **Write:** 16-bit value for deviation threshold	R/W

The MAXDEV_H/MAXDEV_L registers return the maximum absolute position deviation while the position estimator is active. The returned 16-bit value has the same format (resolution) as the fast position channel and is clamped to a maximum of 65535 steps (0xFFFF). The registers are set to the maximum value 0xFFFF at reset.

These registers also allow setting a deviation threshold value for triggering the error event. The threshold value can be written with the same format as the deviation (unsigned 16-bit, same resolution as the fast position channel).

51.3.1.30 MIR_STn : Mirror Status n Register (n = 0 to 7)

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x60 + 0x1 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MIRST[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MIRST[7:0]	Mirror status, byte n The individual bits indicate different errors, events and warnings. The meaning of each individual bit depends on the particular DSL Slave specification. 0: Encoder in normal status 1: Error, event, or warning status	R/W*1

Note 1. Write 0 only to the bit to be cleared.

The MIR_ST encoder status registers contain all slave system errors, events, and warnings from the DSL encoder. This way, the information can be accessed via drive interface without blocking the short message channel.

The allocation between the individual bits and the slave system statuses depends on the DSL Slave specification. MIR_STn registers (n = 0 to 7) are updated only when a read Short Message is initiated on the corresponding remote status register ENC_STn and the response is read from S_PC_DATA. Trigger for updating the MIR_STn, is reading S_PC_DATA.

Background: initiating a Short Message saves the index of the register, but reading S_PC_DATA triggers the actual copy of register content to the corresponding MIR_STn.

Example:

1. Read Short Message on ENC_ST0
2. Read S_PC_DATA
3. Read access on MIR_ST0 and read PC_DATA content – ENC_ST0 content is mirrored to Drive Interface successfully.

Table 51.11 Relationship between mirror status and SAFE_SUM register

Encoder status	SAFE_SUM bit
MIRST[0]	SSUM0
MIRST[1]	SSUM1
MIRST[2]	SSUM2
MIRST[3]	SSUM3
MIRST[4]	SSUM4
MIRST[5]	SSUM5
MIRST[6]	SSUM6
MIRST[7]	SSUM7

51.3.1.31 DUMMY : Dummy Register

Base address: HDSLDm = 0x9020_0000 + 0x1_0000 × m (m = 0 to 11)
HDSLDi = 0x9030_0000 + 0x1_0000 × (i − 12) (i = 12 to 15)
HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x3F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DUMMY[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	DUMMY[7:0]	Due to the transmission of the online-status, Online Status read transactions need less time for transmission. Therefore, dummy read transactions must be inserted to avoid unwanted extra transactions. A read access to this register 0x3F has no effect and must be used for this purpose.	R

When accessing through SPI interface, the dummy read by the DUMMY register in Safe 1 interface is required. The dummy read by the DUMMY register in Drive interface is not required.

51.3.2 Safe Channel 1 Interface Registers

Table 51.12 Safe channel 1 interface register map

Offset address	Register symbol	Register name
0x09	EDGES*1	Edge Register
0x0A	DELAY*1	Run Time Delay Register
0x0B	VERSION*1	Version Register
0x0D	ENC_ID2*1	Encoder ID 2 Register
0x0E	ENC_ID1*1	Encoder ID 1 Register
0x0F	ENC_ID0*1	Encoder ID 0 Register
0x19	VPOS4	Safe Position 4 Register
0x1A	VPOS3	Safe Position 3 Register
0x1B	VPOS2	Safe Position 2 Register
0x1C	VPOS1	Safe Position 1 Register
0x1D	VPOS0	Safe Position 0 Register
0x1E	VPOSCRC_H	Safe Position CRC High Byte Register
0x1F	VPOSCRC_L	Safe Position CRC Low Byte Register
0x35	SAFE_CTRL	Safe System Control Register
0x36	SAFE_SUM	Safe Status Summary Register
0x37	S_PC_DATA	Parameter Channel Short Message Register
0x3D	EVENT_S	Safe Event Register
0x3E	MASK_S	Safe Event Mask Register
0x3F	DUMMY*1	Dummy Register
0x40 + 0x1 × n	ENC_STn	Encoder Status n Register (n = 0 to 7)
0x7C	SRSSI	Slave RSSI Register
0x7E	MAIL	Slave Mail Register
0x7F	PING	Slave Ping Register

Note 1. These registers are shared with Drive Interface.

51.3.2.1 VPOSn : Safe Position n Register (n = 0 to 4)

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x19 (VPOS4)
0x1A (VPOS3)
0x1B (VPOS2)
0x1C (VPOS1)
0x1D (VPOS0)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOS[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOS[7:0]	Safe position, byte n Position value transmitted through safe channel 1 (length: 40 bits), absolute value.	R

The VPOSn registers for the safe position contain the position value from the primary channel of the motor feedback system connected. This safe position is transmitted in every eighth protocol frame.

The safe position is not synchronized with the sync signal.

Only those vpos bits are activated that lie within the range that the motor feedback system has actually measured. All other higher value bits are read as 0. The number of measurable bits can be taken from ENCID bits 9 to 0 in the ENC_ID0 to ENC_ID2 registers.

If Sign is set in the ENC_ID register, the value of the safe position is given signed in the two's complement.

The units of the position value are [steps]. These registers are write-protected.

51.3.2.2 VPOSCRC_H : Safe Position CRC High Byte Register

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x1E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOSCRCH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOSCRCH[7:0]	CRC of the safe position, bit 15 to bit 8 16-bit CRC checksum (CRC 16) of the safe position and status summary in Safe Channel 1.	R

51.3.2.3 VPOSCRC_L : Safe Position CRC Low Byte Register

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x1F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOSCRCL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOSCRCL[7:0]	CRC of the safe position, bit 7 to bit 0 16-bit CRC checksum (CRC 16) of the safe position and status summary in Safe Channel 1.	R

The VPOSCRC_H/VPOSCRC_L registers contain the CRC checksum of the VPOSn registers and the SAFE_SUM register.

The CRC is checked in the DSL Master. In order to guarantee, in a safety-related application, that the CRC machine in the module is functioning, these registers can be checked with an external cross-check in the diagnostics test interval.

The CRC is generated with the following CRC parameters:

Table 51.13 VPOSCRC parameter

Parameter	Value
CRC sequence	16 bits
CRC polynomial	$X^{16} + X^{15} + X^{12} + X^7 + X^6 + X^4 + X^3 + 1$ Normal representation: 0x90D9
Starting value	0x0000
Closing XOR value	0x00FF
Reverse data bytes	No
Reverse CRC before closing XOR	No

The sequence of the bytes to calculate the CRC is shown in the following figure:

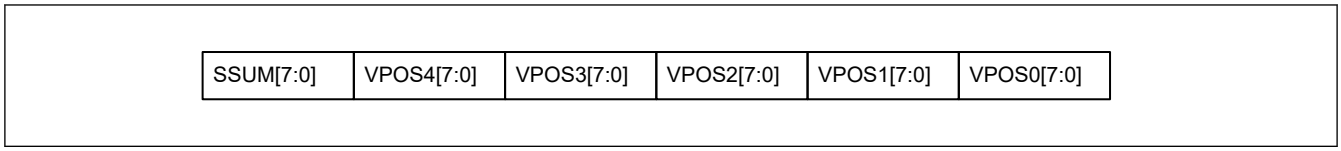


Figure 51.2 CRC calculation for VPOSCRC_H/VPOSCRC_L registers

51.3.2.4 SAFE_CTRL : Safe System Control Register

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x35

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PRST	MRST	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	MRST	Message reset 0: Normal Parameters Channel action 1: The Parameters Channel is reset. Current short and long messages are discarded.	R/W
7	PRST	Protocol reset 0: Normal protocol action 1: A forced reset of the protocol status will be initiated. If the bit is deleted, a restart of the connection is triggered.	R/W

The SAFE_CTRL register contains two control bits of the DSL Master available for the safety-related application.

51.3.2.5 SAFE_SUM : Safety Status Summary Register

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x36

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SSUM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SSUM[7:0]	Status summary bit bit [0] (interface): 0: The DSL Slave protocol has not triggered an error, a warning, or an event. 1: An error, a warning, or an event associated with the DSL Slave protocol interface was triggered. bit [7:1] (external resource): 0: The corresponding error, warning, or event is not active. 1: An error, a warning, or an event associated with DSL Slave external resources was triggered.	R

The SAFE_SUM register contains the summarized DSL Slave status information for the safety-related application. It is based on the encoder status ENC_STn (n = 0 to 7). Each status summary bit contains the summarized information from 8 error, warning, and event modes of the DSL Slave. The bits in the status summary register can be read by the safety-related application only.

51.3.2.6 S_PC_DATA : Parameter Channel Short Message Register

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x37

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPCDATA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SPCDATA[7:0]	"Short message" Parameter Channel data 8-bit value of the requested remote register.	R

The S_PC_DATA register for the Parameters Channel short message contains the results of "short message" transactions.

"Short message" transactions are generated if operations are carried out with remote registers (DSL Slave). Generally, FRES (in the EVENT_S register) must be set after a transaction is started. Only then will SPCDATA contain valid information.

51.3.2.7 EVENT_S : Safe Event Register

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x3D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SINT	SSUM	SCE	VPOS	QMLW	PRST	MIN	FRES
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FRES	Channel free for "short message" If the bit is set, the frequency inverter application can trigger a "short message". Provided no answer has been received from the DSL Slave, this bit remains 0. As the processing duration of a "long message" in the motor feedback system is not specified, a time limit condition is installed in the DSL Master. If the time limit is exceeded, attempts are made again automatically. 0: No "short message" can be sent. 1: A "short message" can be sent on the Parameters Channel.	R/W*1
1	MIN	Message init When this warning is displayed, the Parameters Channel is still in the initialization status and no "short message" or "long message" can be triggered. 0: No acknowledgment for the initialization received. 1: An acknowledgment was received from the Slave for the initialization of a message.	R/W*1
2	PRST	Protocol reset warning This error message indicates that the protocol connection to the DSL Slave has been re-initialized. This error message can be caused by a frequency inverter application request (PRST bit in SYS_CTRL register), a safety-related application request (PRST bit in SAFE_CTRL register), generated by the DSL Master itself, or activated via the reset input. The DSL Master causes a protocol reset if too many transmission errors indicate a connection problem. A protocol reset causes a re-synchronization with the DSL Slave that can improve the connection quality. 0: Normal protocol action 1: The forced protocol reset was triggered.	R/W*1

Bit	Symbol	Function	R/W
3	QMLW	Quality monitoring low-value warning This warning indicates that a transmission error occurred. If this error occurs frequently, the wiring of the DSL connection should be checked. 0: Quality monitoring value, QM[3:0] in the MASTERQM register, greater than or equal to "14" 1: Quality monitoring value below "14"	R/W*1
4	VPOS	Safe position error This error usually indicates an encoder sensor error. If this error occurs continuously, there is probably an error in the motor feedback system. 0: The safe position is correct. 1: Sensor error	R/W*1
5	SCE	Error on the Safe Channel This error usually indicates a transmission error on the DSL connection. If this error occurs frequently, the wiring of the DSL connection should be checked. If this error occurs continuously, there is probably an error in the motor feedback system. This error affects quality monitoring and produces the QMLW warning or a protocol reset. 0: Safe Channel data was correctly transmitted. 1: Data consistency error on the Safe Channel.	R/W*1
6	SSUM	Remote event monitoring When the bit is set, an error or a warning has been transmitted from the DSL Slave. The safety-related application must check the SAFE_SUM register to obtain a detailed description. This bit is level sensitive. 0: All DSL Slave events are cleared. 1: The DSL Slave has signaled an event.	R/W*1
7	SINT	Safe interrupt status This bit reflects the status of the safety-related interrupt.	R

Note 1. Write 0 only to the bit to be cleared.

The EVENT_S register contains the message Quality monitoring bits for all warning and error modes of the DSL system.

All messaging bits are set by the DSL Master if a corresponding status is determined.

The following bit description lists the effects of warning and error conditions as well as the reactions to errors that must be installed in the safety-related application.

An event bit that has been set is not reset by the DSL Master. The safety-related application must delete bits that have been set.

Both edge and level-sensitive flags are present in the EVENT_S register. Edge-sensitive bits are set when the corresponding status arises. They are only set again if the corresponding status disappears and then arises once more. This is the standard action. The level-sensitive bits set a bit as long as the corresponding status exists.

51.3.2.8 MASK_S : Safe Event Mask Register

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x3E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MSSUM	MSCE	MVPOS	MQMLW	MPRST	MMIN	MFRES
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MFRES	Mask for "channel free for "short message"" 0: If a "short message" can be sent on the Parameters Channel, the safety-related interrupt signal is not set. 1: If a "short message" can be sent on the Parameters Channel, the safety-related interrupt signal is set.	W

Bit	Symbol	Function	R/W
1	MMIN	Mask for message initialization confirmation 0: The acknowledgment for the initialization of a DSL Slave message does not set the safety-related interrupt signal. 1: The acknowledgment for the initialization of a DSL Slave message sets the safety-related interrupt signal.	W
2	MPRST	Mask for protocol reset warning 0: A protocol reset does not set the safety-related interrupt signal. 1: A protocol reset sets the safety-related interrupt signal.	W
3	MQMLW	Mask for low-quality monitoring value warning 0: A low-quality monitoring value does not set the safety-related interrupt signal. 1: A low-quality monitoring value sets the safety-related interrupt signal.	W
4	MVPOS	Mask for safe position error 0: An error in the safe position does not set the safety-related interrupt signal. 1: An error in the safe position sets the safety-related interrupt signal.	W
5	MSCE	Mask for transmission errors on the Safe Channel 0: A transmission error on the Safe Channel does not set the safety-related interrupt signal. 1: A transmission error on the Safe Channel sets the safety-related interrupt signal.	W
6	MSSUM	Mask for remote event monitoring 0: DSL Slave events that are set in the SAFE_SUM register do not set the safety-related interrupt signal. 1: DSL Slave events that are set in the SAFE_SUM register sets the safety-related interrupt signal.	W
7	—	The write value should be 0.	W

In the MASK_S register, events are chosen that can trigger the safety-related interrupt signal.

Several events can be masked to trigger an interrupt. In addition, events from the DSL Slave summary can be combined with these events.

51.3.2.9 ENC_STn : Encoder Status n Register (n = 0 to 7)

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x40 + 0x1 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENCST[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ENCST[7:0]	Encoder status, byte n The individual bits indicate different errors, events, and warnings. The meaning of each individual bit is determined by the particular DSL Slave installation. ENCST[0]: bit 7 to bit 0 for Parameters Channel ENCST[1]: bit 15 to bit 8 for Parameters Channel ENCST[2]: bit 23 to bit 16 for Parameters Channel ENCST[3]: bit 31 to bit 24 for Parameters Channel ENCST[4]: bit 39 to bit 32 for Parameters Channel ENCST[5]: bit 47 to bit 40 for Parameters Channel ENCST[6]: bit 55 to bit 48 for Parameters Channel ENCST[7]: bit 63 to bit 56 for Parameters Channel 0: Encoder in normal status 1: Error, event, or warning status	R/W*1

Note 1. Write 0 only to the bit to be cleared.

The ENC_STn encoder status registers contain all slave system errors, events, and warnings from the DSL encoder.

The allocation between the individual bits and the slave system statuses depends on DSL Slave specification.

Table 51.14 Relationship between encoder status and SAFE_SUM register

Encoder status	SAFE_SUM bit
ENCST[0]	SSUM0
ENCST[1]	SSUM1
ENCST[2]	SSUM2
ENCST[3]	SSUM3
ENCST[4]	SSUM4
ENCST[5]	SSUM5
ENCST[6]	SSUM6
ENCST[7]	SSUM7

51.3.2.10 SRSSI : Slave RSSI Register

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x7C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SRSSI[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SRSSI[2:0]	Value of the Slave RSSI The values for the Slave RSSI range from "0" (poorest signal strength) to "7" (best signal strength).	R
7:3	—	These bits are read as 0.	R

The SRSSI register for indicating the received signal strength at the slave (Slave Received Signal Strength Indicator, RSSI) provides an indication of the strength of the signal arriving at the slave.

The value of the register is only updated from frame to frame if the measurement result deteriorates. After a read access to this register, the register is reset to the value "7" (maximum signal strength).

The register is write-protected.

51.3.2.11 MAIL : Slave Mail Register

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x7E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MAIL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MAIL[7:0]	Slave-Mail 8-bit slave mail data for multiple applications	W

The MAIL multi-purpose register of the slave is used for fast communication with the DSL motor feedback system processor. The content of the slave mail register is transmitted to the encoder processor by the most rapid route possible.

The slave mail register is used for the transmission of test messages to the DSL motor feedback system.

51.3.2.12 PING : Slave Ping Register

Base address: HDSLS1m = 0x9020_1000 + 0x1_0000 × m (m = 0 to 11)
HDSLS1i = 0x9030_1000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x7F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PING[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PING[7:0]	Slave-Ping 8-bit hardware version of the DSL encoder at start-up. On first reading, the Ping value of the slave can be used as a connection test on behalf of the slave.	R/W

The PING register of the slave is used to carry out connection tests on behalf of the DSL Slave. The register can be written to and read externally, without this affecting the DSL interface.

On start-up, the register is initialized with the DSL Slave interface.

51.3.3 Safe Channel 2 Interface Registers

51.3.3.1 VERSION2 : Version in Safe Channel 2 Register

Base address: HDSLS2m = 0x9020_2000 + 0x1_0000 × m (m = 0 to 11)
HDSLS2i = 0x9030_2000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CODE[1:0]		MAJOR[1:0]		MINOR[3:0]			
Value after reset:	0	1	0	1	0	1	1	1

Bit	Symbol	Function	R/W
3:0	MINOR[3:0]	Minor release number The current version is 7.	R
5:4	MAJOR[1:0]	Major release number The current version is 1.	R
7:6	CODE[1:0]	Type of module 0 1: DSL Master Others: Reserved	R

The VERSION2 register in safe channel 2 is identical to the VERSION register in the drive and safe channel 1 interfaces.

51.3.3.2 ENC2_ID : Encoder ID in Safe Channel 2 Register

Base address: HDSLS2m = 0x9020_2000 + 0x1_0000 × m (m = 0 to 11)
HDSLS2i = 0x9030_2000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x0F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENC2ID[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ENC2ID[7:0]	Designation of the safe channel 2 channel **0x02/0x22:** SIL2 arrangement. The same position data is transmitted to the safe channel 2 channel as to the safe channel 1 channel. **0x03/0x33:** SIL3 arrangement. Position data is transmitted to the safe channel 2 channel that originates from the secondary sensor in the HIPERFACE DSL® motor feedback system.	R

The ENC2_ID register for the ID of the encoder in safe channel 2 contains the designation code for the safe channel 2 channel of the DSL encoder.

The contents of ENC2_ID are defined during the actual installation of the motor feedback system and can be found in the corresponding data sheet.

The register is write-protected.

51.3.3.3 STATUS2 : Safe Channel 2 Status Register

Base address: HDSLS2m = 0x9020_2000 + 0x1_0000 × m (m = 0 to 11)
HDSLS2i = 0x9030_2000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TOG2	TEST2	ERR2	FIX2[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	FIX2[4:0]	Safe Channel 2 Fixed Bit Pattern FIX2 is a fixed bit pattern for each SPI transaction that indicates errors arising in the SPI interface due to unchanged bit values. The standard value of the fixed bit pattern is 11100b. All other values indicate an error on safe channel 2 of the DSL system. Suitable measures must be installed in the user application.	R
5	ERR2	Safe Channel 2 Position Error 0: The last safe position received in safe channel 2 is correct. 1: The last safe position received in safe channel 2 is invalid. Suitable measures must be installed in the user application.	R
6	TEST2	Safe Channel 2 Test TEST2 bit is set to 1 if a test message is executed via the MAIL register while the safe channel 2 status and position data are valid. TEST2 can only be valid if the user application has previously requested a test. Corresponding error indications for TEST2 are either the ERR2 bit or a discrepancy between the position and the CRC of safe channel 2.	R
7	TOG2	Safe Channel 2 Toggle For successive position transmissions on safe channel 2, TOG2 must always toggle between 0 and 1. The starting value for TOG2 is 0. If the toggle bit does not change its value, it is probable that a transmission error occurred and the transmitted absolute value for safe channel 2 is invalid. Suitable measures must be installed in the user application.	R

The STATUS2 status register contains the status information for safe channel 2 of the HIPERFACE DSL® motor feedback system.

51.3.3.4 VPOS2n : Safe Position Channel 2 Byte n Register (n = 0 to 4)

Base address: HDSLS2m = 0x9020_2000 + 0x1_0000 × m (m = 0 to 11)
HDSLS2i = 0x9030_2000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x19 (VPOS24)
0x1A (VPOS23)
0x1B (VPOS22)
0x1C (VPOS21)
0x1D (VPOS20)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOS2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOS2[7:0]	Safe position, Channel 2, byte n Position value at motor feedback system Safe Channel 2 (length: 40 bits), transmitted as an absolute value complement.	R

The VPOS2n registers contain the position value from the secondary channel of the motor feedback system connected. This safe position is transmitted in every eighth protocol frame if the validity of the data transfer has been checked.

The safe position is not synchronized with the most recent sync signal but to the (user or internally generated) sync signal of one of the last 8 protocol frames. Only those VPOS2 bits are relevant that lie within the range that the motor feedback system has actually measured. Also, typically channel 2 has a lower resolution than channel 1. The operating manual and datasheet of the product specify the resolution of each channel.

The units of the position value are [steps].

These registers are write-protected.

51.3.3.5 VPOSCRC2_H : Safe Position CRC 2 High Byte Register

Base address: HDSLS2m = 0x9020_2000 + 0x1_0000 × m (m = 0 to 11)
HDSLS2i = 0x9030_2000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x1E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOSCRC2H[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOSCRC2H[7:0]	CRC of the safe position 2, bit 15 to bit 8 16-bit CRC checksum (CRC 16) of the safe position and status summary in Safe Channel 2.	R

51.3.3.6 VPOSCRC2_L : Safe Position CRC 2 Low Byte Register

Base address: HDSLS2m = 0x9020_2000 + 0x1_0000 × m (m = 0 to 11)
HDSLS2i = 0x9030_2000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x1F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOSCRC2L[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOSCRC2L[7:0]	CRC of the safe position 2, bit 7 to bit 0 16-bit CRC checksum (CRC 16) of the safe position and status summary in safe channel 2.	R

The VPOSCRC2_H/VPOSCRC2_L registers for the position checksum in safe channel 2 contain the CRC checksum for the safe position VPOS2n.

The CRC is checked in the DSL MasterIP Core. In order to guarantee, in a safety-related application, that the CRC machine is functioning, these registers can be checked with an external cross-check in the diagnostics test interval.

The CRC is generated with the following CRC parameters:

Table 51.15 VPOSCRC2 parameter

Parameter	Value
CRC sequence	16 bits
CRC polynomial	$X^{16} + X^{15} + X^{12} + X^7 + X^6 + X^4 + X^3 + 1$ Normal representation: 0x90D9
Starting value	0x0000
Closing XOR value	0x00FF
Reverse data bytes	No
Reverse CRC before closing XOR	No

The sequence of the bytes to calculate the CRC is shown in the following figure:

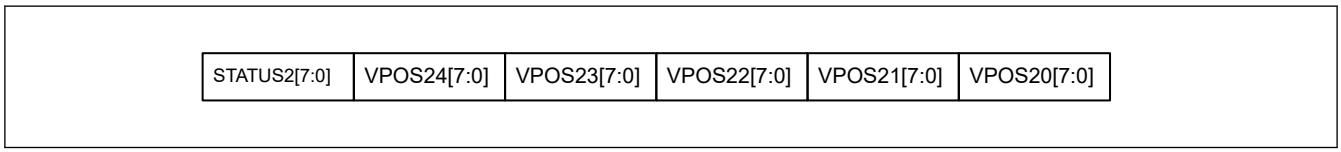


Figure 51.3 CRC calculation for VPOSCRC2_H/VPOSCRC2_L registers

51.3.3.7 DUMMY2 : Dummy in Safe Channel 2 Register

Base address: HDSLS2m = 0x9020_2000 + 0x1_0000 × m (m = 0 to 11)
HDSLS2i = 0x9030_2000 + 0x1_0000 × (i − 12) (i = 12 to 15)

Offset address: 0x3F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DUMMY[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	DUMMY[7:0]	Due to the transmission of the online-status, Online status read transactions need less time for transmission. Therefore, dummy read transactions must be inserted to avoid unwanted extra transactions. A read access to this register 0x3F has no effect and must be used for this purpose.	R

When accessing through SPI interface, the dummy read by the DUMMY2 register in the safe channel 2 is required.

51.4 Operation

51.4.1 System Start

As soon as the motor feedback system is supplied with power, a forced reset ensures that a defined system start status is produced in the DSL Master.

Figure 51.4 shows the status table for system start.

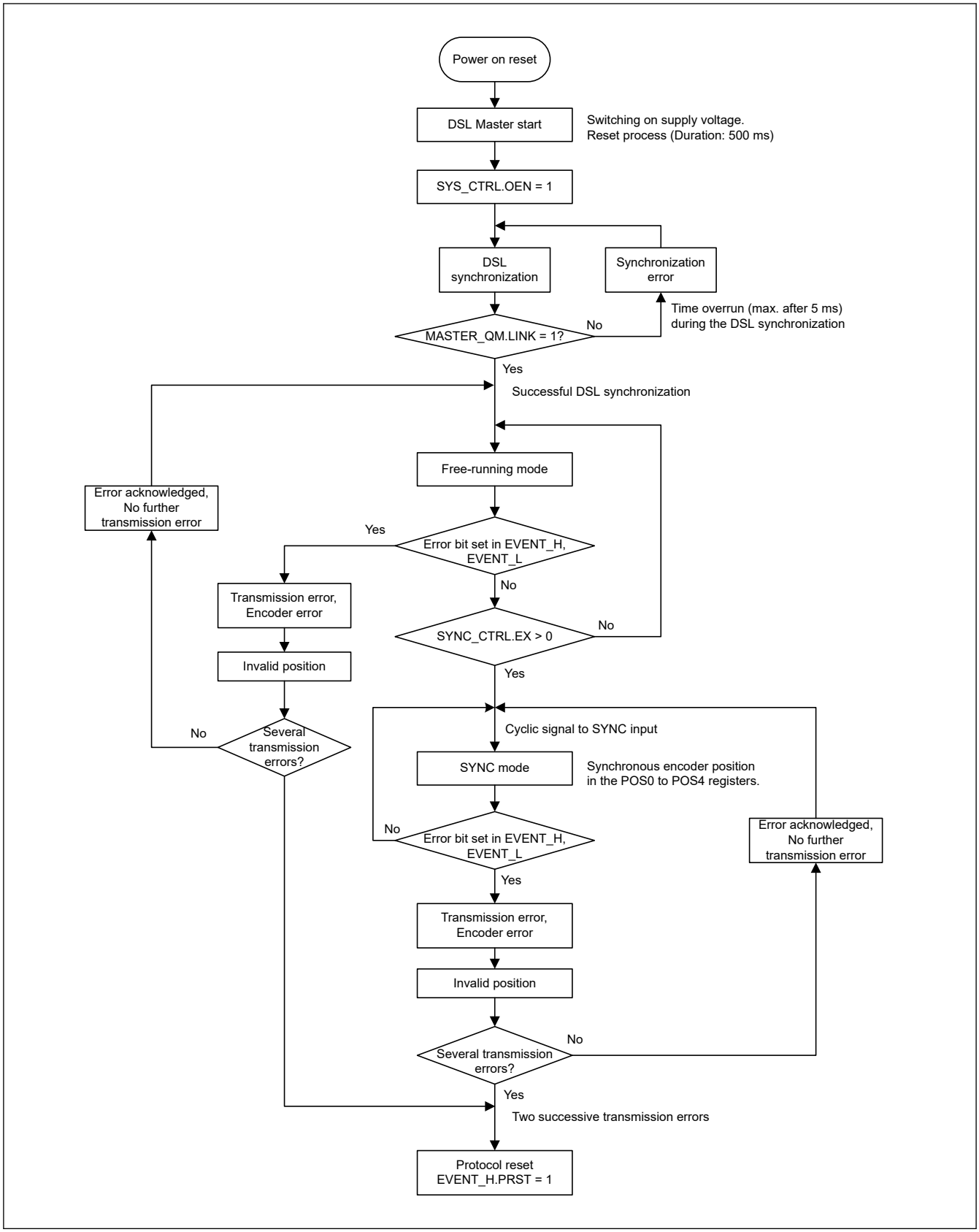


Figure 51.4 Status table for DSL system start

51.4.2 Quality Monitoring

The MASTERQM register contains the quality monitoring value for the data connection.

As soon as the DSL Master detects events that indicate an improvement or degradation of the quality of the data connection, these events are indicated as values higher or lower than the quality monitoring value.

Table 51.16 Quality monitoring events

Protocol event	Value change in quality monitoring
Wrong synchronization in Safe Channel (last byte)	-4
Wrong synchronization in Safe Channel (1st … 7th bytes)	-6
RSSI < 2	-4
Wrong encoding in parameter	-1
Wrong encoding in process data channel	-2
Unknown special characters in the protocol package	-2
Any identified error in Safe Channel 1	-6
Any identified error in Safe Channel 2	-8
Correct synchronization in Safe Channel	+1
Correct CRC value in Safe Channel	+1

Quality monitoring is initiated with the value "8". The maximum quality monitoring value is "15". This is the standard value during operation.

Note: Particular attention must be paid to the quality monitoring value during the development of a DSL drive controller. If a value lower than "15" is indicated, the cause may be a problem with the connection circuit, particularly if the value is continuously displayed.

If the quality monitoring value falls below "14", QMLW information is indicated in the EVENT_L register.

If the quality monitoring value falls to "0", a forced reset of the protocol is carried out. This is indicated by the PRST error bit in the EVENT_H register.

51.4.3 Interrupt

The interrupt signal is set to 1 if an interrupt condition has been fulfilled in the DSL Master. The interrupt conditions are set using the registers MASKH, MASKL, and MASKSUM. The interrupt conditions for safety are set using register MASKS.

Figure 51.5 and Figure 51.6 show an interrupt condition as a diagram.

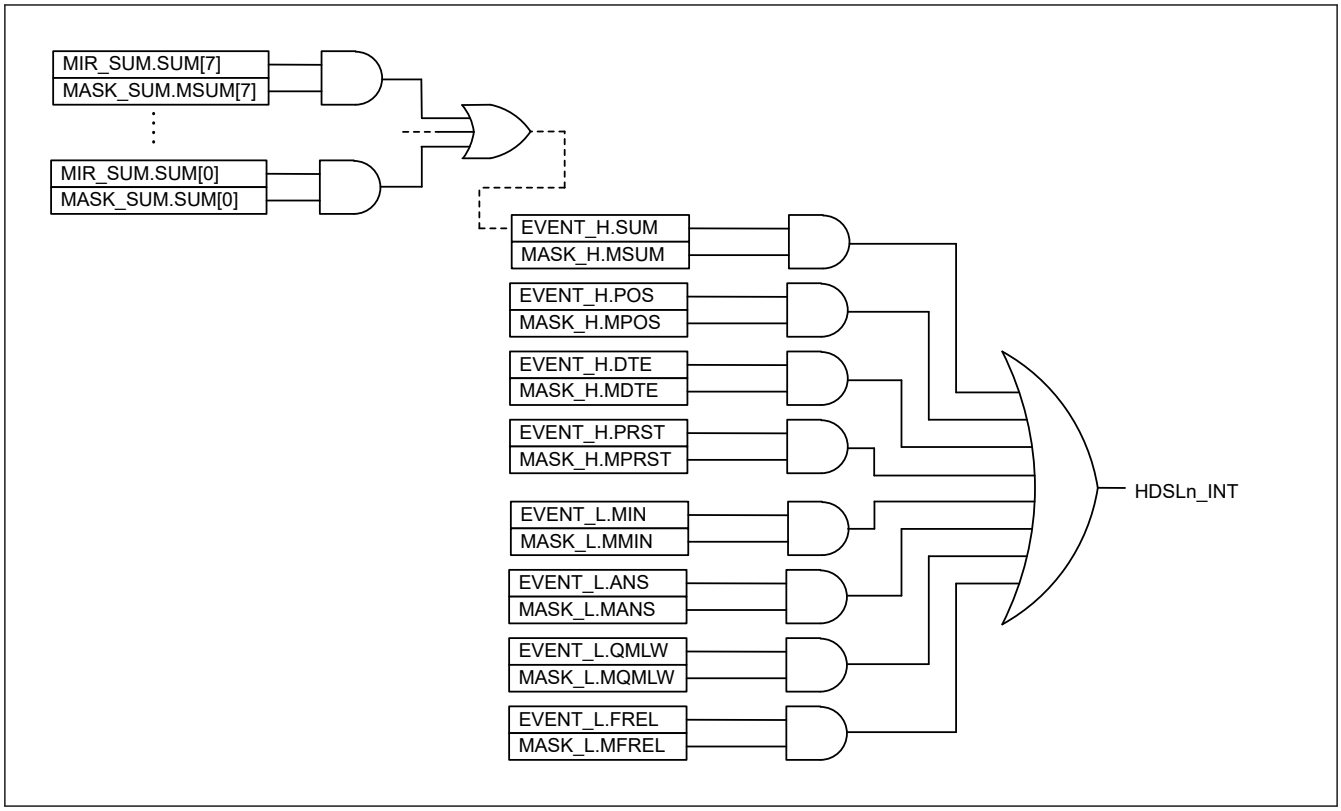


Figure 51.5 Interrupt

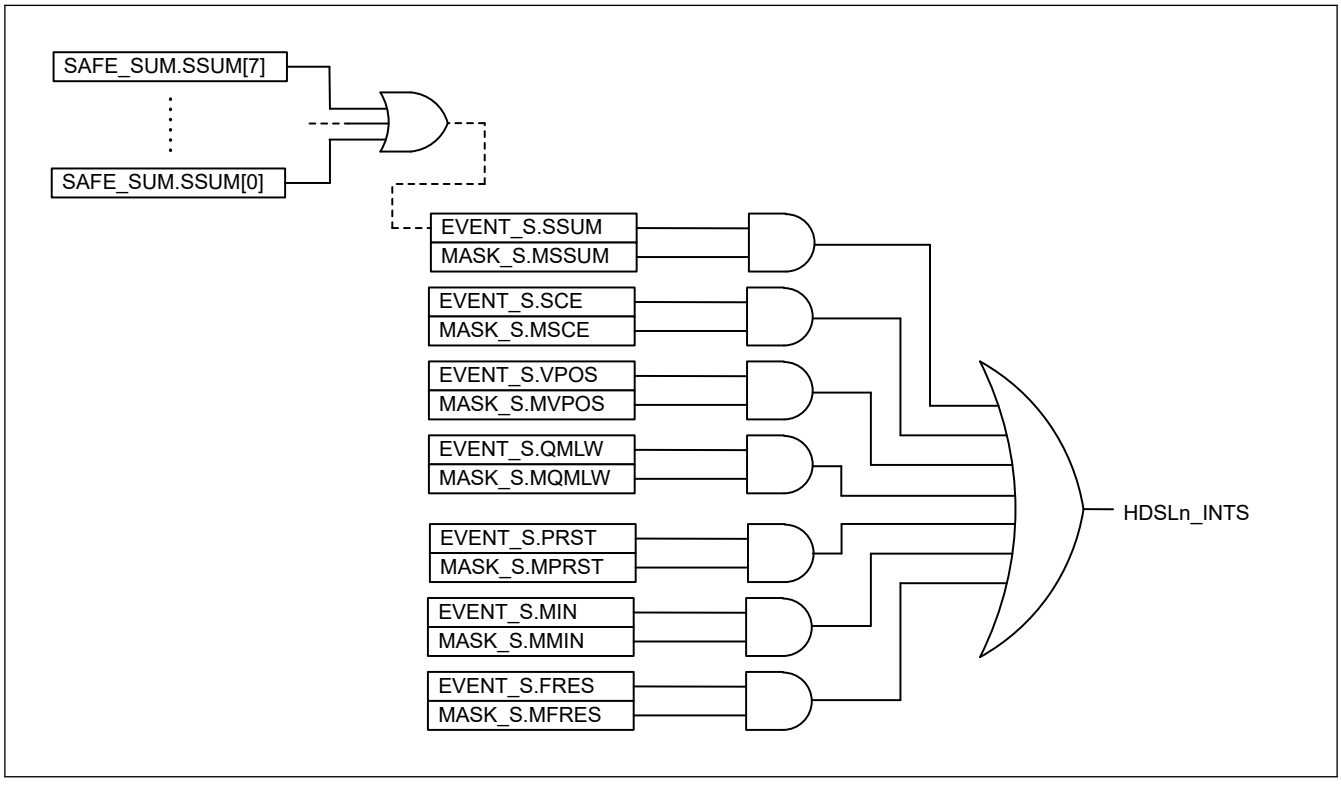


Figure 51.6 Interrupt for Safety

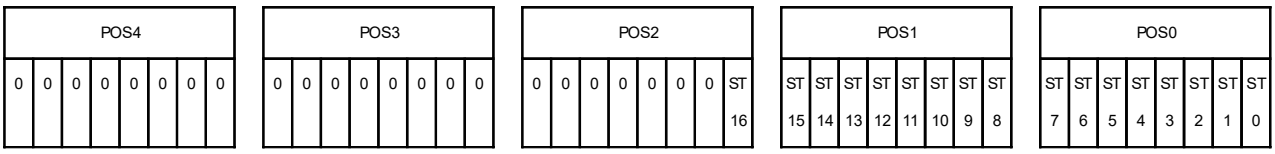
51.4.4 Fast Position

The fast position and the rotation speed of the encoder shaft are transmitted on the DSL motor feedback system process data channel. These values are the main process values for the drive application control circuit.

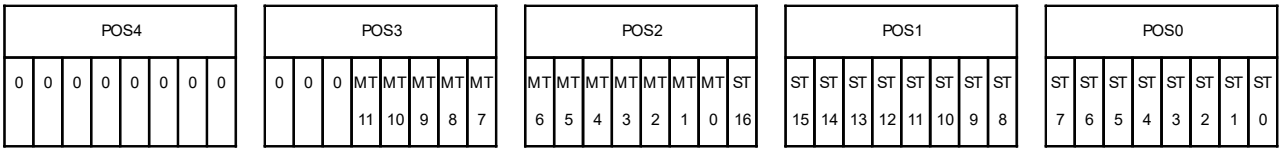
HIPERFACE DSL stores the fast position in the POS0 to POS4 HDSL Master registers and the rotation speed in the VEL0 to VEL2 registers.

The position is given as a 40 bit value that includes the angle setting ("single-turn" value) and the number of rotations ("multi-turn" value). Only the position bits actually measured by the motor feedback system are accessible and are stored in the registers as a right-justified value. The other (higher value) bits are constantly set at "0" (see examples "a" to "c" below).

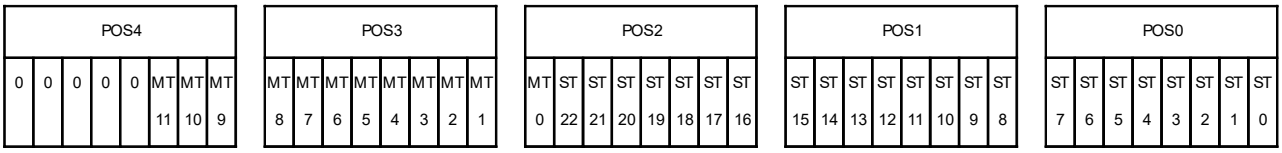
a. Single-turn 17 bit



b. Single-turn 17 bit, Multi-turn 12 bit



c. Single-turn 23 bit, Multi-turn 12 bit



The motor feedback system fast position is sampled and transmitted if the DSL Master receives a SYNC signal.

This SYNC signal can be created in two different ways (see section 51.4.4.1. Free running mode and section 51.4.4.2. SYNC mode).

51.4.4.1 Free running mode

In free-running mode, the SYNC signal is automatically created by the DSL Master, for which the maximum frame transmission frequency is used. The free-running mode is the standard DSL Master operating mode at start-up. This operating mode can also be selected manually, by setting the ES value in the SYNC_CTRL register to 0.

51.4.4.2 SYNC mode

In SYNC mode, the DSL Master depends on a prepared cyclic control signal. This control signal triggers position measurements and enables polling of position and rotation speed values synchronously with the control signal. The control signal must be applied to the SYNC input and have the characteristics prescribed for the DSL Master.

The position is available after a set delay in relation to the leading edge of the control signal.

When SYNC mode is used the following points must be noted:

1. A correct control signal must be applied at the sync input. The signal must correspond with the specifications for pulse width and cycle time.
2. Setting or deleting the SPOL bit in the SYS_CTRL register determines whether the position measurements are to be triggered by the leading or the trailing edge of the control signal. The set latency of the DSL system is measured from this edge.
3. The correct ES divider must be set in the SYNC_CTRL register. This divider determines how many position samplings and transmissions will be undertaken for each control signal. Do not change setting of ES divider during start-up or operation.

52. A-Format (AFMT)

52.1 Overview

The MPU includes A-format (AFMT) master to enable communication between the encoder with A-format interface. Table 52.1 describes the A-format specifications and Figure 52.1 shows a block diagram of the AFMT.

Table 52.1 A-format (AFMT) specifications

Item	Description
Number of units	16 units (12 in LLPP0 and 4 in LLPP1)
Protocol	A-format Ver.2.0
Functions	• Communication with absolute encoder supporting A-format • Up to 8 slaves bus connection each unit. (Individual transmission mode only) • Compliant to A-format Version 2.0 and A-Safety-format Version 1.0 • Trigger input – CPU control (SW trigger) – HW trigger • Encoder status • Communication alarm • Status and error messages • Time out
Interrupt sources	• Two interrupts – Communication completion – Time out error
Event link	nEXT_STRG input for HW trigger provided from ELC
Module-stop function	Module-stop state can be set to reduce power consumption

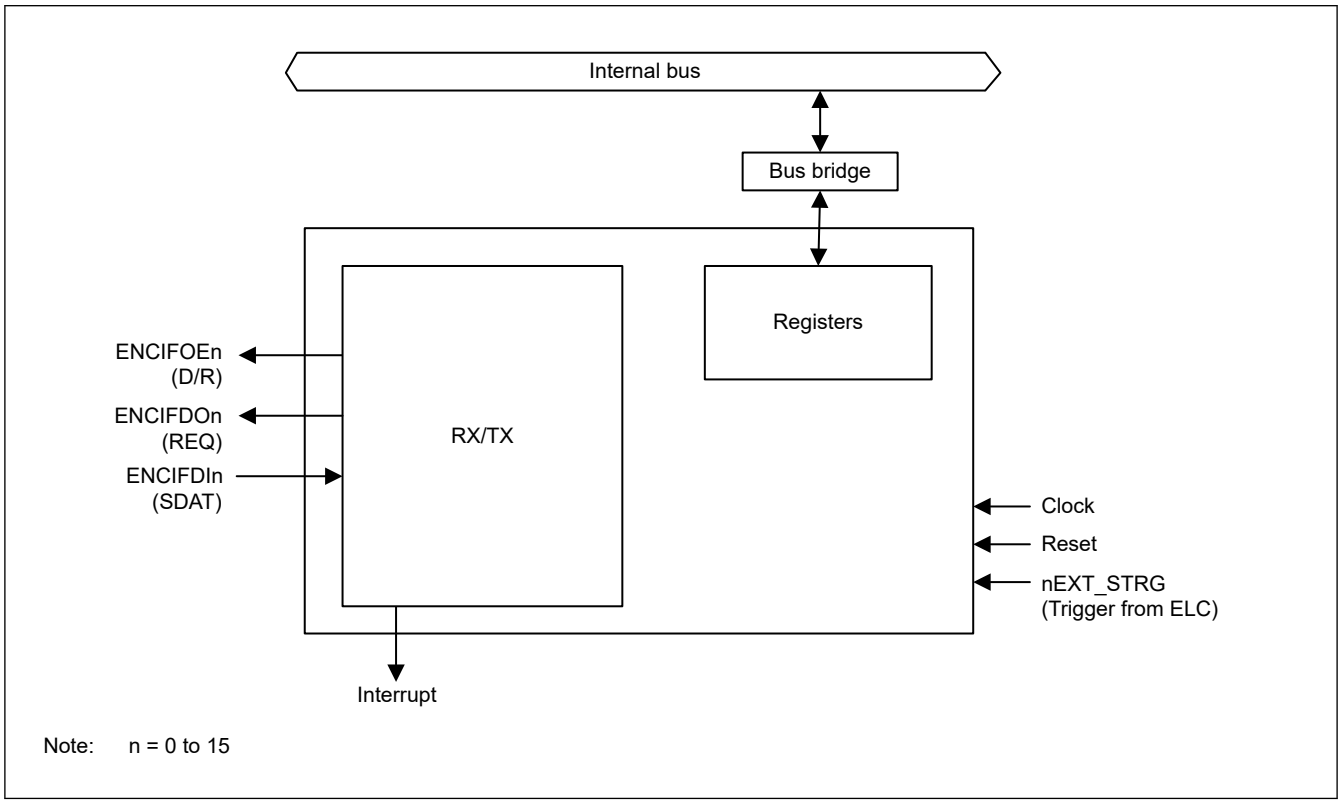


Figure 52.1 Block diagram of A-format (AFMT)

Table 52.2 lists the input/output pins of this module.

Table 52.2 PIN configuration of the AFMT

Unit	Pin name	I/O	Function
AFMTn	ENCIFOEn (D/R)	Output	A-format unit n line transceiver control signal
	ENCIFDOn (REQ)	Output	A-format unit n serial data output
	ENCIFDIn (SDAT)	Input	A-format unit n serial data input

Note: n = 0 to 15

Table 52.3 AFMT interrupt sources

Name	Interrupt sources
AFMTi_EOF	Communication completion
AFMTi_TMOUT	Time out error

Note: i = 0 to 15

52.2 Register Map

A part of the address of the register is shared between writing and reading.

Table 52.4 AFMT register map (for writing)

Address	Register symbol	Register name	Write protection
0x9020_6000 + 0x1_0000 × i (i = 0 to 11) 0x9030_6000 + 0x1_0000 × (i - 12) (i = 12 to 15)	COMMAND	Command Code Encoder Address Register	—
0x9020_6004 + 0x1_0000 × i (i = 0 to 11) 0x9030_6004 + 0x1_0000 × (i - 12) (i = 12 to 15)	ROMDATA	EEPROM Address and Write Data Register	—
0x9020_6008 + 0x1_0000 × i (i = 0 to 11) 0x9030_6008 + 0x1_0000 × (i - 12) (i = 12 to 15)	ID	ID Code Register	—
0x9020_6100 + 0x1_0000 × i (i = 0 to 11) 0x9030_6100 + 0x1_0000 × (i - 12) (i = 12 to 15)	TRGSEL	Trigger Select Register	—
0x9020_6104 + 0x1_0000 × i (i = 0 to 11) 0x9030_6104 + 0x1_0000 × (i - 12) (i = 12 to 15)	TXTRG	Transmission Trigger Register	—
0x9020_6200 + 0x1_0000 × i (i = 0 to 11) 0x9030_6200 + 0x1_0000 × (i - 12) (i = 12 to 15)	BRSEL	Baud Rate Setting Register	—

Table 52.5 AFMT register map (for reading)

Address	Register symbol	Register name	Write protection
0x9020_6000 + 0x004 × m + 0x020 × (n - 1) + 0x1_0000 × i (i = 0 to 11) 0x9030_6000 + 0x004 × m + 0x020 × (n - 1) + 0x1_0000 × (i - 12) (i = 12 to 15)	ENCnRXDATAm	ENCn Receive Data m Register (n = 1 to 8, m = 0 to 7)	—

Table 52.6 AFMT related system control register (1 of 2)

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0 (LLPP0)	—	MSTPCRJ.MSTPCRJ00	SLVACCCTL7.LLPP_SL*1
1 (LLPP0)	—	MSTPCRJ.MSTPCRJ04	SLVACCCTL7.LLPP_SL*1
2 (LLPP0)	—	MSTPCRJ.MSTPCRJ08	SLVACCCTL7.LLPP_SL*1
3 (LLPP0)	—	MSTPCRJ.MSTPCRJ12	SLVACCCTL7.LLPP_SL*1
4 (LLPP0)	—	MSTPCRJ.MSTPCRJ16	SLVACCCTL7.LLPP_SL*1
5 (LLPP0)	—	MSTPCRJ.MSTPCRJ20	SLVACCCTL7.LLPP_SL*1
6 (LLPP0)	—	MSTPCRJ.MSTPCRJ24	SLVACCCTL7.LLPP_SL*1
7 (LLPP0)	—	MSTPCRJ.MSTPCRJ28	SLVACCCTL7.LLPP_SL*1

Table 52.6 AFMT related system control register (2 of 2)

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
8 (LLPP0)	—	MSTPCRK.MSTPCRK00	SLVACCCTL7.LLPP_SL*1
9 (LLPP0)	—	MSTPCRK.MSTPCRK04	SLVACCCTL7.LLPP_SL*1
10 (LLPP0)	—	MSTPCRK.MSTPCRK08	SLVACCCTL7.LLPP_SL*1
11 (LLPP0)	—	MSTPCRK.MSTPCRK12	SLVACCCTL7.LLPP_SL*1
12 (LLPP1)	—	MSTPCRK.MSTPCRK16	SLVACCCTL7.LLPP_SL*1
13 (LLPP1)	—	MSTPCRK.MSTPCRK20	SLVACCCTL7.LLPP_SL*1
14 (LLPP1)	—	MSTPCRK.MSTPCRK24	SLVACCCTL7.LLPP_SL*1
15 (LLPP1)	—	MSTPCRK.MSTPCRK28	SLVACCCTL7.LLPP_SL*1

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

52.3 Register Descriptions

52.3.1 COMMAND : Command Code Encoder Address Register

Base address: AFMTi = 0x9020_6000 + 0x1_0000 × i (i = 0 to 11)
AFMTj = 0x9030_6000 + 0x1_0000 × (j - 12) (j = 12 to 15)

Offset address: 0x000

Bit position:	31																			12				8	7			3	2		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SID[4:0]					CC[4:0]					XEA[2:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	XEA[2:0]	Encoder address Please see Table 52.7.	W
7:3	CC[4:0]	Command code Please see Table 52.8.	W
12:8	SID[4:0]	Subcommand ID Please see Table 52.9.	W
31:13	—	The write value should be 0.	W

Table 52.7 Encoder address

XEA2 / EA2	XEA1 / EA1	XEA0 / EA0	Encoders
0	0	0	ENC1
0	0	1	ENC2
0	1	0	ENC3
0	1	1	ENC4
1	0	0	ENC5
1	0	1	ENC6
1	1	0	ENC7
1	1	1	ENC8

Table 52.8 Command code

Frame name	CC[4:0]	Classification	Remarks
CDF0	00000b	Data request (Individual mode)	ABS full 40-bit data request
CDF1	00001b		ABS lower 24-bit data request
CDF2	00010b		ABS upper 24-bit data request
CDF3	00011b		Encoder status request
CDF4	00100b	Reserved	These commands are not supported.
CDF5	00101b		These commands are not supported.
CDF6	00110b		These commands are not supported.
CDF7	00111b		These commands are not supported.
CDF8	01000b	Operating request (Individual mode)	Status flag clear request (8 times in a row)
CDF9	01001b		Multi-rotation data clear request (8 times in a row)
CDF10	01010b		Status and multi-rotation data clear requests (8 times in a row)
CDF11	01011b		Encoder address setting I (8 times in a row)
CDF12	01100b		One-rotation data zero preset (8 times in a row)
CDF13	01101b	Memory access request (Individual mode)	Memory read request (2 frames)
CDF14	01110b		Memory write request (4 frames)
CDF15	01111b	Data request (Individual mode)	10-bit temperature data request
CDF16	10000b	ID code read	ID code read I (Individual)
CDF17	10001b		ID code read II (peer-to-peer connection)
CDF18	10010b	ID code write	ID code write I (Individual, 4 frames)
CDF19	10011b		ID code write II (peer-to-peer connection, 4 frames)
CDF20	10100b	Operating request (Individual mode)	Encoder address setting II (4 frames)
CDF21	10101b	Data request (Individual)	ABS lower 17-bit data request
CDF22	10110b	Reserved	This command is not supported.
CDF27	11011b	Data request (Individual)	ABS lower 24-bit data and status requests
CDF28	11100b	Reserved	This command is not supported.
CDF29	11101b	Data request (Individual)	ABS lower 24-bit data and temperature data requests
CDF30	11110b	Reserved	This command is not supported.
CDF31	11111b	Operating request (Individual)	A-Safety Format
CDF23 - 26	Others	No definition	

Table 52.9 Subcommand ID (1 of 2)

Frame name	SID[4:0]	Classification	Remarks
SID0	00000b	Data request (Individual transmission)	Pos1(39:0), Pos2(31:0), Diag_p(15:0), Diag_n(15:0), DB(7:0)
SID1	00001b	Reserved	
SID2	00010b	Data request (Individual transmission)	Pos1(39:0), Pos2(31:0), Diag_p(15:0), Diag_n(15:0)
SID3	00011b	Reserved	
SID4	00100b	Data request (Individual transmission)	Pos1(23:0), Pos2(31:0), Diag_p(15:0), Diag_n(15:0), DB(7:0)
SID5	00101b	Reserved	
SID6	00110b	Data request (Individual transmission)	Pos1(23:0), Pos2(31:0), Diag_p(15:0), Diag_n(15:0)
SID7	00111b	Reserved	
SID24 - 30	Command for power-on test of Safety encoder		

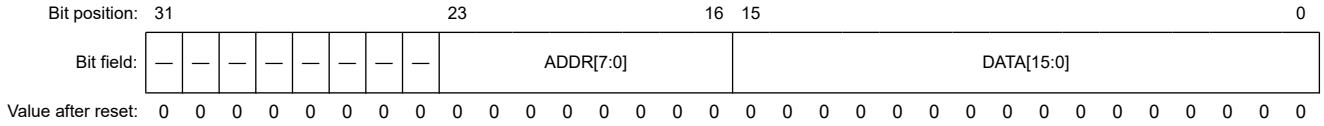
Table 52.9 Subcommand ID (2 of 2)

Frame name	SID[4:0]	Classification	Remarks
	Others	No definition	

52.3.2 ROMDATA : EEPROM Address and Write Data Register

Base address: AFMTi = 0x9020_6000 + 0x1_0000 × i (i = 0 to 11)
AFMTj = 0x9030_6000 + 0x1_0000 × (j - 12) (j = 12 to 15)

Offset address: 0x004

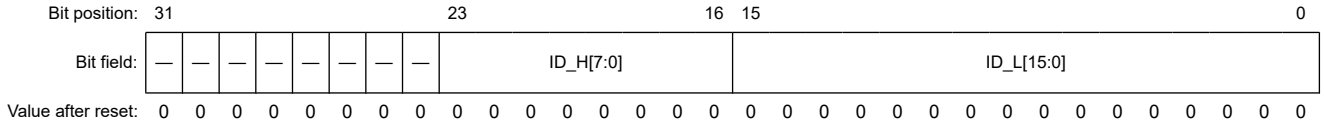


Bit	Symbol	Function	R/W
15:0	DATA[15:0]	16-bit EEPROM data When the command code is a memory writing request (CDF14), 16-bit write data value should be written in the ROMDATA register with writing in the ROMADDR register.	W
23:16	ADDR[7:0]	8-bit EEPROM address When the command code is a memory reading request (CDF13), 16-bit read address should be written in the ROMADDR register. When command code is a memory writing request (CDF14), 16-bit read address should be written as well.	W
31:24	—	The write value should be 0.	W

52.3.3 ID : ID Code Register

Base address: AFMTi = 0x9020_6000 + 0x1_0000 × i (i = 0 to 11)
AFMTj = 0x9030_6000 + 0x1_0000 × (j - 12) (j = 12 to 15)

Offset address: 0x008

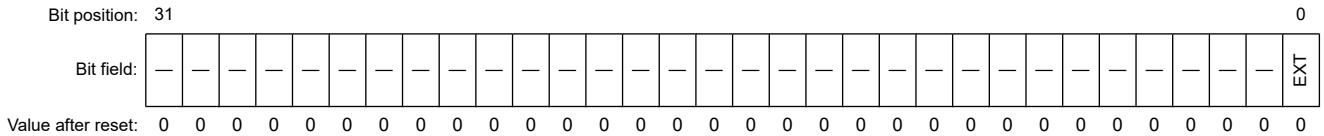


Bit	Symbol	Function	R/W
15:0	ID_L[15:0]	24-bit ID code, low word When the command code is ID code writing request (CDF18 or 19), 24-bit ID code should be written in the ID_H and ID_L bits.	W
23:16	ID_H[7:0]	24-bit ID code, high byte When the command code is ID code writing request (CDF18 or 19), 24-bit ID code should be written in ID_H and ID_L bits.	W
31:24	—	The write value should be 0.	W

52.3.4 TRGSEL : Trigger Select Register

Base address: AFMTi = 0x9020_6000 + 0x1_0000 × i (i = 0 to 11)
AFMTj = 0x9030_6000 + 0x1_0000 × (j - 12) (j = 12 to 15)

Offset address: 0x100



Bit	Symbol	Function	R/W
0	EXT	Trigger select 0: Set TRG bit in ENCnTXSEL register 1: Trigger from ELC	W
31:1	—	The write value should be 0.	W

52.3.5 TXTRG : Transmission Trigger Register

Base address: AFMTi = 0x9020_6000 + 0x1_0000 × i (i = 0 to 11)
AFMTj = 0x9030_6000 + 0x1_0000 × (j - 12) (j = 12 to 15)

Offset address: 0x104

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRG	Command data transmission trigger generation When EXT bit in ECNnTRGSEL is set to 0, transmission trigger is generated by this bit setting and command data transmission to specified Encoder starts. When EXT = 1, this bit setting is ignored. 0: Do nothing. 1: Transmission trigger is generated.	W
31:1	—	The write value should be 0.	W

52.3.6 BRSEL : Baud Rate Setting Register

Base address: AFMTi = 0x9020_6000 + 0x1_0000 × i (i = 0 to 11)
AFMTj = 0x9030_6000 + 0x1_0000 × (j - 12) (j = 12 to 15)

Offset address: 0x200

Bit position:	31																			11								4	3	2	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TM[7:0]									AS	BR[2:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	BR[2:0]	Baud rate setting The baud rate is required a value depending on the encoder used. 0 0 0: 2.5 Mbps 0 0 1: 4 Mbps 0 1 0: 6.67 Mbps 0 1 1: 8 Mbps 1 0 0: 16 Mbps Others: Setting prohibited	W
3	AS	A-Safety enable 0: Disable 1: Enable	W
11:4	TM[7:0]	t2 time setting N t2 time is set to 1 µs + N × 0.25 µs	W
31:12	—	The write value should be 0.	W

52.3.7 ENCnRXDATAm : ENCn Receive Data m Register (n = 1 to 8, m = 0 to 7)

Base address: AFMTi = 0x9020_6000 + 0x1_0000 × i (i = 0 to 11)
AFMTj = 0x9030_6000 + 0x1_0000 × (j - 12) (j = 12 to 15)

Offset address: 0x004 × m + 0x020 × (n - 1)

Bit position:	31																												0
Bit field:	DATA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Encoder read data For details, see Table 52.10 and Table 52.11.	R

The ENCnRXDATAm registers are used to receive data which encoder outputs according to the transmitted command. Note that index n is encoder category which is specified by encoder address (ENC1 to ENC8). Index m is data register number each encoder category.

Table 52.10 Receive data (except CDF31)

Command code	ENCnRXDATAm	DATA[31:0]																															
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDF0	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	ABS 40-bit full data (ABS[31:0])																															
	ENCnRXDATA2																	0	0	0	0	0	0	0	0	ABS 40-bit full data (ABS[39:32])							
CDF1	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	ABS lower 24-bit data (ABS[23:0])																							
CDF2	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1									ABS upper 24-bit data (ABS[39:16])																							
CDF3, CDF8 to CDF12	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Status code (ALM[15:0])															
CDF13, CDF14	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	8-bit EEPROM address								16-bit EEPROM data															
CDF15	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10-bit temperature data (DB[9:0])									
CDF16 to CDF19	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	24-bit ID code [23:0]																							
CDF21	ENCnRXDATA0	CA[3:1]			0	ES	EA[2:0]			0	0	0	0	0	0	0	ABS lower 17-bit data (ABS[16:0])																
CDF27	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	ABS lower 24-bit data (ABS[23:0])																							
	ENCnRXDATA2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Status code (ALM[15:0])															
CDF29	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	ABS lower 24-bit data (ABS[23:0])																							
	ENCnRXDATA2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10-bit temperature data (DB[9:0])									

Note: CA[3:0]: Communication alarm. For details, see Table 52.12.
ES[3:0]: Encoder status. For details, see Table 52.13.

CC[4:0]: Command code. For details, see Table 52.8.
EA[2:0]: Encoder address. For details, see Table 52.7.
ALM[15:0]: Status code. For details, see Table 52.14.

Table 52.11 Receive data (CDF31)

Command code	ENCnRXDATAm	DATA[31:0]																															
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID0	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[4:0]					SQD[2:0]			SID[4:0]					EA[2:0]		
	ENCnRXDATA1	Pos1[31:0]																															
	ENCnRXDATA2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DTB[7:0]								Pos1[39:32]							
	ENCnRXDATA3	Pos2[31:0]																															
	ENCnRXDATA4	Diag_N[15:0]																Diag_P[15:0]															
SID2	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[4:0]					SQD[2:0]			SID[4:0]					EA[2:0]		
	ENCnRXDATA1	Pos1[31:0]																															
	ENCnRXDATA2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Pos1[39:32]							
	ENCnRXDATA3	Pos2[31:0]																															
	ENCnRXDATA4	Diag_N[15:0]																Diag_P[15:0]															
SID4	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[4:0]					SQD[2:0]			SID[4:0]					EA[2:0]		
	ENCnRXDATA1	DTB[7:0]								Pos1[23:0]																							
	ENCnRXDATA2	Pos2[31:0]																															
	ENCnRXDATA3	Diag_N[15:0]																Diag_P[15:0]															
SID6	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[4:0]					SQD[2:0]			SID[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	Pos1[23:0]																							
	ENCnRXDATA2	Pos2[31:0]																															
	ENCnRXDATA3	Diag_N[15:0]																Diag_P[15:0]															
SID24 to SID30	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[4:0]					SQD[2:0]			SID[4:0]					EA[2:0]		
	ENCnRXDATA1	Pos1[31:0]																															
	ENCnRXDATA2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DTB[7:0]								Pos1[39:32]							
	ENCnRXDATA3	Pos2[31:0]																															
	ENCnRXDATA5	0	0	0	0	0	0	0	0	ALM[23:0]																							

Table 52.12 Communication alarm (CA[4:0]) (1 of 2)

Bit	Symbol	Function
0	CA[0]	CMD status flag When this bit is set to 1, indicates transmitted command is not matched with received command.
1	CA[1]	FORM status flag When this bit is set to 1, indicate start bit and stop bit mismatch for the field or a lack of receive data length.
2	CA[2]	SYNC status flag When this bit is set to 1, indicates receive data does not come within the designated time after command code transmission.
3	CA[3]	CRC status flag When this bit is set to 1, indicates transmitted CRC is not matched with CRC calculated from received data.

Table 52.12 Communication alarm (CA[4:0]) (2 of 2)

Bit	Symbol	Function
4	CA[4]	SQN status flag When this bit is set to 1, indicates transmitted sequence number is not matched with received sequence number. (CDF31 only)

Table 52.13 Encoder status (ES[3:0] and ES)

Bit	Symbol	Function
0	ES[0]	BUSY + MEMBUSY
1	ES[1]	BATT
2	ES[2]	OVSPD + MEMERR
3	ES[3]	STERR + PSERR + MTERR
—	ES	ORed ES[3:0]. (CDF21 only)

Note: Confirm the specification document of the encoder to be used.

Table 52.14 Status code (ALM[15:0])

Bit	Symbol	Function
0	ALM[0]	BATT
1	ALM[1]	MTERR
2	ALM[2]	0 fixed
3	ALM[3]	OVSPD
4	ALM[4]	MEMERR
5	ALM[5]	STERR
6	ALM[6]	PSERR
7	ALM[7]	BUSY
8	ALM[8]	MEMBUSY
9	ALM[9]	OVTEMP
10	ALM[10]	0 fixed
11	ALM[11]	0 fixed
12	ALM[12]	0 fixed
13	ALM[13]	0 fixed
14	ALM[14]	0 fixed
15	ALM[15]	0 fixed

Note: Confirm the specification document of the encoder to be used.

The status when the main power is turned on (250 min^{-1} or less) when the external battery is not connected and the internal backup capacitor is completely discharged is as follows:

BUSY, STERR, MEMBUSY, MEMERR, OVTEMP = 0

MTERR, BATT = 1

OVSPD, PSERR = x (indefinite value)

52.4 Operation

52.4.1 Basic Operation

After writing a command and memory data such as read address and write data, command data transmission starts by register-write or trigger input according to the TRGSEL register setting.

(1) Startup by register access

When the EXT bit in the TRGSEL register is set to 0, the register trigger is selected.

In this mode, when the TRG bit in the TXTRG register is set to 1, command data transmission starts.

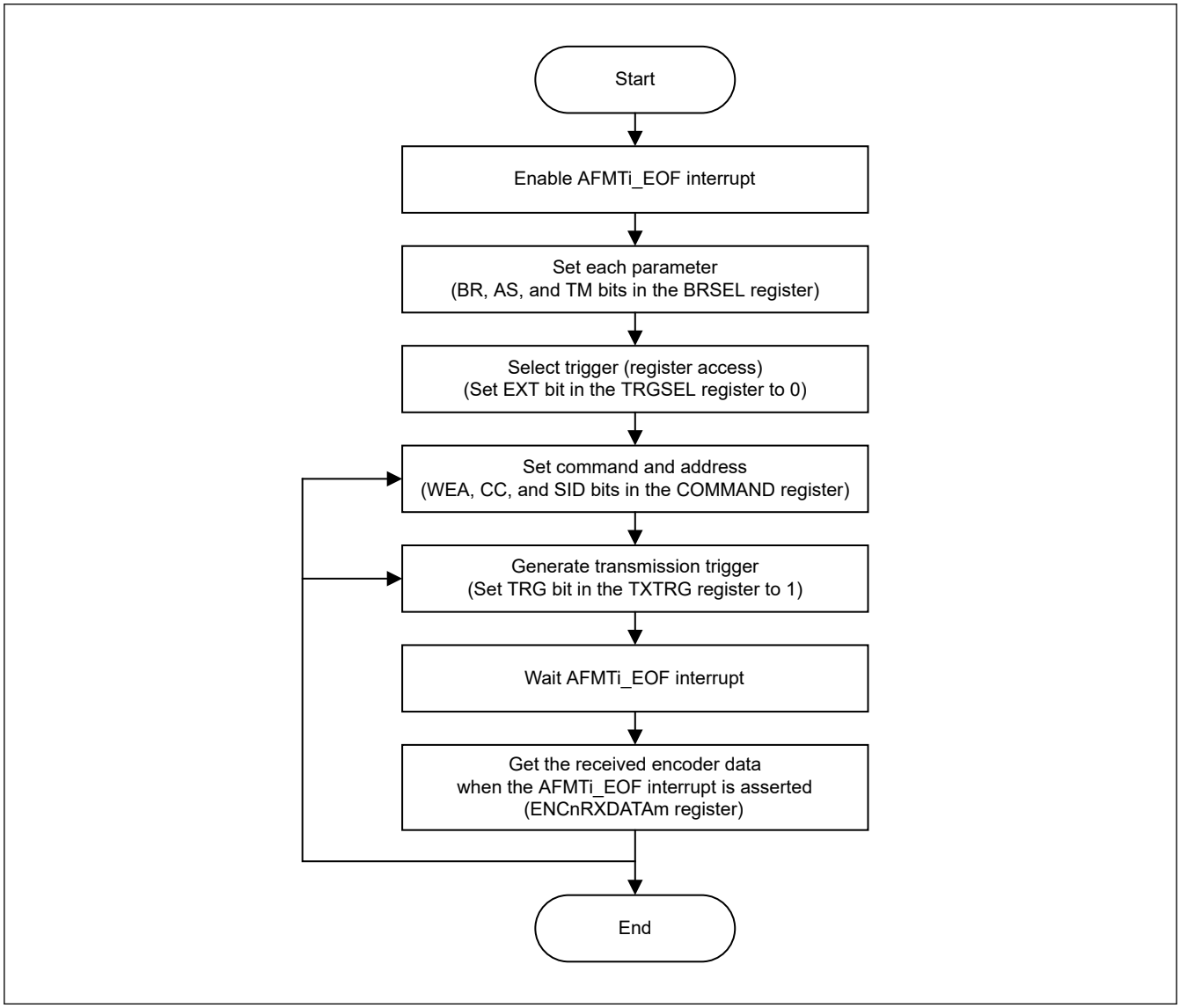


Figure 52.2 Basic operation (startup by register access)

(2) Startup by trigger input from ELC

When the EXT bit in the TRGSEL register is set to 1, the trigger from ELC is selected.

In this mode, when the trigger input is received from ELC, command data transmission starts.

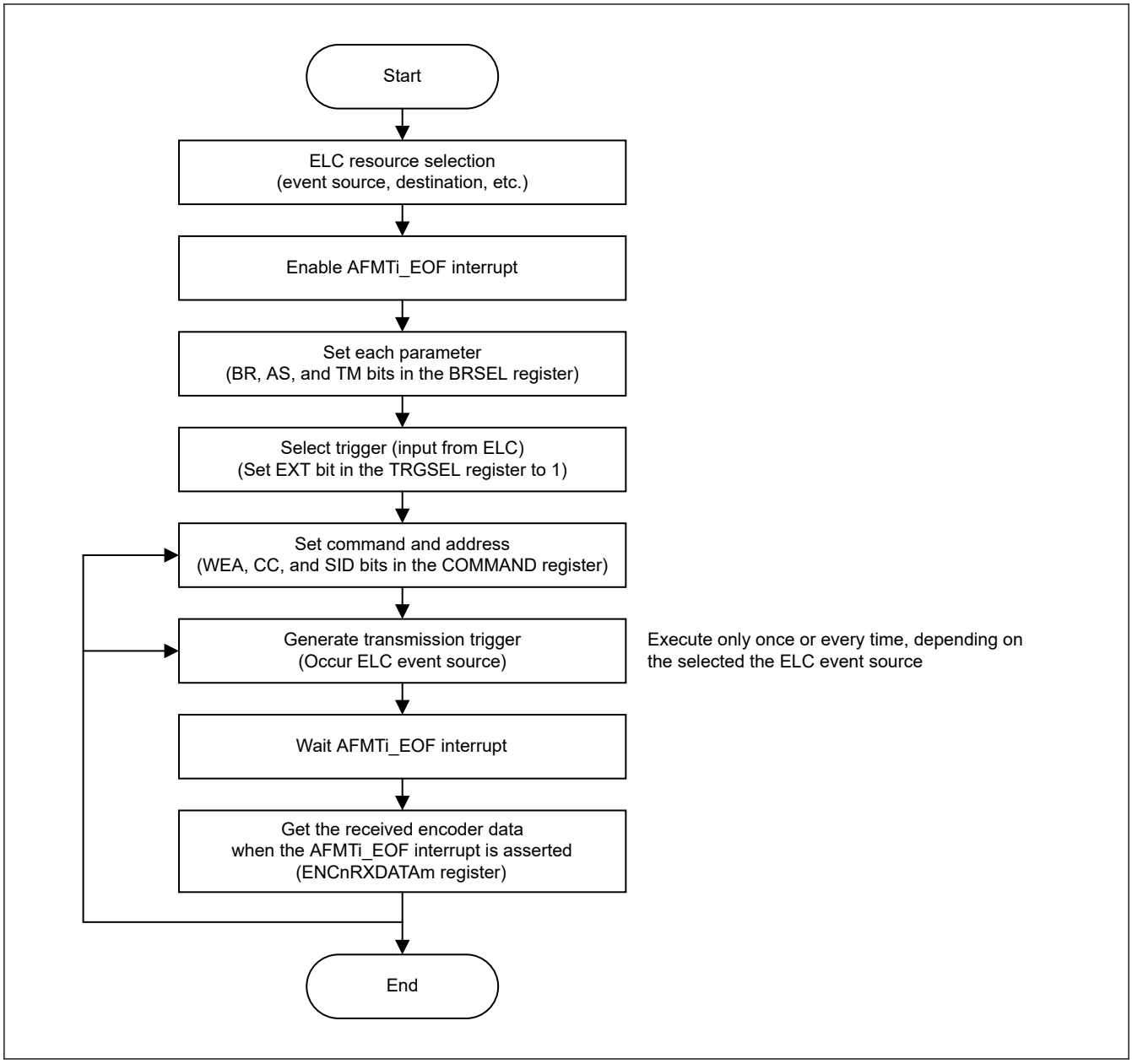


Figure 52.3 Basic operation (startup by trigger input from ELC)

A received encoder data is stored to the ENCnRXDATAm register and the AFMTi_EOF interrupt is asserted when communication is completed. The AFMTi_EOF interrupt is canceled when the first output data is read out.

The baud rate is required a value depending on the encoder used.

52.4.2 Command Configuration

(1) One frame command (CDF0 to CDF12, CDF15 to CDF17, CDF21, CDF22, and CDF27 to CDF31)
Position data request / various clear request / EC address setting / one-rotate data zero preset

Command data

(2) Two frame command (CDF13)
EEPROM data read

Command data	Read address (8 bits)

(2) Four frame command (CDF14, and CDF18 to CDF19)
EEPROM data write

Command data	Write data (lower 8 bits)	Write data (upper 8 bits)	Read address (8 bits)

Figure 52.4 Command configuration

52.4.3 Interrupts

52.4.3.1 Communication Completion Interrupt

When a received encoder data is set to the ENCnRXDATAm (n = 1 to 8, m = 0 to 7) registers and the sequence of receiving and sending operations is completed, the communication completion interrupt, AFMTi_EOF (i = 0 to 15), is asserted. When the first output data readout is executed, the communication completion interrupt is canceled.

The TRGSEL register is reset when the communication completion interrupt happens.

52.4.3.2 Time Out Error Interrupt

When data is not returned within the predetermined time for a request, the time out error interrupt, AFMTi_TMOUT (i = 0 to 15), is asserted. When next communication start, the time out error interrupt is canceled.

52.4.4 Individual Transmission Mode

Figure 52.5 shows the serial data transmission/reception timing for the individual mode.

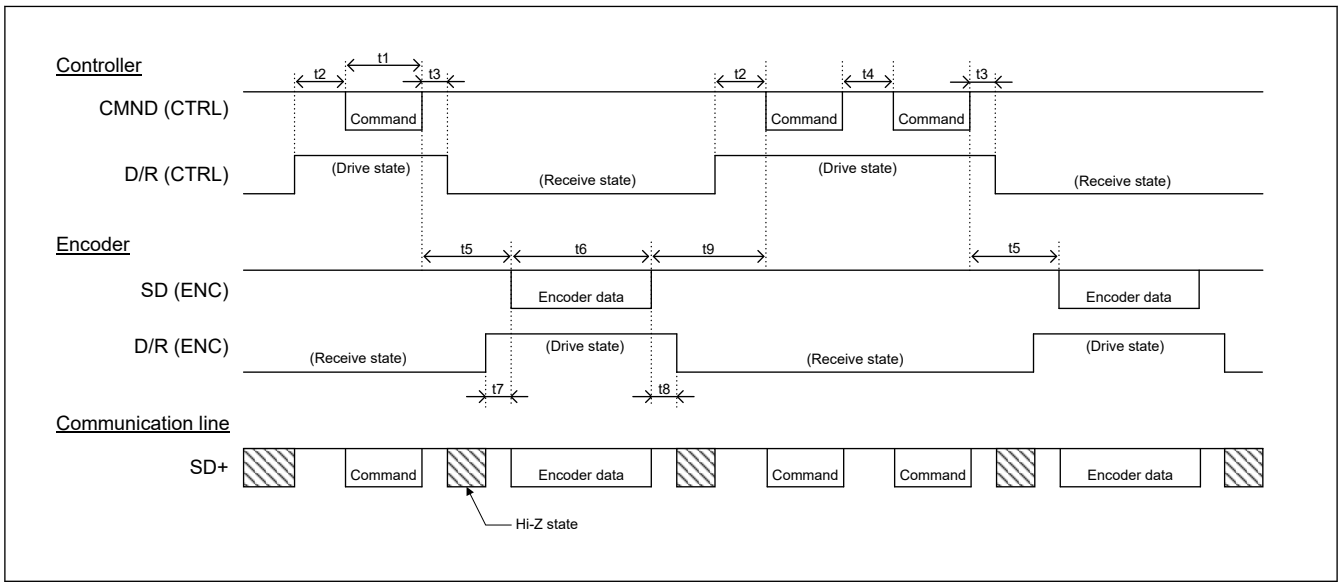


Figure 52.5 Individual transmission mode

Table 52.15 Timing of individual transmission mode

Time	2.5 Mbps	4 Mbps	6.67 Mbps	8 Mbps	16 Mbps	Remarks
t1	7.2 µs	4.5 µs	2.7 µs	2.25 µs	1.125 µs	—
t2	1 µs (MIN)	1 µs (MIN)	1 µs (MIN)	1 µs (MIN)	1 µs (MIN)	Reference value
t3	1.5 µs	1.0 µs	1.0 µs	0.7 µs	0.7 µs	Reference value
t4	1 µs (MIN)	1 µs (MIN)	1 µs (MIN)	1 µs (MIN)	1 µs (MIN)	—
t5	3 µs	2 µs	2 µs	1.5 µs	1.5 µs	—
t6	7.2 µs × n	4.5 µs × n	2.7 µs × n	2.25 µs × n	1.125 µs × n	n = 2 to 6*2
t7	387.5 ns	237.5 ns	137.5 ns	112.5 ns	50 ns	—
t8	1. 2 µs	0.8 µs	0.8 µs	0.4 µs	0.4 µs	—
t9	3.5 µs (MIN)	2.5 µs (MIN)	2.5 µs (MIN)	2.2 µs (MIN)	2.2 µs (MIN)	—

Note 1. Delay of transmission path and transceiver/receiver is assumed zero.
Note 2. Data transmission time is different dependent on commands.
Note 3. t2 is changeable via register setting.

52.5 Usage Note

52.5.1 Multiple Transmission Mode

This LSI does not support multiple transmission mode (Command: CDF4 to CDF7, CDF22, CDF28, CDF30, CDF31. Subcommand for CDF31: SID1, SID3, SID5 and SID7). Individual transmission mode is only available.

53. Encoder Divided-Output (ENCOUT) Module

53.1 Overview

The encoder divided-output (ENCOUT) module of this MPU generates Phase A, Phase B, and Phase Z (index) output signals. The waveforms are similar to the output signals from an incremental encoder, and reflect the positional information set in registers. Especially, by setting the results of reading positional information from an incremental or absolute encoder in the registers of ENCOUT, these signals can be converted to the desired Phase A, Phase B, and Phase Z output signals.

Table 53.1 describes the ENCOUT specifications and Figure 53.1 shows a block diagram of the ENCOUT module.

Table 53.1 ENCOUT specifications

Item	Description
Number of units	1 unit
Operation clock	20 MHz or 80 MHz selectable
Functions	• Number of edges per rotation: 16 to 65,536 (a multiple of 4) • Carrier period: 350 ns to 3,276.75 µs (20 MHz), 88 ns to 819.188 µs (80 MHz) • Changing the active sense of Phase B • Changing the phase and width of Phase Z
Event link	Support for the input of carrier period event signals from ELC
Module-stop function	Module-stop state can be set to reduce power consumption

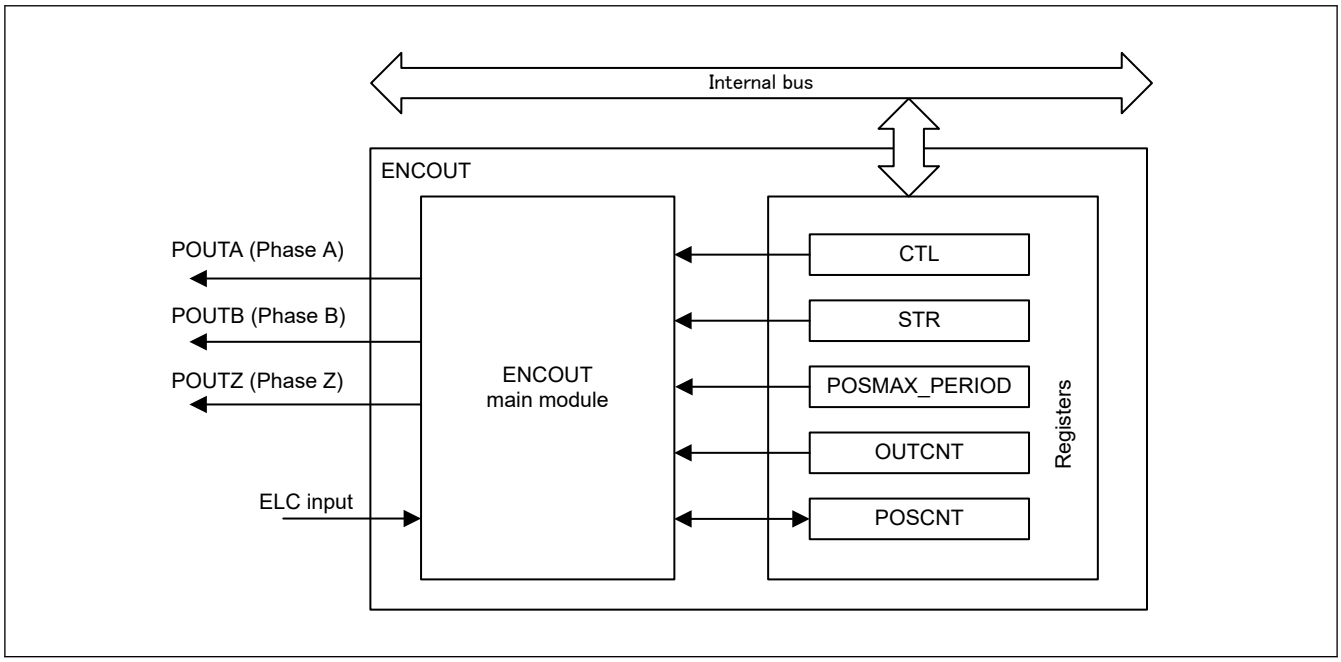
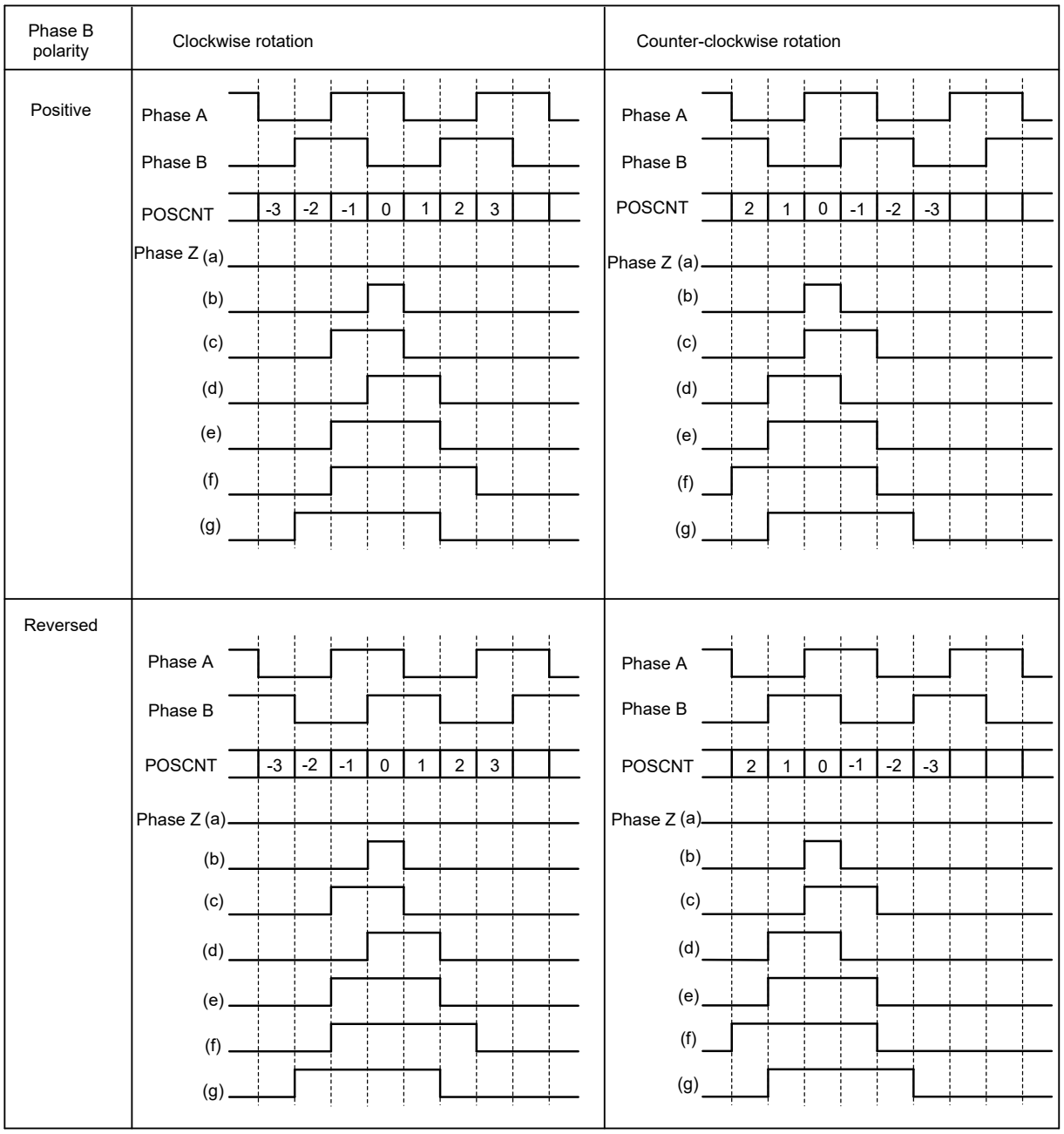


Figure 53.1 ENCOUT block diagram

Table 53.2 lists the input/output pins of the ENCOUT module.

Table 53.2 ENCOUT pin configuration

Pin name	I/O	Function
POUTA	Output	Phase A output pin
POUTB	Output	Phase B output pin
POUTZ	Output	Phase Z output pin



Note: Each output waveform of Phase Z is based on the following settings:
(a) Disable Phase Z output; CTL.ZW = 0
(b) Pulse output when current position, POSCNT, is 0; CTL.ZW = 1
(c) Pulse output when current position, POSCNT, is −1 and 0; CTL.ZW = 2 and CTL.ZS = 0
(d) Pulse output when current position, POSCNT, is 0 and 1; CTL.ZW = 2 and CTL.ZS = 1
(e) Pulse output when current position, POSCNT, is −1, 0, and 1; CTL.ZW = 3
(f) Pulse output when current position, POSCNT, is −1, 0, 1, and 2; CTL.ZW = 4 and CTL.ZS = 0
(g) Pulse output when current position, POSCNT, is −2, −1, 0, and 1; CTL.ZW = 4 and CTL.ZS = 1

Note: Negative values for current position (POSCNT).
POSCNT = −1: POSMAX setting value
POSCNT = −2: POSMAX setting value − 1

Figure 53.2 Phase A, Phase B, and Phase Z output waveforms

53.2 Register Map

Table 53.3 ENCOUT register map

Address	Register symbol	Register name	Write protection
0x902C_0000	CTL	Control Register	—
0x902C_0004	STR	Start Register	—
0x902C_000C	POSMAX_PERIOD	Maximum Position and Carrier Period Register	—
0x902C_0010	OUTCNT	Output Count Register	—
0x902C_0014	POSCNT	Position Count Register	—

Table 53.4 ENCOUT related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	—	MSTPCRL.MSTPCRL00	SLVACCCTL7.LLPP_SL*1

Note 1. Access from Cortex-R52 CPU0 and CPU1 is not protected by TrustZone. This slave access control is applied to access from other masters.

53.3 Register Descriptions

53.3.1 CTL : Control Register

Base address: ENCOUT = 0x902C_0000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ZS	ZW[2:0]			POL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	POL	Polarity of Phase B Set this bit when STR.ENCE = 0. The bit setting is immediately reflected to Phase B output. 0: Positive phase 1: Reversed phase	R/W
3:1	ZW[2:0]	Pulse Width of Phase Z*1 0x0: Disable Phase Z output 0x1: Pulse output when current position, POSCNT, is 0 0x2: Pulse output when current position, POSCNT, is -1 and 0 (sync with Phase A) Pulse output when current position, POSCNT, is 0 and 1 (sync with Phase B) 0x3: Pulse output when current position, POSCNT, is -1, 0 and 1 0x4: Pulse output when current position, POSCNT, is -1, 0, 1 and 2 (sync with Phase A) Pulse output when current position, POSCNT, is -2, -1, 0 and 1 (sync with Phase B) Others: Disable Phase Z output	R/W
4	ZS	Synchronization phase of Phase Z 0: Phase A 1: Phase B	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

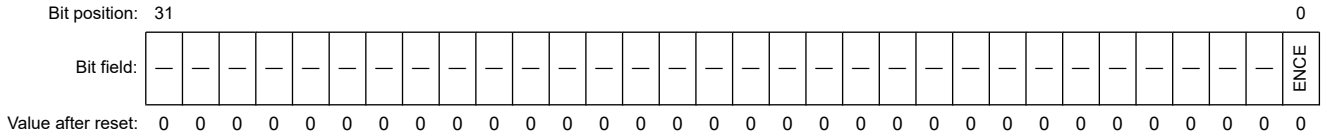
Note 1. Negative values for current position (POSCNT).
POSCNT = −1: POSMAX setting value
POSCNT = −2: POSMAX setting value − 1

The CTL register is used to set ENCOUT operation. This register should be set when STR.ENCE = 0. When STR.ENCE = 1, the write operation is disregarded.

53.3.2 STR : Start Register

Base address: ENCOUT = 0x902C_0000

Offset address: 0x004



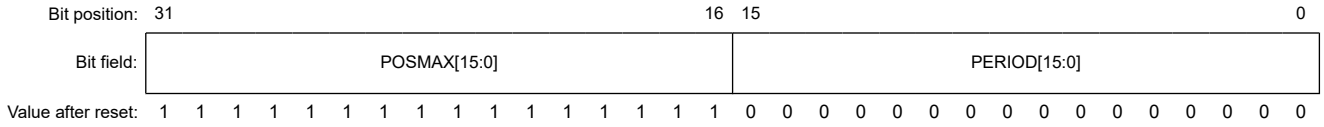
Bit	Symbol	Function	R/W
0	ENCE	ENCOUT Operation Start When the operation is stopped, the outputs of Phase A and Phase B correspond to POSCNT register, and Phase Z output is 0. 0: ENCOUT operation stop 1: ENCOUT operation start	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The STR register is used to start/stop ENCOUT operation.

53.3.3 POSMAX_PERIOD : Maximum Position and Carrier Period Register

Base address: ENCOUT = 0x902C_0000

Offset address: 0x00C



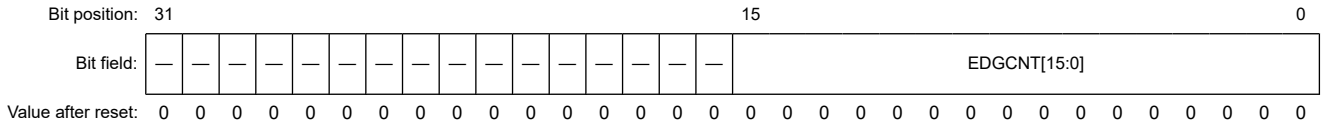
Bit	Symbol	Function	R/W
15:0	PERIOD[15:0]	Carrier Period Setting Carrier period length should be set in units of operation clock period (50 ns for 20 MHz or 12.5 ns for 80 MHz) according to the event input period from ELC. Settable value is from 7 (0x0007) to 65535 (0xFFFF). If event period from ELC cannot be divided evenly by operation clock period, the value should be set to ("event period from ELC" / "operation clock period") + 1 (round up the decimal point).	R/W
31:16	POSMAX[15:0]	Maximum Position Setting Maximum position value (number of edges) for one revolution. The maximum position value is set a multiple of 4. If maximum position value is 4 × N, this value should be set to 4 × N – 1. Settable value is 15 (0x000F), 19 (0x0013), 23 (0x0017), 27 (0x001B), 31 (0x001F), …, 65531 (0xFFFB), 65535 (0xFFFF).	R/W

The POSMAX_PERIOD register is used to set the maximum position value and carrier period. This register should be set when STR.ENCE = 0. When STR.ENCE = 1, the write operation is ignored.

53.3.4 OUTCNT : Output Count Register

Base address: ENCOUT = 0x902C_0000

Offset address: 0x010



Bit	Symbol	Function	R/W
15:0	EDGCNT[15:0]	Edge Count Setting Set the number of edges for Phase A and Phase B in the next carrier period. Set in two's complement representation, negative values indicate counter-clockwise rotation. Settable value is from -32768 (0x8000) to 32767 (0x7FFF). This value should be set to less than or equal to the setting value in PERIOD[15:0]. The update interval of edge count can be calculated from (PERIOD[15:0] / EDGCNT[15:0]) × operation clock period (50 ns for 20 MHz or 12.5 ns for 80 MHz). If PERIOD[15:0] / EDGCNT[15:0] cannot be divided evenly, the EDGCNT[15:0] is updated to disperse the error over the carrier period. See section 53.4.6. Uniformization of Waveform Output Accumulation Error.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The OUTCNT register is used to set the number of edges for Phase A and Phase B. When STR.ENCE = 1, the value should be set before an event occurs for the next carrier period.

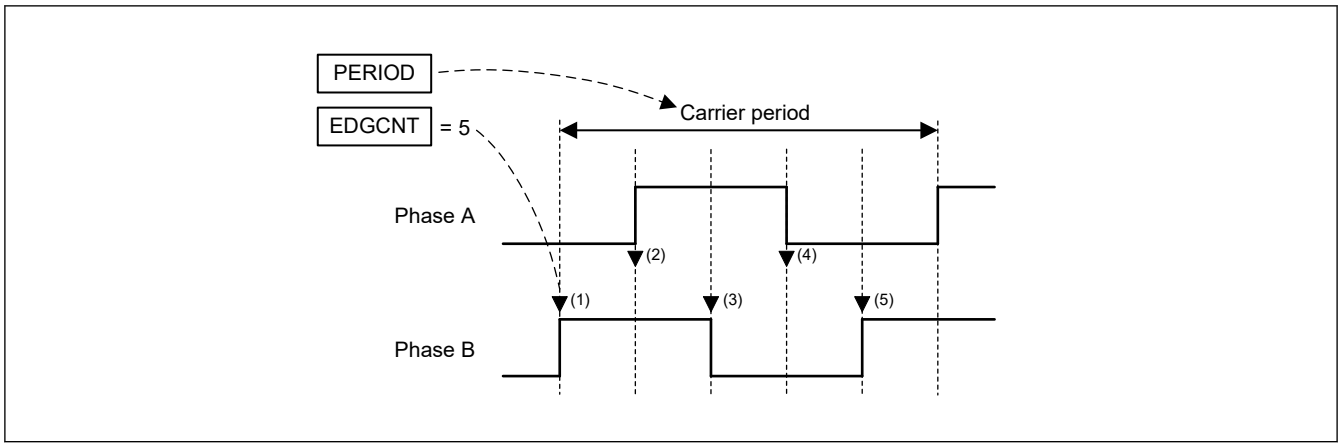


Figure 53.3 EDGCNT (clockwise rotation) and PERIOD setting sample

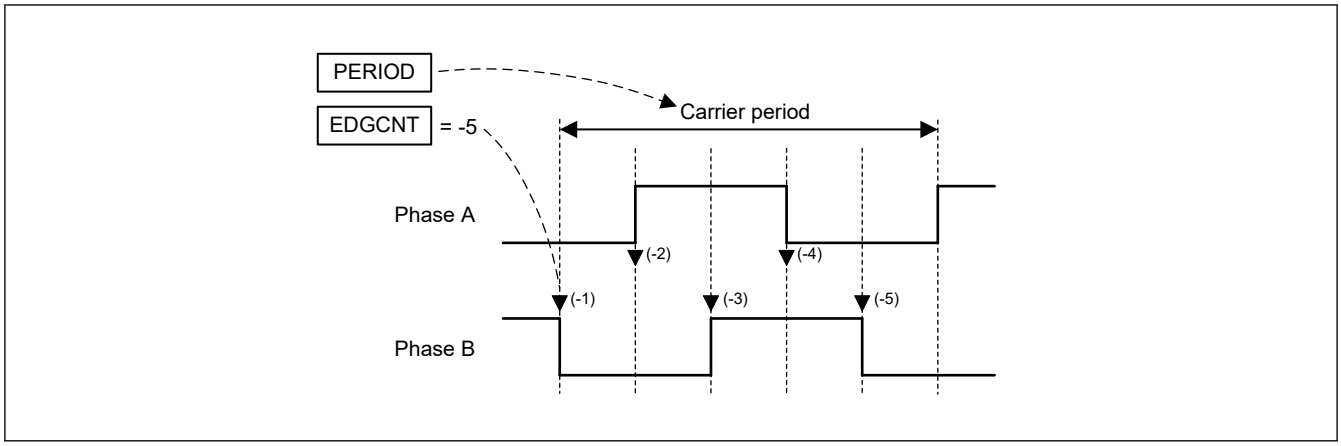


Figure 53.4 EDGCNT (counter-clockwise rotation) and PERIOD setting sample

53.3.5 POSCNT : Position Count Register

Base address: ENCOUT = 0x902C_0000

Offset address: 0x014

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	POSCNT[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	POSCNT[15:0]	Current Position Set a current position value. Settable value is from 0 to value in POSMAX[15:0]. After starting ENCOUT operation, the value is updated according to the position.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The POSCNT register is used to set a current position value and check the current position. This register should be written when STR.ENCE = 0. When STR.ENCE = 1, the value is updated according to the position associated with Phase A and Phase B output.

53.4 Operation

ENCOUT generates Phase A/B and Phase Z outputs based on the register settings. The carrier period of Phase A/B and Phase Z outputs are generated by input from ELC.

Phase A/B and Phase Z output formats are shown in Figure 53.2. And Phase Z output is generated by matching POSCNT register value according to Table 53.6.

53.4.1 Phase A and Phase B Output Format

Phase A and Phase B output are represented by the four patterns shown in the table below according to the value of POSCNT[1:0]. The patterns are shifted right with the value of POSCNT[1:0] as Phase A pattern = 1001b (0x9) and Phase B pattern = 1100b (0xC), the first bit will be the output value of Phase A and Phase B. When reverse phase output (CTL.POL = 1), Phase B output pattern is reversed.

Table 53.5 Phase A and Phase B output pattern

POSCNT[1:0] value	3	2	1	0
Phase A output	1	0	0	1
Phase B output (CTL.POL = 0: Positive)	1	1	0	0
Phase B output (CTL.POL = 1: Reversed)	0	0	1	1

53.4.2 Phase Z Output Format

Phase Z output is generated by matching POSCNT register value to the table values below.

Table 53.6 Phase Z output pattern

Phase Z output width (CTL.ZW[2:0])	POSCNT[15:0] value when Phase Z output is 1
0	(Disable Phase Z output)
1	0
2 (Synchronize with Phase A)	-1, 0
2 (Synchronize with Phase B)	0, 1
3	-1, 0, 1
4 (Synchronize with Phase A)	-1, 0, 1, 2
4 (Synchronize with Phase B)	-2, -1, 0, 1

Note: Negative values for current position (POSCNT)
POSCNT = −1: POSMAX setting value
POSCNT = −2: POSMAX setting value − 1

53.4.3 Carrier Period Generation by Events from ELC

Event inputs from ELC are recognized as a carrier period. Generate a carrier period with a timer, etc., and input it using ELC.

53.4.4 PERIOD Setting Value 1

If the ELC event period is not divisible by the number of operating clocks in POSMAX_PERIOD.PERIOD configuration, set the integer value + 1 to POSMAX_PERIOD.PERIOD.

The figure below is an operation example. The operation clock period is 50 ns (20 MHz), the carrier period is 1000 ns (PERIOD = 20), ELC event period is 980 ns. ELC event is checked on every clock, and if the cumulative error between the carrier period and ELC event period exceeds 50 ns, an edge update is performed by 50 ns shorter than the 1000 ns carrier period and the carrier period counter is reset. Therefore, the cumulative error is less than 50 ns (one clock) as shown in the following table.

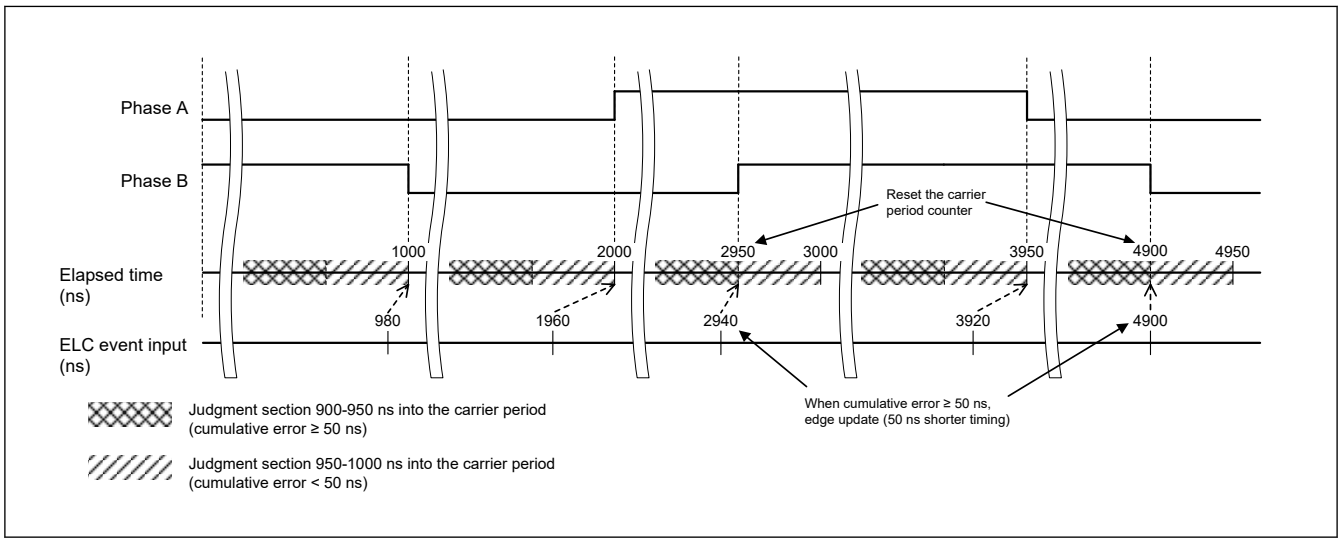


Figure 53.5 Operation example in the case ELC event period is 980 ns

Table 53.7 Actual carrier period in the case ELC event period is 980 ns

ELC event period	Actual carrier period	Remarks
980	1000	—
1960	2000	—
2940	2950	If the cumulative error is ≥ 50 ns, the edge is updated by 50 ns earlier.
3920	3950	—
4900	4900	If the cumulative error is ≥ 50 ns, the edge is updated by 50 ns earlier.
5880	5900	—
6860	6900	—
7840	7850	If the cumulative error is ≥ 50 ns, the edge is updated by 50 ns earlier.
8820	8850	—
9800	9800	If the cumulative error is ≥ 50 ns, the edge is updated by 50 ns earlier.

53.4.5 PERIOD Setting Value 2

The minimum setting for POSMAX_PERIOD.PERIOD is 7. The carrier period takes the edge of the ELC signal asynchronously at the operation clock (20 MHz or 80 MHz). The next ELC input signal cannot be accepted from the edge detection to the completion of operation clock synchronization. Therefore, the minimum setting value of PERIOD is 7 or more, so that the ELC signal can be reliably detected. Operation cannot be guaranteed when the setting value is set to 6 or less.

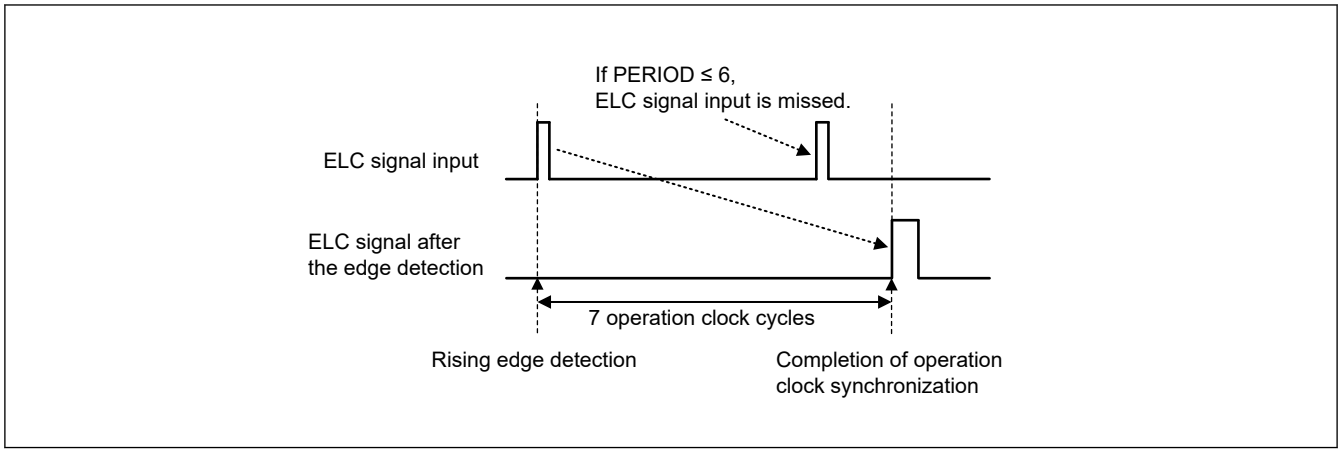


Figure 53.6 Example of the operation when PERIOD value is 6 or less

53.4.6 Uniformization of Waveform Output Accumulation Error

The edge-to-edge operation cycle truncates the decimal point of the calculated value, resulting in an error at the end of the carrier signal.

Counter for waveform update = carrier period / operation clock period / number of edges (OUTCNT.EDGCNT)

Example of the error accumulation with waveform output

The following figure is an operation example. The carrier period = 5000 ns, the operation clock period = 20 MHz (50 ns), the number of edges = 7. In this case, the error occurs finally because the decimal point is truncated by the edge interval calculation as follows.

Edge interval = 5000 / 50 / 7 = 14.2857 (the decimal point is truncated, and it results in 14)

Error = 5000 / 50 – 14 × 7 = 2

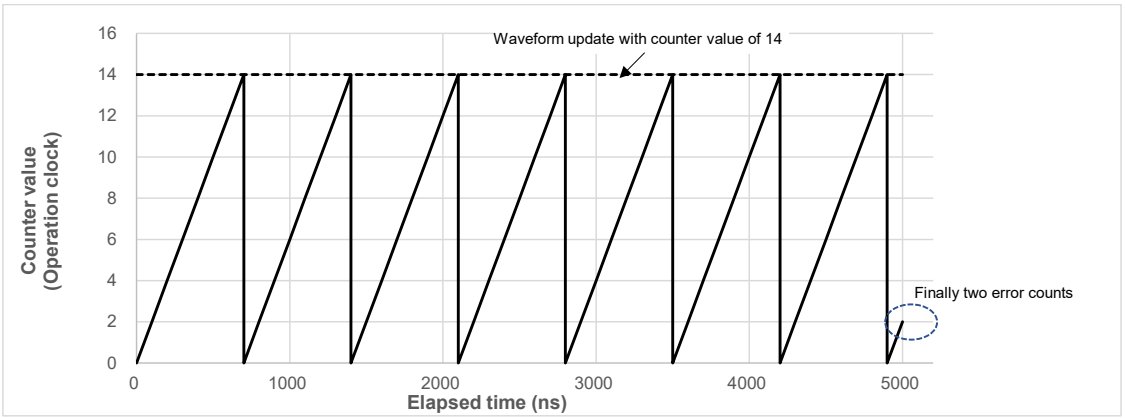


Figure 53.7 Waveform output and counter (accumulation error)

Therefore, the settings of the waveform output are changed as follows, and the error of the output waveform is uniformed throughout the carrier period.

- Add the "number of edges" at each operating clock
- The total counts of the carrier period = the number of edges × the number of operating clocks during the carrier period
- Update the waveform when the counter value of the operating clock exceeds the "number of clocks during the carrier period"

Example of the waveform output with the error accumulation measure

The following figure is an operation example. The carrier period = 5000 ns, the operation clock period = 20 MHz (50 ns), the number of edges = 7. The number of operating clocks during the carrier period is 100, counting by 7 per clock cycle. When the counter exceeds 100, it is subtracted by 100 and the waveform is updated. The counter increases by 700 in one carrier period, and the waveform changes exactly seven times in one carrier period.

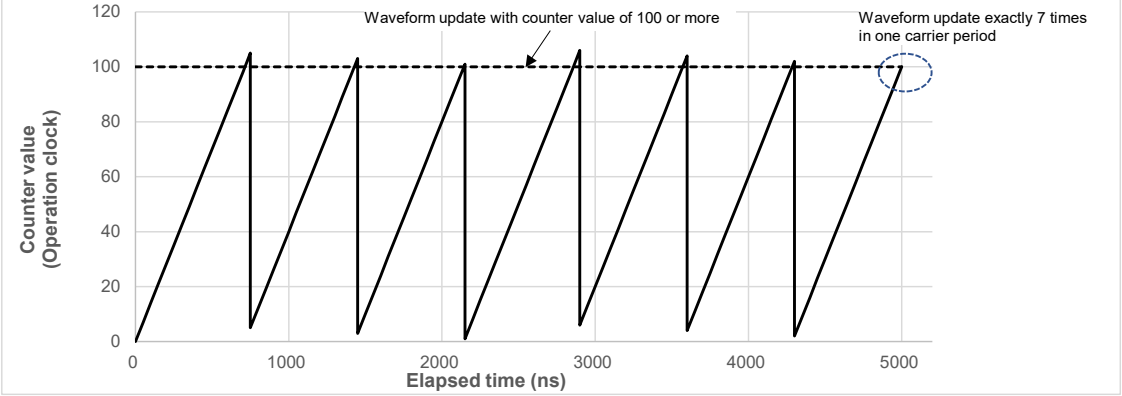


Figure 53.8 Waveform output and counter (with the error accumulation measure)

In the case of this setting, once the "number of clocks during the carrier period" is initially set, the waveform will be output by simply updating the "number of edges" setting during operation. In addition, since the assumption of the carrier period is 50000 to 3276000 (ns), the "number of clocks for carrier period" can be handled within the range of 2 bytes data.

(Carrier period: 3276000 ns) / (Operation clock period: 50 ns) = 65520 (0xFFF0)

Maximum carrier period setting: (POSMAX_PERIOD.PERIOD setting: 65535 (0xFFFF)) × 50 = 3276750 [ns]

53.4.7 Setting Procedure

53.4.7.1 System Configuration

The following figure shows a system configuration example that uses ENCOUT to convert the position information of an incremental encoder or an absolute encoder into Phase A/B and Phase Z outputs. The CPU obtains position information from the encoder for each period of the current loop and position control loop notified by the timer module. Then by updating ENCOUT setting periodically due to the difference from the previous position information, Phase A/B and Phase Z output is generated. The timer module events that generate periods of the current loop or position control loop should be input to ENCOUT through the ELC.

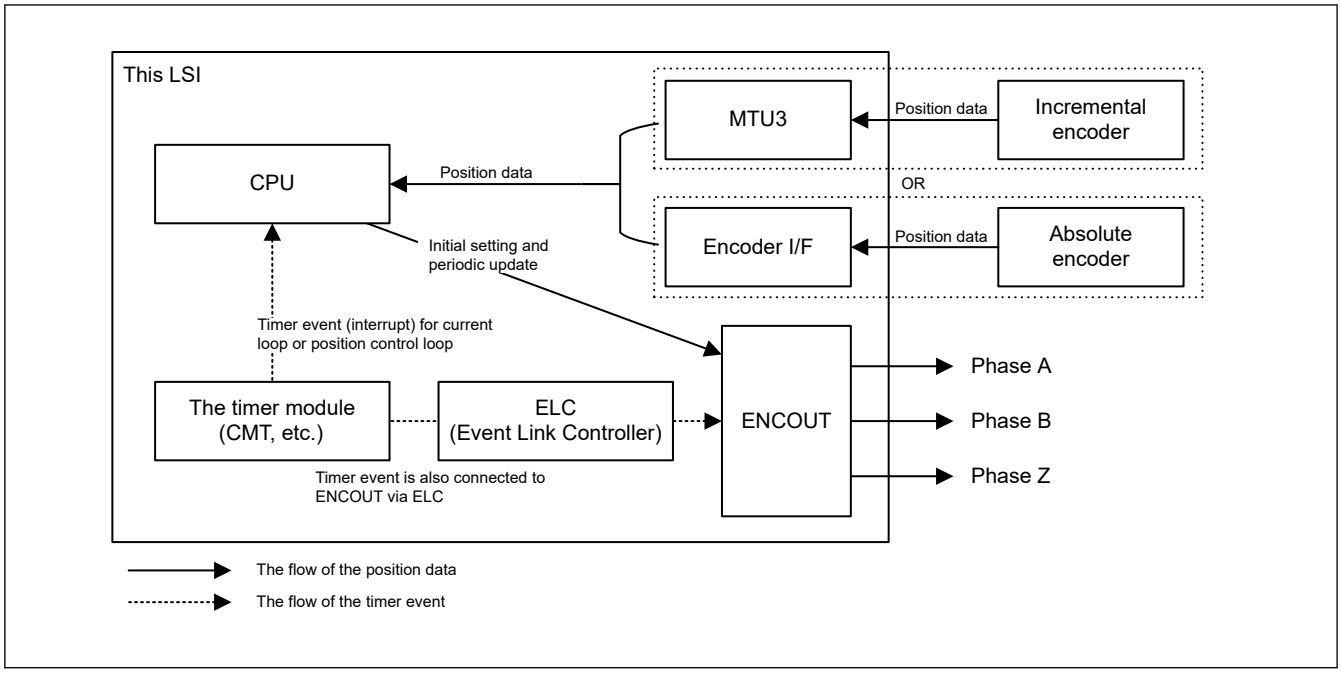


Figure 53.9 Example of phase output system configuration

53.4.7.2 Initialization

The following figure shows the initialization flow of ENCOUT.

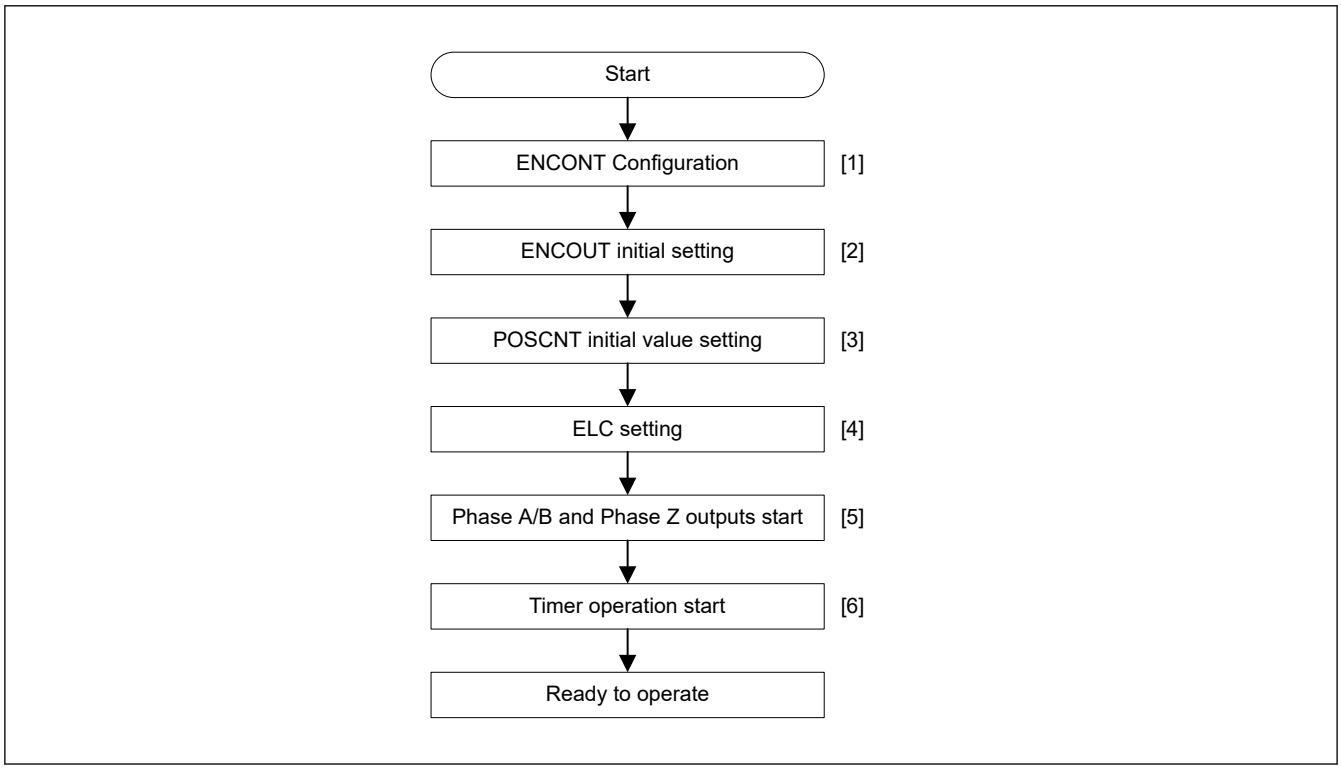


Figure 53.10 ENCOUT initialization flow

[1] ENCOUT configuration

Set the output pin assignment using the I/O port registers. Set the operation clock frequency by ENCOUTCLK bit in System Clock Control Register 4 (SCKCR4). Release from the module-stop state by Module Stop Control Register.

[2] ENCOUT initial setting

Initialize CTL, POSMAX_PERIOD, and OUTCNT registers. Set OUTCNT register to 0. Set the value of POSMAX_PERIOD.PERIOD to match the event input period from the ELC. If the event period from ELC is not divisible by operation clock period (50 ns or 12.5 ns), the value should be set to ["event period from ELC" / "operation clock period" + 1]. (Round up the decimal point)

Example 1: Operation clock period is 50 ns. The number of edges per revolution is 100, Phase B output is positive, and the carrier period is 500 (ns).
POSMAX_PERIOD.POSMAX = 99 (4 × 25 − 1)
POSMAX_PERIOD.PERIOD = 10 (500 / 50)
CTL = 0, OUTCNT = 0

Example 2: Operation clock period is 50 ns. The number of edges per revolution is 100, Phase B output is positive, and the carrier period is 520 (ns).
POSMAX_PERIOD.POSMAX = 99 (4 × 25 − 1)
POSMAX_PERIOD.PERIOD = 11 (520 / 50 + 1)
CTL = 0, OUTCNT = 0

[3] POSCNT initial value setting

Set the POSCNT register to the initial position in the range from 0 to POSMAX_PERIOD.POSMAX of an absolute encoder or an incremental encoder.

The position values = [Encoder position value] × [Number of edges per rotation] / [Encoder position value resolution] (truncate the decimal point).

Example: The number of edges per revolution is 100, encoder position value resolution is 20 bits (1048576), and initial position value is 943719.
POSCNT = 90 (943719 × 100 / 1048576)

[4] ELC setting

The event from the timer is input to ENCOUT by the setting of ELC registers.

[5] Phase A/B and Phase Z outputs start

Set the STR.ENCE bit to 1 to start Phase A/B and Phase Z output.

[6] Timer operation start

Operate a timer to generate a carrier period event.

53.4.7.3 Main Processing

The following figure shows the main processing flow of ENCOUT. This flow should be performed for each carrier period.

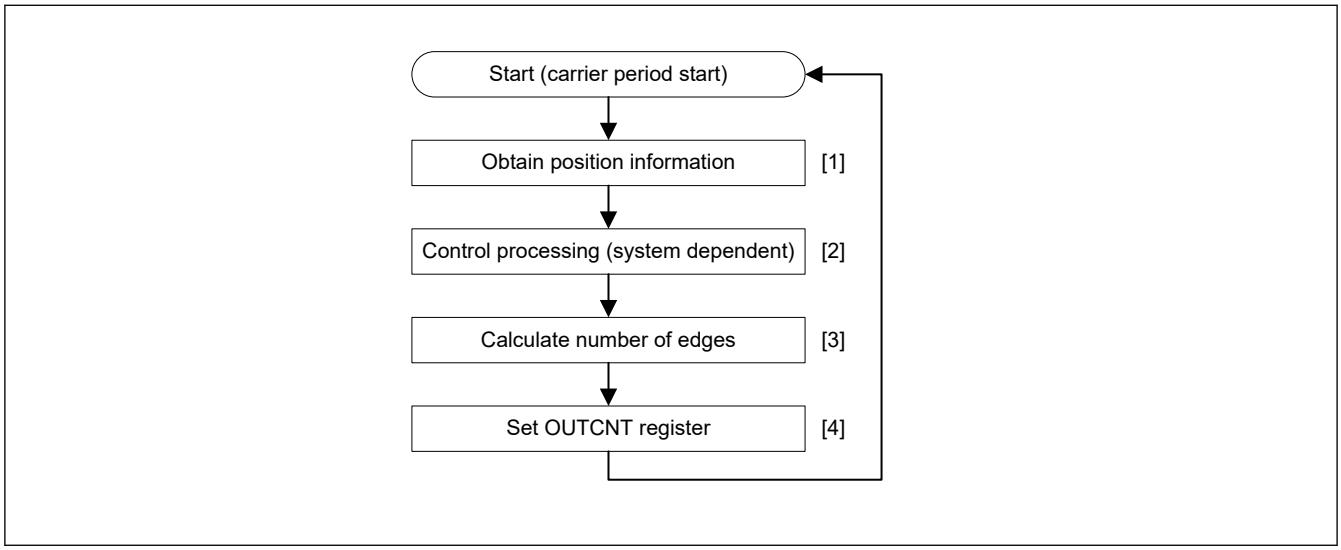


Figure 53.11 ENCOUT main processing flow

[1] Obtain position information

Obtain position information from the absolute or incremental encoders.

[2] Control processing (system dependent)

Performs motor control processing, etc. based on position information. This process is system dependent.

[3] Calculate number of edges

Calculates the setting value for OUTCNT.EDGCNT.

Current position value with the maximum position as POSMAX = [Position value obtained in step [1]] × [Number of edges per rotation] / [Encoder position value resolution] (truncate the decimal point)

Setting value of OUTCNT.EDGCNT = current position − previous position (the maximum position as POSMAX)

However, if the absolute value of above is greater than half of the number of edges per revolution:

- When setting value > 0, setting value − [Number of edges per rotation]
- When setting value < 0, setting value + [Number of edges per rotation]

Example: In the following cases:
Encoder position value = 20972, encoder position value resolution = 20 bits (1048576), number of edges per rotation = 100, previous position (the maximum position as POSMAX) = 95, carrier period = 100 µs
Current position (the maximum position as POSMAX) = 20972 × 100 / 1048576 = 2
Setting value of OUTCNT.EDGCNT = 2 − 95 = −93
This absolute value is greater than half of the number of edges per revolution (93 > 50).
Setting value is (−93) < 0, setting value + [Number of edges per rotation] = −93 + 100 = 7.

[4] Set OUTCNT register

Set OUTCNT.EDGCNT to the value calculated in step [3].

53.4.8 Setting and Output Examples

The following figure shows an example of Phase A/B and Phase Z output waveforms that correspond to the ENCOUT settings in case operation clock period is 50 ns.

This is an example when the carrier period is 100 µs, the number of edges per revolution is 100, Phase B is the positive phase, and initial position value is 90 (the maximum position as POSMAX). This is an example of a case the position transitions to 95, 2, and 98.

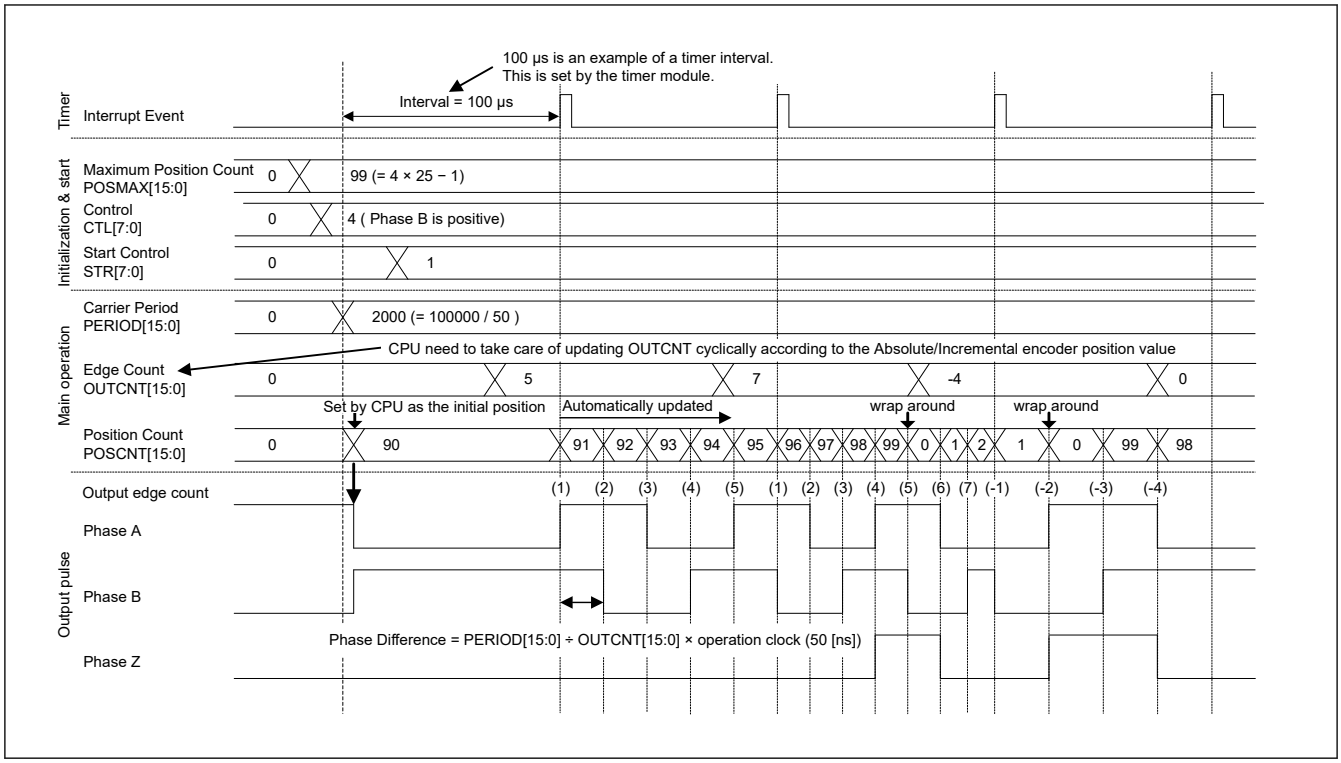


Figure 53.12 Example of Phase A/B and Phase Z output waveforms by ENCOUT setting value

54. LCD Controller (LCDC)

54.1 Overview

This chapter describes the features of the LCDC unit of this LSI.

This unit is LCD controller, it is composed of Frame Compression Processor (FCPVD), Video Signal Processor (VSPD), and Display Unit (DU).

The following is key features of this unit.

- FCPVD
  - Supports out-of-order for the whole outstanding transactions
  - Read linear addressing image data
  - Read display list data
  - Write image data
- VSPD
  - Supports various data formats and conversion
    - Supports YCbCr444/422/420, RGB, α RGB, α plane
    - Color space conversion and changes to the number of colors by dithering
    - Color keying
    - Supports combination between pixel alpha and global alpha
    - Supports generating pre-multiplied alpha
  - Video processing
    - Blending of two picture layers and raster operations (ROPs)
    - Clipping
    - 1D look up table
    - Vertical flipping in case of output to memory
  - Direct connection to display module
    - Supports up to 1920 pixels in horizontal direction
    - Writing back image data which is transferred to Display Unit (DU) to memory
- DU
  - Supports Display Parallel Interface (DPI)
  - Display timing master
    - Generates video timings (Front porch, Back porch, Sync active, Active video area)
    - Selects the polarity of output DCLK, HSYNC, VSYNC, and DE
  - Supports Progressive (Non-interlace)
  - Not supports Interlace
  - Input data format (from VSPD): RGB888, RGB666 (not supports dithering of RGB565)
  - Output data format: same as Input data format
  - Supports up to 100 MHz of the pixel clock frequency
  - Examples of the supported resolutions is below:
    - RGB888, 1920 pixels × 1080 lines, 30 fps (FHD)
    - RGB888, 1280 pixels × 1024 lines, 60 fps

Figure 54.1 shows a block diagram.

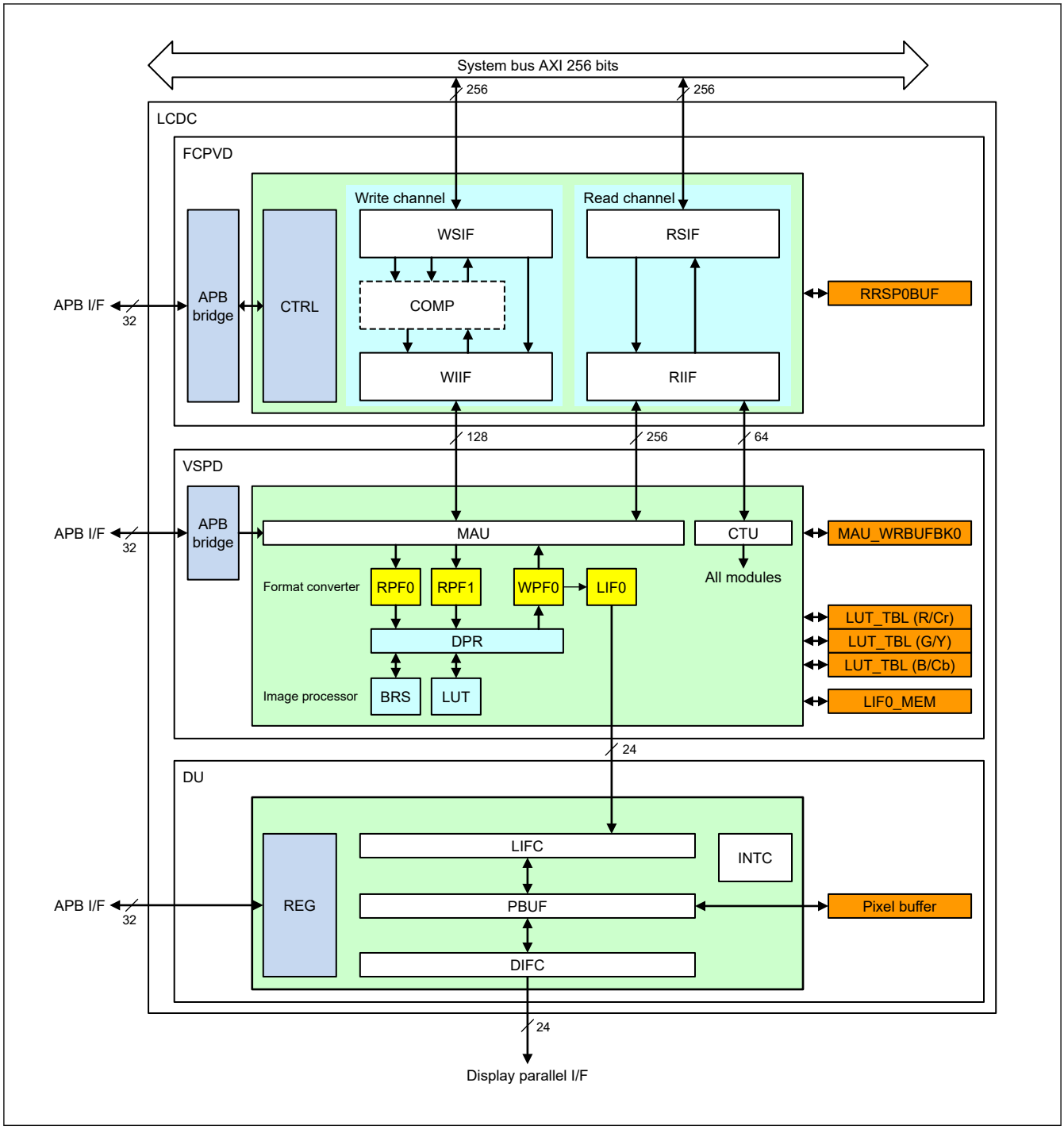


Figure 54.1 Block diagram

Table 54.1 Input/output pins of the LCD controller (1 of 2)

Name	I/O	Description
DISP_CLK	Output	Display Parallel Interface pixel clock
DISP_HSYNC	Output	Display Parallel Interface Horizontal sync pulse
DISP_VSYNC	Output	Display Parallel Interface Vertical sync pulse
DISP_DE	Output	Display Parallel Interface data enable
DISP_DATAR0 to DISP_DATAR7	Output	Display Parallel Interface pixel data output (R/Cr)
DISP_DATAG0 to DISP_DATAG7	Output	Display Parallel Interface pixel data output (G/Y)

Table 54.1 Input/output pins of the LCD controller (2 of 2)

Name	I/O	Description
DISP_DATAB0 to DISP_DATAB7	Output	Display Parallel Interface pixel data output (B/Cb)

54.1.1 FCPVD

FCPVD has the following sub modules.

Table 54.2 Sub modules in FCPVD

Abbreviation	Description
CTRL	FCPVD controller
WIIF	VSPD write channel interface
WSIF	System AXI write channel interface
RIIF	VSPD read channel interface
RSIF	System AXI read channel interface
COMP	Compression module (This is a dummy module)

54.1.2 VSPD

VSPD has the following sub modules.

54.1.2.1 Memory Access Unit (MAU)

The VSPD applies processing to the image data stored in the external memory and writes the resultant data back to the external memory. The data transfer between the external memory and VSPD necessary for this operation is done by the MAU, which works as the bus master, according to the register settings. The MAU executes this data transfer between the external memory and VSPD.

54.1.2.2 Command Transfer Unit (CTU)

The VSPD can directly read register parameters for image processing by display lists stored in external memory. The CTU module is a bus interface and controls display lists when the CTU reads display lists as a bus master.

54.1.2.3 Read Pixel Formatter (RPF)

The RPF reads image data from the external memory through the MAU, unpacks data according to the specified format, converts the color space, converts the number of colors, executes color keying, ROP operation, Multiply-alpha and OSD processing, and outputs the resultant data to the DPR. The RPF has an input format unpacking unit, a 1-bit mask generator, a raster operation unit (ROP unit), a color keying unit, a color space converter, and multiply-alpha. Figure 54.2 shows the processing flow in the RPF.

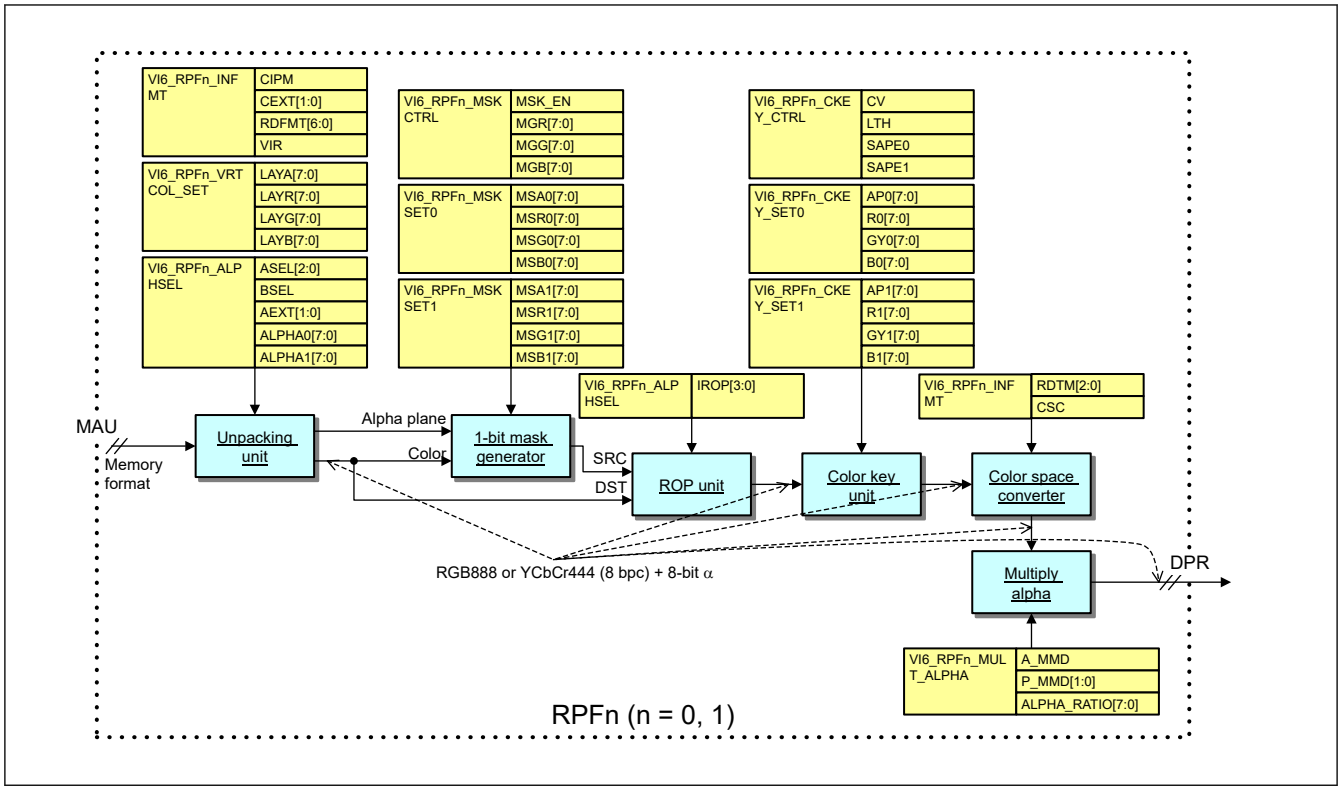


Figure 54.2 RPF processing flow

The input format unpacking unit expand the image data input from the MAU into the image format for internal processing (YCbCr444 8-bpc or RGB888 8-bpc), and the 1-bit mask generator generates a 1-bit image mask from the image data expanded through the unpacking unit. Alternately, a 1-bit image mask can be generated from the alpha plane that is different from the picture plane read through the MAU.

The raster operation unit (ROP unit) executes raster operation between the data from the 1-bit mask generator and the image data expanded from the input format, and the color keying unit applies color replacement and specifies the transparent color for the image data input from the ROP unit. The color space converter converts the color space (RGB-YCbCr) of the image data input from the color keying unit as necessary. The multiply-alpha unit multiplies pixel-alpha by specified alpha or/and multiplies image data by pixel-alpha or/and specified alpha.

The VSPD provides maximum two RPF modules (RPF0 to RPF1).

54.1.2.4 Data Path Router (DPR)

The DPR controls the data paths among RPFs, function modules, and WPFs. The DPR selects one of the images input from RPFs, outputs it to a function module (BRS or LUT), and selects one of WPFs as the destination where the image data processed in the function module will be output. Before output to the WPF, the output from each function module can be input to another function module, which enables multiple image processing functions to be executed continuously without involving the external memory.

54.1.2.5 Look Up Table (LUT)

This is a 1D-LUT that converts each of three color components by using a lookup table. The LUT is connected to the DPR and can be used for gamma correction, negative-positive conversion, posterization, and binarization through desired tone curve settings.

The LUT supports local area processing for Y component. The LUT can apply 1D look up table for Y component per area where a image is divided into multiple area specified by registers and command list.

54.1.2.6 Blend ROP Sub Unit (BRS)

The BRS is a module connected to the DPR, which executes the image blending processing and ROP operation. The BRS has two blend/ROP operation units (blend/ROP unit m, m = A to B), a blend/ROP input switch (SEL) for selecting the input to these operation units, and a divider for normalization (div unit).

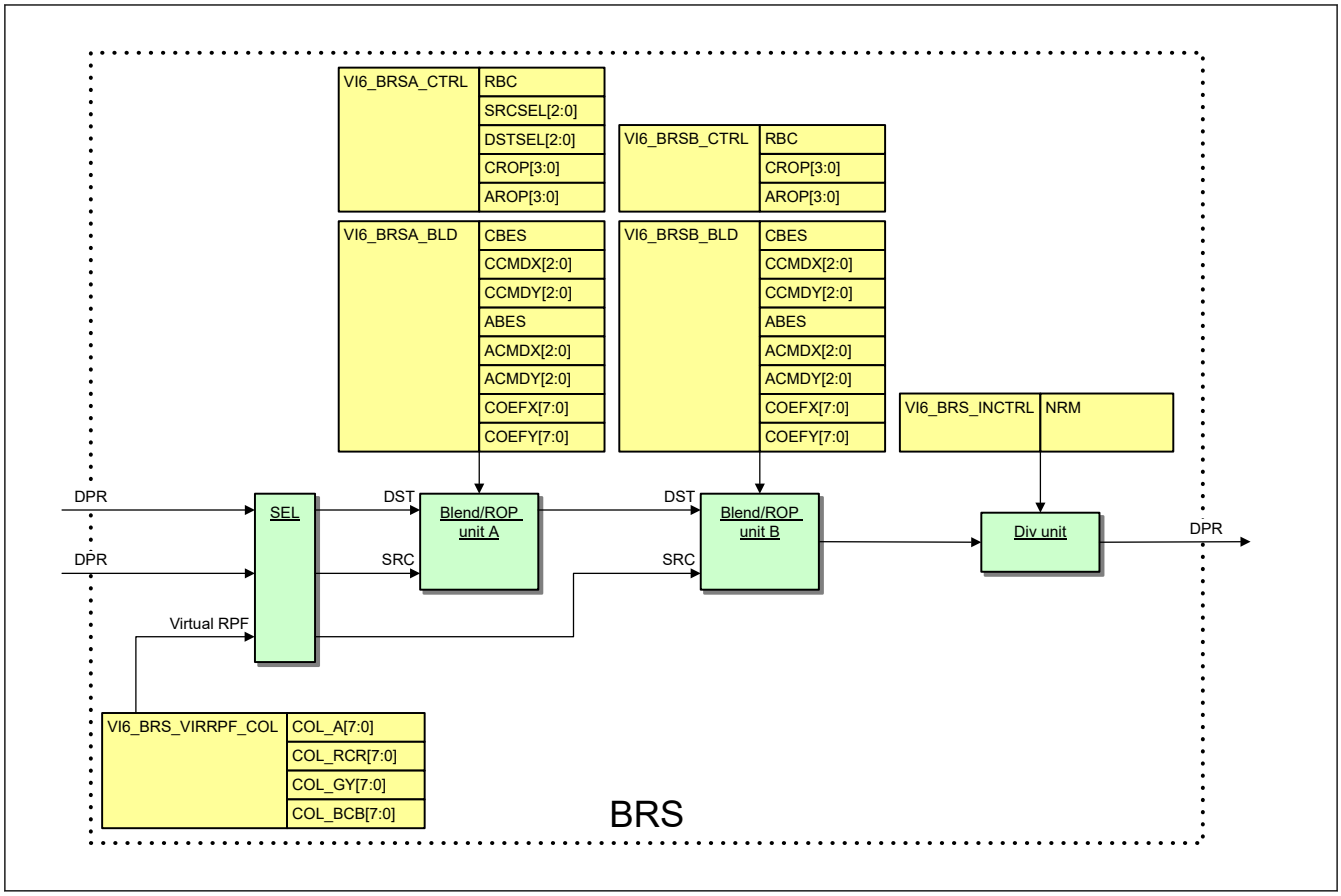


Figure 54.3 BRS processing flow

Each of the two blend-ROP operation units (blend/ROP unit m) receives the output from the SEL or blend/ROP Unit m, and executes blending or raster operation (ROP) of images.

The divider for normalization (div unit) divides the pixel value by the α value.

54.1.2.7 Write Pixel Formatter (WPF)

The WPF is an output module that receives 32 bits image data (YCbCr444 or RGB888 + 8-bit α) from the DPR, converts the color space, number of colors, and format of the data, and outputs the results of VSPD image processing to external memory through the MAU. The WPF is mainly configured from a color space converter and an output format converter (the packing unit).

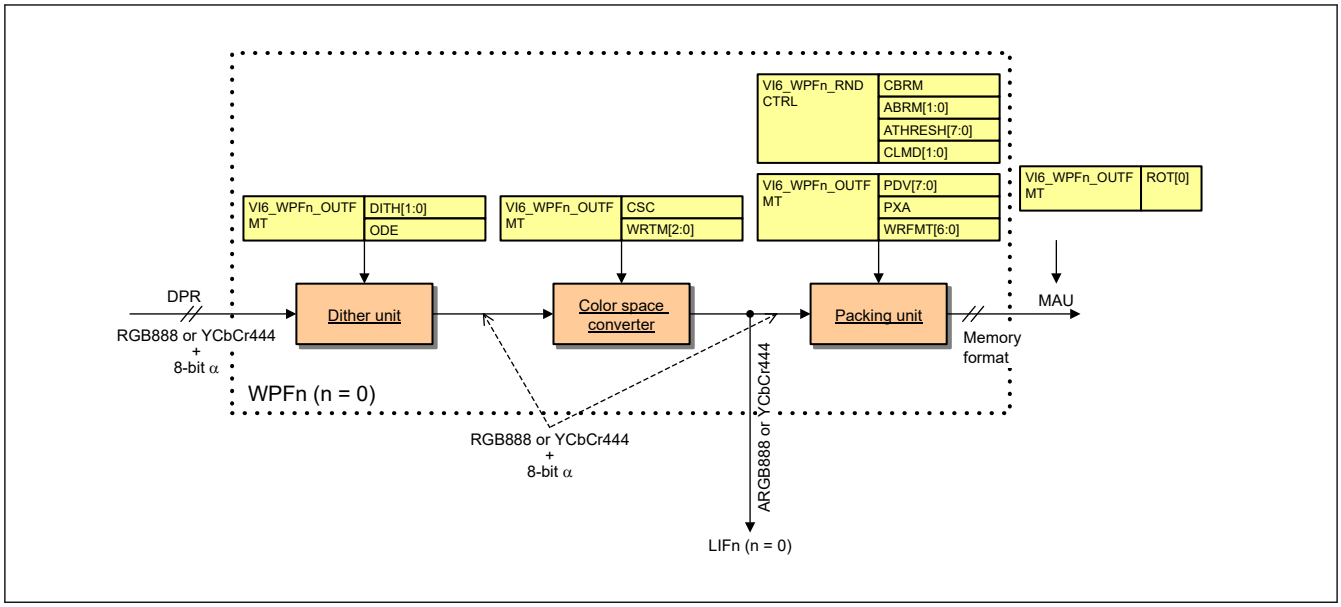


Figure 54.4 WPF processing flow

The color space converter converts the color space between RGB and YCbCr, and the packing unit converts the format into the picture plane storing format. The VSPD provides one WPF (WPF0). The WPF has the output to LIF module after the color space conversion for transferring the video data to the display module. Image data to LIF is 24 bits (YCbCr444 or RGB888) format, and its' color space is after color space conversion.

54.1.2.8 Display Unit InterFace (LIF)

The LIF module is used for transferring image data to the display module. The input port of the LIFn (n = 0) module is connected to WPFn (n = 0), and the output port of the LIF module is connected to DU (Display Unit). Data flow in LIF is shown in Figure 54.30.

54.1.2.9 Detail Function

Table 54.3 Detail function of VSPD (1)

Image data transfer function			
Bus interface	Protocol		AXI 256 bits through FCPVD
	Data alignment	Conversion method	Byte, Word, LW, or LLW data swapping
		Channel	Data alignment can be specified separately for each input/output channel.
Image memory	Input	Address setting	1-byte units
	Output	Address setting	1-byte units
		Memory area	Images can be written to the same memory area where the master layer is stored. Note the restrictions shown in Table 54.44.
Tile transfer mode			Supported by RPF0 to RPF1
YCbCr memory storage format			Interleaved, planar, or semi-planar

Table 54.4 Detail function of VSPD (2)

Display list / Extended display list transfer function			
Bus interface	Protocol		AXI 256 bits through FCPVD
	Data alignment*1	Conversion method	Byte, Word, LW data swapping
Data memory	Load	Address setting	8-byte units
	Store	—	Not available

Note 1. Data alignment can be specified separately for display list and each entry of extended display list.

Table 54.5 Detail function of VSPD (3) (1 of 2)

Read Pixel Formatter (RPF)			
Number of channels			Two channels (RPF0 to RPF1)
Input color bit-depth		Color	2- to 8-bit
		Alpha	1-/2-/4-/8-bit
Operation bit-depth		Color	8-bit
		Alpha	8-bit
Image format	Input	RGB	RGB888, RGB565, RGB666, αRGB8888, αRGB4444, αRGB1555 α plane (8 bpp, 1 bpp)
		YCbCr	YCbCr4:4:4 (8 bpc) Planar/Semi planar/Interleaved YCbCr4:2:2 (8 bpc) Planar/Semi planar/Interleaved YCbCr4:2:0 (8 bpc) Planar/Semi planar/Interleaved α plane (8 bpp, 1 bpp)
		Maximum size	1920 × 1080 pixels The internal data path modules have separate restrictions on the maximum image size. For details, see section 54.7.2. Input Image Size.
		Minimum size	1 × 1 pixel The internal data path modules have separate restrictions on the minimum image size. For details, see section 54.7.2. Input Image Size.
		Size setting unit	YCbCr420: 2-pixel units both horizontally and vertically. YCbCr422: 2-pixel units horizontally and 1-pixel units vertically. In other formats, the size can be set in 1-pixel units.
Color keying	Color replacement	Compared data	RGB or Y (8 bpc)
		Replaced data	αRGB or αYCbCr (8 bpc for R, G, B, Y, Cb, Cr, α)
		Comparison Mode	Matched color mode
		Input source	RPF0 to RPF1
	Transparent color	Compared data	RGB or Y (8 bpc)
		Replaced data	α 8 bpp
		Comparison Mode	Matched color mode, Luma threshold mode
		Input source	RPF0 to RPF1
Raster operation	ROP2 (within input channels)	Operator	16 types (OpenGL2.0 is supported)
		Sources of operation	ROP2 operation between the 1-bpp α plane and RGB/YCbCr data in RPF0 to RPF1. Note that 1-bpp α is converted to αRGB or αYCbCr4:4:4. (8 bpc)
Color space conversion	RGB to YCbCr	Conversion expression	RGB (0, $2^n - 1$) to BT601 (2^{n-4}, $235 \times 2^{n-8}$ / $240 \times 2^{n-8}$) RGB (0, $2^n - 1$) to BT709 (2^{n-4}, $235 \times 2^{n-8}$ / $240 \times 2^{n-8}$) RGB (0, $2^n - 1$) to BT601 (0, $2^n - 1$) RGB (2^{n-4}, $235 \times 2^{n-8}$) to BT709 (2^{n-4}, $235 \times 2^{n-8}$ / $240 \times 2^{n-8}$) (n = 8)
		Target of conversion	RPF0 to RPF1
	YCbCr to RGB	Conversion expression	BT.601 (2^{n-4}, $235 \times 2^{n-8}$ / $240 \times 2^{n-8}$) to RGB (0, $2^n - 1$) BT.709 (2^{n-4}, $235 \times 2^{n-8}$ / $240 \times 2^{n-8}$) to RGB (0, $2^n - 1$) BT.601 (0, $2^n - 1$) to RGB (0, $2^n - 1$) BT.709 (2^{n-4}, $235 \times 2^{n-8}$ / $240 \times 2^{n-8}$) to RGB (2^{n-4}, $235 \times 2^{n-8}$) (n = 8)
		Target of conversion	RPF0 to RPF1

Table 54.5 Detail function of VSPD (3) (2 of 2)

Read Pixel Formatter (RPF)			
Changing number of colors	Bit extension	Extended bits	Input bit-depth (2 bpc to 6 bpc) to operation bit-depth 8 bpc
		Target format	RGB666, RGB565, RGB555, RGB444, or RGB332
		Bit reduction method	Padded with 0. Copied from the most significant bits.
		Target of conversion	RPF0 to RPF1
	YCbCr444 generation	Vertical	CbCr copying
		Horizontal	Copying or interpolation.
α bit count conversion	Bit extension	Extended bits	Input bit-depth 1 bpc to operation bit-depth 8 bpc
		Method	Padded with 0. Copied from the most significant bits.
Multiply-alpha function	Fade-alpha		RPF0 to RPF1 Available for straight pixel and pre-multiplied pixel.
	Generate pre-multiplied alpha		RPF0 to RPF1
Virtual display	RPF0 to RPF1	Color format	αRGB8888 or αYCbCr4:4:4 single-color
		Display size	Same as the size of the input channel
	Virtual RPF	Color format	αRGB8888 or αYCbCr4:4:4 single-color
		Display size	Maximum: 1920 × 1080 pixels Minimum: 4 × 4 pixels

Table 54.6 Detail function of VSPD (4)

Write Pixel Formatter (WPF)			
Number of channels			One channel (WPF0)
Image format	Output	RGB	RGB332, RGB444, RGB565, RGB666, RGB888, αRGB8666, αRGB8888, αRGB4444, αRGB1555
		YCbCr	YCbCr4:4:4, YCbCr4:2:2, YCbCr4:2:0
		Maximum size	1920 × 1080 pixels The internal data path modules have separate restrictions on the maximum image size. For details, see section 54.7.3. Output Image Size.
		Minimum size	1 × 1 pixel The internal data path modules have separate restrictions on the minimum image size. For details, see section 54.7.3. Output Image Size.
		Size setting unit	YCbCr420: 2-pixel units both horizontally and vertically. YCbCr422: 2-pixel units horizontally and 1-pixel units vertically. In other formats, the size can be set in 1-pixel units.
Color space conversion	RGB to YCbCr	Conversion expression	RGB (0, 255) to BT.601 (16, 235/240) RGB (0, 255) to BT.709 (16, 235/240) RGB (0, 255) to BT.601 (0, 255) RGB (16, 235) to BT.709 (16, 235/240)
		Target of conversion	WPF0
	YCbCr to RGB	Conversion expression	BT.601 (16, 235/240) to RGB (0, 255) BT.709 (16, 235/240) to RGB (0, 255) BT.601 (0, 255) to RGB (0, 255) BT.709 (16, 235/240) to RGB (16, 235)
		Target of conversion	WPF0
Changing number of colors	Output	Reducing RGB color depth	Dithering, lower-order bit truncation, or rounding
		YCbCr422/420	CbCr skipping or CbCr vertical skipping and horizontal skipping
α bit count conversion	Output	Bit reduction	Truncation, rounding, or comparison with threshold (for 1 bpp)

Table 54.7 Detail function of VSPD (5)

Image compositing			
α blending	Input α value selection	RGB	Pixel α, fixed α value, α plane, or 1-bit α converted from the color specified for pixels
		YCbCr	α plane, fixed α value, or 1-bit α converted from the color specified for pixels
	α blending expression	Layer A: Upper layer Layer B: Lower layer	xA + yB, xA − yB Coefficients x and y should be selected from the following. Fixed α value, (α for layer A), (1 − α for layer A), (α for layer B), (1 − α for layer B)
	Output α value selection	RGB	Fixed α value, x (α for layer A) + y (α for layer B), x (α for layer A) – y (α for layer B) Coefficients x and y should be selected from the following. Fixed α value, (α for layer A), (1 − α for layer A), (α for layer B), (1 − α for layer B)
	Blending layers	Number of layers	Two layers selected from RPF0 to RPF1 and video processing function output, and virtual RPF; three layers in total
		Order of layers	The order of three layers selected from RPF0 to RPF1, virtual RPF, and video processing function output can be changed as desired.
	α plane	Format	8 bpp or 1 bpp (α value can be specified through register)
		Input source	RPF0 to RPF1
	Fixed α value	Format	8 bpp
		Input source	RPF0 to RPF1, virtual RPF, or video processing function output
Raster operation	ROP2 (between input channels)	Operator	16 types (OpenGL2.0 is supported)
		Sources of operation	RPF0 to RPF1, virtual RPF, and video processing function output
		Operation control	RGB/YCbCr and α are operated separately.
	ROP3 (between input channels)	Operator	256-type ROP3 is available by combining ROP2 operations
		Sources of operation	RPF0 to RPF1, virtual RPF, and video processing function output
		Operation control	RGB/YCbCr and α are operated separately.

Table 54.8 Detail function of VSPD (6)

Color adjustment function		
1D-LUT	LUT configuration	Independent R/Y, G/Cb, and B/Cr. 256 entries each

Table 54.9 Detail function of VSPD (7)

Direct connection to display module		
Display I/F	Destination	DU
	Direct connection	Can transfer data to DU without going through external memory
Line padding	Padding cycles	1 to 32 cycles
	Padding pattern	Arbitrarily setting up to 32 cycles

54.1.3 DU

DU has the following sub modules.

54.1.3.1 REG

The REG controls registers for DU. The DU has APB4 interface, and its transaction is transferred with no wait states.

54.1.3.2 LIFC

The LIFC is the interface block from/to VSPD. The LIFC receives image data from VSPD, and sends to PBUF in a pixel unit. The LIFC continues to receive the image data from VSPD unless FIFO will be full.

54.1.3.3 PBUF

The PBUF is the asynchronous FIFO which converts clock domain of image data from System Bus Clock to Video Clock. The PBUF watches over FIFO underflow.

54.1.3.4 DIFC

The DIFC is the display timing master. The DIFC controls the polarity and the timing of the Video Output signals. Controllable timings are the followings.

- Hfront: Horizontal front porch (pixels)
- Hback: Horizontal back porch (pixels)
- Hsync: Horizontal sync active (pixels)
- Hactive: Horizontal active area (pixels)
- Vfront: Vertical front porch (lines)
- Vback: Vertical back porch (lines)
- Vsync: Vertical sync active (lines)
- Vactive: Vertical active area (lines)

54.1.3.5 INTC

The INTC generates the interrupt when PBUF will detect FIFO underflow. The interrupt is level signal.

54.1.4 Clock and Interrupt Signal

The following clocks are supplied to the LCDC.

Table 54.10 Clock list

Clock name	Description	Frequency (MHz)
PCLKAH	AXI clock	400
PCLKAL	APB clock	100
LCDC_clkd	Video clock	5 to 100

The LCDC transmits the following interrupt signals to the interrupt controller.

Table 54.11 Interrupt list

Signal name	Interrupt condition	Interrupt source register
LCDC_VSPD_INT	Frame End	VI6_WPF0_IRQ_STA.FRE
	Display List Frame End	VI6_WPF0_IRQ_STA.DFE
	DU Connection UnderRun Error	VI6_WPF0_IRQ_STA.UND
LCDC_DU_INT	RBUF FIFO Underflow	DU_MSR0.ST_PB_RUF

Note: See section 54.4.4.13. VI6_WPF0_IRQ_STA : WPF0 Interrupt Status Register and section 54.5.1. DU_MCR0 : DU Module Control Register 0 for details.

54.2 Register Map

Table 54.12 LCDC register map (1 of 3)

Address	Register symbol	Register name	Write protection
0x920D_0000	FCP_VCR	FCPVD Version Control Register	—
0x920E_0000	VI6_CMD0	VSPD Start Register 0	—
0x920E_0010	VI6_CLK_CTRL0	Clock Control Register 0	—
0x920E_0014	VI6_CLK_CTRL1	Clock Control Register 1	—
0x920E_0018	VI6_CLK_DCSWT	Dynamic Clock Stop Control Register	—
0x920E_001C	VI6_CLK_DCSM0	Dynamic Clock Stop Disable Register 0	—
0x920E_0020	VI6_CLK_DCSM1	Dynamic Clock Stop Disable Register 1	—
0x920E_0028	VI6_SRESET	Software Reset Register	—
0x920E_002C	VI6_MRESET_ENB0	Module Reset Enable Register 0	—
0x920E_0030	VI6_MRESET_ENB1	Module Reset Enable Register 1	—
0x920E_0034	VI6_MRESET	Module Reset Issuing Register	—
0x920E_0038	VI6_STATUS	Operating Status Register	—
0x920E_0048	VI6_WPF0_IRQ_ENB	WPF0 Interrupt Enable Register	—
0x920E_004C	VI6_WPF0_IRQ_STA	WPF0 Interrupt Status Register	—
0x920E_0078	VI6_DISP0_IRQ_ENB	Display-0 Interrupt Enable Register	—
0x920E_007C	VI6_DISP0_IRQ_STA	Display-0 Interrupt Status Register	—
0x920E_0100	VI6_DL_CTRL	Display List Control Register	—
0x920E_0104	VI6_DL_HDR_ADDR0	Display List-0 Header Address Register	—
0x920E_0114	VI6_DL_SWAP0	Display List-0 Data Swapping Register	—
0x920E_011C	VI6_DL_EXT_CTRL0	Extended Display List-0 Control Register	—
0x920E_0120	VI6_DL_BODY_SIZE0	Display List Body Size Register 0	—
0x920E_0130	VI6_DL_HDR_REF_ADDR0	Display List-0 Header Reference Address Register	—
0x920E_0300 + 0x100 × n	VI6_RPFn_SRC_BSIZE	RPFn Basic Read Size Register (n = 0, 1)	—
0x920E_0304 + 0x100 × n	VI6_RPFn_SRC_ESIZE	RPFn Extended Read Size Register (n = 0, 1)	—
0x920E_0308 + 0x100 × n	VI6_RPFn_INFMT	RPFn Input Format Register (n = 0, 1)	—
0x920E_030C + 0x100 × n	VI6_RPFn_DSWAP	RPFn Data Swapping Register (n = 0, 1)	—
0x920E_0310 + 0x100 × n	VI6_RPFn_LOC	RPFn Display Location Register (n = 0, 1)	—
0x920E_0314 + 0x100 × n	VI6_RPFn_ALPH_SEL	RPFn α Plane Selection Control Register (n = 0, 1)	—
0x920E_0318 + 0x100 × n	VI6_RPFn_VRTCOL_SET	RPFn Virtual Plane Color Information Register (n = 0, 1)	—
0x920E_031C + 0x100 × n	VI6_RPFn_MSKCTRL	RPFn Mask Control Register (n = 0, 1)	—
0x920E_0320 + 0x100 × n	VI6_RPFn_MSKSET0	RPFn IROP-SRC Input Value Register 0 (n = 0, 1)	—
0x920E_0324 + 0x100 × n	VI6_RPFn_MSKSET1	RPFn IROP-SRC Input Value Register 1 (n = 0, 1)	—
0x920E_0328 + 0x100 × n	VI6_RPFn_CKEY_CTRL	RPFn Color Keying Control Register (n = 0, 1)	—
0x920E_032C + 0x100 × n	VI6_RPFn_CKEY_SET0	RPFn Color Keying Color Setting Register 0 (n = 0, 1)	—
0x920E_0330 + 0x100 × n	VI6_RPFn_CKEY_SET1	RPFn Color Keying Color Setting Register 1 (n = 0, 1)	—
0x920E_0334 + 0x100 × n	VI6_RPFn_SRCM_PSTRIDE	RPFn Source Picture Memory Stride Setting Register (n = 0, 1)	—
0x920E_0338 + 0x100 × n	VI6_RPFn_SRCM_ASTRIDE	RPFn Source α Memory Stride Setting Register (n = 0, 1)	—
0x920E_033C + 0x100 × n	VI6_RPFn_SRCM_ADDR_Y	RPFn Source Y/RGB Address Register (n = 0, 1)	—
0x920E_0340 + 0x100 × n	VI6_RPFn_SRCM_ADDR_C0	RPFn Source Chroma Address Register 0 (n = 0, 1)	—

Table 54.12 LCDC register map (2 of 3)

Address	Register symbol	Register name	Write protection
0x920E_0344 + 0x100 × n	VI6_RPFn_SRCM_ADDR_C1	RPFn Source Chroma Address Register 1 (n = 0, 1)	—
0x920E_0348 + 0x100 × n	VI6_RPFn_SRCM_ADDR_AI	RPFn Source α Address Register (n = 0, 1)	—
0x920E_0350 + 0x100 × n	VI6_RPFn_BAC	RPFn Bus Access Control Register (n = 0, 1)	—
0x920E_036C + 0x100 × n	VI6_RPFn_MULT_ALPH	RPFn Multiply Alpha Control Register (n = 0, 1)	—
0x920E_1000	VI6_WPF0_SRCRPF	WPF0-Source-RPF Register	—
0x920E_1004	VI6_WPF0_HSZCLIP	WPF0 Horizontal Input Size Clipping Register	—
0x920E_1008	VI6_WPF0_VSZCLIP	WPF0 Vertical Input Size Clipping Register	—
0x920E_100C	VI6_WPF0_OUTFMT	WPF0 Output Format Register	—
0x920E_1010	VI6_WPF0_DSWAP	WPF0 Data Swapping Register	—
0x920E_1014	VI6_WPF0_RNDCTRL	WPF0 Rounding Control Register	—
0x920E_101C	VI6_WPF0_DSTM_STRIDE_Y	WPF0 Destination Y Plane Memory Stride Register	—
0x920E_1020	VI6_WPF0_DSTM_STRIDE_C	WPF0 Destination C Plane Memory Stride Register	—
0x920E_1024	VI6_WPF0_DSTM_ADDR_Y	WPF0 Destination Y/RGB Address Register	—
0x920E_1028	VI6_WPF0_DSTM_ADDR_C0	WPF0 Destination Chroma Address Register 0	—
0x920E_102C	VI6_WPF0_DSTM_ADDR_C1	WPF0 Destination Chroma Address Register 1	—
0x920E_1034	VI6_WPF0_WRBCK_CTRL	WPF0 LIF Write Back Control Register	—
0x920E_2000	VI6_DPR_RPF0_ROUTE	RPF0 Routing Register	—
0x920E_2004	VI6_DPR_RPF1_ROUTE	RPF1 Routing Register	—
0x920E_2014	VI6_DPR_WPF0_FPORCH	WPF0 Timing Control Register	—
0x920E_203C	VI6_DPR_LUT_ROUTE	LUT Routing Register	—
0x920E_2050	VI6_DPR_BRS_ROUTE	BRS Routing Register	—
0x920E_2800	VI6_LUT_CTRL	LUT Control Register	—
0x920E_3900	VI6_BRS_INCTRL	BRS Input Control Register	—
0x920E_3904	VI6_BRS_VIRRPF_SIZE	Size Register of BRS Input Virtual RPF	—
0x920E_3908	VI6_BRS_VIRRPF_LOC	Display Location Register of BRS Input Virtual RPF	—
0x920E_390C	VI6_BRS_VIRRPF_COL	Color Information Register of BRS Input Virtual RPF	—
0x920E_3910	VI6_BRSA_CTRL	BRS Control Register A	—
0x920E_3914	VI6_BRSA_BLD	BRS Blend Control Register A	—
0x920E_3918	VI6_BRSB_CTRL	BRS Control Register B	—
0x920E_391C	VI6_BRSB_BLD	BRS Blend Control Register B	—
0x920E_3B00	VI6_LIF0_CTRL	LIF0 Control Register	—
0x920E_3B04	VI6_LIF0_CSBTH	LIF0 Clock Stop Buffer Control Register	—
0x920E_3B0C	VI6_LIF0_LBA	LIF0 Buffer Attribute Register	—
0x920E_3B30	VI6_LIF0_PADLN_CYC	LIF0 Padding Line Cycle Register	—
0x920E_3B34	VI6_LIF0_PADLN_PT	LIF0 Padding Line Pattern Register	—
0x920E_3B38	VI6_LIF0_PADLN_VAL	LIF0 Padding Line Value Register	—
0x920E_3B3C	VI6_LIF0_PADLN_SIZE	LIF0 Padding Line Size Register	—
0x920C_0000	DU_MCR0	DU Module Control Register 0	—
0x920C_0004	DU_MSR0	DU Module Status Register 0	—
0x920C_0008	DU_MSR1	DU Module Status Register 1	—
0x920C_000C	DU_IMR0	DU Interrupt Mask Register 0	—

Table 54.12 LCDC register map (3 of 3)

Address	Register symbol	Register name	Write protection
0x920C_0010	DU_DITR0	DU Display I/F Timing Register 0	—
0x920C_0014	DU_DITR1	DU Display I/F Timing Register 1	—
0x920C_0018	DU_DITR2	DU Display I/F Timing Register 2	—
0x920C_001C	DU_DITR3	DU Display I/F Timing Register 3	—
0x920C_0020	DU_DITR4	DU Display I/F Timing Register 4	—
0x920C_0040	DU_MCR1	DU Module Control Register 1	—
0x920C_004C	DU_PBCR0	DU PBUF Control Register 0	—
0x920C_0050	DU_PBCR1	DU PBUF Control Register 1	—
0x920C_0054	DU_PBCR2	DU PBUF Control Register 2	—

Table 54.13 LCDC related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	—	MSTPCRM.MSTPCRM04	SLVACCCTL4.LCDC_SL

54.3 FCPVD Register Descriptions

54.3.1 FCP_VCR : FCPVD Version Control Register

Base address: FCPVD = 0x920D_0000

Offset address: 0x0000

Bit position:	31																15							8	7						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CATEGORY[7:0]								REVISION[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1

Bit	Symbol	Function	R/W
7:0	REVISION[7:0]	Version of LSI product 0x09: FCP revision is the Simple LCDC	R
15:8	CATEGORY[7:0]	FCP use case category	R
31:16	—	These bits are read as 0.	R

FCP_VCR is the version control register of the FCPVD. The value read is fixed.

54.4 VSPD Register Descriptions

54.4.1 Notational Conventions for Registers and Bit Fields

This section uses the following notational conventions for the VSPD registers and bit fields.

1. The names of registers and bits are written in uppercase.
2. A bit or bit field in a register is indicated as [register name.bit name]. For example, the STRCMD bit in the VI6_CMD0 register is indicated as VI6_CMD0.STRCMD.
3. Lowercase "n" in a register name or a bit name indicates an integer and the range of value n is defined when necessary. For RPFn and WPFn, when the range of value n is not defined, RPFn (n = 0, 1), WPFn (n = 0) are assumed.
4. In each subsection for register description in section 54.4. VSPD Register Descriptions, when only a bit name is written without showing its register name, the bit is in the register described in that subsection.

5. A wildcard (*) indicates any characters in a name and represents all registers or bits that match the specified first part of a name. For example, when there are two registers VI6_RPF_SRC_BSIZE and VI6_RPF_SRC_ESIZE, VI6_RPF_SRC_* indicates both registers.

54.4.2 Register Classification

The VSPD registers are arranged in the following order; the general control registers to display list control registers control operation of the entire VSPD, and the other registers control each image processing and specify parameters for the processing. The functions of the registers are described in this order starting from section 54.4.4. General Control Registers.

1. General control registers (VI6_CMDn (n = 0, 1), VI6_SRESET, VI6_STATUS, VI6_WPFn_IRQ_* (n = 0))
2. Display list control registers (VI6_DL_*)
3. RPF control registers (VI6_RPFn_* (n = 0, 1))
4. WPF control registers (VI6_WPFn_* (n = 0))
5. DPR control registers (VI6_DPR_*)
6. LUT control register (VI6_LUT_CTRL)
7. BRS control registers (VI6_BRS_*)
8. LIFn control registers (VI6_LIFn_* (n = 0))

The VSPD has two RPF channels and the register configuration is the same for all of RPFn (n = 0, 1). However, some bit fields have restrictions in certain RPFs. These restrictions are included in the description of the corresponding bit fields and registers. Likewise, the register configuration is the same for all WPFn (n = 0), but some bit fields have restrictions in certain WPFs; the restrictions are included in the description of the corresponding bit fields and registers.

54.4.3 Restrictions on Access to Registers and Lookup Tables

The VSPD has control registers and lookup tables. All VSPD registers are writable and readable by only 32 bits unit. To write partial bits in each register, read-modify-write is needed. When accessing the addresses where these registers and lookup tables are allocated, the following restrictions should be satisfied. If any restriction is violated, the VSPD will not operate correctly.

1. For the read-only bits and reserved bits in all VSPD registers, writing 1 is prohibited unless otherwise specified.
2. Undefined addresses are reserved areas and write access is prohibited in these areas.
3. For all registers and lookup tables, except VI6_CMDn, VI6_SRESET, VI6_*IRQ* and two plane registers such as VI6_DL_HDR_ADDRn, VI6_DL_BODY_SIZE0, modifying register values during operation of the module is prohibited. Modify registers while the corresponding module is stopped. For the operating status of the target module, see section 54.6.3. Concept of VSPD Operation Starting and Stopping.
4. There are three types about General control registers (section 54.4.4. General Control Registers and section 54.4.5. Display List Control Registers) as below.
   - Controlling WPF0 (ex. VI6_CMD0, VI6_SRESET.SRST0)
   - Common setting of WPF0 (ex. VI6_DL_CTRL.AR_WAIT[15:0])

Table 54.14 Correspondence between modules and register names

Module name	Register name
RPFn (n = 0, 1)	VI6_RPFn_*
WPFn (n = 0)	VI6_WPFn_*
DPR	VI6_DPR_*
LUT	VI6_LUT_CTRL
BRS	VI6_BRS_*
LIFn (n = 0)	VI6_LIFn_*

54.4.4 General Control Registers

54.4.4.1 VI6_CMD0 : VSPD Start Register 0

Base address: VSPD = 0x920E_0000

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	UPDHDR	—	—	—	STRCMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STRCMD	Start reservation of WPF VI6_CMD0.STRCMD controls WPF0. Writing 1 to this bit starts WPF0 in VSPD. Set this bit for activation only after all register settings in each output channel have been completed. If WPF0 is idle, WPF0 starts right after this bit is set to 1. If WPF0 is active, writing 1 to this bit reserves starting WPF0 operation. Wait to set VI6_CMD0.STRCMD to 1 until this bit is read as 0. VI6_CMD0.STRCMD can be negated by software reset (VI6_SRESET.SRST0). 0: Start reservation of WPF0 is canceled. [Write] Starting VSPD is not reserved. [Read] 1: Start reservation of WPF0 is set. [Write] Starting VSPD is reserved. [Read]	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	UPDHDR	Reserved state of updating Display List Header (DLH) address UPDHDR can be negated by software reset (VI6_SRESET.SRST0). Set this bit to 0 when STRCMD bit is set to 1. 0: NOP [Write] Not reserved to update DLH address [Read] 1: Do not set this bit to 1 [Write] Reserved to update DLH address [Read]	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

The basic concept of image processing operation started by activating the VSPD is shown in Figure 54.5. The actual data input/output is executed by the MAU, which is the bus interface module, as described in section 54.1.2. VSPD, but conceptually the RPF works as the data entry point to the VSPD, and the WPF works as the data exit point. To process images through the VSPD, the RPF (entrance) and WPF (exit) should be connected, and a data path from the RPF to the WPF should be formed.

To connect RPFn to WPFn, specify RPFn as the source RPF for WPFn in VI6_WPFn_SRCRPF, which is a register for WPFn (see section 54.4.7.1. VI6_WPF0_SRCRPF : WPF0-Source-RPF Register). This setting determines that RPFn will be started when WPFn is started through VI6_CMDn.

A data path to execute desired image processing should then be formed between the RPF (entrance) and WPF (exit). To form a data path, connect the necessary function modules in the VSPD between RPF and WPF. This function is provided by the DPR; specify the information for each module connection in data path routing registers VI6_DPR_*_ROUTE (see section 54.4.8. DPR Control Registers).

After a data path is formed (RPFn → WPFn) as described above, starting output module WPFn in the VSPD through VI6_CMDn starts all function modules connected to WPFn, and the desired image processing is executed. According to this design concept, starting a WPF module means starting the VSPD.

There are two types of data path configuration in the VSPD; one is "a single input module to a single output module" as shown in Figure 54.5 (A), and the other is "multiple input modules to a single output module" as shown in Figure 54.5 (B).

Figure 54.5 (A) shows an example of a configuration where modules with single input and single output are implemented through the DPR.

Figure 54.5 (B) shows another configuration example where the module with multiple input and single output is implemented through the DPR.

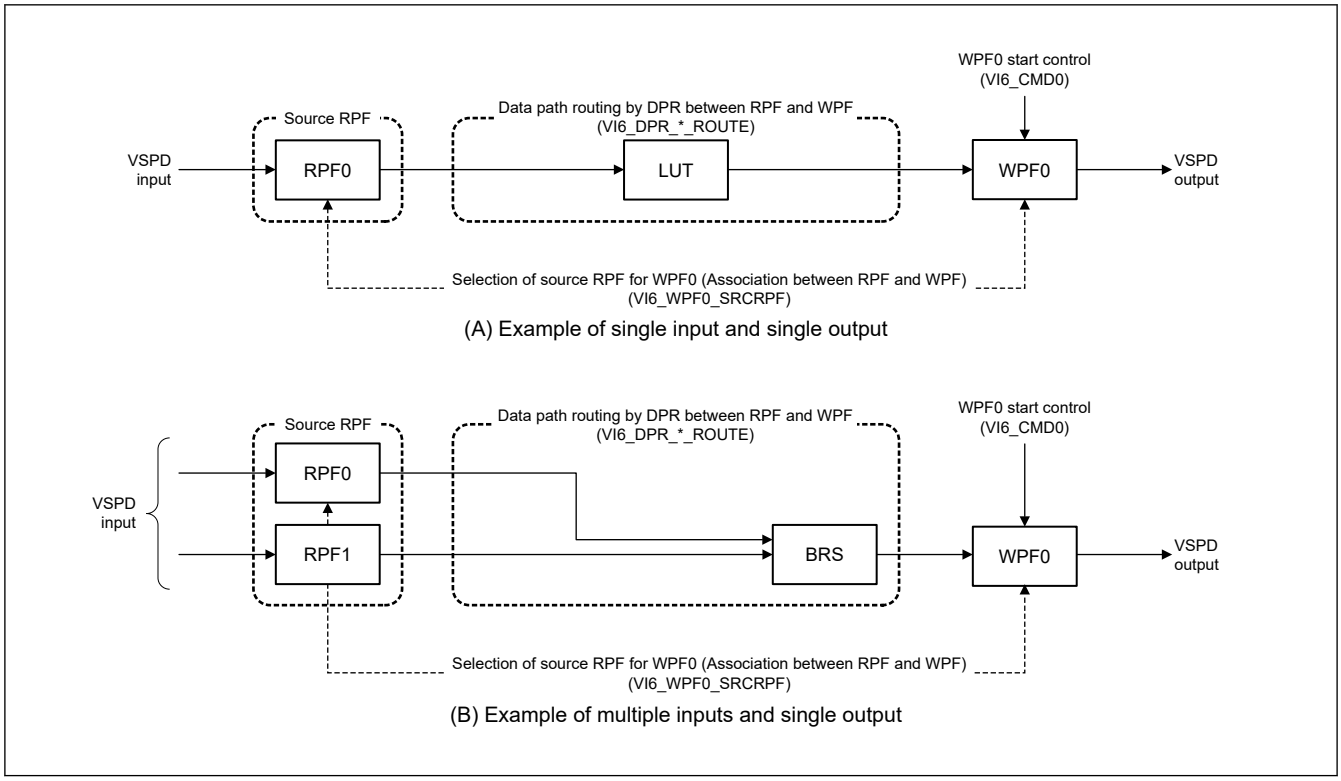


Figure 54.5 Basic concept of VSPD startup

54.4.4.2 VI6_CLK_CTRL0 : Clock Control Register 0

Base address: VSPD = 0x920E_0000

Offset address: 0x0010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	GCS0	—	—	—	—	—	—	—	—	—	—	—	GCS1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	GCS2[3:0]				—	—	—	GCS3[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	GCS3[4:0]	Clock Control Setting 3	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
11:8	GCS2[3:0]	Clock Control Setting 2	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	GCS1	Clock Control Setting 1	R/W
27:17	—	These bits are read as 0. The write value should be 0.	R/W
28	GCS0	Clock Control Setting 0	R/W

Bit	Symbol	Function	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

VSPD can stop its operating clock for reducing power consumption.

To enable clock stop function, set following registers:

- VI6_CLK_CTRL0 = 0x0000 0000
- VI6_CLK_CTRL1 = 0x0000 0000
- VI6_CLK_DCSWT = 0x0000 0808
- VI6_CLK_DCSM0 = 0x0000 0000
- VI6_CLK_DCSM1 = 0x0000 0000

To disable clock stop function, set following registers:

- VI6_CLK_CTRL0 = 0x1001 0F1F
- VI6_CLK_CTRL1 = 0xFF10 FFFF
- VI6_CLK_DCSWT = 0x0033 0808
- VI6_CLK_DCSM0 = 0x1FFF 0F1F
- VI6_CLK_DCSM1 = 0xFF10 FFFF

When the clock gating is enabled, the supply of the clock signal is stopped for the module which is not operated. If the bit field of this register is changed from 0 to 1, the supply of the clock signal for the module is immediately started. If the bit field of the register is changed from 1 to 0, the clock supply is immediately stopped. Therefore, please do not change this register field from 1 to 0 in operation because the VSPD may be stalled.

54.4.4.3 VI6_CLK_CTRL1 : Clock Control Register 1

Base address: VSPD = 0x920E_0000

Offset address: 0x0014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GCS4[7:0]								—	—	—	GCS6	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GCS5[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	GCS5[15:0]	Clock Control Setting 5	R/W
19:16	—	These bits are read as 0. The write value should be 0.	R/W
20	GCS6	Clock Control Setting 6	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
31:24	GCS4[7:0]	Clock Control Setting 4	R/W

See section 54.4.4.2. VI6_CLK_CTRL0 : Clock Control Register 0 for detail.

54.4.4.4 VI6_CLK_DCSWT : Dynamic Clock Stop Control Register

Base address: VSPD = 0x920E_0000

Offset address: 0x0018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	DCC0[1:0]		—	—	DCC1	DCC2
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CSTPW[7:0]								CSTRW[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CSTRW[7:0]	Dynamic Clock Stop Control 2 Always set these bits to 8.	R/W
15:8	CSTPW[7:0]	Dynamic Clock Stop Control 1 Always set these bits to 8.	R/W
16	DCC2	Dynamic Clock Control Setting 2	R/W
17	DCC1	Dynamic Clock Control Setting 1	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
21:20	DCC0[1:0]	Dynamic Clock Control Setting 0	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

See section 54.4.4.2. VI6_CLK_CTRL0 : Clock Control Register 0 for detail.

54.4.4.5 VI6_CLK_DCSM0 : Dynamic Clock Stop Disable Register 0

Base address: VSPD = 0x920E_0000

Offset address: 0x001C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	DCD0[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	DCD1[3:0]				—	—	—	DCD2[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DCD2[4:0]	Dynamic Clock Stop Disable Setting 2	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
11:8	DCD1[3:0]	Dynamic Clock Stop Disable Setting 1	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
28:16	DCD0[12:0]	Dynamic Clock Stop Disable Setting 0	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

See section 54.4.4.2. VI6_CLK_CTRL0 : Clock Control Register 0 for detail.

54.4.4.6 VI6_CLK_DCSM1 : Dynamic Clock Stop Disable Register 1

Base address: VSPD = 0x920E_0000

Offset address: 0x0020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DCD3[7:0]								—	—	—	DCD5	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DCD4[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DCD4[15:0]	Dynamic Clock Stop Disable Setting 4	R/W
19:16	—	These bits are read as 0. The write value should be 0.	R/W
20	DCD5	Dynamic Clock Stop Disable Setting 5	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
31:24	DCD3[7:0]	Dynamic Clock Stop Disable Setting 3	R/W

See section 54.4.4.2. VI6_CLK_CTRL0 : Clock Control Register 0 for detail.

54.4.4.7 VI6_SRESET : Software Reset Register

Base address: VSPD = 0x920E_0000

Offset address: 0x0028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRST 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRST0	WPF0 Software Reset Writing 1 to this bit aborts the current processing in WPF0 (the partially-completed image undergoing processing is output). The period until this software reset processing is completed depends on the bus state. When this reset processing is completed, the VI6_WPF0_IRQ_STA.FRE interrupt source bit is set to 1; when the FRE interrupt is enabled, the FRE end interrupt is output to notify the end of the reset processing. This bit is read as 0. 0: NOP 1: WPF0 software reset*1	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Applying a software reset to each WPF has the following restrictions.

1. A software reset can be applied to only one of WPF0 through single write access to VI6_SRESET.
2. After a software reset is issued, no more software reset can be issued to another WPF until the issued software reset processing is completed.
3. The end of software reset processing is notified through the FRE bit in VI6_WPFn_IRQ_STA, but the software reset issued while WPF is stopped is ignored as NOP. As it takes a while until the reset is actually issued after the reset bit is set, the VSPD may

complete operation before the reset is actually issued. In this case, no interrupt is output for the software reset that is issued after the VSPD completes operation.

4. If a software reset is issued during downloading of a display list, the downloading processing is not aborted. After the end of downloading that is in progress when a software reset is issued, a frame end interrupt is output.
5. When software reset is issued to VSPD, issue software reset to also FCPVD. See Figure 54.40 for VSPD software reset sequence.

54.4.4.8 VI6_MRESET_ENB0 : Module Reset Enable Register 0

Base address: VSPD = 0x920E_0000

Offset address: 0x002C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	MRSTE0[1:0]		—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MRSTE1[3:0]				—	—	—	MRSTE2[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	MRSTE2[4:0]	Module Reset Enable 2	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
11:8	MRSTE1[3:0]	Module Reset Enable 1	R/W
27:12	—	These bits are read as 0. The write value should be 0.	R/W
29:28	MRSTE0[1:0]	Module Reset Enable 0	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

This register is for purpose of H/W debugging.

54.4.4.9 VI6_MRESET_ENB1 : Module Reset Enable Register 1

Base address: VSPD = 0x920E_0000

Offset address: 0x0030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MRSTE3[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MRSTE4[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MRSTE4[15:0]	Module Reset Enable 4	R/W
23:16	—	These bits are read as 0. The write value should be 0.	R/W
31:24	MRSTE3[7:0]	Module Reset Enable 3	R/W

This register is for purpose of H/W debugging.

54.4.4.10 VI6_MRESET : Module Reset Issuing Register

Base address: VSPD = 0x920E_0000

Offset address: 0x0034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MRST	Module Reset Assertion	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

This register is for purpose of H/W debugging.

54.4.4.11 VI6_STATUS : Operating Status Register

Base address: VSPD = 0x920E_0000

Offset address: 0x0038

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	FLDST0	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SYS0_ACT	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0.	R
8	SYS0_ACT	WPF0 Operating Status This bit indicates the operating or stopped state of control channel n (WPF0). 0: WPF0 is stopped. 1: WPF0 is operating.	R
27:9	—	These bits are read as 0.	R
28	FLDST0	Field status of previous frame of WPF0 This bit can be referred in case of AUTO-FLD or AUTO-DISP. This bit is changed at the timing of VI6_WPF0_IRQ_STA.FRE. 0: Previous finished frame is TOP field. 1: Previous finished frame is BOT field.	R
31:29	—	These bits are read as 0.	R

54.4.4.12 VI6_WPF0_IRQ_ENB : WPF0 Interrupt Enable Register

Base address: VSPD = 0x920E_0000

Offset address: 0x0048

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UNDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DFEE	FREE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FREE	Interrupt Enable for WPF0 Frame End 0: Interrupt disabled 1: Interrupt enabled	R/W
1	DFEE	Interrupt Enable for WPF0 Display List Frame End 0: Interrupt disabled 1: Interrupt enabled	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	UNDE	Interrupt Enable for WPF0 Underrun in case of DU connection 0: Interrupt disabled 1: Interrupt enabled	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: Each bit controls the interrupt enable of the corresponding interrupt source.

Each bit in VI6_WPFn_IRQ_STA is set to 1 when the corresponding interrupt source is generated.

VI6_WPF0_IRQ_ENB specifies whether to output an interrupt signal for the generated source. When an interrupt is disabled in this register, no interrupt signal is generated even when the corresponding bit in VI6_WPFn_IRQ_STA is set to 1. When an interrupt is enabled in this register, an interrupt signal is output when the corresponding bit in VI6_WPFn_IRQ_STA is set to 1.

54.4.4.13 VI6_WPF0_IRQ_STA : WPF0 Interrupt Status Register

Base address: VSPD = 0x920E_0000

Offset address: 0x004C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UND
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DFE	FRE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FRE	Interrupt Status and Clear for WPF0 Frame End*1 This interrupt source bit is set to 1 when VSPD completes one-frame processing. This bit is also set to 1 when one-frame processing using a display list is completed. The interrupt status is set to 1 by any of the following conditions. a) Processing one frame is finished normally. b) Software reset is issued during VSPD is processing. c) One frame's data from DU is displayed while VSPD could not transfer one frame data to DU (in case of Linked with DU). 0: No interrupt. [Read] The interrupt status is cleared to 0. [Write] 1: Interrupt activated. [Read] Hold the interrupt status value. [Write]	R/W
1	DFE	Interrupt Status and Clear for WPF0 Display List Frame End*1 This interrupt source bit is set to 1 when VSPD completes one-frame processing while the current frame in enable value stored in the display list header is 1 (see section 54.6.3. Concept of VSPD Operation Starting and Stopping). When display lists are not used, this bit is not used. In this case, clear VI6_WPF0_IRQ_ENB.DFEE to 0 to mask the interrupt generation by this interrupt source. 0: No interrupt. [Read] The interrupt status is cleared to 0. [Write] 1: Interrupt activated. [Read] Hold the interrupt status value. [Write]	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	UND	Interrupt Status and Clear for WPF0 Underrun in case of DU connection This interrupt source bit is set to 1 when data underrun occurs in case of DU connection. Timing that this bit is set to 1 is at end of 1 frame. 0: No interrupt. [Read] The interrupt status is cleared to 0. [Write] 1: Interrupt activated. [Read] Hold the interrupt status value. [Write]	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: The read value from each bit is the status of the interrupt source, and the write access to each bit controls the interrupt status.
Note 1. This bit can hold the most recent of two times of interrupt status at the maximum. When writing 0 to this bit, the oldest interrupt status is cleared. And the status bit becomes 0 after all interrupt statuses are cleared.

VI6_WPF0_IRQ_STA indicates the state of the interrupt sources generated in the VSPD. Whether to output a VSPD interrupt when an interrupt source is generated and the corresponding bit is set to 1 is determined by the corresponding bit setting in VI6_WPFn_IRQ_ENB. While an interrupt is disabled in VI6_WPFn_IRQ_ENB, the VSPD does not output an interrupt signal even when an interrupt source is generated, but the source flag in this register is set to 1.

Note that the interrupt source bits in this register cannot be cleared by write access using a display list.

54.4.4.14 VI6_DISP0_IRQ_ENB : Display-0 Interrupt Enable Register

Base address: VSPD = 0x920E_0000

Offset address: 0x0078

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DSTE	—	—	MAEE	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
5	MAEE	Interrupt Enable for Display Read Data End 0: Interrupt disabled 1: Interrupt enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	DSTE	Interrupt Enable for Display Start 0: Interrupt disabled 1: Interrupt enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: Each bit controls the interrupt enable of the corresponding interrupt source.

54.4.4.15 VI6_DISP0_IRQ_STA : Display-0 Interrupt Status Register

Base address: VSPD = 0x920E_0000

Offset address: 0x007C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DST	—	—	MAE	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0. The write value should be 0.	R/W
5	MAE	Interrupt Status and Clear for Display Read Data End This bit is set to 1 when all RPFs transfer the last data of the frame to LIF module. The RPF module which is not used by LIF module does not affect this bit. 0: No interrupt. [Read] The interrupt status is cleared to 0. [Write] 1: Interrupt activated. [Read] Hold the interrupt status value. [Write]	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	DST	Interrupt Status and Clear for Display Start This bit is set to 1 when LIF module transfers the first data to the display module at the beginning of each frame. The timing depends on the output buffer status of LIF module. 0: No interrupt. [Read] The interrupt status is cleared to 0. [Write] 1: Interrupt activated. [Read] Hold the interrupt status value. [Write]	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: The read value from each bit is the status of the interrupt source, and the write access to each bit controls the interrupt status.

54.4.5 Display List Control Registers

54.4.5.1 VI6_DL_CTRL : Display List Control Register

Base address: VSPD = 0x920E_0000

Offset address: 0x0100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AR_WAIT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DC2	—	—	—	DC1	—	—	—	DLE1	RLM0	CFM0	NH0	DLE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DLE0	Display List Enable/Disable for WPF0 Enables or disables the WPF0 display list function. When the display list function is enabled through this bit, all WPF processing channels work in display list mode. When using display lists, note the restrictions in section 54.4.3. Restrictions on Access to Registers and Lookup Tables. 0: The display list function is disabled. 1: The display list function is enabled.	R/W
1	NH0	Header-less Display List Mode*1 *2 This bit is used for specifying the header-less display list mode. In case of header-less mode, the number of the display lists is 1. The address of the display body is set in VI6_DL_HDR_ADDR0 register, and the body size is set in VI6_DL_BODY_SIZE0 register. When this bit is changed, make sure that VSPD is stopped. And also make sure the following before starting VSPD. ● Header Address (VI6_DL_HDR_ADDR0) ● Body Size (VI6_DL_BODY_SIZE0) in case of header-less mode 0: Use Display List Header (Normal DL Mode). 1: Do not use Display List Header (Header-less Mode).	R/W
2	CFM0	Continuous Frame Mode for Header-less Display List for WPF0 This bit determines whether the next frame is automatically started or not. When the updated flag of the display list, VI6_DL_BODY_SIZE0.UPD0, is not updated, the display list of the next frame is not transferred and the same register values are used for the next frame. When the value of VI6_DL_BODY_SIZE0.UPD0 is updated, the new display list is transferred. 0: Stopped at the end of every frame. 1: The next frame is automatically started.	R/W
3	RLM0	Loading two plane Registers Mode for WPF0 0: Reserved status (VI6_CMD0.UPDHDR) is accepted by next_frame_auto_start. Two plane registers are also downloaded by next_frame_auto_start. 1: Reserved status (VI6_CMD0.UPDHDR) is not accepted by next_frame_auto_start. pNext_header in previous frame's DLH is used for loading Display List. Two plane registers are not downloaded by next_frame_auto_start either.	R/W
4	DLE1	This bit does not affect anything VSPD. (NOP)	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	DC1	This bit does not affect anything VSPD. (NOP)	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	DC2	This bit does not affect anything VSPD. (NOP)	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
31:16	AR_WAIT[15:0]	Display List Control Setting Always set these bits to 256.	R/W

Note 1. Only WPF0 supports header-less display list. WPF1 works as the normal display list mode even if the WPF0 is set to header-less display list mode.
Note 2. When DLE0 bit is 0, set NH0 bit to 0.

54.4.5.2 VI6_DL_HDR_ADDR0 : Display List-0 Header Address Register

Base address: VSPD = 0x920E_0000

Offset address: 0x0104

Bit position:	31																												0			
Bit field:																													0	0	0	0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Display List-0 Header Address These bits specify the address of the display list header to be read for display list-n in 16-byte units (the lower-order four bits are read-only). When WPF0 is first started in display list mode, the display list header is loaded from the address specified in this register. After loading of the header is completed, the register value of the display list address is updated to the next header address stored in the loaded header to prepare for loading of the next display list header. After that, this header address updating is repeated. A value from 0x00000000 to 0xFFFFFFF0 can be specified.	R/W

54.4.5.3 VI6_DL_SWAP0 : Display List-0 Data Swapping Register

Base address: VSPD = 0x920E_0000

Offset address: 0x0114

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	IND	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	LWS	WDS	BTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BTS	Display List Data Swapping in Byte Units The effect of this bit setting is defined in Table 54.15. 0: Data swapping in byte (8-bit) units is disabled 1: Data swapping in byte (8-bit) units is enabled	R/W
1	WDS	Display List Data Swapping in Word Units The effect of this bit setting is defined in Table 54.15. 0: Data swapping in word (16-bit) units is disabled 1: Data swapping in word (16-bit) units is enabled	R/W
2	LWS	Display List Data Swapping in long word Units The effect of this bit setting is defined in Table 54.15. 0: Data swapping in long word (32-bit) units is disabled 1: Data swapping in long word (32-bit) units is enabled	R/W
30:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31	IND	Enabling independent swap setting per WPF This bit is available only for VI6_DL_SWAP1. This bit is reserved for VI6_DL_SWAP0. 0: Display list swap for WPF1 is specified by LWS, WDS, and BTS in VI6_DL_SWAP0. 1: Display list swap for WPF1 is specified by LWS, WDS, and BTS in VI6_DL_SWAP1.	R/W

Table 54.15 shows the data order before and after swapping according to the long word, word, and byte swapping settings.

When data order in memory for each format is the same as Table 54.40, set 111b to {LWS, WDS, BTS}. If data order is not the same as the definition, change data order within 8-byte unit by these bits as shown in Table 54.15.

Table 54.15 Changing data order according to Display List Swap register

Data order in memory											
Byte address	8n + 0	8n + 1	8n + 2	8n + 3	8n + 4	8n + 5	8n + 6	8n + 7	*_LWS	*_WDS	*_BTS
Data	0	1	2	3	4	5	6	7	1	1	1
	1	0	3	2	5	4	7	6	1	1	0
	2	3	0	1	6	7	4	5	1	0	1
	3	2	1	0	7	6	5	4	1	0	0
	4	5	6	7	0	1	2	3	0	1	1
	5	4	7	6	1	0	3	2	0	1	0
	6	7	4	5	2	3	0	1	0	0	1
	7	6	5	4	3	2	1	0	0	0	0

↓

Data order defined in Table 54.40								
Byte address	8n + 0	8n + 1	8n + 2	8n + 3	8n + 4	8n + 5	8n + 6	8n + 7
Data	0	1	2	3	4	5	6	7

54.4.5.4 VI6_DL_EXT_CTRL0 : Extended Display List-0 Control Register

Base address: VSPD = 0x920E_0000

Offset address: 0x011C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NWE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	POLINT[5:0]						—	—	DLPRI	EXPRI	—	—	—	EXT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EXT	Extended Display List for WPF0*1 Enables or disables the extended display list function. When extended display lists are used, the display list header size is 96 bytes; when they are not used, the header size is 80 bytes. 0: No extended display lists are used. 1: Extended display lists are used.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	EXPRI	Display List Control 1 Always set this bit to 0.	R/W
5	DLPRI	Display List Control 0 Always set this bit to 1.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	POLINT[5:0]	Extended Display List Command Control Always set these bits to 2.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	NWE	No Wait for Polling When this bit is set to 1, the polling condition for extended display lists is always assumed to be true.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When using extended display lists, be sure to also use normal display list mode (VI6_DL_CTRL.DLE0); executing only extended display lists is not possible.
When the header-less display list mode is activated, this bit should be set to 0. The extended display list cannot be used with the header-less display list mode.
Extended Display List is available only for VSPD/LIF0 to realized AUTO-FLD or AUTODISP.

54.4.5.5 VI6_DL_BODY_SIZE0 : Display List Body Size Register 0

Base address: VSPD = 0x920E_0000

Offset address: 0x0120

Bit position:	31							24							16														0
Bit field:	—	—	—	—	—	—	—	UPD0	—	—	—	—	—	—	—	BS0[16:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
16:0	BS0[16:0]	Header-less Display List Body Size (WPF0) These bits are used for specifying the body size of the display list in case of header-less display list mode. The unit of the size is byte. The value should be set in multiples of 8.	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	UPD0	Update Flag This bit controls the download of the display list at the next downloading timing in case that the header-less display list mode is used. Set this bit to 1 when display list need to be downloaded for next frame. This bit is automatically cleared to 0 after VI6_DL_HDR_ADDRn and VI6_DL_BODY_SIZE0.BS0 is downloaded in H/W side. When this bit is set to 1, the value of VI6_DL_HDR_ADDR0 and VI6_DL_BODY_SIZE0.BS0 should not be changed. 0: Updating display list for the next frame is not reserved. 1: Updating display list for the next frame is reserved.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

See section 54.6.7.2. Controlling Two Register Planes using Display Lists for detail.

54.4.5.6 VI6_DL_HDR_REF_ADDR0 : Display List-0 Header Reference Address Register

Base address: VSPD = 0x920E_0000

Offset address: 0x0130

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

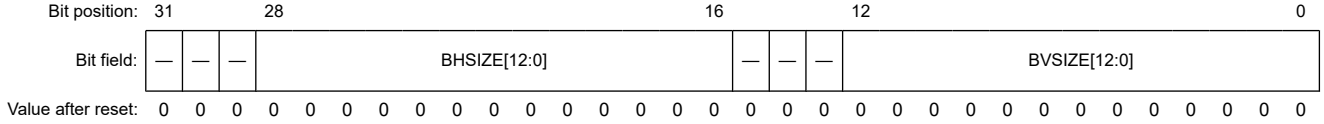
Bit	Symbol	Function	R/W
31:0	n/a	Display List-n reference Header Address Following value is read out from the VI6_DL_HDR_REF_ADDR0 for each period. 1. When H/W is reading display list from external memory, header address of the display list referred by VSPD-H/W is read out. 2. When H/W is not reading display list from external memory, the value of VI6_DL_HDR_ADDR0 is read out. For details, see section 54.6.7.1. Operation flow of VSPD and DU.	R

54.4.6 RPF Control Registers

54.4.6.1 VI6_RPFn_SRC_BSIZE : RPFn Basic Read Size Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0300 + 0x100 × n



Bit	Symbol	Function	R/W
12:0	BVSIZE[12:0]	Vertical Size of RPF Basic Read Area*1 These bits specify the vertical size of the basic area to be read from the external RAM by the RPFn. When the input format is YCbCr4:2:0, specify the size in 2-pixel units. Specify a value equal to or smaller than the extended read size (VI6_RPFn_SRC_ESIZE.EVSIZE).	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	BHSIZE[12:0]	Horizontal Size of RPF Basic Read Area*1 These bits specify the horizontal size of the basic area to be read from the external RAM by the RPFn. When the input format is YCbCr4:2:2 or YCbCr4:2:0, specify the size in 2-pixel units. Specify a value equal to or smaller than the extended read size (VI6_RPFn_SRC_ESIZE.EHSIZE).	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. A value from 1 to 8190 can be specified.

Figure 54.6 shows the relationship between the basic read size and extended read size. The RPF reads data from the source memory area specified by the basic read size. The RPF repeats reading the basic read area in the horizontal and vertical directions up to the extended read size and sends the read data to the processing modules in the VSPD.

For basic read size reading, the reading start address, called the RPFn source image storing address, should be specified in VI6_RPFn_SRCM_ADDR_*. In the memory area where the basic read area image is stored, the distance (number of bytes) between addresses for lines n and n + 1 of two-dimensional image data, called the memory stride, should be specified in VI6_RPFn_SRCM_PSTRIDE.

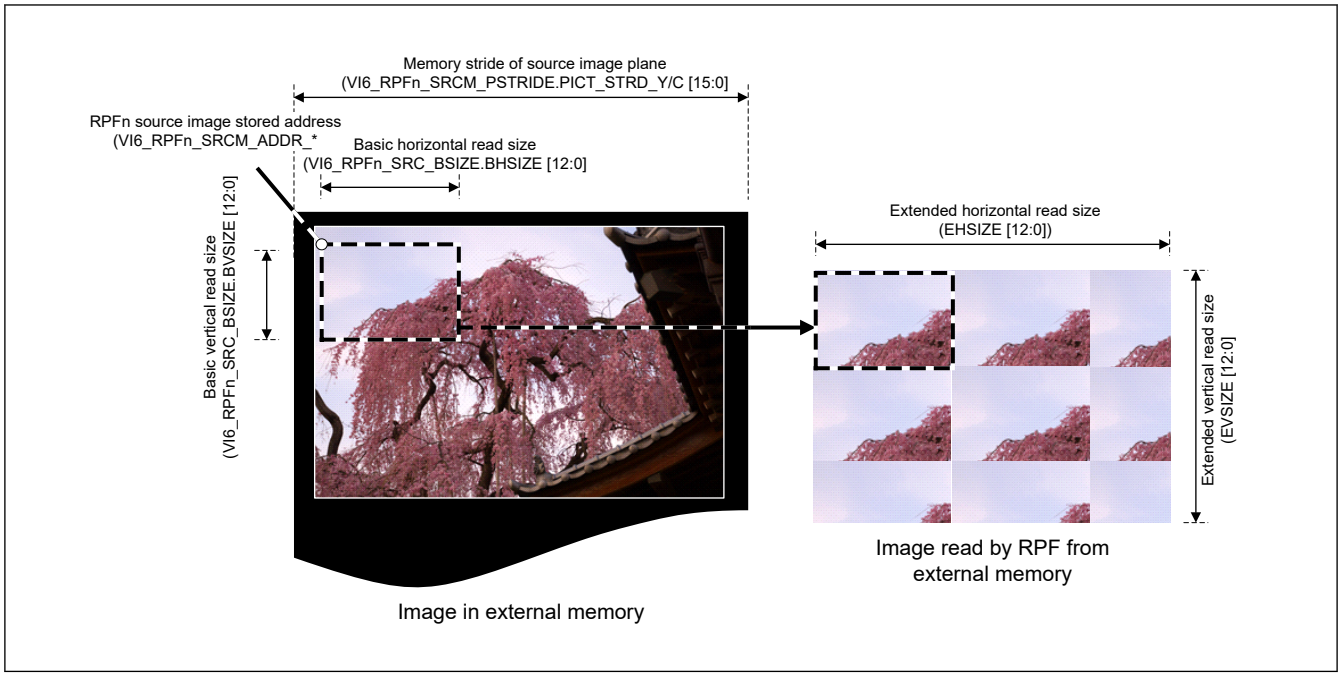


Figure 54.6 Relationship between Basic Read Size and Extended Read Size

Also see the following sections.

- section 54.4.6.2. VI6_RPFn_SRC_ESIZE : RPFn Extended Read Size Register (n = 0, 1)
- section 54.4.6.14. VI6_RPFn_SRCM_PSTRIDE : RPFn Source Picture Memory Stride Setting Register (n = 0, 1)
- section 54.4.6.16. VI6_RPFn_SRCM_ADDR_Y : RPFn Source Y/RGB Address Register (n = 0, 1)
- section 54.4.6.17. VI6_RPFn_SRCM_ADDR_C0 : RPFn Source Chroma Address Register 0 (n = 0, 1)
- section 54.4.6.18. VI6_RPFn_SRCM_ADDR_C1 : RPFn Source Chroma Address Register 1 (n = 0, 1)

54.4.6.2 VI6_RPFn_SRC_ESIZE : RPFn Extended Read Size Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0304 + 0x100 × n

Bit position:	31			28												16			12												0
Bit field:	—	—	—	EHSIZE[12:0]												—	—	—	EVSIZE[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	EVSIZE[12:0]	RPF Extended Vertical Read Size These bits specify the vertical size of the extended read area to which the RPFn reads data from the external RAM. As shown in Figure 54.6, the basic read area image is repeatedly placed in the extended read area; in the EVSIZE bits, specify a value not smaller than the vertical size of the basic read area. When the input format is YCbCr4:2:0, specify the size in 2-pixel units (an even value). A value from 1 to 1080 can be specified.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	EHSIZE[12:0]	RPF Extended Horizontal Read Size These bits specify the horizontal size of the extended read area to which the RPFn reads data from the external RAM. As shown in Figure 54.6, the basic read area image is repeatedly placed in the extended read area; in the EHSIZE bits, specify a value not smaller than the horizontal size of the basic read area. When the input format is YCbCr4:2:2 or YCbCr4:2:0, specify the size in 2-pixel units (an even value). A value from 1 to 1920 can be specified.	R/W

Bit	Symbol	Function	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

VI6_RPFn_SRC_ESIZE specifies the extended size for RPFn. The extended horizontal and vertical sizes should be equal to or greater than the basic sizes specified in VI6_RPFn_SRC_BSIZE. The RPF internal data processing described later and image processing described in section 54.4.8. DPR Control Registers and later sections are all applied to the image in the extended read size shown on the right side in Figure 54.6.

54.4.6.3 VI6_RPFn_INFMT : RPFn Input Format Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0308 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	VIR	—	—	—	—	—	—	—	—	—	—	—	CIPM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPYCS	SPUVS	CEXT[1:0]		RDTM[2:0]			CSC	—	RDFMT[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	RDFMT[6:0]	RPF Input Image Format Setting These bits select the format of the image input from the external RAM to the RPF0. Select a value corresponding to the desired format from those shown in Table 54.17, Table 54.18, and Table 54.19. When the virtual input function is used (VIR = 1), the color information for the virtual input should be specified in VI6_RPF0_VRTCOL_SET. If this color information is in the RGB format, set the RDFMT bits to 19. If the color information is in the YCbCr format, set these bits to 64.*1	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CSC	Color Space Conversion Enable Enables or disables color space conversion between YCbCr and RGB to be executed in RPF0. The characteristics of color space conversion are determined by the RDTM bit setting.*2 When using the virtual input (VIR = 1), set this bit to 0. 0: Color space conversion is disabled. 1: Color space conversion is enabled.	R/W
11:9	RDTM[2:0]	CSC Conversion Expression Setting These bits select the expression used for color space conversion. The conversion direction is RGB → YCbCr when RGB is selected through the RDFMT bits; the direction is YCbCr → RGB when YCbCr is selected. 0 0 0: BT.601 YCbCr [16,235/240] ↔ RGB [0,255] 0 0 1: BT.601 YCbCr [0,255] ↔ RGB [0,255] 0 1 0: BT.709 YCbCr [16,235/240] ↔ RGB [0,255] 0 1 1: BT.709 YCbCr [16,235/240] ↔ RGB [16,235] Others: Setting prohibited.	R/W
13:12	CEXT[1:0]	Lower-Bit Color Data Extension Method Setting When an RGB input format where each color component is expressed in less than 8 bits are selected from Table 54.17 through the RDFMT bits, VSPD internally extends each color component to 8 bits before using the data. These bits select this extension method. 0 0: Lower-order bits of color data are extended with 0. 0 1: Upper-order bits of color data are copied to the lower-order bits. 1 0: Lower-order bits of color data are extended with 0. The maximum value is limited to 0xFF. 1 1: Setting prohibited.	R/W

Bit	Symbol	Function	R/W
14	SPUVS	RPF Input Mode Setting 2 When the input format is NV61, set this bit to 1 and set the RDFMT bits to 65 (0x41). When the input format is NV21, set this bit to 1 and set the RDFMT bits to 66 (0x42). When the input format is YVYU, set this bit and the SPYCS bit to 1 and set the RDFMT bits to 71 (0x47). In other cases, set this bit to 0.	R/W
15	SPYCS	RPF Input Mode Setting 1 When the input format is YUY2, set this bit to 1 and set the RDFMT bits to 71 (0x47). When the input format is YVYU, set this bit and the SPUVS bit to 1 and set the RDFMT bits to 71 (0x47). In other cases, set this bit to 0.	R/W
16	CIPM	Horizontal Chrominance Interpolation Method Setting Image data is processed in the YCbCr444 format inside VSPD in case of YCbCr color space. When the chrominance format of the input image is YCbCr422 or YCbCr420, data is up-sampled as shown in Figure 54.7 for internal processing. This bit specifies the method of up-sampling for this purpose. 0: The nearest-neighbor method is used for horizontal chrominance interpolation. 1: The bilinear method is used for horizontal chrominance interpolation.	R/W
27:17	—	These bits are read as 0. The write value should be 0.	R/W
28	VIR	Virtual Input Enable Enables or disables the virtual input function of the RPF0. The image to be processed by the RPF0 is usually read from the external memory by the MAU. Instead of this input, the virtual input function generates a single-color image within the RPF0 and sends it to the modules in VSPD. When the virtual input function is enabled, the fixed value specified in VI6_RPF0_VRTCOL_SET is used as the input to the RPF0. While the virtual input function is enabled, data is not read from the external memory; that is, the α plane is not read and the IROP calculation thus cannot be executed. In this case, set VI6_RPF0_ALPH_SEL.ASEL to 4. Neither the color space conversion through CSC nor the color keying described in section 54.4.6.11. VI6_RPFn_CKEY_CTRL : RPFn Color Keying Control Register (n = 0, 1) can be used. 0: RPF0 uses general input. 1: RPF0 uses virtual input.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Number of input pixels
When YCbCr4:2:2 is selected through the RDFMT bits, the horizontal size of the input image should be specified in 2-pixel units. When YCbCr4:2:0 is selected, the vertical and horizontal sizes should be specified in 2-pixel units. Observe these restrictions when specifying the image size in VI6_RPF0_SRC_BSIZE and VI6_RPF0_SRC_ESIZE.
Note 2. Note on color space settings.

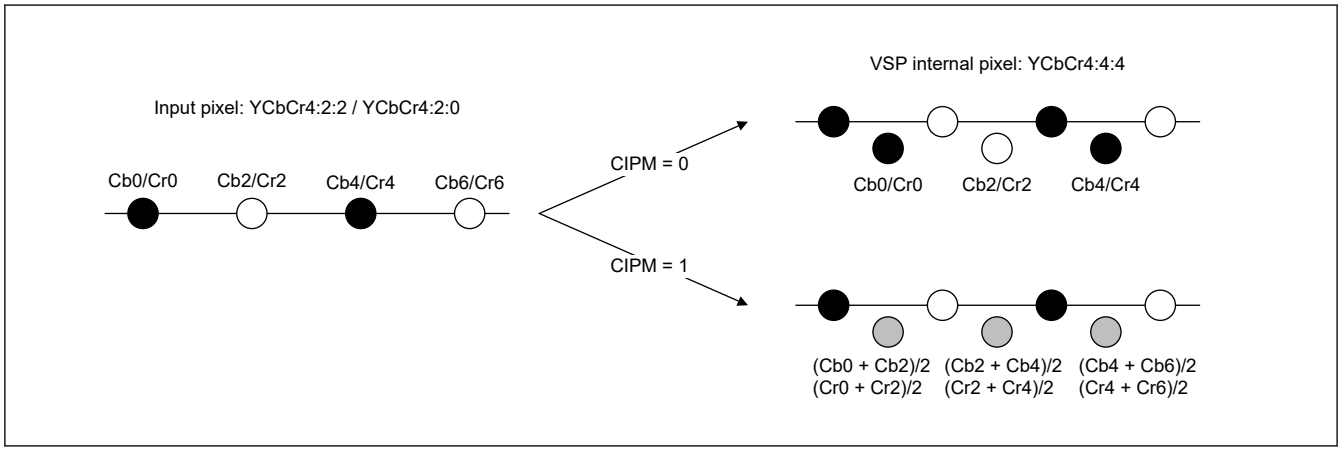


Figure 54.7 Chrominance interpolation methods selectable through CIPM setting

The color space for the image output from the RPF to VSPD internal modules is determined by the combination of the color space for the image input to the RPF, which is selected through the VI6_RPFn_INFMT.RDFMT setting, and the enabled or disabled state of the color space conversion function, which is selected through the VI6_RPFn_INFMT.CSC setting (Table 54.16). For example, when the image input to the RPF is in the YCbCr format, the RPF outputs data to VSPD internal modules in the YCbCr format if color space conversion is disabled through the CSC bit, and the RPF outputs data in the RGB format if color space conversion is enabled. When the image input to the RPF is in the RGB format, the relationship

between the output format and the color space conversion setting is the opposite of the YCbCr case. For some VSPD internal modules, the YCbCr format is recommended for image processing because of the characteristics of the processing, or the same color space needs to be specified between multiple RPF outputs. In these cases, set VI6_RPFn_INFMT.RDFMT and VI6_RPFn_INFMT.CSC appropriately so that the RPFs can output the required color space according to the color space conditions described above.

Table 54.16 RPFn input color space and output color space

RPFn input color space (VI6_RPFn_INFMT. RDFMT)		Color space conversion setting (VI6_RPFn_INFMT.CSC)		RPFn output color space
RGB	(0x00 to 0x3F)*1	Disabled	(0)*1	RGB
		Enabled	(1)*1	YCbCr
YCbCr	(0x40 to 0x7F)*1	Disabled	(0)*1	YCbCr
		Enabled	(1)*1	RGB

Note 1. Value specified in the register

A color space conversion function equivalent to that in the RPFn is also provided by the WPF. As shown in Table 54.16, the color space (YCbCr or RGB) output from the RPF becomes the input format for the WPF. Here, the color space of the output image obtained by the color space conversion function of the WPF must match the color space of the format specified through VI6_WPFn_OUTFMT.WRFMT.

Figure 54.8 shows the relationship between the input/output format and color space. The input color space for the RPF is determined when the input image format for the RPF is specified through the RDFMT bits. The color space for the image output from the RPF to subsequent VSPD internal modules depends on the combination of the RPF input format and CSC (color space conversion function) enabled or disabled state in the RPF as shown in Table 54.16 and Figure 54.8. The user should first determine whether the image processing in the VSPD is done in YCbCr or RGB, and then specify the RPF input format and CSC enabled or disabled state to obtain the desired color space. The color space of the RPF output image is also that of the WPF input image; the color space of the data output from the WPF to the outside of VSPD depends on the combination of the WPF input color space and the enabled or disabled state of the CSC implemented in the WPF as shown in Figure 54.8. The color space of the WPF output image must match that of the WPF output format (determined by VI6_WPFn_OUTFMT.WRFMT). For example, in the flow shown in Figure 54.8, YCbCr should not be specified as the WPF output format regardless of the fact that the color space of the WPF output image is in RGB format.

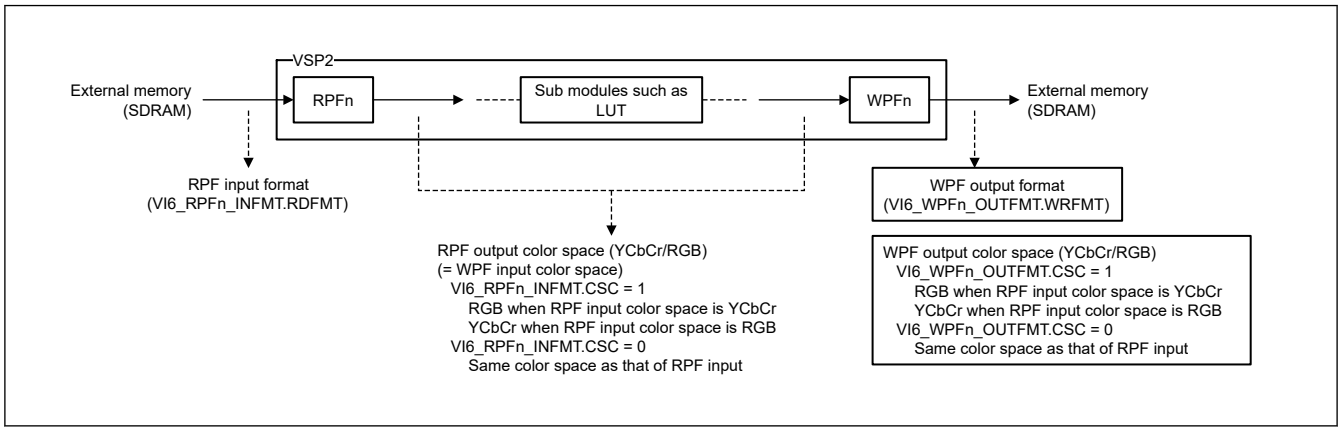


Figure 54.8 Relationship between input/output format and color space

Table 54.17 Packed formats for RPF input (1 of 2)

RDFMT [6:0]	Bit per pixel	Phase	upper row - address / bottom row - bit field																													
			n								n + 1								n + 2								n + 3							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0x00	8	—	R0	R0	R0	G0	G0	G0	B0	B0	R1	R1	R1	G1	G1	G1	B1	B1	R2	R2	R2	G2	G2	G2	B2	B2	R3	R3	R3	G3	G3	G3	B3	B3
0x01	12	—					R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0					R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1
0x02			R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0					R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1				
0x03	—	—	Reserved								Reserved								Reserved								Reserved							

Table 54.17 Packed formats for RPF input (2 of 2)

RDFMT [6:0]	Bit per pixel	Phase	upper row - address / bottom row - bit field																													
			n								n + 1								n + 2								n + 3							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0x04	15	—		R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0		R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1
0x05			R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0		R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	
0x06	16	—	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1
0x07	18	—	A	A	A	A	A	A	A	A							R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0
0x08			R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0							A	A	A	A	A	A	A	A
0x09									R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	A	A	A	A	A	A	A	A
0x0A			A	A	A	A	A	A	A	A	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0						
0x0B			A	A	A	A	A	A	A	A			R0	R0	R0	R0	R0	R0			G0	G0	G0	G0	G0	G0			B0	B0	B0	B0	B0	B0
0x0C					R0	R0	R0	R0	R0	R0			G0	G0	G0	G0	G0	G0			B0	B0	B0	B0	B0	B0	A	A	A	A	A	A	A	A
0x0D			A	A	A	A	A	A	A	A	R0	R0	R0	R0	R0	R0			G0	G0	G0	G0	G0	G0			B0	B0	B0	B0	B0	B0		
0x0E			R0	R0	R0	R0	R0	R0			G0	G0	G0	G0	G0	G0			B0	B0	B0	B0	B0	B0			A	A	A	A	A	A	A	A
0x0F		0							R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0							R1	R1
		1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1							R2	R2	R2	R2	R2	R2	G2	G2	G2	G2
		2	G2	G2	B2	B2	B2	B2	B2	B2							R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3
0x10		0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0							R1	R1	R1	R1	R1	R1	G1	G1
		1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1							R2	R2	R2	R2	R2	R2	G2	G2	G2	G2	G2	G2	B2	B2	B2	B2
		2	B2	B2							R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3	B3						
0x11		0			R0	R0	R0	R0	R0	R0			G0	G0	G0	G0	G0	G0			B0	B0	B0	B0	B0	B0			R1	R1	R1	R1	R1	R1
		1			G1	G1	G1	G1	G1	G1			B1	B1	B1	B1	B1	B1			R2	R2	R2	R2	R2	R2			G2	G2	G2	G2	G2	G2
		2			B2	B2	B2	B2	B2	B2			R3	R3	R3	R3	R3	R3			G3	G3	G3	G3	G3	G3			B3	B3	B3	B3	B3	B3
0x12		0	R0	R0	R0	R0	R0	R0			G0	G0	G0	G0	G0	G0			B0	B0	B0	B0	B0	B0			R1	R1	R1	R1	R1	R1		
		1	G1	G1	G1	G1	G1	G1			B1	B1	B1	B1	B1	B1			R2	R2	R2	R2	R2	R2			G2	G2	G2	G2	G2	G2		
		2	B2	B2	B2	B2	B2	B2			R3	R3	R3	R3	R3	R3			G3	G3	G3	G3	G3	G3			B3	B3	B3	B3	B3	B3		
0x13	24	—	A	A	A	A	A	A	A	A	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0
0x14			R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	A	A	A	A	A	A	A	A
0x15		0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	R1	R1	R1
		1	G1	G1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1	B1	B1	R2	R2	R2	R2	R2	R2	R2	R2	G2	G2	G2	G2	G2	G2	G2	G2
		2	B2	B2	B2	B2	B2	B2	B2	B2	R3	R3	R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3	B3	B3
0x16	18	—								R0	R0	R0	R0	R0	R0	G0	G0	G0								G0	G0	G0	B0	B0	B0	B0	B0	B0
0x17																	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0
0x18	24	0	B0	B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	R0	B1	B1	B1	B1	B1	B1	B1	B1
		1	G1	G1	G1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1	R1	R1	R1	B2	B2	B2	B2	B2	B2	B2	B2	G2	G2	G2	G2	G2	G2	G2	G2
		2	R2	R2	R2	R2	R2	R2	R2	R2	B3	B3	B3	B3	B3	B3	B3	B3	G3	G3	G3	G3	G3	G3	G3	G3	R3	R3	R3	R3	R3	R3	R3	R3
0x19	12	—	A	A	A	A	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	A	A	A	A	R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1
0x1A		—	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	A	A	A	A	R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1	A	A	A	A
0x1B	15	—	A	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	A	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1
0x1C		—	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	A	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	A
0x1D	12	—	A	A	A	A	B0	B0	B0	B0	G0	G0	G0	G0	R0	R0	R0	R0	A	A	A	A	B1	B1	B1	B1	G1	G1	G1	G1	R1	R1	R1	R1
0x1E		—	B0	B0	B0	B0	G0	G0	G0	G0	R0	R0	R0	R0	A	A	A	A	B1	B1	B1	B1	G1	G1	G1	G1	R1	R1	R1	R1	A	A	A	A
0x1F	15	—	A	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	A	B1	B1	B1	B1	B1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1
0x20		—	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	A	B1	B1	B1	B1	B1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1	A
0x21	18	0			B0	B0	B0	B0	B0	B0			G0	G0	G0	G0	G0	G0			R0	R0	R0	R0	R0	R0			B1	B1	B1	B1	B1	B1
		1			G1	G1	G1	G1	G1	G1			R1	R1	R1	R1	R1	R1			B2	B2	B2	B2	B2	B2			G2	G2	G2	G2	G2	G2
		2			R2	R2	R2	R2	R2	R2			B3	B3	B3	B3	B3	B3			G3	G3	G3	G3	G3	G3			R3	R3	R3	R3	R3	R3
0x22	24	—	A	A	A	A	A	A	A	A	B0	B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	R0
0x23	16	—																	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0
0x24 to 0x2F	—	—	Reserved								Reserved								Reserved								Reserved							
0x30	*1	*1	*1								*1								*1								*1							
0x31 to 0x3F	—	—	Reserved								Reserved								Reserved								Reserved							

Note 1. When the RDFMT[6:0] bits are set to 0x30, see Table 54.18 for the additional settings and the packed formats

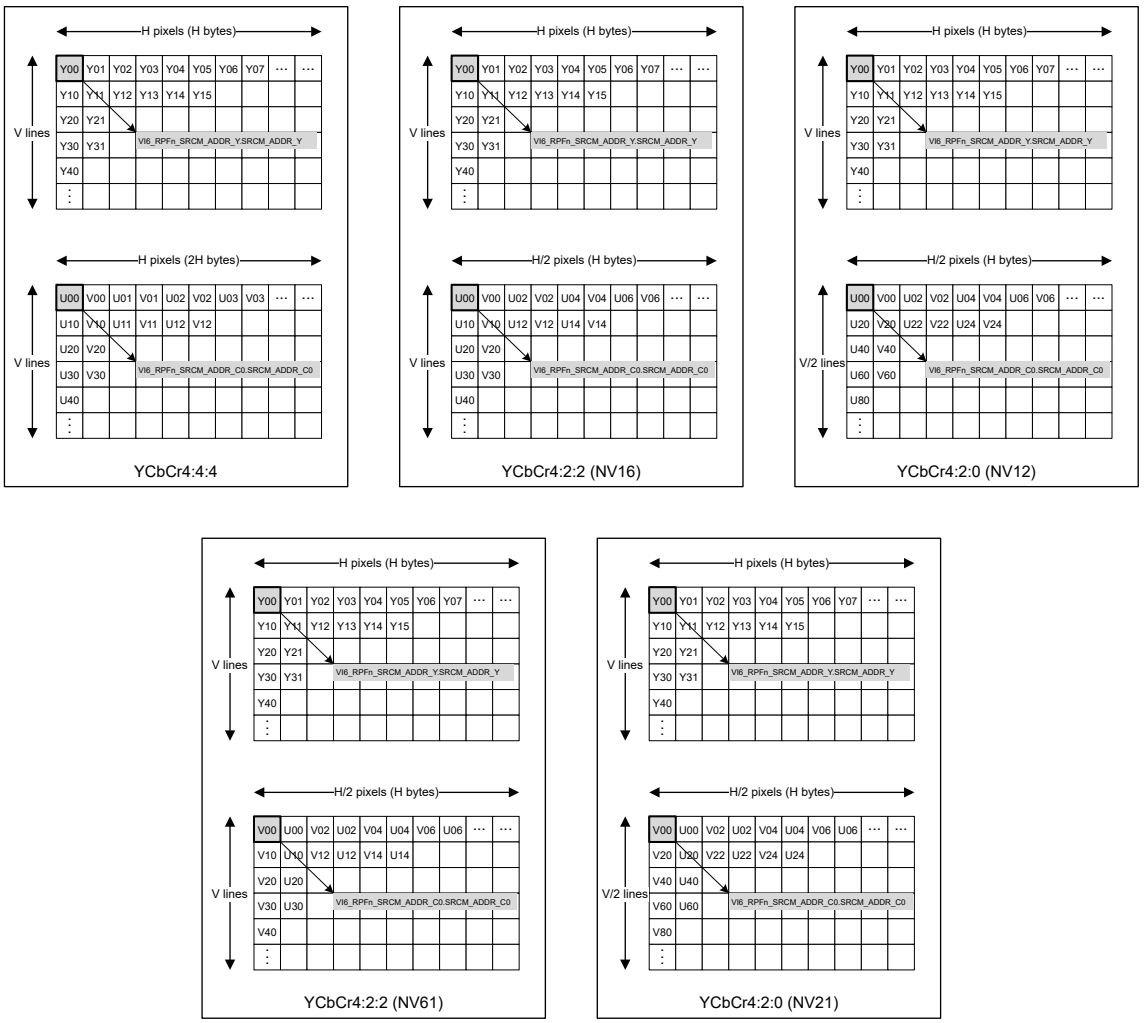
Table 54.18 Packed formats of RGB10, RGB10A2, A2RGB10

RDFMT [6:0]	Format name	BYPP_M1[2:0] CPOS[31:0] CLEN[31:0]	upper row - address / bottom row - bit field																													
			n								n + 1								n + 2								n + 3							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0x30	RGB10	0x3 0x000A1400 0x0A0A0A00	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0		
0x30	RGB10A2	0x3 0x000A141E 0x0A0A0A02	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0	A0	A0
0x30	A2RGB10	0x3 0x020C1600 0x0A0A0A02	A0	A0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0

Table 54.19 Packed YCbCr formats for RPF input

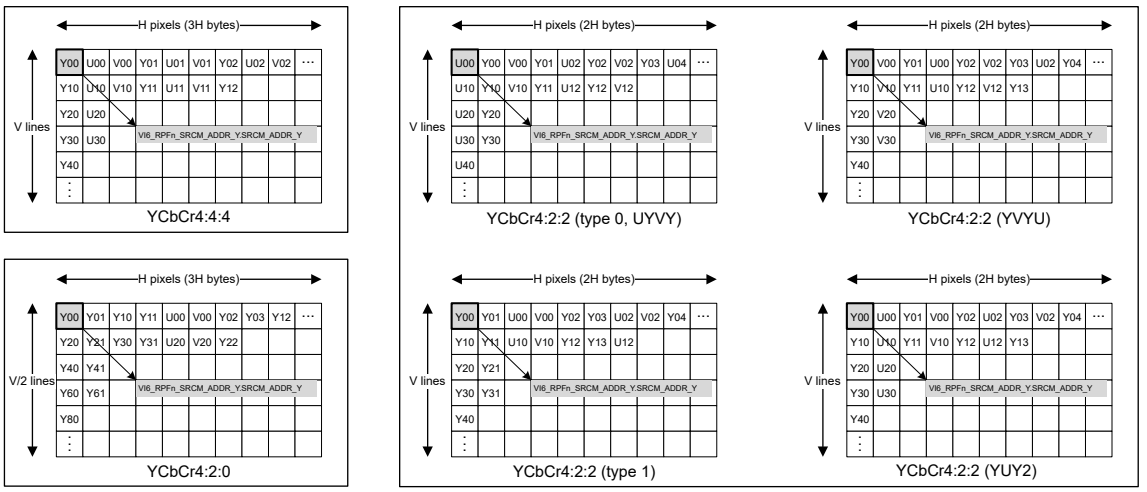
RDFMT[6:0]	Packed YCbCr input format	Reference
0x40	YCbCr4:4:4 semi-planar	Figure 54.9*4
0x41	YCbCr4:2:2 semi-planar (NV16, NV61*1)	
0x42	YCbCr4:2:0 semi-planar (NV12, NV21*1)*5	
0x43 to 0x45	Reserved	—
0x46	YCbCr4:4:4 interleaved	Figure 54.10*4
0x47	YCbCr4:2:2 interleaved type 0 (UYVY, YUY2*2, YVYU*3)	
0x48	YCbCr4:2:2 interleaved type 1	
0x49	YCbCr4:2:0 interleaved*6	
0x4A	YCbCr4:4:4 planar	Figure 54.11*4
0x4B	YCbCr4:2:2 planar (YV16)	
0x4C	YCbCr4:2:0 planar (YV12, I420)*5	
0x4D to 0x7E	Reserved	—
0x7F	Reserved	—

Note 1. When the input format is NV61 or NV21, set the SPUVS bit to 1.
Note 2. When the input format is YUY2, set the SPYCS bit to 1.
Note 3. When the input format is YVYU, set the SPUVS bit to 1 and SPYCS bit to 1.
Note 4. Figure 54.12 shows the definition of memory address for each pixel in Figure 54.9 to Figure 54.11.
Note 5. Each line of C plane is read twice, so byte/pixel of YCbCr420 is 2 byte/pixel (same as YCbCr422).
Note 6. Each line of plane is read twice, so byte/pixel of YCbCr420ITL is 3 byte/pixel (same as YCbCr444ITL).



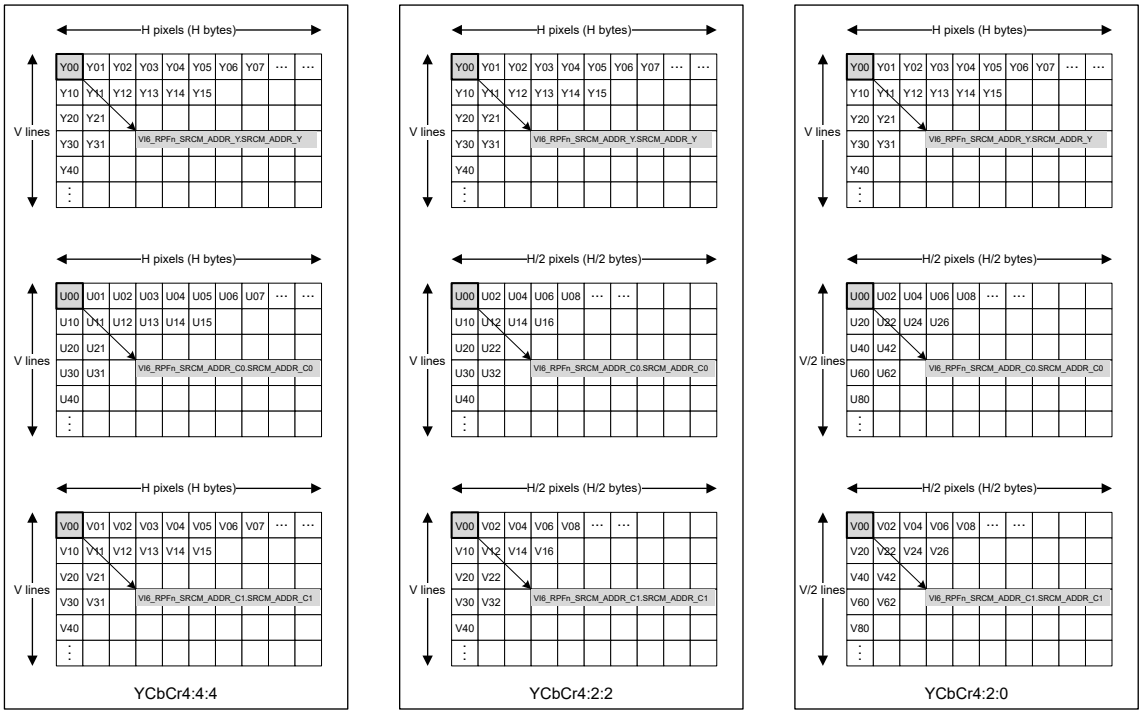
Note: This figure is for 8 bpc.

Figure 54.9 YCbCr Semi-Planar formats



Note: This figure is for 8 bpc.

Figure 54.10 YCbCr Interleaved formats



Note: This figure is for 8 bpc.

Figure 54.11 YCbCr Planar formats

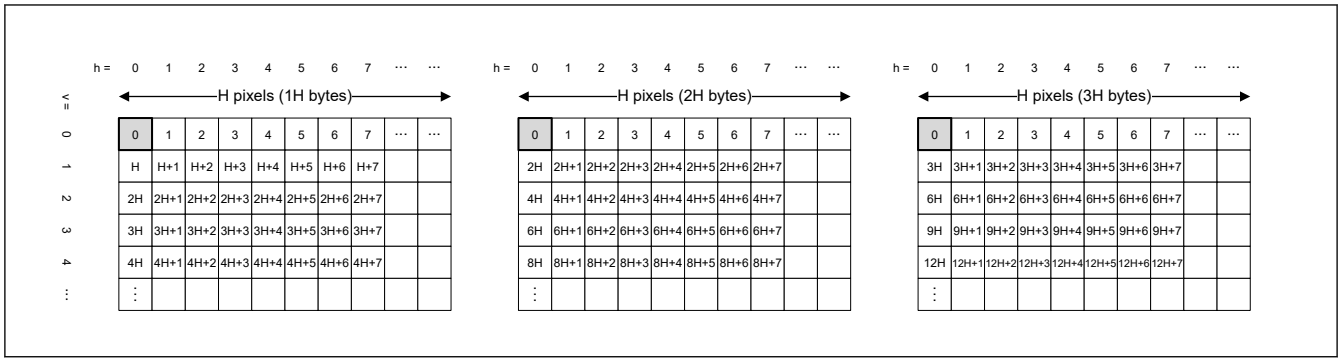


Figure 54.12 Memory address for the corresponding pixel's position

Legend of Figure 54.12

- H is horizontal image size in pixel unit.
- 1H, 2H, and 3H is corresponding to the case 1 pixel has 1 byte, 2 bytes, and 3 bytes of data, respectively.

54.4.6.4 VI6_RPFn_DSWAP : RPFn Data Swapping Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x030C + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	A_LLS	A_LWS	A_WDS	A_BTS	—	—	—	—	P_LLS	P_LWS	P_WDS	P_BTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	P_BTS	Picture Plane Data Swapping in Byte Units*1 The effect of this bit setting is defined in Table 54.20. 0: Data swapping in Byte (8-bit) units is disabled. 1: Data swapping in Byte (8-bit) units is enabled.	R/W
1	P_WDS	Picture Plane Data Swapping in Word Units*1 The effect of this bit setting is defined in Table 54.20. 0: Data swapping in Word (16-bit) units is disabled. 1: Data swapping in Word (16-bit) units is enabled.	R/W
2	P_LWS	Picture Plane Data Swapping in long word Units*1 The effect of this bit setting is defined in Table 54.20. 0: Data swapping in long word (32-bit) units is disabled. 1: Data swapping in long word (32-bit) units is enabled.	R/W
3	P_LLS	Picture Plane Data Swapping in LONG LWORD Units*1 The effect of this bit setting is defined in Table 54.20. 0: Data swapping in LONG LWORD (64-bit) units is disabled. 1: Data swapping in LONG LWORD (64-bit) units is enabled.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	A_BTS	α Plane Data Swapping in Byte Units The effect of this bit setting is defined in Table 54.20. 0: Data swapping in Byte (8-bit) units is disabled. 1: Data swapping in Byte (8-bit) units is enabled.	R/W

Bit	Symbol	Function	R/W
9	A_WDS	α Plane Data Swapping in Word Units The effect of this bit setting is defined in Table 54.20. 0: Data swapping in Word (16-bit) units is disabled. 1: Data swapping in Word (16-bit) units is enabled.	R/W
10	A_LWS	α Plane Data Swapping in long word Units The effect of this bit setting is defined in Table 54.20. 0: Data swapping in long word (32-bit) units is disabled. 1: Data swapping in long word (32-bit) units is enabled.	R/W
11	A_LLS	α Plane Data Swapping in LONG LWORD Units The effect of this bit setting is defined in Table 54.20. 0: Data swapping in LONG LWORD (64-bit) units is disabled. 1: Data swapping in LONG LWORD (64-bit) units is enabled.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This register is available for only Luma plane (Y) in the case input data format is YUV Planar or Semi-Planar and input bit-depth is 2 bpc (IPBD_Y ≠ 0 or IPBD_C ≠ 0), and available for both Luma plane (Y) and Chroma plane (CbCr/U/V) in other cases.

When the virtual input function of the RPFn is used (VI6_RPFn_INFMT.VIR = 1), this register setting is ignored. Swapping of RPF input data can be specified separately for the α plane and picture plane.

Table 54.20 shows the data order before and after swapping according to the long long word, long word, word, and byte swapping settings.

When data order in memory for each format is the same as Table 54.17 for RGB format and Figure 54.9 to Figure 54.12 for YCbCr format, set 1111b to {*_LLS, *_LWS, *_WDS, *_BTS}. If data order is not the same as the definition, change data order within 16-byte unit by these bits as shown in Table 54.20.

Table 54.20 Changing data order according to swap register

Data order in memory																				
Byte address	16n + 0	16n + 1	16n + 2	16n + 3	16n + 4	16n + 5	16n + 6	16n + 7	16n + 8	16n + 9	16n + 10	16n + 11	16n + 12	16n + 13	16n + 14	16n + 15	*_LLS	*_LWS	*_WDS	*_BTS
Data	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1	1	1	1
	1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14	1	1	1	0
	2	3	0	1	6	7	4	5	10	11	8	9	14	15	12	13	1	1	0	1
	3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12	1	1	0	0
	4	5	6	7	0	1	2	3	12	13	14	15	8	9	10	11	1	0	1	1
	5	4	7	6	1	0	3	2	13	12	15	14	9	8	11	10	1	0	1	0
	6	7	4	5	2	3	0	1	14	15	12	13	10	11	8	9	1	0	0	1
	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	1	0	0	0
	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	0	1	1	1
	9	8	11	10	13	12	15	14	1	0	3	2	5	4	7	6	0	1	1	0
	10	11	8	9	14	15	12	13	2	3	0	1	6	7	4	5	0	1	0	1
	11	10	9	8	15	14	13	12	3	2	1	0	7	6	5	4	0	1	0	0
	12	13	14	15	8	9	10	11	4	5	6	7	0	1	2	3	0	0	1	1
	13	12	15	14	9	8	11	10	5	4	7	6	1	0	3	2	0	0	1	0
	14	15	12	13	10	11	8	9	6	7	4	5	2	3	0	1	0	0	0	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0

↓

Table 54.17 for RGB format and Figure 54.9 to Figure 54.12 for YCbCr format																
Byte address	16n + 0	16n + 1	16n + 2	16n + 3	16n + 4	16n + 5	16n + 6	16n + 7	16n + 8	16n + 9	16n + 10	16n + 11	16n + 12	16n + 13	16n + 14	16n + 15
Data	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

54.4.6.5 VI6_RPFn_LOC : RPFn Display Location Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0310 + 0x100 × n

Bit position:	31			28												16			12												0
Bit field:	—	—	—	HCOORD[12:0]												—	—	—	VCOORD[12:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	VCOORD[12:0]	Vertical Coordinate of Sublayer Display Location on Master Layer*1 These bits specify the top-end location of the sublayer displayed by the RPFn and the subsequent module connected through the DPR. Specify the vertical coordinate of the location in pixel units with the top-end pixel of the master layer set at coordinate 0. When the RPFn is the master layer, set these bits to 0. If the sublayer extends beyond the master layer according to the VCOORD setting, the extended section is cut off at the bottom-end of the master layer. Even in this case, however, a bus transfer that is unnecessary for output image generation is executed since the whole sublayer data is read from the external memory. Appropriate coordinate setting is required so that the sublayer does not extend beyond the bottom-end of the master layer.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	HCOORD[12:0]	Horizontal Coordinate of Sublayer Display Location on Master Layer*1 These bits specify the left-end location of the sublayer displayed by the RPFn and the subsequent module connected through the DPR. Specify the horizontal coordinate of the location in pixel units with the left-end pixel of the master layer set at coordinate 0. When the RPFn is the master layer, set these bits to 0. If the sublayer extends beyond the master layer according to the HCOORD setting, the extended section is cut off at the right-end of the master layer. Even in this case, however, a bus transfer that is unnecessary for output image generation is executed since the whole sublayer data is read from the external memory. Appropriate coordinate setting is required so that the sublayer does not extend beyond the right-end of the master layer.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. A value from 0 to 8189 can be specified.

Figure 54.13 shows an example of RPF1 offsets with respect to master layer RPF0. Although this figure only shows sublayers RPF1, specify offsets for all RPFs other than the master layer in the same way as shown in this example.

Whether an RPFn is the master layer or a sublayer is determined through the selection of the source RPF for WPFn (the VI6_WPFn_SRCRPF setting). For details, see section 54.4.7.1. VI6_WPF0_SRCRPF : WPF0-Source-RPF Register.

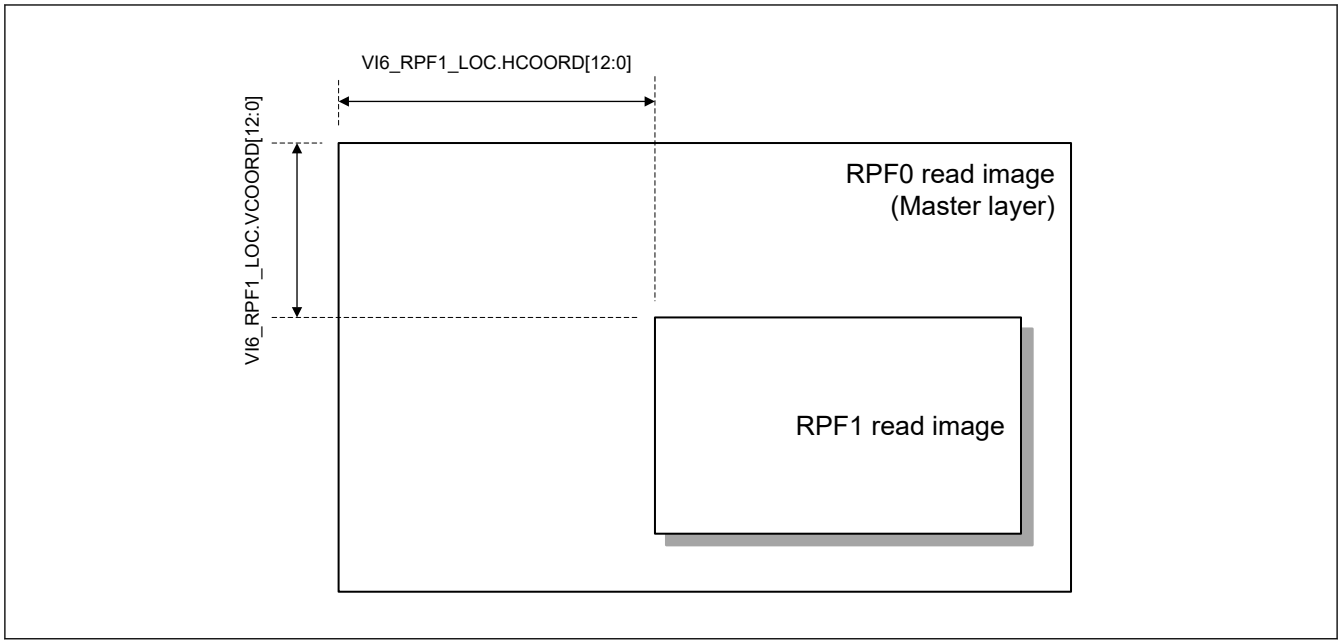


Figure 54.13 RPF1 offsets from Master layer

54.4.6.6 VI6_RPFn_ALPH_SEL : RPFn α Plane Selection Control Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0314 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ASEL[2:0]			IROP[3:0]				BSEL	—	—	—	AEXT[1:0]		—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ALPHA1[7:0]								ALPHA0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ALPHA0[7:0]	8-Bit α Value Output when 1-Bit α Value is 0*1 These bits specify the 8-bit α value to be output when 1-bit α data is input and the α value input to the 8-bit transparent α generator shown in Figure 54.14 is 0b. This setting is valid when the ASEL bits are set to 010b or 011b.	R/W
15:8	ALPHA1[7:0]	8-Bit α Value Output when 1-Bit α Value is 1*1 These bits specify the 8-bit α value to be output when 1-bit α data is input and the α value input to the 8-bit transparent α generator shown in Figure 54.14 is 1b. This setting is valid when the ASEL bits are set to 010b or 011b.	R/W
17:16	—	These bits are read as 0. The write value should be 0.	R/W
19:18	AEXT[1:0]	Lower-Bit α Value Extension Method Set These bits specify the method for extending the input α data to 8 bits through the unpack processing. 0 0: The lower-order bits of α value are extended with 0. 0 1: The upper-order bits of α value are copied to the lower-order bits. 1 0: The lower-order bits of α value are extended with 0. The maximum value is limited to 0xFF. 1 1: Setting prohibited.	R/W
22:20	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
23	BSEL	α Bit Count Conversion Selection for 1-Bit Mask Generator Specifies the number of bits in the α plane to be read as mask information from the external RAM. The α value in mask information is used for the source (S) in IROP. When α plane data is 8 bits, it is converted to one bit through the 1-bit mask generator shown in Figure 54.14. Note that this bit setting is valid when the ASEL bits are set to 0 or 2 and VI6_RPFn_MSKCTRL.MSK_EN is set to 0. In other cases, this bit setting has no effect. 0: 8-bit α is converted to 1-bit α through the 1-bit mask generator. When the 8-bit α value input to the RPF is not 0, it is converted to 1b; when the value is 0, it is converted to 0b. 1: α value goes through the 1-bit mask generator. The 1-bit α value input to the RPF is output through the 1-bit mask generator without change.	R/W
27:24	IROP[3:0]	IROP Operation Setting These bits specify the operator to be executed in the IROP operation unit shown in Figure 54.14. The source (S) for the IROP operation is the pixel data and α data specified in the VI6_RPFn_MSKSET0 or VI6_RPFn_MSKSET1 IROP input value register, which is selected according to the value (0 or 1) generated by the 1-bit mask generator. The destination (D) is the image data (RGB/Y CbCr) and 8-bit α data output from the unpack/OSD processor. IROP operation is applied both for the image data and α data between the source and destination data. If these bits are set to the operation that involves the source (S) (IROP setting other than 0, 5, 10, or 15) while VI6_RPFn_MSKCTRL.MSK_EN is 0, the α plane is read from the external RAM to be used for the α value for IROP operation; specify the α plane read start address (VI6_RPFn_SRCM_ADDR_A). When the virtual input function is used (VI6_RPFn_INFMT.VIR = 1), IROP operation is not available; set these bits to 0x0. 0x0: NOP(D) 0x1: AND(S & D) 0x2: AND_REVERSE(S & ~D) 0x3: COPY(S) 0x4: AND_INVERTED(~S & D) 0x5: CLEAR(0) 0x6: XOR(S ^ D) 0x7: OR(S | D) 0x8: NOR(~(S | D)) 0x9: EQUIV(~(S ^ D)) 0xA: INVERT(~D) 0xB: OR_REVERSE(S | ~D) 0xC: COPY_INVERTED(~S) 0xD: OR_INVERTED(~S | D) 0xE: NAND(~(S & D)) 0xF: SET(all 1)	R/W

Bit	Symbol	Function	R/W
30:28	ASEL[2:0]	α Format and Processing Method Select These bits select how to handle the α value to be used. The RPF handles two types of α value; 8-bit and 1-bit values. When a 1-bit α value is used, VSPD assumes that the 1-bpp α value for each pixel is stored in the order from MSB to LSB in each byte (big endian). The α value is used as either transparency information or mask information. Transparency information is included in the α plane read from external memory when the ASEL bits are set to 1 or 3 and in the α value stored in the packed RGB bit field when these bits are set to 0 or 2. The α value as transparency information is sent as the destination value to the IROP as shown in Figure 54.14 and then output to the subsequent modules. The output α value is used, for example, for blending in the BRS. The α value as mask information is used for IROP operation in the RPF. The mask information is included in the α plane read from external RAM when the ASEL bits are set to 0 or 2 and the source value is used in IROP operation (IROP setting other than 0, 5, 10, or 15). This α value is sent as the source value to the IROP as shown in Figure 54.14. Note that the α value selected through the ASEL bits has a lower priority than the VI6_RPFn_CKEY_SET*.AP* value replaced through the color keying function. When the color keying function is used, the α value may be replaced with the VI6_RPFn_CKEY_SET*.AP* value regardless of the ASEL bit setting. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), set these bits to 4. 0 0 0: 1, 4, or 8-bit packed α + plane α (IROP ≠ 0, 5, 10, 15) The α bit field in 1, 4, or 8-bit packed α is handled as transparency information. Be sure to specify the packed format that includes α through VI6_RPFn_INFMT.RDFMT. When VI6_RPFn_MSKCTRL.MSK_EN is 0 and the IROP bit value is not 0, 5, 10, or 15, the α plane should be read as mask information. Specify the number of α data bits (BSEL) stored in the α plane and the α plane read start address (VI6_RPFn_SRCM_ADDR_AI). When the IROP bits are set to 0, 5, 10, or 15, the α plane is not read. 0 0 1: 8-bit plane α The 8-bit α plane is read from external RAM as transparency information. When the packed RGB format has a bit field for α, the information in the α bit field is discarded. The α plane read start address (VI6_RPFn_SRCM_ADDR_AI) should be specified. The α value goes through the 8-bit transparent α generator shown in Figure 54.14 without change. When VI6_RPFn_MSKCTRL.MSK_EN is 0, IROP operation cannot be executed; set the IROP bits to 0 in this case. When VI6_RPFn_MSKCTRL.MSK_EN is 1. IROP operation can be executed. 0 1 0: 1-bit packed α + plane α (IROP ≠ 0, 5, 10, 15) The 1-bit packed α input is converted by the 8-bit transparent α generator shown in Figure 54.14 according to the ALPHA0/1 setting into the 8-bit α value as transparency information. Select the packed input format that includes a 1-bit α field. When VI6_RPFn_MSKCTRL.MSK_EN is 0 and the IROP value is not 0, 5, 10, or 15, the α plane should be read as mask information. Specify the number of α data bits (BSEL) stored in the α plane and the α plane read start address (VI6_RPFn_SRCM_ADDR_AI). When the IROP bits are set to 0, 5, 10, or 15, the α plane is not read. 0 1 1: 1-bit plane α The 1-bit α plane is read from external RAM and converted by the 8-bit transparent α generator shown in Figure 54.14 according to the ALPHA0/1 setting into the 8-bit α value as transparency information. When the packed RGB format has a bit field for α, the information in the α bit field is discarded. The α plane read start address (VI6_RPFn_SRCM_ADDR_AI) should be specified. When VI6_RPFn_MSKCTRL.MSK_EN is 0, IROP operation cannot be executed; set the IROP bits to 0 in this case. When VI6_RPFn_MSKCTRL.MSK_EN is 1. IROP operation can be executed. 1 0 0: Fixed α The fixed α value (VI6_RPFn_VRTCOL_SET.LAY A value) is output from the RPF. IROP operation cannot be executed; set the IROP bits to 0 in this case. Others: Setting prohibited.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. A value from 0 to 255 can be specified.

Figure 54.14 shows the relationship between the α selector, IROP operation unit, color keying unit, and related registers. The IROP operation unit receives two inputs, source and destination. The image data input from the external memory is processed through the unpack processor and 8-bit transparent α generator and then input to the IROP operation unit as destination data. The α plane data input from the external memory is sent to the 8-bit transparent α generator when

the ASEL bits are set to 1 or 3, or sent to the 1-bit mask α generator when the ASEL bits are set to 0 or 2. For the pixel data and 8-bit α value on the source side of the IROP operation unit, either the VI6_RPFn_MSKSET0 value or VI6_RPFn_MSKSET1 values will be selected according to the 1-bit α value output by the 1-bit mask α generator.

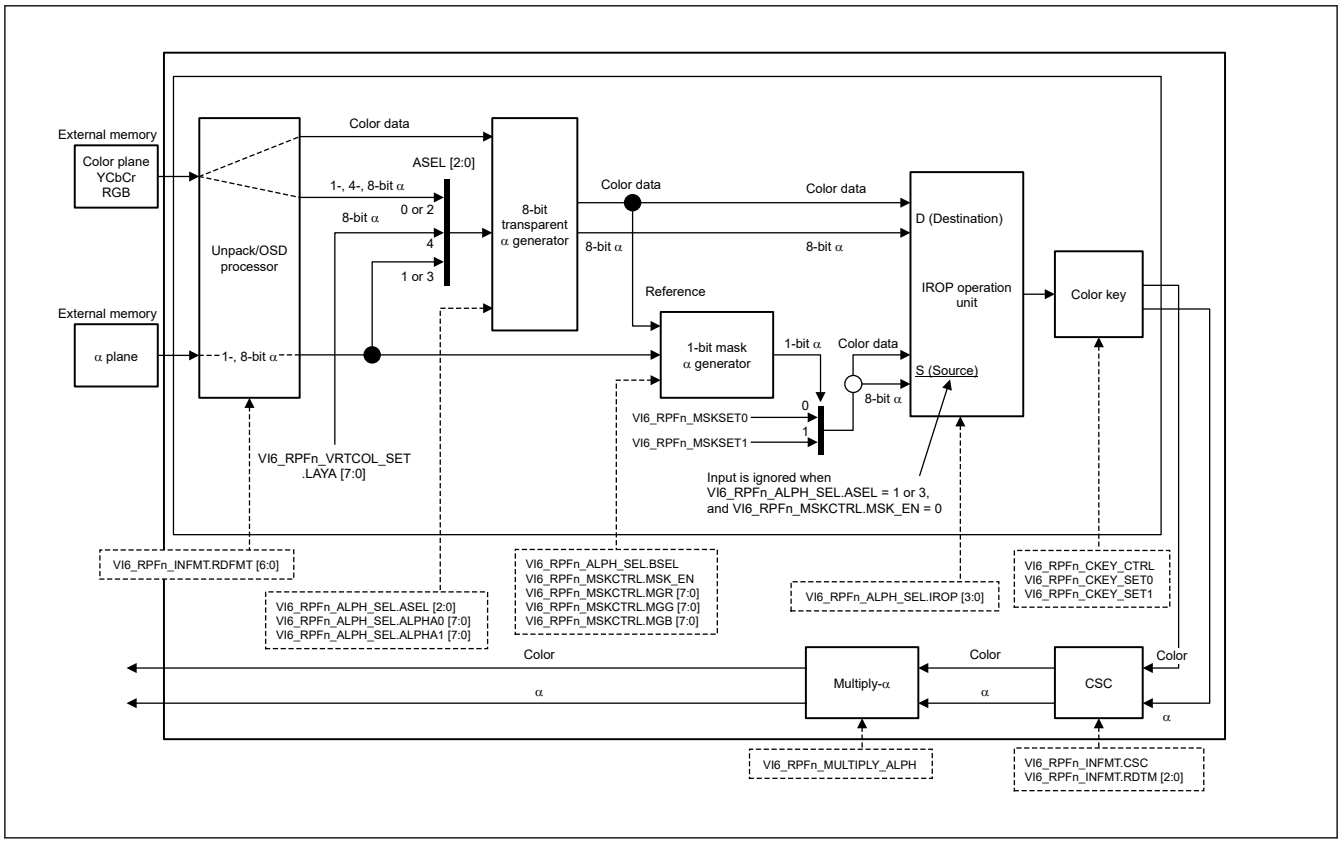


Figure 54.14 Configuration of Alpha selector and IROP operation unit in RPF

The following describes the function of each block shown in Figure 54.14.

Unpack/OSD processor:

Unpacks each component and α value of the image data according to the packed format specified in VI6_RPFn_INFMT.RDFMT.

8-bit transparent α generator:

Converts the input α value into 8-bit α when the input α is four bits or one bit.

When VI6_RPFn_ALPH_SEL.ASEL are set to 0 (8-, 4-, or 1-bit packed α), this generator outputs the input α value without change if the α bit field in the packed α data is 8 bits; if the α bit field is less than 8 bits, it is converted to an 8-bit α value by extending the LSB side according to the VI6_RPFn_ALPH_SEL.AEXT setting.

When VI6_RPFn_ALPH_SEL.ASEL are set to 1 (8-bit plane α) or 4 (fixed α), this generator outputs the input 8-bit plane α without change. If a packed α value is included in RGB data, it is discarded.

When VI6_RPFn_ALPH_SEL.ASEL are set to 2 (1-bit packed α) or 3 (8-bit α generated from 1-bit plane α), an 8-bit α value is generated by using the VI6_RPFn_ALPH_SEL.ALPHA0[7:0] value when the input 1-bit α value is 0 or by using the VI6_RPFn_ALPH_SEL.ALPHA1[7:0] value when the input 1-bit α value is 1. When VI6_RPFn_ALPH_SEL.ASEL is set to 3, a packed α value that is included in RGB data is discarded.

1-bit mask α generator:

Generates 1-bit α data from the input 8-bit α data or pixel data. When the input α data is one bit, this generator outputs it without change.

When VI6_RPFn_ALPH_SEL.ASEL are set to 0 (8-, 4-, or 1-bit packed α) or 2 (plane α) and VI6_RPFn_MSKCTRL.MSK_EN is set to 0, the α plane read from the external memory to be used in IROP is converted to 1-bit α data when necessary. When the α plane data read from the external RAM is 8 bits (BSEL = 0), if the value is 0,

a 1-bit α value of 0b is generated; if the value is not 0, a 1-bit α value of 1b is generated. When the α plane data is one bit (BSEL = 1), this generator outputs it without change.

When the value of the 1-bit α generated by the 1-bit mask α generator is 0b, the 8-bit α and pixel data specified in VI6_RPFn_MSKSET0 are output as the source. When the generated 1-bit α value is 1b, the 8-bit α and pixel data specified in VI6_RPFn_MSKSET1 are output as the source.

As shown in Figure 54.14, when VI6_RPFn_ALPH_SEL.ASEL are set to 1 (8-bit plane α) or 3 (1-bit plane α), the α plane read from the external RAM is sent to the 8-bit transparent α generator as transparency information. When VI6_RPFn_MSKCTRL.MSK_EN is set to 0, the 1-bit α for masking is generated according to the input α plane (see section 54.4.6.8. VI6_RPFn_MSKCTRL : RPFn Mask Control Register (n = 0, 1)), but the 1-bit mask α generator does not refer to the input α plane because it is input to the 8-bit transparent α generator as transparency information. Accordingly, the 1-bit mask α generator does not generate a 1-bit α for masking and the data on the source side becomes invalid; that is, IROP operation cannot be executed. Set the IROP bits to 0 in this case. In contrast, when VI6_RPFn_MSKCTRL.MSK_EN is set to 1, the 1-bit mask α generator creates α data for masking according to the pixel data instead of the input α plane data, and IROP operation can be executed in this case.

IROP operation unit:

Executes ROP operation according to the opcode specified in VI6_RPFn_ALPH_SEL.IROP. For ROP operation (other than NOP), valid values should be input both for the source and destination. As described in the above (description of the 1-bit mask α generator), when VI6_RPFn_ALPH_SEL.ASEL are set to 1 or 3 and VI6_RPFn_MSKCTRL.MSK_EN is set to 0, the source data for the IROP operation unit is treated as invalid; set VI6_RPFn_ALPH_SEL.IROP to 0 (NOP). When VI6_RPFn_ALPH_SEL.ASEL are set to 4, a fixed α value is output from the RPF and IROP operation is not available. In the same way as the above case, set VI6_RPFn_ALPH_SEL.IROP to 0 (NOP).

To specify a valid source value for the IROP operation unit and execute IROP operation (specify an opcode other than NOP in the IROP bits), specify register values as shown in Table 54.21. Where the source input state is indicated as "Valid" in the table, IROP operation can be executed. In the cases where IROP operation is not available, set the IROP bits to 0 (NOP).

Table 54.21 Source input state in IROP operation unit

VI6_RPFn_ALPH_SEL.ASEL[2:0]		VI6_RPFn_MSKCTRL.MSK_EN		
		0 (Source data is generated according to input α plane)		1 (Source data is generated according to the destination-side pixel data)
000b	(1-, 4-, or 8-bit packed α + plane α)	Valid	(α plane input)	Valid
001b	(8-bit α plane)	Invalid	(IROP operation is not available; α plane is output to the subsequent modules behind RPF)	Valid
010b	(8-bit α generated from 1-bit packed α + plane α)	Valid	(α plane input)	Valid
011b	(8-bit α generated from 1-bit plane α)	Invalid	(IROP operation is not available; α plane is output to the subsequent modules behind RPF)	Valid
100b	(Fixed α)	Invalid (IROP operation is not available; fixed α is output to the subsequent modules behind RPF)		

For the handling of the α values shown in Figure 54.14 and Table 54.21, the relationship between the RPF input format and RPF output α value is shown in Table 54.22. Where only bit names are shown in the table, the bits are in VI6_RPFn_ALPH_SEL described in this section.

Table 54.22 α value selected and output according to ASEL bits in each input format (1 of 2)

ASEL setting		α Value Output for each input format	
		RGB	YCbCr
000b	(8-, 4-, or 1-bit packed α is input)	1-, 4-, or 8-bit pixel α	0xFF*1
001b	(8-bit plane α is input)	8-bit α plane	8-bit α plane
010b	(8-bit α is generated from the 1-bit packed α input)	ALPHA0 or ALPHA1 setting	0xFF*1
011b	(8-bit α is generated from the 1-bit plane α input)	ALPHA0 or ALPHA1 setting	ALPHA0 or ALPHA1 setting

Table 54.22 α value selected and output according to ASEL bits in each input format (2 of 2)

ASEL setting		α Value Output for each input format	
		RGB	YCbCr
100b	(Fixed α is output)	VI6_RPFn_VRTCOL_SET.LAYA setting	

Note 1. Fixed value 0xFF is output because packed α is not included in YCbCr.

54.4.6.7 VI6_RPFn_VRTCOL_SET : RPFn Virtual Plane Color Information Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0318 + 0x100 × n

Bit position:	31 to 24	23 to 16	15 to 8	7 to 0
Bit field:	LAYA[7:0]	LAYR[7:0]	LAYG[7:0]	LAYB[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	LAYB[7:0]	Virtual-Input Fixed B/Cb Component Value*1 These bits specify the fixed B or Cb value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting. When VI6_RPFn_INFMT.RDFMT is set to 19, the value specified in these bits is used as the B value. When VI6_RPFn_INFMT.RDFMT is set to 64, the value specified in these bits is used as the Cb value.	R/W
15:8	LAYG[7:0]	Virtual-Input Fixed G/Y Component Value*1 These bits specify the fixed G or Y value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting. When VI6_RPFn_INFMT.RDFMT is set to 19, the value specified in these bits is used as the G value. When VI6_RPFn_INFMT.RDFMT is set to 64, the value specified in these bits is used as the Y value.	R/W
23:16	LAYR[7:0]	Virtual-Input Fixed R/Cr Component Value*1 These bits specify the fixed R or Cr value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting. When VI6_RPFn_INFMT.RDFMT is set to 19, the value specified in these bits is used as the R value. When VI6_RPFn_INFMT.RDFMT is set to 64, the value specified in these bits is used as the Cr value.	R/W
31:24	LAYA[7:0]	Virtual-Input Fixed α Value*1 These bits specify the fixed α value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting. When the virtual input function is disabled (VI6_RPFn_INFMT.VIR = 0), these bits are used to specify the fixed α value to be output from the RPF while VI6_RPFn_ALPH_SEL.ASEL are set to 4.	R/W

Note 1. A value from 0 to 255 can be specified.

54.4.6.8 VI6_RPFn_MSKCTRL : RPFn Mask Control Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x031C + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23 to 16
Bit field:	—	—	—	—	—	—	—	MSK_EN	MGR[7:0]
Value after reset:	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0

Bit position:	15 to 8	7 to 0
Bit field:	MGG[7:0]	MGB[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	MGB[7:0]	B/Cb Comparison Value for 1-Bit α Generation*1 *2 These bits specify the B/Cb value to be compared for 1-bit α generation by using the pixel data on the destination side. When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a B value for comparison. When YCbCr is specified, specify a Cb value for comparison.	R/W
15:8	MGG[7:0]	G/Y Comparison Value for 1-Bit α Generation*1 *2 These bits specify the G/Y value to be compared for 1-bit α generation by using the pixel data on the destination side. When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a G value for comparison. When YCbCr is specified, specify a Y value for comparison.	R/W
23:16	MGR[7:0]	R/Cr Comparison Value for 1-Bit α Generation*1 *2 These bits specify the R/Cr value to be compared for 1-bit α generation by using the pixel data on the destination side. When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify an R value for comparison. When YCbCr is specified, specify a Cr value for comparison.	R/W
24	MSK_EN	Mask Generation Specification Specifies the method of α value generation in the 1-bit mask α generator shown in Figure 54.14. 0: A 1-bit mask value is generated according to the input α plane value. When the input α is in the 1-bit format (VI6_RPFn_ALPH_SEL.BSEL = 1), the 1-bit mask value is output without change. When the input α is in the 8-bit format (VI6_RPFn_ALPH_SEL.BSEL = 0), the 1-bit mask value is 0 if the α value is 0x00; otherwise, the 1-bit mask value is 1. 1: The R/Cr, G/Y, and B/Cb components of the image input to the destination side of the IROP operation unit are compared with the values specified in the MGR, MGG, and MGB bits, respectively. When all values match, 1 is output as the 1-bit mask value, and in other cases, 0 is output. When the generated 1-bit mask data is not used, set VI6_RPFn_ALPH_SEL.IROP to 0.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This setting is ignored when MSK_EN is set to 0.
Note 2. A value from 0 to 255 can be specified.

54.4.6.9 VI6_RPFn_MSKSET0 : RPFn IROP-SRC Input Value Register 0 (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0320 + 0x100 × n

Bit position:	31								24	23							16	15							8	7							0
Bit field:	MSA0[7:0]									MSR0[7:0]								MSG0[7:0]								MSB0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MSB0[7:0]	IROP-Source Input B/Cb Value when 1-Bit α is 0*1 These bits specify the B/Cb value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (Figure 54.14). When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a B component value. When YCbCr is specified, specify a Cb component value.	R/W
15:8	MSG0[7:0]	IROP-Source Input G/Y Value when 1-Bit α is 0*1 These bits specify the G/Y value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (Figure 54.14). When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a G component value. When YCbCr is specified, specify a Y component value.	R/W
23:16	MSR0[7:0]	IROP-Source Input R/Cr Value when 1-Bit α is 0*1 These bits specify the R/Cr value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (Figure 54.14). When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify an R component value. When YCbCr is specified, specify a Cr component value.	R/W

Bit	Symbol	Function	R/W
31:24	MSA0[7:0]	IROP-Source Input α Value when 1-Bit α is 0*1 These bits specify the 8-bit α value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (Figure 54.14).	R/W

Note 1. A value from 0 to 255 can be specified.

54.4.6.10 VI6_RPFn_MSKSET1 : RPFn IROP-SRC Input Value Register 1 (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0324 + 0x100 × n

Bit position:	31								24	23							16	15							8	7							0
Bit field:	MSA1[7:0]									MSR1[7:0]								MSG1[7:0]								MSB1[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MSB1[7:0]	IROP-Source Input B/Cb Value when 1-Bit α is 1*1 These bits specify the B/Cb value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1 (Figure 54.14). When RGB is specified as the RPF1 input format through VI6_RPF1_INFMT.RDFMT, specify a B component value. When YCbCr is specified, specify a Cb component value.	R/W
15:8	MSG1[7:0]	IROP-Source Input G/Y Value when 1-Bit α is 1*1 These bits specify the G/Y value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1 (Figure 54.14). When RGB is specified as the RPF1 input format through VI6_RPF1_INFMT.RDFMT, specify a G component value. When YCbCr is specified, specify a Y component value.	R/W
23:16	MSR1[7:0]	IROP-Source Input R/Cr Value when 1-Bit α is 1*1 These bits specify the R/Cr value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1 (Figure 54.14). When RGB is specified as the RPF1 input format through VI6_RPF1_INFMT.RDFMT, specify an R component value. When YCbCr is specified, specify a Cr component value.	R/W
31:24	MSA1[7:0]	IROP-Source Input α Value when 1-Bit α is 1*1 These bits specify the 8-bit α value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1 (Figure 54.14).	R/W

Note 1. A value from 0 to 255 can be specified.

54.4.6.11 VI6_RPFn_CKEY_CTRL : RPFn Color Keying Control Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0328 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

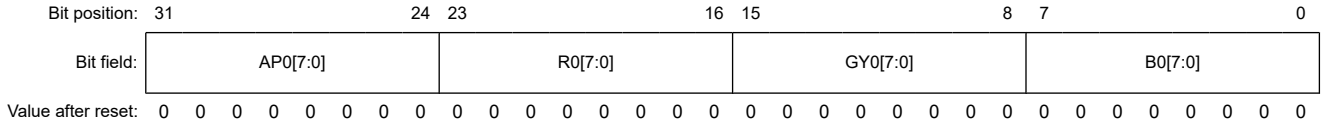
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LTH	—	—	—	CV	—	—	SAPE 1	SAPE 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SAPE0	Comparison Color Data Setting 0 Enable/Disable This bit enables or disables comparison color data setting 0 in the transparent color mode for the color keying module. This bit setting is valid only when the CV bit is set to 0; it is ignored when the CV bit is set to 1. In transparent color mode, color information 0 (VI6_RPFn_CKEY_SET0.R0/GY0/B0) specified in VI6_RPFn_CKEY_SET0 is compared with the input component values. When the input data is in an RGB format, and if all input values match the specified color information, the input α value is replaced with the value specified in VI6_RPFn_CKEY_SET0.AP. When the input data is in YCbCr format, only the Y data is compared. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), clear this bit to 0. 0: Comparison color data setting 0 is disabled. 1: Comparison color data setting 0 is enabled.	R/W
1	SAPE1	Comparison Color Data Setting 1 Enable/Disable This bit enables or disables comparison color data setting 1 in the transparent color mode for the color keying module. This bit setting is valid only when the CV bit is set to 0; it is ignored when the CV bit is set to 1. In transparent color mode, color information 1 (VI6_RPFn_CKEY_SET1.R1/GY1/B1) specified in VI6_RPFn_CKEY_SET1 is compared with the input component values. When the input data is in an RGB format, and if all input values match the specified color information, the input α value is replaced with the value specified in VI6_RPFn_CKEY_SET1.AP. When the input data is in YCbCr format, only the Y data is compared. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), set this bit to 0. 0: Comparison color data setting 1 is disabled. 1: Comparison color data setting 1 is enabled.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	CV	Color Replacement Control This bit controls the color replacement function in the color keying module shown in Figure 54.2. When an RGB format is specified as the color space of the RPFn input data through VI6_RPFn_INFMT.RDFMT, and if all components of an input pixel match the color components specified in VI6_RPFn_CKEY_SET0, the color replacement function replaces the values of the input α and all RGB components with the α and color components specified in VI6_RPFn_CKEY_SET1. When a YCbCr format is specified as the color space of the RPFn input data through VI6_RPFn_INFMT.RDFMT, only the Y data is compared; if the luminance component of an input pixel matches the value specified in VI6_RPFn_CKEY_SET0.GY0, the color replacement function replaces the values of the input α and all Y CbCr components with the α and color components specified in VI6_RPFn_CKEY_SET1. When the CV bit is set to 1, the color replacement function is enabled. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), set this bit to 0. 0: Color replacement function is disabled (transparent color mode). 1: Color replacement function is enabled.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	LTH	Transparent color-luma Threshold Mode Enable/Disable This bit enables or disables transparent color-luma threshold mode for the color keying module. In transparent color-luma threshold mode, color information 0 (GY0) specified in VI6_RPFn_CKEY_SET0 register is compared with the input luma values. When the input data is in YCbCr format, and if input Y value is equal to or smaller than specified color information (VI6_RPFn_CKEY _SET0.GY 0), the input α value is replaced with the value specified in VI6_RPFn_CKEY_SET0.AP0. This bit is available only when the input data is in YCbCr format. When the input data is in an RGB format, set this bit to 0. This bit setting is valid only when the CV bit is set to 0; it is ignored when the CV bit is set to 1. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), set this bit to 0. 0: Luma threshold mode is disabled (Matched color mode). 1: Luma threshold mode is enabled.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

54.4.6.12 VI6_RPFn_CKEY_SET0 : RPFn Color Keying Color Setting Register 0 (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x032C + 0x100 × n



Bit	Symbol	Function	R/W
7:0	B0[7:0]	B[*1]/Cb Component Data in Color Keying Color Information-0[*2] ● In transparent color-matched color mode for color keying Specify the B component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPE0 setting. When the RPFn input is in YCbCr format, the color keying module does not compare the Cb component, and the setting of these bits is ignored. ● In transparent color-luma threshold mode for color keying The color keying module does not refer this bit field, and the setting of these bits is ignored. ● In color replacement mode for color keying Specify the B component value to be compared with the input data in the color replacement function of the color keying module.	R/W
15:8	GY0[7:0]	G[*1]/Y Component Data in Color Keying Color Information-0[*2] ● In transparent color-matched color mode for color keying Specify the G/Y component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPE0 setting. ● In transparent color-luma threshold mode for color keying Specify the Y component value in the GY0 to be compared. ● In color replacement mode for color keying Specify the G/Y component value to be compared with the input data in the color replacement function of the color keying module.	R/W
23:16	R0[7:0]	R[*1]/Cr Component Data in Color Keying Color Information-0[*2] ● In transparent color-matched color mode for color keying Specify the R component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPE0 setting. When the RPFn input is in YCbCr format, the color keying module does not compare the Cr component, and the setting of these bits is ignored. ● In transparent color-luma threshold mode for color keying The color keying module does not refer this bit field, and the setting of these bits is ignored. ● In color replacement mode for color keying Specify the R component value to be compared with the input data in the color replacement function of the color keying module.	R/W
31:24	AP0[7:0]	α Data in Color Keying Color Information-0[*2] ● In transparent color-matched color mode for color keying When the input data matches color setting-0 (R0, GY0, and B0) for transparent color comparison in the color keying module, the input α value is replaced with the value specified in these bits. Specify the α value to replace the input value. ● In transparent color-luma threshold mode for color keying When the input data is in YCbCr format, and if input Y value is equal to or smaller than GY 0, the input α value is replaced with the value specified in this bit field (AP0). Specify the α value in this bit field (AP0) to replace the input value. ● In color replacement mode for color keying These bits are not used in this mode. Clear them to 0. α value replacement through these bits in transparent color mode for color keying takes priority over the α value selected through the VI6_RPFn_ALPH_SEL.ASEL setting.	R/W

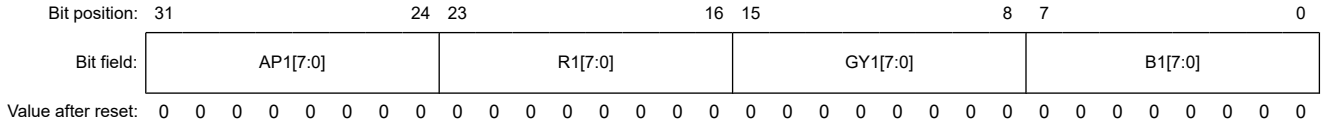
Note 1. When comparison data is specified in an RGB format, if a packed format is selected for RGB input and each of the RGB components is not 8 bits, the lower-order bits of input data are extended as specified through VI6_RPFn_INFMT.CEXT before comparison. The RGB components to be compared with the input should also be extended in the same way and the extended values should be specified in this register.

Note 2. A value from 0 to 255 can be specified.

54.4.6.13 VI6_RPFn_CKEY_SET1 : RPFn Color Keying Color Setting Register 1 (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0330 + 0x100 × n



Bit	Symbol	Function	R/W
7:0	B1[7:0]	B[*1]/Cb Component Data in Color Keying Color Information-1[*2] ● In transparent color-matched color mode for color keying Specify the B component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPE1 setting. When the RPFn input is in YCbCr format, the color keying module does not compare the Cb component, and the setting of these bits is ignored. ● In transparent color-luma threshold mode for color keying The color keying module does not refer this bit field, and the setting of these bits is ignored. ● In color replacement mode for color keying Specify the B component value to be compared with the input data in the color replacement function of the color keying module.	R/W
15:8	GY1[7:0]	G[*1]/Y Component Data in Color Keying Color Information-1[*2] ● In transparent color-matched color mode for color keying Specify the G/Y component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPE1 setting. ● In transparent color-luma threshold mode for color keying Specify the Y component value in the GY1 to be compared. ● In color replacement mode for color keying Specify the G/Y component value to be compared with the input data in the color replacement function of the color keying module.	R/W
23:16	R1[7:0]	R[*1]/Cr Component Data in Color Keying Color Information-1[*2] ● In transparent color-matched color mode for color keying Specify the R component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPE1 setting. When the RPFn input is in YCbCr format, the color keying module does not compare the Cr component, and the setting of these bits is ignored. ● In transparent color-luma threshold mode for color keying The color keying module does not refer this bit field, and the setting of these bits is ignored. ● In color replacement mode for color keying Specify the R component value to be compared with the input data in the color replacement function of the color keying module.	R/W
31:24	AP1[7:0]	α Data in Color Keying Color Information-1[*2] ● In transparent color-matched color mode for color keying When the input data matches color setting-1 (R1, GY1, and B1) for transparent color comparison in the color keying module, the input α value is replaced with the value specified in these bits. Specify the α value to replace the input value. ● In transparent color-luma threshold mode for color keying When the input data is in YCbCr format, and if input Y value is equal to or smaller than GY 0, the input α value is replaced with the value specified in this bit field (AP1). Specify the α value in this bit field (AP1) to replace the input value. ● In color replacement mode for color keying These bits are not used in this mode. Clear them to 0. α value replacement through these bits in transparent color mode for color keying takes priority over the α value selected through the VI6_RPFn_ALPH_SEL.ASEL setting.	R/W

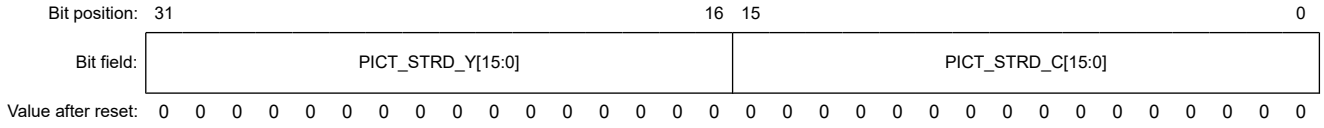
Note 1. When comparison data is specified in an RGB format, if a packed format is selected for RGB input and each of the RGB components is not 8 bits, the lower-order bits of input data are extended as specified through VI6_RPFn_INFMT.CEXT before comparison. The RGB components to be compared with the input should also be extended in the same way and the extended values should be specified in this register.

Note 2. A value from 0 to 255 can be specified.

54.4.6.14 VI6_RPFn_SRCM_PSTRIDE : RPFn Source Picture Memory Stride Setting Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0334 + 0x100 × n



Bit	Symbol	Function	R/W
15:0	PICT_STRD_C[15:0]	Memory Stride of Source Picture C Plane*1 For a YCbCr source picture read by the RPFn, these bits specify the memory stride of the C planes in byte units. Specifically, this setting is used as the memory stride of the Cb and Cr planes in a picture in a YCbCr planar format. When an RGB-format picture is read, these bits do not require setting.	R/W
31:16	PICT_STRD_Y[15:0]	Memory Stride of Source Picture Y/RGB Plane*1 For a Y/RGB plane of source picture read by the RPFn, these bits specify the memory stride in byte units.	R/W

Note 1. A value from 0 to 65535 can be specified. See Figure 54.15 for settings.

This register specifies the memory stride of the picture planes in the source area as shown in Figure 54.15. The memory stride of the α plane should be specified through VI6_RPFn_SRCM_ASTRIDE.ALPH_STRD. When the RPF input is in an RGB format, only the RGB plane is read; when the input is in YCbCr format, the Y and C planes are read as shown in Figure 54.15. When the α plane is used, the α plane is also read. According to the image format and the necessity of the α plane, specify the necessary addresses where the source image is stored (VI6_RPFn_SRCM_ADDR_Y, VI6_RPFn_SRCM_ADDR_C, and VI6_RPFn_SRCM_ADDR_AI).

Whether the α plane needs to be read is determined according to the α plane selection method and IROP operation type. For details, see section 54.4.6.6. VI6_RPFn_ALPH_SEL : RPFn α Plane Selection Control Register (n = 0, 1).

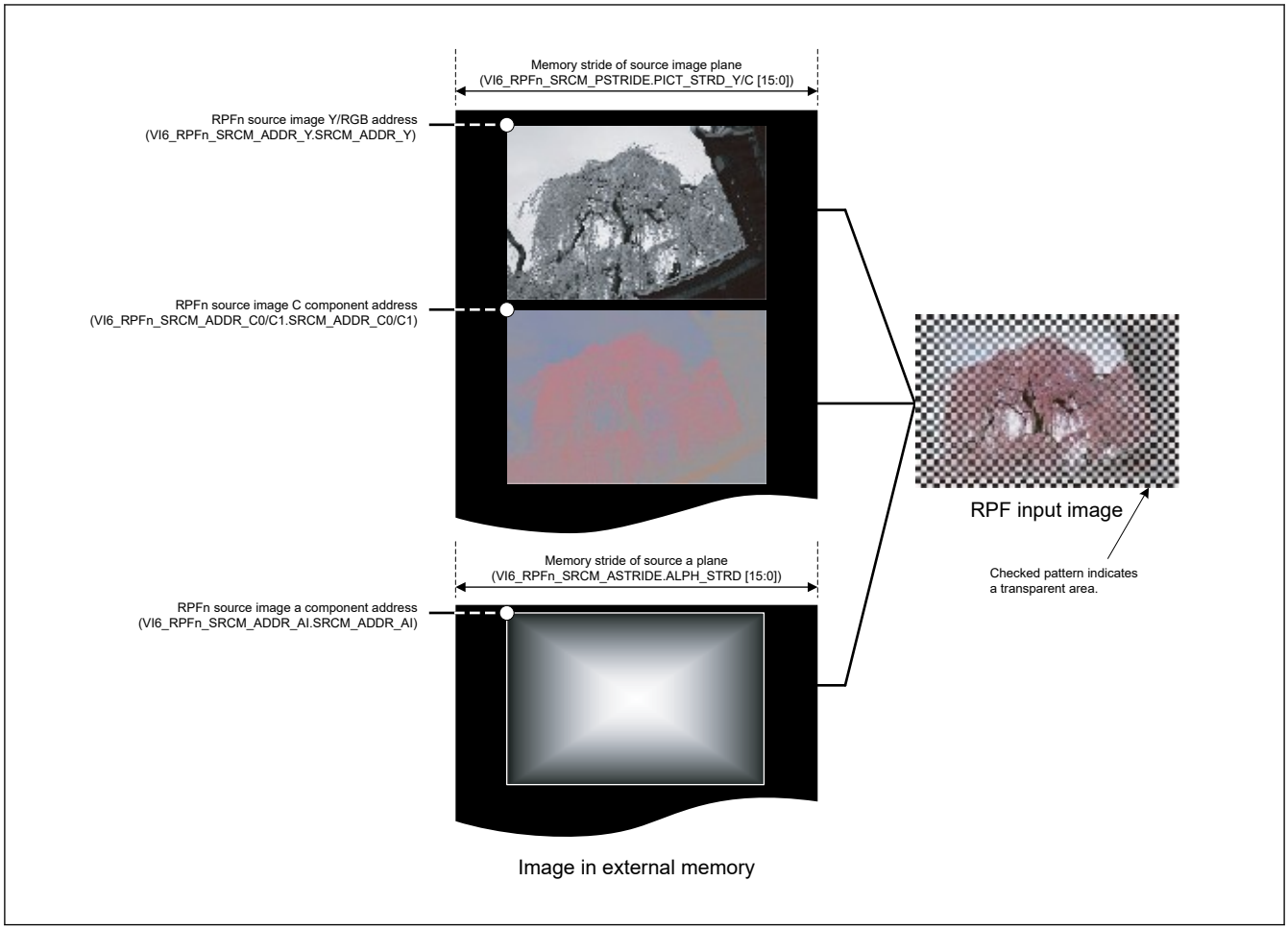


Figure 54.15 Reading an Image from RPFn Source Area

54.4.6.15 VI6_RPFn_SRCM_ASTRIDE : RPFn Source α Memory Stride Setting Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0338 + 0x100 × n

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ALPH_STRD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ALPH_STRD[15:0]	Memory Stride of Source α Plane*1 For a source α plane read by the RPFn, these bits specify the memory stride in byte units.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. A value from 0 to 65535 can be specified. See Figure 54.15 for settings.

54.4.6.16 VI6_RPFn_SRCM_ADDR_Y : RPFn Source Y/RGB Address Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x033C + 0x100 × n

Bit position:	31																												0
Bit field:	SRCM_ADDR_Y[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SRCM_ADDR_Y[31:0]	Source Image Y/RGB Plane Storing Address*1 For a Y plane and packed RGB plane of source image read by the RPFn, these bits specify the start address in byte units.	R/W

Note 1. A value from 0x0000_0000 to 0xFFFF_FFFF can be specified. See Figure 54.15 in section 54.4.6.14. VI6_RPFn_SRCM_PSTRIDE : RPFn Source Picture Memory Stride Setting Register (n = 0, 1) for settings.

54.4.6.17 VI6_RPFn_SRCM_ADDR_C0 : RPFn Source Chroma Address Register 0 (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0340 + 0x100 × n

Bit position:	31																												0
Bit field:	SRCM_ADDR_C0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SRCM_ADDR_C0[31:0]	Source Image C Plane Storing Address 0*1 For a C plane of source image read by the RPFn, these bits specify the start address in byte units. Here, the C plane indicates the combined CbCr plane when a semi-planar format is selected from the packed YCbCr formats shown in Figure 54.15 or the Cb plane when a planar format is selected. When an interleaved format is selected or the RPF input is in an RGB format, this setting is not used.	R/W

Note 1. A value from 0x0000_0000 to 0xFFFF_FFFF can be specified. See Figure 54.15 in section 54.4.6.14. VI6_RPFn_SRCM_PSTRIDE : RPFn Source Picture Memory Stride Setting Register (n = 0, 1) for settings.

54.4.6.18 VI6_RPFn_SRCM_ADDR_C1 : RPFn Source Chroma Address Register 1 (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0344 + 0x100 × n

Bit position:	31																												0
Bit field:	SRCM_ADDR_C1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SRCM_ADDR_C1[31:0]	Source Image C Plane Storing Address 1*1 For a picture in one of the planar YCbCr formats shown in Figure 54.11 read by the RPFn, these bits specify the address where the Cr plane starts in byte units. This setting is not used when the RPF input is in a semi-planer or interleaved YCbCr format or in an RGB format.	R/W

Note 1. A value from 0x0000_0000 to 0xFFFF_FFFF can be specified. See Figure 54.15 in section 54.4.6.14. VI6_RPFn_SRCM_PSTRIDE : RPFn Source Picture Memory Stride Setting Register (n = 0, 1) for settings.

54.4.6.19 VI6_RPFn_SRCM_ADDR_AI : RPFn Source α Address Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0348 + 0x100 × n

Bit position:	31																												0
Bit field:	SRCM_ADDR_AI[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SRCM_ADDR_AI[31:0]	Source Image α Plane Storing Address*1 For an α plane of the source image read by the RPFn, these bits specify the start address in byte units. Specify these bits in the same way as the start address of the picture plane. When the α plane is not read from the source area, these bits do not need to be set.	R/W

Note 1. A value from 0x0000_0000 to 0xFFFF_FFFF can be specified. See Figure 54.15 in section 54.4.6.14. VI6_RPFn_SRCM_PSTRIDE : RPFn Source Picture Memory Stride Setting Register (n = 0, 1) for settings.

54.4.6.20 VI6_RPFn_BAC : RPFn Bus Access Control Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x0350 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	B512
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	B512	Burst access in 512 pixels' enable Always set this bit to 0. 0: Burst access in 256 pixels 1: Burst access in 512 pixels	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

54.4.6.21 VI6_RPFn_MULT_ALPH : RPFn Multiply Alpha Control Register (n = 0, 1)

Base address: VSPD = 0x920E_0000

Offset address: 0x036C + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	A_MMD	—	—	P_MMD[1:0]		ALPHA_RATIO[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ALPHA_RATIO[7:0]	Multiply-alpha value	R/W
9:8	P_MMD[1:0]	P_MMD[1:0]*1 0 0: Image data goes through multiply-alpha unit 0 1: Multiply-alpha unit multiplies image data by specified alpha (ALPHA_RATIO[7:0]) 1 0: Multiply-alpha unit multiplies image data by alpha data 1 1: Multiply-alpha unit multiplies image data by alpha data and specified alpha (ALPHA_RATIO[7:0])	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	A_MMD	A_MMD*1 0: Alpha data go through multiply-alpha unit. 1: Multiply-alpha unit multiplies alpha data by specified alpha (ALPHA_RATIO[7:0]).	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When output format from CSC unit is YCbCr, set this bit to 0.

ALPin: Input Alpha data to Multiply-alpha unit

PIXin(R): Input R data to Multiply-alpha unit

PIXin(G): Input G data to Multiply-alpha unit

PIXin(B): Input B data to Multiply-alpha unit

ALPout: Output Alpha data from Multiply-alpha unit

PIXout(R): Output R data from Multiply-alpha

PIXout(G): Output G data from Multiply-alpha

PIXout(B): Output B data from Multiply-alpha

Table 54.23 Expression of output alpha data from Multiply-alpha unit

A_MMD	ALPHA_RATIO[7:0]	Expression
0	Don't care	ALPout = ALPin
1	Not 255	ALPout = ALPin × ALPHA_RATIO / 256
	255	ALPout = ALPin

Table 54.24 Expression of output pixel data from Multiply-alpha unit (1 of 2)

P_MMD[1:0]	ALPHA_RATIO[7:0]	ALPin	Expression
0	Don't care	Don't care	PIXout(R) = PIXin(R) PIXout(G) = PIXin(G) PIXout(B) = PIXin(B)
1	Not 255	Don't care	PIXout(R) = PIXin(R) × ALPHA_RATIO / 256 PIXout(G) = PIXin(G) × ALPHA_RATIO / 256 PIXout(B) = PIXin(B) × ALPHA_RATIO / 256
	255	Don't care	PIXout(R) = PIXin(R) PIXout(G) = PIXin(G) PIXout(B) = PIXin(B)
2	Don't care	Not 255	PIXout(R) = PIXin(R) × ALPin / 256 PIXout(G) = PIXin(G) × ALPin / 256 PIXout(B) = PIXin(B) × ALPin / 256
	Don't care	255	PIXout(R) = PIXin(R) PIXout(G) = PIXin(G) PIXout(B) = PIXin(B)

Table 54.24 Expression of output pixel data from Multiply-alpha unit (2 of 2)

P_MMD[1:0]	ALPHA_RATIO[7:0]	ALPin	Expression
3	Not 255	Not 255	PIXout(R) = PIXin(R) × ALPin × ALPHA_RATIO / 256 / 256 PIXout(G) = PIXin(G) × ALPin × ALPHA_RATIO / 256 / 256 PIXout(B) = PIXin(B) × ALPin × ALPHA_RATIO / 256 / 256
	255	Not 255	PIXout(R) = PIXin(R) × ALPin / 256 PIXout(G) = PIXin(G) × ALPin / 256 PIXout(B) = PIXin(B) × ALPin / 256
	Not 255	255	PIXout(R) = PIXin(R) × ALPHA_RATIO / 256 PIXout(G) = PIXin(G) × ALPHA_RATIO / 256 PIXout(B) = PIXin(B) × ALPHA_RATIO / 256
	255	255	PIXout(R) = PIXin(R) PIXout(G) = PIXin(G) PIXout(B) = PIXin(B)

54.4.7 WPF Control Registers

54.4.7.1 VI6_WPF0_SRCRPF : WPF0-Source-RPF Register

Base address: VSPD = 0x920E_0000

Offset address: 0x1000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	VIR_ACT2[1:0]		—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	RPF1_ACT[1:0]		RPF0_ACT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RPF0_ACT[1:0]	RPF0 Start Enable These bits enable start of RPFn as the source RPF for the WPF0 when the WPF0 is started. When RPF0 is not started by the WPF0, set the VI6_DPR_RPF0_ROUTE.RT_RPF0 bits to 63. 0 0: RPF0 is not started. 0 1: RPF0 is started as a sublayer source RPF for the WPF0. 1 0: RPF0 is started as the master-layer source RPF for the WPF0. 1 1: Setting prohibited.	R/W
3:2	RPF1_ACT[1:0]	RPF1 Start Enable These bits enable start of RPF1 as the source RPF for the WPF0 when the WPF0 is started. When RPF1 is not started by the WPF0, set the VI6_DPR_RPF1_ROUTE.RT_RPF1 bits to 63. 0 0: RPF1 is not started. 0 1: RPF1 is started as a sublayer source RPF for the WPF0. 1 0: RPF1 is started as the master-layer source RPF for the WPF0. 1 1: Setting prohibited.	R/W
23:4	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
25:24	VIR_ACT2[1:0]	Virtual RPF Start Enable in BRS These bits enable start of the virtual RPF in the BRS as the source RPF for the WPF0 when the WPF0 is started. For details of the virtual RPF, see the following. section 54.4.10.1. VI6_BRS_INCTRL : BRS Input Control Register section 54.4.10.2. VI6_BRS_VIRRPF_SIZE : Size Register of BRS Input Virtual RPF section 54.4.10.3. VI6_BRS_VIRRPF_LOC : Display Location Register of BRS Input Virtual RPF section 54.4.10.4. VI6_BRS_VIRRPF_COL : Color Information Register of BRS Input Virtual RPF Note that the virtual RPF is in the BRS as shown in Figure 54.28 and there are no register bits for DPR setting related to the virtual RPF. 0 0: The virtual RPF in the BRS is not started. 0 1: The virtual RPF in the BRS is started as a sublayer source RPF for the WPF0. 1 0: The virtual RPF in the BRS is started as the master-layer source RPF for the WPF0. 1 1: Setting prohibited.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

When the WPFn is started through the VSPD start register n (VI6_CMDn: n = 0), the RPF and virtual RPF in the BRS specified as the source RPF in this register are also started to supply data to the VSPD internal modules.

Note the following when specifying the source RPF.

1. When blending or ROP operation is applied to multiple images through the BRS, multiple source RPFs are necessary for one WPF. When multiple source RPFs are used, images should be classified into a master layer and sublayers; assign one of the source RPFs as the master-layer source RPF and other RPFs as sublayer source RPFs. Do not assign all RPFs as sublayer source RPFs (VI6_WPF1_SRCRPF = 0x00000015) or two or more RPFs as the master-layer source RPF (VI6_WPF0_SRCRPF = 0x0000025A) (such settings are prohibited).
2. When the BRS is not used, there should be only one source RPF for one WPF. In this case, the source RPF should be assigned as the master-layer source RPF.

54.4.7.2 VI6_WPF0_HSZCLIP : WPF0 Horizontal Input Size Clipping Register

Base address: VSPD = 0x920E_0000

Offset address: 0x1004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	HCEN	—	—	—	—	HCL_OFST[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	HCL_SIZE[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	HCL_SIZE[12:0]	Horizontal Clipping Size Setting When the HCEN bit is 1, these bits specify the clipping size for horizontal clipping processing. Through this processing, the area of the horizontal size specified through the HCL_SIZE bits starting from the offset position specified through the HCL_OFST bits is determined as the valid image area. Accordingly, the right-side pixels beyond the (HCL_OFST + HCL_SIZE) size in the WPF0 input image are discarded. When the HCEN bit is 0, this setting is ignored. A value from 1 to 8190 can be specified. (HCL_OFST + HCL_SIZE) should not exceed the horizontal size of the WPF input. If the setting shown in the bottom example in Figure 54.16 is made, VSPD does not operate correctly. When the WPF0 output format is YCbCr4:2:2 or YCbCr4:2:0, specify an even value in these bits.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
23:16	HCL_OFST[7:0]	Horizontal Size Clipping Offset Value Setting These bits specify the offset size (pixels) from the left-end of the image in horizontal size clipping when the HCEN bit is 1 (Figure 54.16). The left-side of the image input to the WPF is cut off for the size specified in these bits. When the HCEN bit is 0, this setting is ignored. A value from 0 to 255 can be specified. (HCL_OFST + HCL_SIZE) should not exceed the horizontal size of the WPF input. If the setting shown in the bottom example in Figure 54.16 is made, VSPD does not operate correctly.	R/W
27:24	—	These bits are read as 0. The write value should be 0.	R/W
28	HCEN	Horizontal Size Clipping Enable/Disable Enables or disables clipping of the horizontal size of the WPF0 input image. 0: Horizontal size clipping is disabled. 1: Horizontal size clipping is enabled.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

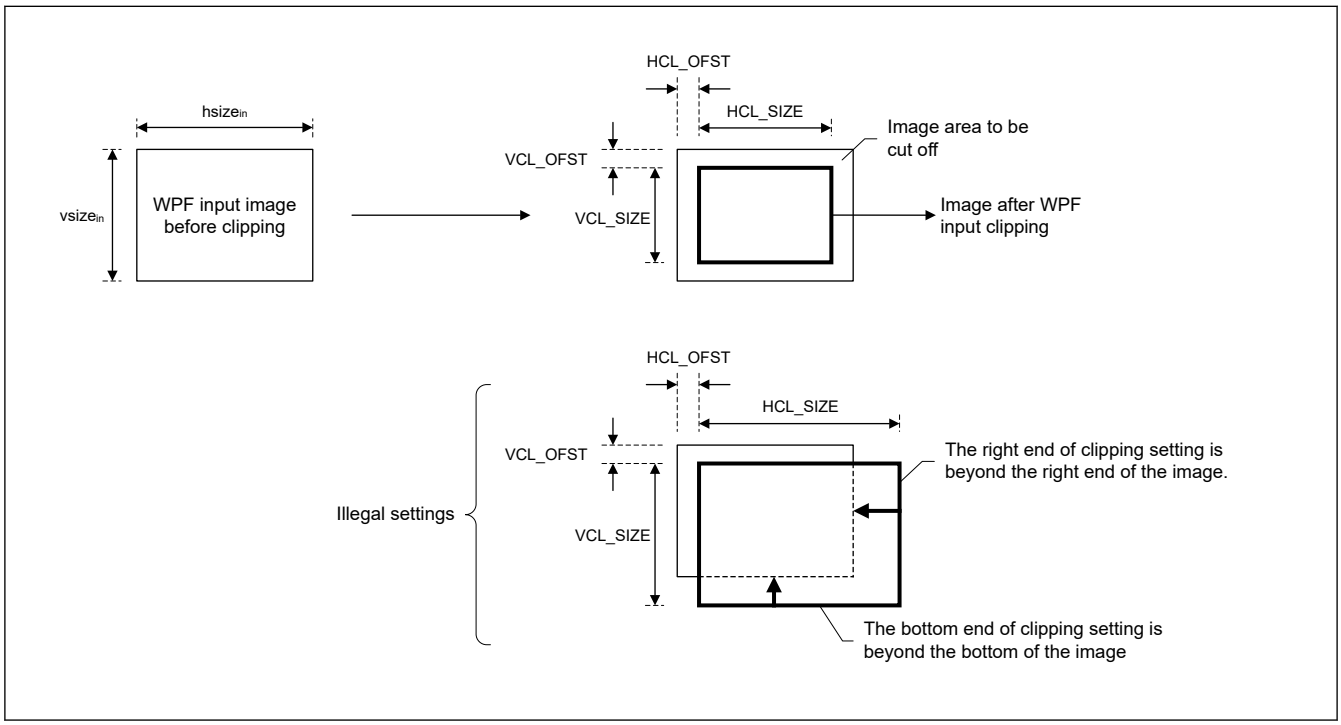


Figure 54.16 Image clipping in WPF input section

54.4.7.3 VI6_WPF0_VSZCLIP : WPF0 Vertical Input Size Clipping Register

Base address: VSPD = 0x920E_0000

Offset address: 0x1008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	VCEN	—	—	—	—	VCL_OFST[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	VCL_SIZE[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	VCL_SIZE[12:0]	Vertical Clipping Size Setting When the VCEN bit is 1, these bits specify the clipping size for vertical clipping processing. Through this processing, the area of the vertical size specified through the VCL_SIZE bits starting from the offset position specified through the VCL_OFST bits is determined as the valid image area. Accordingly, the bottom pixels beyond the (VCL_OFST + VCL_SIZE) size in the WPF0 input image are discarded. When the VCEN bit is 0, this setting is ignored. A value from 1 to 8190 can be specified. (VCL_OFST + VCL_SIZE) should not exceed the vertical size of the WPF input. If the setting shown in the bottom example in Figure 54.16 is made, VSPD does not operate correctly. When the WPF0 output format is YCbCr4:2:0, specify an even value in these bits.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
23:16	VCL_OFST[7:0]	Vertical Size Clipping Offset Value Setting These bits specify the offset size (pixels) from the top-end of the image in vertical size clipping when the VCEN bit is 1 (Figure 54.16). The top of the image input to the WPF is cut off for the size specified in these bits. When the VCEN bit is 0, this setting is ignored. A value from 0 to 255 can be specified. (VCL_OFST + VCL_SIZE) should not exceed the vertical size of the WPF input. If the setting shown in the bottom example in Figure 54.16 is made, VSPD does not operate correctly.	R/W
27:24	—	These bits are read as 0. The write value should be 0.	R/W
28	VCEN	Vertical Size Clipping Enable/Disable Enables or disables clipping of the vertical size of the WPF0 input image. 0: Vertical size clipping is disabled. 1: Vertical size clipping is enabled.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

54.4.7.4 VI6_WPF0_OUTFMT : WPF0 Output Format Register

Base address: VSPD = 0x920E_0000

Offset address: 0x100C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PDV[7:0]								PXA	ODE	—	—	—	ROT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPYCS	SPUVS	DITH[1:0]		WRTM[2:0]			CSC	—	WRFMT[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	WRFMT[6:0]	WPF Output Image Format Setting*1 *2 *3 These bits select the format of the image output from the WPF0 to the external memory from among those listed in Table 54.25 and Table 54.26.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CSC	Color Space Conversion Setting Enables or disables Y CbCr ↔ RGB color space conversion to be executed in the WPF0. The characteristics of color space conversion are determined by the WRTM setting. There are some points to be noted about the relationship between the CSC setting and output format (WRFMT). For details, see (*2) in section 54.4.6.3. VI6_RPFn_INFMT : RPFn Input Format Register (n = 0, 1). 0: Color space is not converted. 1: Color space is converted.	R/W

Bit	Symbol	Function	R/W
11:9	WRTM[2:0]	CSC Conversion Expression Setting These bits select the expression for color space conversion. The conversion direction is RGB to YCbCr when the format specified in the WRFMT bits is RGB, or YCbCr to RGB when the format is YCbCr. 0 0 0: BT.601 Y CbCr [16,235/240] ↔ RGB [0,255] 0 0 1: BT.601 Y CbCr [0,255] ↔ RGB [0,255] 0 1 0: BT.709 Y CbCr [16,235/240] ↔ RGB [0,255] 0 1 1: BT.709 Y CbCr [16,235/240] ↔ RGB [16,235] Others: Setting prohibited.	R/W
13:12	DITH[1:0]	Ordered Dither (mode B) Enable/Disable When the output format specified through the WRFMT bits is RGB with 18 bpp (260000 colors) or less, the color reduction processing is applied to match the number of colors. The color reduction processing may generate the artifacts of pseudo gradation, which can be suppressed through dithering. The DITH bits enable or disable dithering during color reduction. When the output format specified through the WRFMT bits is YCbCr, set these bits to 0. And when VI6_WPF0_OUTFMT.CSC is set to 1, set these bits to 0 even in the case that the output format specified through the WRFMT bits is RGB. When ODE bit in the register is 1, set these bits to 0. 0 0: Dithering (mode B) is disabled. 1 1: Dithering (mode B) is enabled. Others: Setting prohibited.	R/W
14	SPUVS	WPF Output Mode Setting 2 When the output format is NV61, set this bit to 1 and set the WRFMT bits to 65 (0x41). When the output format is NV21, set this bit to 1 and set the WRFMT bits to 66 (0x42). When the output format is YVYU, set this bit and the SPYCS bit to 1 and set the WRFMT bits to 71 (0x47). In other cases, set this bit to 0.	R/W
15	SPYCS	WPF Output Mode Setting 1 When the output format is YUY2, set this bit to 1 and set the WRFMT bits to 71 (0x47). When the output format is YVYU, set this bit and the SPUVS bit to 1 and set the WRFMT bits to 71 (0x47). In other cases, set this bit to 0.	R/W
18:16	ROT[2:0]	Rotation Processing Select These bits select the flipping processing to be applied to the WPF0 output image. Figure 54.18 shows the correspondence between the original image and the flipping result according to each setting. Note that the destination address setting should be changed according to the setting of these bits. For details, see section 54.4.7.9. VI6_WPF0_DSTM_ADDR_Y : WPF0 Destination Y/RGB Address Register. When the LIF module is used (VI6_LIF_CTRL.LIF_EN = 1), set ROT[2:0] to 0. 0: No flipping 1: Vertical flipping	R/W
21:19	—	These bits are read as 0. The write value should be 0.	R/W
22	ODE	Ordered Dither (mode A) Enable/Disable When the output format specified through the WRFMT bits is YCbCr, set this bit to 0. And when VI6_WPF0_OUTFMT.CSC is set to 1, set this bit to 0 even in the case that the output format specified through the WRFMT bits is RGB. Ordered dither is available only for 18 bpp. So, when ODE bit is equal to 1, set WRFMT at 18 bpp format. When ODE bit is equal to 0, WPF dither method is specified by DITH[1:0] in the register. Ordered dither (mode A) is recommended rather than ordered dither (mode B) in case of 18 bpp. 0: Ordered dither (mode A) is disabled. 1: Ordered dither (mode A) is enabled.	R/W

Bit	Symbol	Function	R/W
23	PXA	PAD Data Select Selects the value to be stored in the bit field indicated as PAD or P in the packed RGB output formats shown in Table 54.26. Both the value specified in the PDV bits and the α data input from the DPR to WPF are 8 bits, but some of the PAD and P bit fields shown in Table 54.25 are four bits or one bit. When the target bit field is not 8 bits, the number of bits in the PDV value and the α data input from the DPR to WPF is reduced according to the VI6_WPF0_RNDCTRL.ABRM setting. For bit count reduction, see Figure 54.17 and the description of VI6_WPF0_RNDCTRL.ABRM. 0: The value specified in the PDV bits is stored in the PAD shown in Table 54.25. 1: The α value output from DPR in pixel units is stored in the PAD shown in Table 54.25.	R/W
31:24	PDV[7:0]	PAD Value in Output Packed Data*4 These bits specify the value to be stored in the bit field indicated as PAD or P in the output formats shown in Table 54.25. To store this value in PAD, set the PXA bit to 0.	R/W

Note 1. Number of output pixels
When YCbCr4:2:2 is specified through WRFMT, the horizontal size of the output image should be a multiple of 2 pixels. When YCbCr4:2:0 is specified, the vertical and horizontal sizes of the output image should be multiples of 2 pixels. Specify an appropriate data flow of the source RPF® DPR® target WPF so that the size of the image input to the target WPF satisfies the above restrictions. In particular, when the data flow includes a module or a function that modifies (up-scales, down-scales, or clips) the image size, take special care about the module or function settings.

Note 2. Output lines in YCbCr4:2:0
In the YCbCr4:2:0 output format, the number of chrominance lines in the vertical direction is one-half the number of luminance lines. For this reason, the WPF outputs only even-numbered chrominance lines (lines 0, 2, 4, 6, ...) (conversion from (A) to (B) in Figure 54.19). When vertical flipping is also specified through the ROT bits, the flipping processing is executed last and the chrominance line locations are inverted (lines 1, 3, 5, 7, …) in the output image ((C) in Figure 54.19).

Note 3. Down sampling of CbCr in horizontal direction in YCbCr4:2:0 or YCbCr4:2:2
In the YCbCr4:2:2 or YCbCr4:2:0 output format, method of down sample of Cb/Cr in horizontal direction is average of neighbor two pixels.

Note 4. A value from 0 to 255 can be specified.

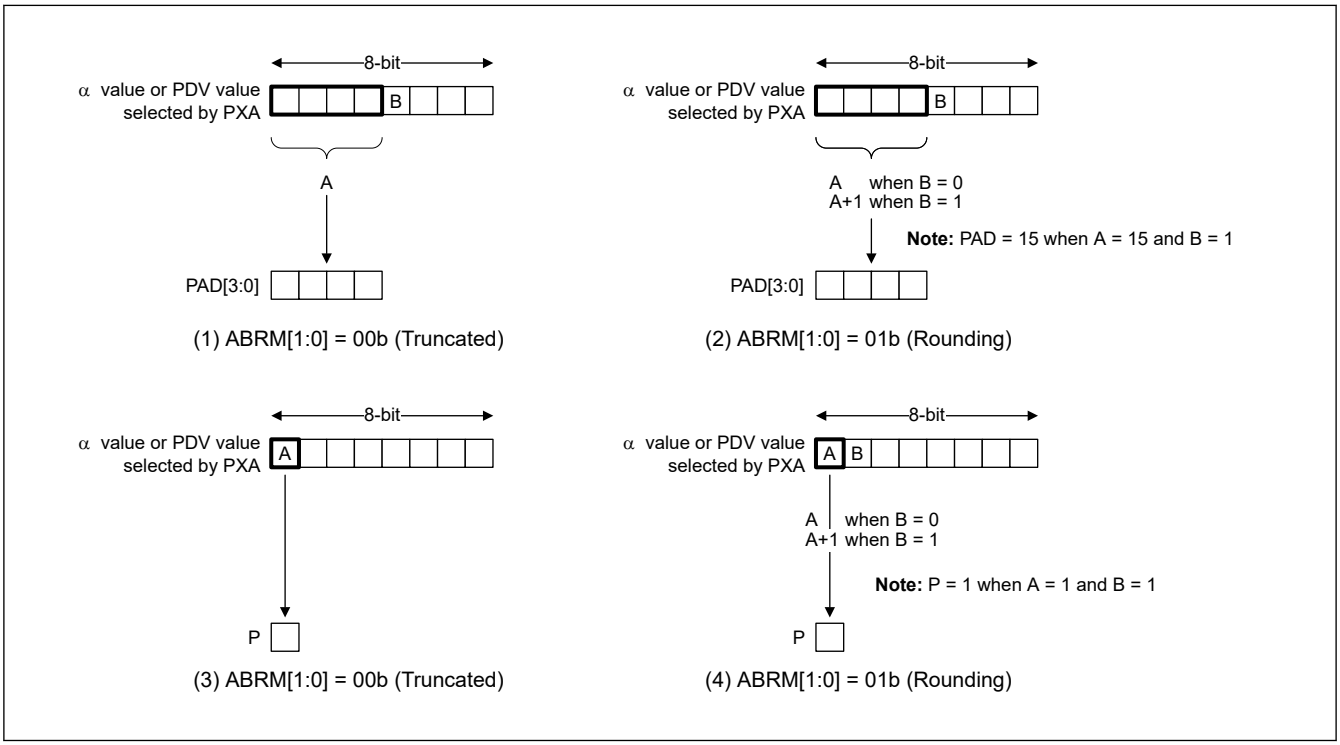
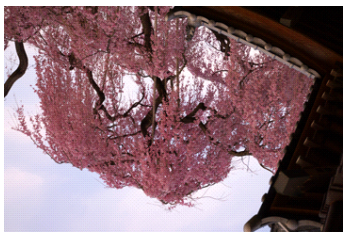


Figure 54.17 Selection of PAD value and reduction of bit count through PXA setting

Without flipping (ROT = 0)

Vertical flipping



Vertical flipping (ROT = 1)

Figure 54.18 Correspondence between original image and flipping result according to ROT setting

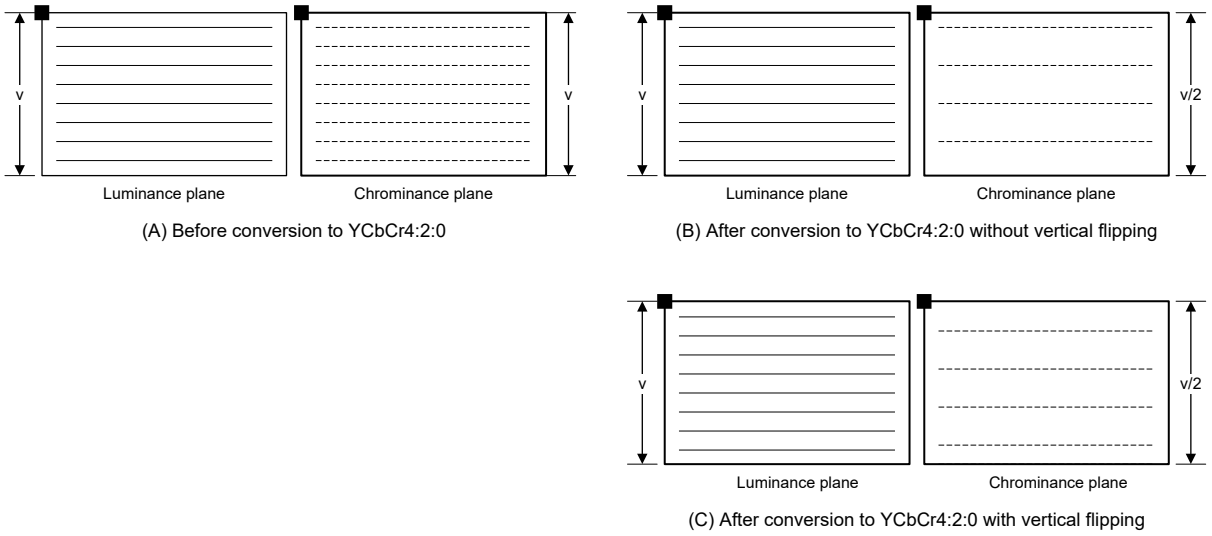


Figure 54.19 Chrominance output lines in YCbCr4:2:0 and vertical flipping result

Table 54.25 Packed RGB formats for WPF output (1 of 2)

WRFMT [6:0]	Bit per pixel	Phase	upper row - address / bottom row - bit field																													
			n								n + 1								n + 2								n + 3							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0x00	8	—	R0	R0	R0	G0	G0	G0	B0	B0	R1	R1	R1	G1	G1	G1	B1	B1	R2	R2	R2	G2	G2	G2	B2	B2	R3	R3	R3	G3	G3	G3	B3	B3
0x01	12	—	0	0	0	0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	0	0	0	0	R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1
0x02			R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	0	0	0	0	R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1	0	0	0	0
0x03	—	—	Reserved								Reserved								Reserved								Reserved							

Table 54.25 Packed RGB formats for WPF output (2 of 2)

WRFMT [6:0]	Bit per pixel	Phase	upper row - address / bottom row - bit field																													
			n								n + 1								n + 2								n + 3							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0x04	15	—	0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	0	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1
0x05			R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	0	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	0
0x06	16	—	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1
0x07	18	—	PAD								0	0	0	0	0	0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0
0x08			R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	0	0	0	0	0	0	PAD							
0x09			0	0	0	0	0	0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	PAD							
0x0A			PAD								R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	0	0	0	0	0	0
0x0B			PAD								0	0	R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	B0
0x0C			0	0	R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	B0	PAD							
0x0D			PAD								R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	B0	0	0
0x0E			R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	B0	0	0	PAD							
0x0F		0	0	0	0	0	0	0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	0	0	0	0	0	0	R1	R1
		1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1	0	0	0	0	0	0	R2	R2	R2	R2	R2	R2	G2	G2	G2	G2
		2	G2	G2	B2	B2	B2	B2	B2	B2	0	0	0	0	0	0	R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3
0x10		0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	0	0	0	0	0	0	R1	R1	R1	R1	R1	R1	G1	G1
		1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1	0	0	0	0	0	0	R2	R2	R2	R2	R2	R2	G2	G2	G2	G2	G2	G2	B2	B2	B2	B2
		2	B2	B2	0	0	0	0	0	0	R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3	0	0	0	0	0	0
0x11		0	0	0	R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	B0	0	0	R1	R1	R1	R1	R1	R1
		1	0	0	G1	G1	G1	G1	G1	G1	0	0	B1	B1	B1	B1	B1	B1	0	0	R2	R2	R2	R2	R2	R2	0	0	G2	G2	G2	G2	G2	G2
		2	0	0	B2	B2	B2	B2	B2	B2	0	0	R3	R3	R3	R3	R3	R3	0	0	G3	G3	G3	G3	G3	G3	0	0	B3	B3	B3	B3	B3	B3
0x12		0	R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	B0	0	0	R1	R1	R1	R1	R1	R1	0	0
		1	G1	G1	G1	G1	G1	G1	0	0	B1	B1	B1	B1	B1	B1	0	0	R2	R2	R2	R2	R2	R2	0	0	G2	G2	G2	G2	G2	G2	0	0
		2	B2	B2	B2	B2	B2	B2	0	0	R3	R3	R3	R3	R3	R3	0	0	G3	G3	G3	G3	G3	G3	0	0	B3	B3	B3	B3	B3	B3	0	0
0x13	24	—	PAD								R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0
0x14			R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	PAD							
0x15		0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	R1	R1	R1
		1	G1	G1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1	B1	B1	R2	R2	R2	R2	R2	R2	R2	R2	G2	G2	G2	G2	G2	G2	G2	G2
		2	B2	B2	B2	B2	B2	B2	B2	B2	R3	R3	R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3	B3	B3
0x16	18	—	0	0	0	0	0	0	0	R0	R0	R0	R0	R0	R0	G0	G0	G0	0	0	0	0	0	0	0	G0	G0	G0	B0	B0	B0	B0	B0	B0
0x17			0	0	0	0	0	0	0	0	0	0	0	0	0	0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0
0x18	24	0	B0	B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	R0	B1	B1	B1	B1	B1	B1	B1	B1
		1	G1	G1	G1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1	R1	R1	R1	B2	B2	B2	B2	B2	B2	B2	B2	G2	G2	G2	G2	G2	G2	G2	G2
		2	R2	R2	R2	R2	R2	R2	R2	R2	B3	B3	B3	B3	B3	B3	B3	B3	G3	G3	G3	G3	G3	G3	G3	G3	R3	R3	R3	R3	R3	R3	R3	R3
0x19	12	—	PAD				R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	PAD				R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1
0x1A		—	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	PAD				R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1	PAD			
0x1B	15	—	P	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	P	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1
0x1C		—	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	P	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	P
0x1D	12	—	PAD				B0	B0	B0	B0	G0	G0	G0	G0	R0	R0	R0	R0	PAD				B1	B1	B1	B1	G1	G1	G1	G1	R1	R1	R1	R1
0x1E		—	B0	B0	B0	B0	G0	G0	G0	G0	R0	R0	R0	R0	PAD				B1	B1	B1	B1	G1	G1	G1	G1	R1	R1	R1	R1	PAD			
0x1F	15	—	P	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	P	B1	B1	B1	B1	B1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1
0x20		—	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	P	B1	B1	B1	B1	B1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1	P
0x21	18	0	0	0	B0	B0	B0	B0	B0	B0	0	0	G0	G0	G0	G0	G0	G0	0	0	R0	R0	R0	R0	R0	R0	0	0	B1	B1	B1	B1	B1	B1
		1	0	0	G1	G1	G1	G1	G1	G1	0	0	R1	R1	R1	R1	R1	R1	0	0	B2	B2	B2	B2	B2	B2	0	0	G2	G2	G2	G2	G2	G2
		2	0	0	R2	R2	R2	R2	R2	R2	0	0	B3	B3	B3	B3	B3	B3	0	0	G3	G3	G3	G3	G3	G3	0	0	R3	R3	R3	R3	R3	R3
0x22	24	—	PAD								B0	B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	R0
0x23	16	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0
0x24 to 0x3F	—	—	Reserved								Reserved								Reserved								Reserved							

Table 54.26 Packed YCbCr formats for WPF output

WRFMT[6:0]	Packed YCbCr output format	Reference
0x40	YCbCr4:4:4 semi-planar	Figure 54.9*4
0x41	YCbCr4:2:2 semi-planar (NV16, NV61*1)	
0x42	YCbCr4:2:0 semi-planar (NV12, NV21*1)	
0x43 to 0x45	Reserved	—
0x46	YCbCr4:4:4 interleaved	Figure 54.10*4
0x47	YCbCr4:2:2 interleaved type 0 (UYVY, YUY2*2, YVYU*3)	
0x48	YCbCr4:2:2 interleaved type 1	
0x49	YCbCr4:2:0 interleaved*5	
0x4A	YCbCr4:4:4 planar	Figure 54.11*4
0x4B	YCbCr4:2:2 planar (YV16)	
0x4C	YCbCr4:2:0 planar (YV12, I420)	
0x4D to 0x7F	Reserved	—

Note 1. When the output format is NV61 or NV21, set SPUVS (bit 14) to 1.
Note 2. When the output format is YUY2, set SPYCS (bit 15) to 1.
Note 3. When the output format is YVYU, set SPUVS (bit 14) to 1 and SPYCS (bit 15) to 1.
Note 4. Figure 54.12 shows the definition of memory address for each pixel in Figure 54.9 to Figure 54.11.
Note 5. Each line of plane is written twice, so byte/pixel of YCbCr420ITL is 3 byte/pixel (same as YCbCr444ITL).

For details of each YCbCr format, see Figure 54.9 to Figure 54.12. In these figures, registers for the RPF are indicated; read them as registers for the WPF as follows.

(RPF registers in the figures)	→	(Corresponding WPF registers)
VI6_RPFn_SRCM_ADDR_Y.SRCM_ADDR_Y	→	VI6_WPFn_DSTM_ADDR_Y.DSTM_ADDR_Y
VI6_RPFn_SRCM_ADDR_C0.SRCM_ADDR_C0	→	VI6_WPFn_DSTM_ADDR_C0.DSTM_ADDR_C0
VI6_RPFn_SRCM_ADDR_C1.SRCM_ADDR_C1	→	VI6_WPFn_DSTM_ADDR_C1.DSTM_ADDR_C1

54.4.7.5 VI6_WPF0_DSWAP : WPF0 Data Swapping Register

Base address: VSPD = 0x920E_0000

Offset address: 0x1010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	P_LLS	P_LWS	P_WDS	P_BTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	P_BTS	WPF Output Data Swapping in Byte Units*1 0: Data swapping in byte (8-bit) units is disabled. 1: Data swapping in byte (8-bit) units is enabled.	R/W
1	P_WDS	WPF Output Data Swapping in Word Units*1 0: Data swapping in word (16-bit) units is disabled. 1: Data swapping in word (16-bit) units is enabled.	R/W
2	P_LWS	WPF Output Data Swapping in long word Units*1 0: Data swapping in long word (32-bit) units is disabled. 1: Data swapping in long word (32-bit) units is enabled.	R/W

Bit	Symbol	Function	R/W
3	P_LLS	WPF Output Data Swapping in LONG LWORD Units*1 0: Data swapping in LONG LWORD (64-bit) units is disabled. 1: Data swapping in LONG LWORD (64-bit) units is enabled.	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The effect of this bit setting is the same as data swapping in the RPF; see Table 54.20.

Table 54.20 shows the data order before and after swapping according to the long long word, long word, word, and byte swapping settings.

When data order in memory for each format is the same as in Table 54.25 for RGB format and Figure 54.9 to Figure 54.12 for YCbCr format, set 1111b to {*_LLS, *_LWS, *_WDS, *_BTS}. If data order is not the same as the definition, change data order within 16-byte unit by these bits as shown in Table 54.20.

54.4.7.6 VI6_WPF0_RNDCTRL : WPF0 Rounding Control Register

Base address: VSPD = 0x920E_0000

Offset address: 0x1014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CBRM	—	—	ABRM[1:0]		ATHRESH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CLMD[1:0]		—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0. The write value should be 0.	R/W
13:12	CLMD[1:0]	Color Data Clipping These bits specify the method for clipping the YCbCr color data output from the WPF. When RGB color data is output from the WPF, set these bits to 0. 0 0: Output value is not clipped (0-255) 0 1: Output value is clipped: YCbCr mode 1 (16-235 (Y), 16-240 (Cb/Cr)) 1 0: Output value is clipped: YCbCr mode 2 (Y/Cb/Cr = 1-254) 1 1: Setting prohibited.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
23:16	ATHRESH[7:0]	Threshold for Conversion to 1-Bit α Data*1 These bits specify the threshold value used for conversion from 8-bit α data to one bit when the ABRM bits are set to 10b. When the 8-bit α value before bit count reduction is equal to or smaller than the ATHRESH value, 0 is stored as the reduced 1-bit α data. In other cases, 1 is stored as the 1-bit α data.	R/W
25:24	ABRM[1:0]	Bit Count Reduction Method Selection for Data Storage in PAD These bits specify the method for reducing the number of bits when the data selected through VI6_WPF0_OUTFMT.PXA is stored in the bit fields indicated as PAD or P in Table 54.25 and the target bit field is four bits or one bit. A value of 10b can be specified only when the packed RGB format specified through VI6_WPF0_OUTFMT.WRFMT includes a 1-bit P field. In this case, when the data selected through VI6_WPF0_OUTFMT.PXA is greater than the ATHRESH value, 1 is stored in the P field; when the selected data is not greater than the ATHRESH value, 0 is stored. 0 0: Bit count conversion: The lower-order bits are truncated. 0 1: Bit count conversion: Rounding (rounding off) 1 0: Bit count conversion: Comparison with the threshold value (this setting is allowed only when the storage field is one bit) 1 1: Setting prohibited.	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
28	CBRM	Bit Count Reduction Method Selection for Data Storage in Packed RGB This bit specifies the method for reducing the number of bits when data is stored in the bit fields indicated as R, G, and B in Table 54.25 and the target bit fields are not 8 bits. 0: Bit count conversion: The lower-order bits are truncated. 1: Bit count conversion: Rounding (rounding off)	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. A value from 0 to 255 can be specified.

54.4.7.7 VI6_WPF0_DSTM_STRIDE_Y : WPF0 Destination Y Plane Memory Stride Register

Base address: VSPD = 0x920E_0000

Offset address: 0x101C

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PICT_STRD_Y[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PICT_STRD_Y[15:0]	Memory Stride of Destination Picture Y/RGB Plane*1 For a destination picture in the external memory to be written to by the WPF0, these bits specify the memory stride in byte units as shown in Figure 54.20.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. A value from 0x0000 to 0xFFFF can be specified.

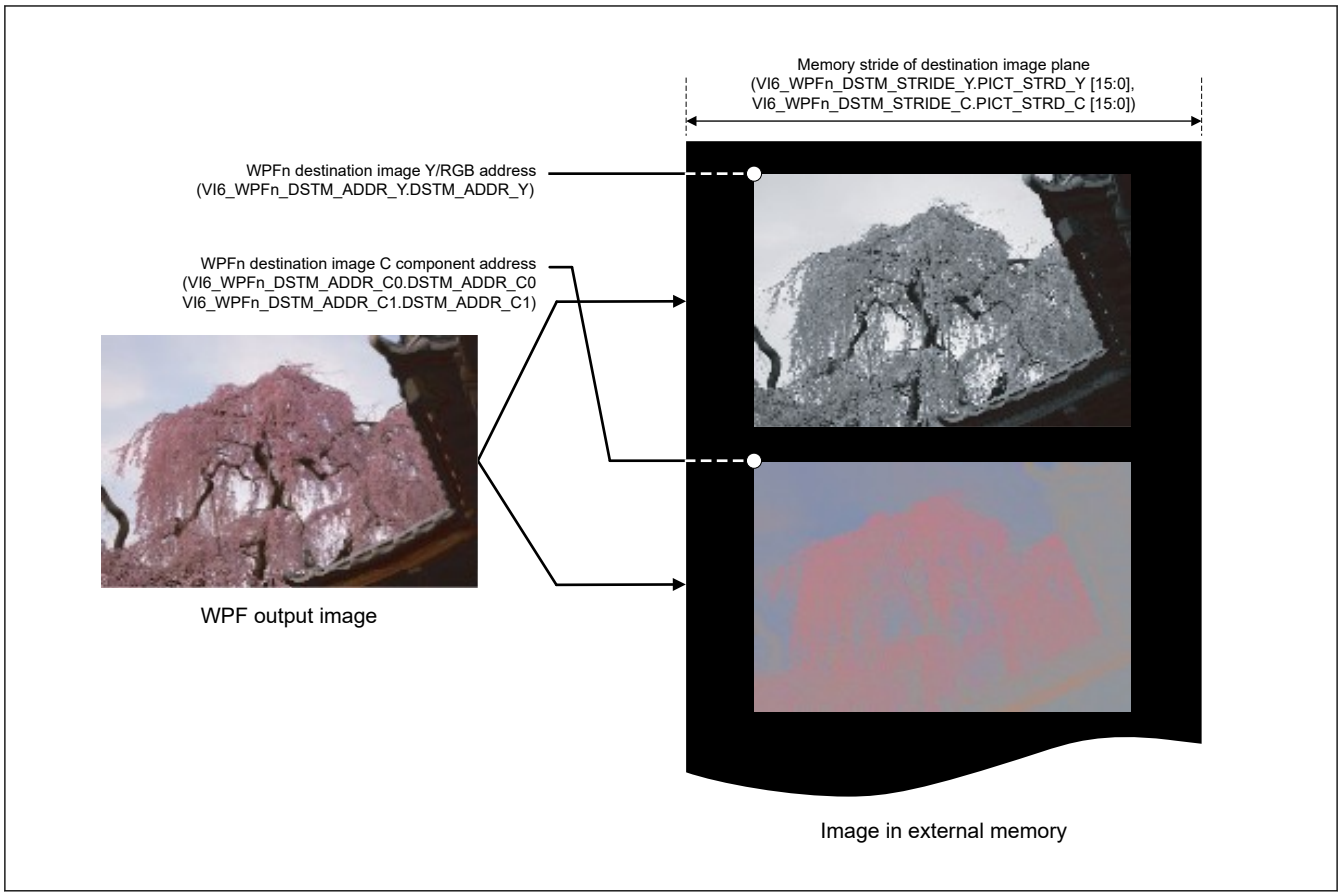


Figure 54.20 Writing image data to destination area in WPFn

54.4.7.8 VI6_WPF0_DSTM_STRIDE_C : WPF0 Destination C Plane Memory Stride Register

Base address: VSPD = 0x920E_0000

Offset address: 0x1020

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PICT_STRD_C[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PICT_STRD_C[15:0]	Memory Stride of Destination Picture C Plane*1 For a C plane of the destination picture in the external memory to be written to by the WPF0, these bits specify the memory stride in byte units as shown in Figure 54.20. When the WPF0 outputs images in an RGB format, this setting is not used. When the WPF0 outputs images in YCbCr planar format, this setting is applied to both the Cb and Cr planes.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. A value from 0x0000 to 0xFFFF can be specified.

54.4.7.9 VI6_WPF0_DSTM_ADDR_Y : WPF0 Destination Y/RGB Address Register

Base address: VSPD = 0x920E_0000

Offset address: 0x1024

Bit position:	31																													0
Bit field:	DSTM_ADDR_Y[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DSTM_ADDR_Y[31:0]	Destination Image Y/RGB Plane Storing Address*1 For storing a Y plane or packed RGB plane of destination-image to be written to by the WPF0, these bits specify the address in byte units in the method described later.	R/W

Note 1. A value from 0x0000 0000 to 0xFFFF FFFF can be specified.

[Destination Address Specification Method]

When flipping is not used, the start address of a frame (address FHA shown in Figure 54.21) should be specified as the destination address. When flipping is used, the destination address is not the frame start address (FHA); one of addresses A0 and A2 shown in Figure 54.21 should be selected according to the combination of desired flipping (VI6_WPFn_OUTFMT.ROT setting).

To strictly define locations A0 and A2, let the horizontal size of the output image be H, the vertical size of the output image be V, and the memory stride (VI6_WPFn_DSTM_STRIDE_Y/C setting) be S as shown in Figure 54.21. Calculate the destination address (one of A0 and A2) using the formula shown in Table 54.27 and specify it in the destination address storing register.

The values of variables L in Table 54.27 depend on the other register settings and luminance and chrominance components. These values should be obtained by Table 54.28 when calculating the address to be specified in VI6_WPFn_DSTM_ADDR_Y, or Table 54.29 when calculating the address to be specified in VI6_WPFn_DSTM_ADDR_C0 or VI6_WPFn_DSTM_ADDR_C1.

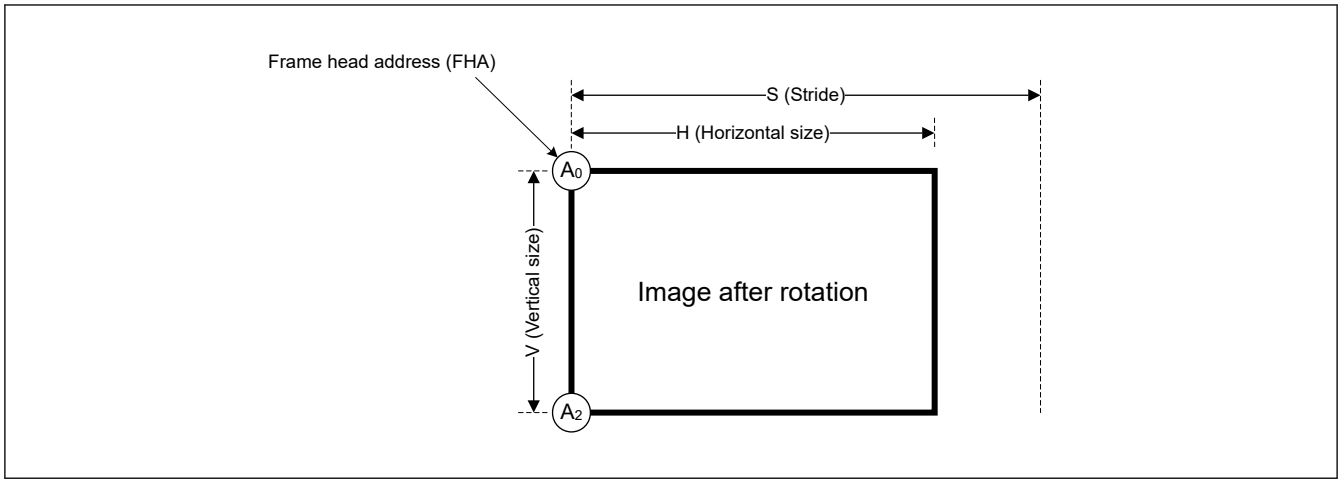


Figure 54.21 Location of destination address to be specified for flipping

Table 54.27 Destination address A0, A1, A2, and A3 calculation formulas

VI6_WPFn_OUTFMT.ROT setting	Formula for calculating address to be set in VI6_WPFn_DSTM_ADDR_Y, VI6_WPFn_DSTM_ADDR_C0, and VI6_WPFn_DSTM_ADDR_C1
0	A0 = FHA
1	A2 = FHA + (V × L − 1) × S

Table 54.28 Value of L according to VI6_WPFn_OUTFMT.WRFMT setting (for RGB and Luminance Y address calculation)

VI6_WPFn_OUTFMT.WRFMT	L
73	0.5
0 to 2, 4 to 35, 64 to 66, 70 to 72, or 74 to 76	1

Table 54.29 Value of L according to VI6_WPFn_OUTFMT.WRFMT setting (for Chrominance C0 and C1 address calculation)

VI6_WPFn_OUTFMT.WRFMT	L
70 to 73	Not defined
66, 76	0.5
64 to 65, 74 to 75	1

54.4.7.10 VI6_WPF0_DSTM_ADDR_C0 : WPF0 Destination Chroma Address Register 0

Base address: VSPD = 0x920E_0000

Offset address: 0x1028

Bit position:	31																												0
Bit field:	DSTM_ADDR_C0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DSTM_ADDR_C0[31:0]	Destination Image C Plane Storing Address 0*1 For storing a C plane of destination-image to be written to by the WPF0, these bits specify the address in byte units. See the description of VI6_WPF0_DSTM_ADDR_Y for settings. Here, the C plane indicates the combined CbCr plane when a semi-planar format is selected from the packed YCbCr formats shown in Table 54.26 or the Cb plane when a planar format is selected. When an interleaved format is selected or the output is in an RGB format, this setting is not used.	R/W

Note 1. A value from 0x0000 0000 to 0xFFFF FFFF can be specified. See Figure 54.21 in section 54.4.7.9. VI6_WPF0_DSTM_ADDR_Y : WPF0 Destination Y/RGB Address Register for settings.

54.4.7.11 VI6_WPF0_DSTM_ADDR_C1 : WPF0 Destination Chroma Address Register 1

Base address: VSPD = 0x920E_0000

Offset address: 0x102C

Bit position:	31																												0
Bit field:	DSTM_ADDR_C1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DSTM_ADDR_C1[31:0]	Destination Image C Plane Storing Address 1*1 For storing the Cr plane, these bits specify the address in byte units when the WPF0 outputs images to the external memory in YCbCr planar format shown in Table 54.26. See the description of VI6_WPF0_DSTM_ADDR_Y for settings. This setting is not used when the WPF outputs in YCbCr format that is not a planar format or in an RGB format.	R/W

Note 1. A value from 0x0000 0000 to 0xFFFF FFFF can be specified. See Figure 54.21 in section 54.4.7.9. VI6_WPF0_DSTM_ADDR_Y : WPF0 Destination Y/RGB Address Register for settings.

54.4.7.12 VI6_WPF0_WRBCK_CTRL : WPF0 LIF Write Back Control Register

Base address: VSPD = 0x920E_0000

Offset address: 0x1034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WBMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	WBMD[1:0]	Display Data Write Back Control These bits are used for selecting the write back mode when the value of VI6_LIF0_CTRL.LIF_EN bit is set to 1. 0: Write Back Disabled. 1: Write Back Enabled.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

54.4.8 DPR Control Registers

54.4.8.1 Concept of DPR Settings

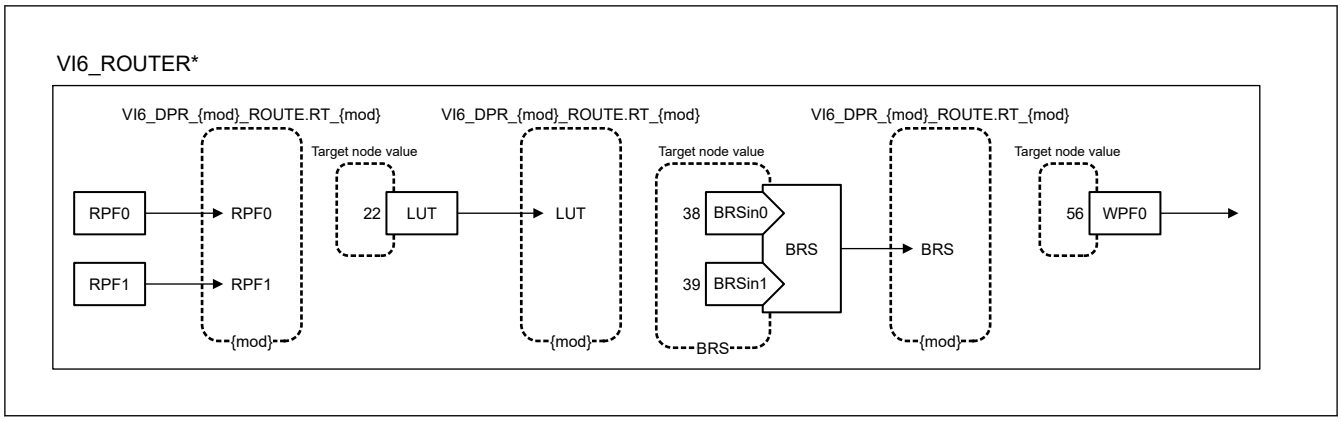


Figure 54.22 Node register names and target node values on data path router

In the VSPD internal data path, the order of processes can be specified as desired. The module for performing each process has a unique node value. Set each bit field in VI6_DPR_*_ROUTE to an appropriate node value shown in Figure 54.22 to specify the target node to be connected behind each module.

For DPR settings, all of the following restrictions should be observed. If any of them is violated, even the WPF paths operating correctly at that time will be affected as well as the WPF paths connected through the DPR, and correct operation will not be guaranteed.

1. Specify 63 for the output node values of all RPFs and processing modules that are not used in the DPR. Here, make sure that no module is connected to a module for which 63 is specified as the node value.
   Note: VI6_DPR_{mod}_ROUTE.RT are not implemented in VSPD which does not have {mod}. See Figure 54.1 whether VSPD has {mod} or not.
2. When specifying a value other than 63 for an output node value in the DPR, make sure that valid inputs (RPF0 to RPF1 or virtual RPF) and a target WPF are determined.
3. Only one module can be connected to each module; specifying the same target node value for two or more modules is prohibited.
4. Desired modules can be connected between each RPF and BRS input port, but all RPFs specified as the sources for a BRS input port should have the same target WPF.
5. Make appropriate routing or RPF register settings so that the color space formats (RGB/YCbCr) for all BRS input ports are the same.
6. Do not connect the output of any module as the input to the same module (in the BRS case, any input port) even when there is another module between the output and input (creating a loop is prohibited).
7. Each node can be used only once throughout all paths from RPFn to WPFn. When a module shown in Figure 54.22 is assigned in one RPF to WPF path, it cannot be used in another RPF to WPF path.
8. While a WPF is operating, modifying the DPR connection settings in VI6_DPR_*_ROUTE is prohibited for modules used by the WPF but allowed for modules not used by the WPF. Be careful not to accidentally modify the settings of the modules included in the WPF path that is operating.
9. Be careful to do the following setting when BRS is used.
   VI6_DPR_BRS_ROUTE.BRSSEL = 1

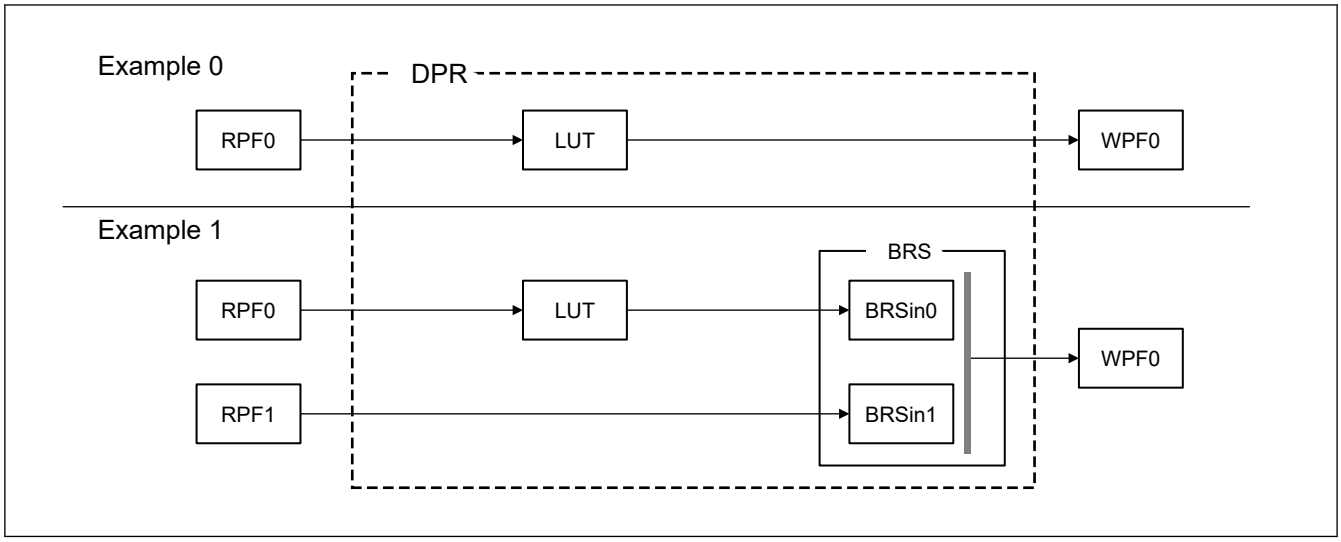


Figure 54.23 Examples of internal data path routing

Figure 54.23 shows examples of internal data path routing. Example 0 is WPF0 processing (RPF0 is the source RPF), and example 1 is also WPF0 (RPF0 to RPF1 are the source RPFs). Each example has the configuration shown in Figure 54.23. Example 0 performs LUT (e.g. γ correction) processing. Example 1 performs LUT processing (e.g. γ correction), for input 0 (RPF0) and then applies blending or raster operation between the resultant data and input data 1 (RPF1). The VI6_DPR_*_ROUTE settings for these examples are shown in Table 54.30. The bit fields for the modules that are not used in the examples should be set to 63.

Table 54.30 VI6_DPR_*_ROUTE Register Settings in Connection Examples Shown in Figure 54.23. Although both examples show in Figure 54.23 include operation in image processing modules, connect the RPF to the WPF directly when only image format conversion or packed format conversion is required.

Table 54.30 Examples of internal data path routing

	Register name	Setting	
Example 0	VI6_DPR_RPF0_ROUTE.RT_RPF0	22	(To LUT)
	VI6_DPR_RPF1_ROUTE.RT_RPF1	63	(UNUSED)
	VI6_DPR_LUT_ROUTE.RT	56	(To WPF0)
	VI6_DPR_BRS_ROUTE.RT	63	(UNUSED)
Example 1	VI6_DPR_RPF0_ROUTE.RT_RPF0	22	(To LUT)
	VI6_DPR_RPF1_ROUTE.RT_RPF1	39	(To BRSin1)
	VI6_DPR_LUT_ROUTE.RT	38	(To BRSin0)
	VI6_DPR_BRS_ROUTE.RT	56	(To WPF0)

54.4.8.2 VI6_DPR_RPF0_ROUTE : RPF0 Routing Register

Base address: VSPD = 0x920E_0000

Offset address: 0x2000

Bit position:	31																								5						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RT_RPF0[5:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	RT_RPF0[5:0]	RPF0 Target Node Value These bits specify the target node value for RPF0. When using RPF0, see Figure 54.22 for settings. When RPF0 is not started through the VI6_WPF0_SRCRPF setting, set these bits to 63.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

54.4.8.3 VI6_DPR_RPF1_ROUTE : RPF1 Routing Register

Base address: VSPD = 0x920E_0000

Offset address: 0x2004

Bit position:	31																								5					0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RT_RPF1[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	RT_RPF1[5:0]	RPF1 Target Node Value These bits specify the target node value for RPF1. When using RPF1, see Figure 54.22 for settings. When RPF1 is not started through the VI6_WPF0_SRCRPF setting, set these bits to 63.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

54.4.8.4 VI6_DPR_WPF0_FPORCH : WPF0 Timing Control Register

Base address: VSPD = 0x920E_0000

Offset address: 0x2014

Bit position:	31																	13					8							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FP_WPF0[5:0]						—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
13:8	FP_WPF0[5:0]	WPF0 Internal Operation Timing Setting Set these bits to 5.	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

54.4.8.5 VI6_DPR_LUT_ROUTE : LUT Routing Register

Base address: VSPD = 0x920E_0000

Offset address: 0x203C

Bit position:	31								23							16		13					8				5			0
Bit field:	—	—	—	—	—	—	—	—	FXA[7:0]								—	—	FP[5:0]						—	—	RT[5:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	RT[5:0]	LUT Target Node Value These bits specify the target node value for the LUT. When using the LUT, see Figure 54.22 for settings. When not using the LUT, set these bits to 63.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
13:8	FP[5:0]	LUT Internal Operation Timing Setting Set these bits to 0.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
23:16	FXA[7:0]	Fixed α Output Value for LUT*1 The LUT does not support input/output of the α value. The α value input to the LUT is discarded, and the fixed α value specified in these bits is always output from the LUT.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. A value from 0 to 255 can be specified.

54.4.8.6 VI6_DPR_BRS_ROUTE : BRS Routing Register

Base address: VSPD = 0x920E_0000

Offset address: 0x2050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	BRSSEL	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	FP[5:0]						—	—	RT[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	RT[5:0]	BRS Target Node Value These bits specify the target node value for the BRS. When using the BRS, see Figure 54.22 for settings. When not using the BRS, set these bits to 63.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	FP[5:0]	BRS Internal Operation Timing Setting Set these bits to 0.	R/W
27:14	—	These bits are read as 0. The write value should be 0.	R/W
28	BRSSEL	Always set this bit to 1.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

54.4.9 LUT Control Register

54.4.9.1 VI6_LUT_CTRL : LUT Control Register

Base address: VSPD = 0x920E_0000

Offset address: 0x2800

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LUT_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LUT_EN	1D-LUT Enable/Disable Enables or disables the 1D-LUT function by the LUT. When the 1D-LUT is used, the color component curve information needs to be set separately in the LUT table. For the LUT table settings, see section 54.6.6.1. LUT. 0: 1D-LUT function is disabled. 1: 1D-LUT function is enabled.	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

In the LUT, various image processing, such as curves with high operation load (e.g., γ correction), negative-positive conversion, and gain adjustment of images, can be achieved by the data replacement processing by the 1D-LUT. As shown in Figure 54.24, the LUT replaces each component of the input pixel data using the set replacement table of 256 entries. For example, if the LUT is set as in Figure 54.25, when there is an input of 150, the data stored in address 150 of the 1D-LUT is read and output as the LUT output. Figure 54.25 shows a case in which the input and output become equal for convenience in explaining.

In the LUT settings shown in Figure 54.26, the input bits are reversed. This has the effect of negative-positive flipping. In Figure 54.27, γ correction (γ = 1.8 is shown as an example) is possible. As described above, information to be set in the LUT indicates LUT processing characteristics. If the same value is set for each component in the LUT, an equal effect can be obtained for each component of the input image when it is processed. If the LUT is set with different characteristics for each component, the processing characteristics can be changed for each component.

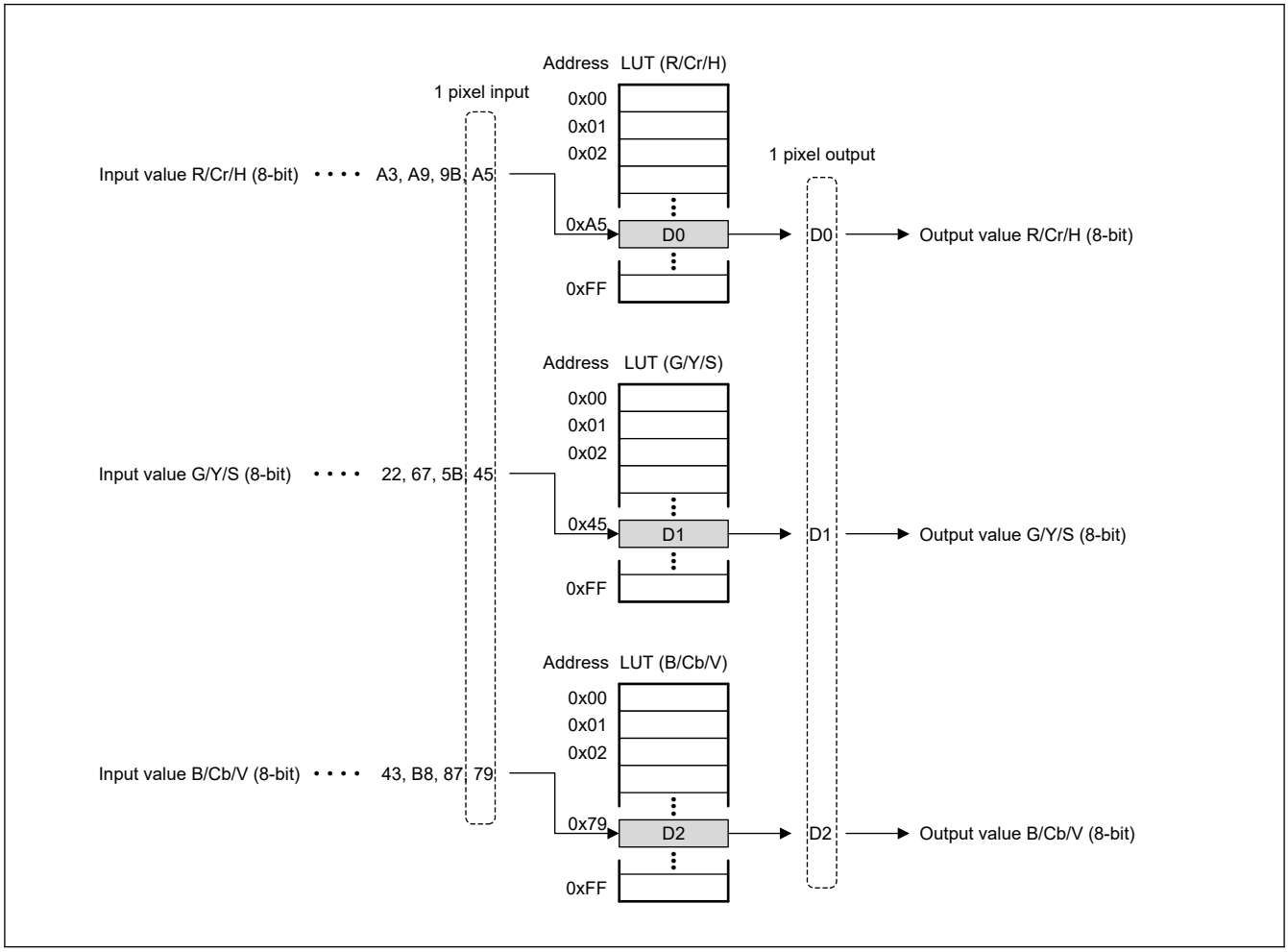


Figure 54.24 Relationship between input and output for 1D-LUT table

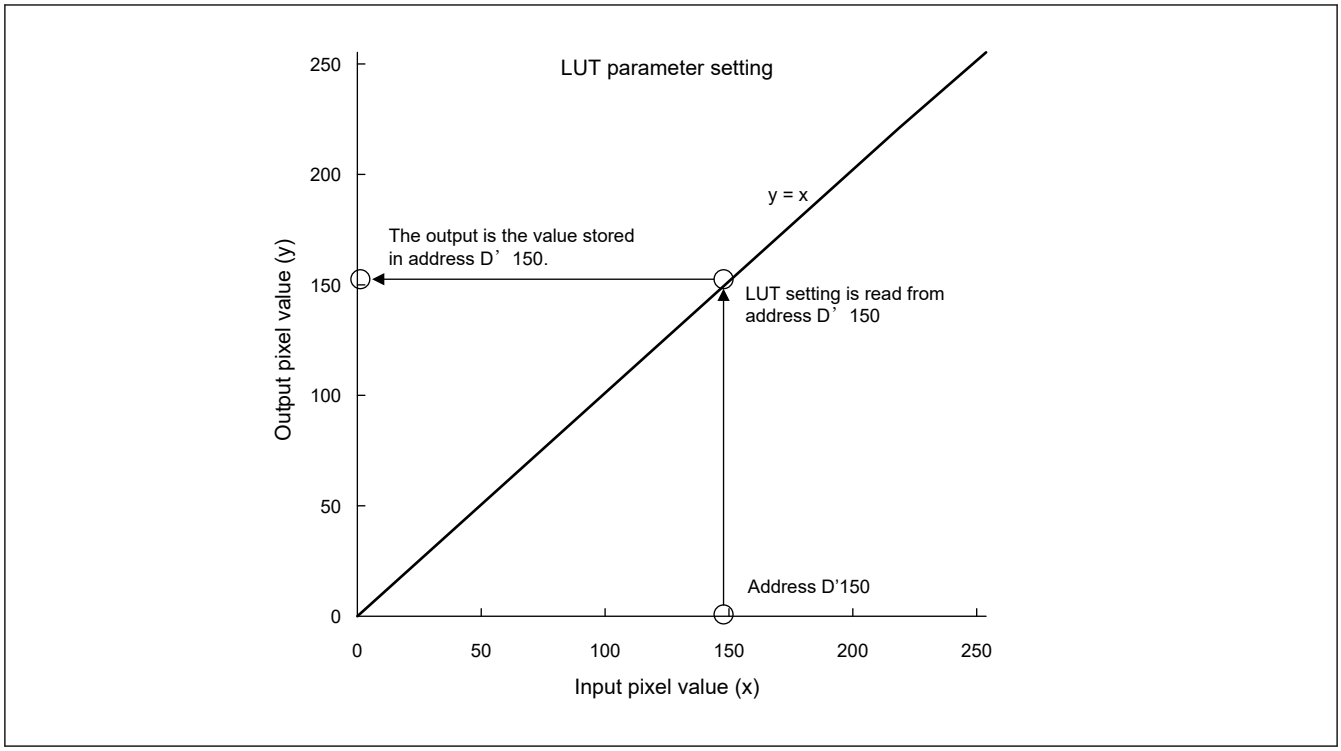


Figure 54.25 Setting example in which output becomes equal to input

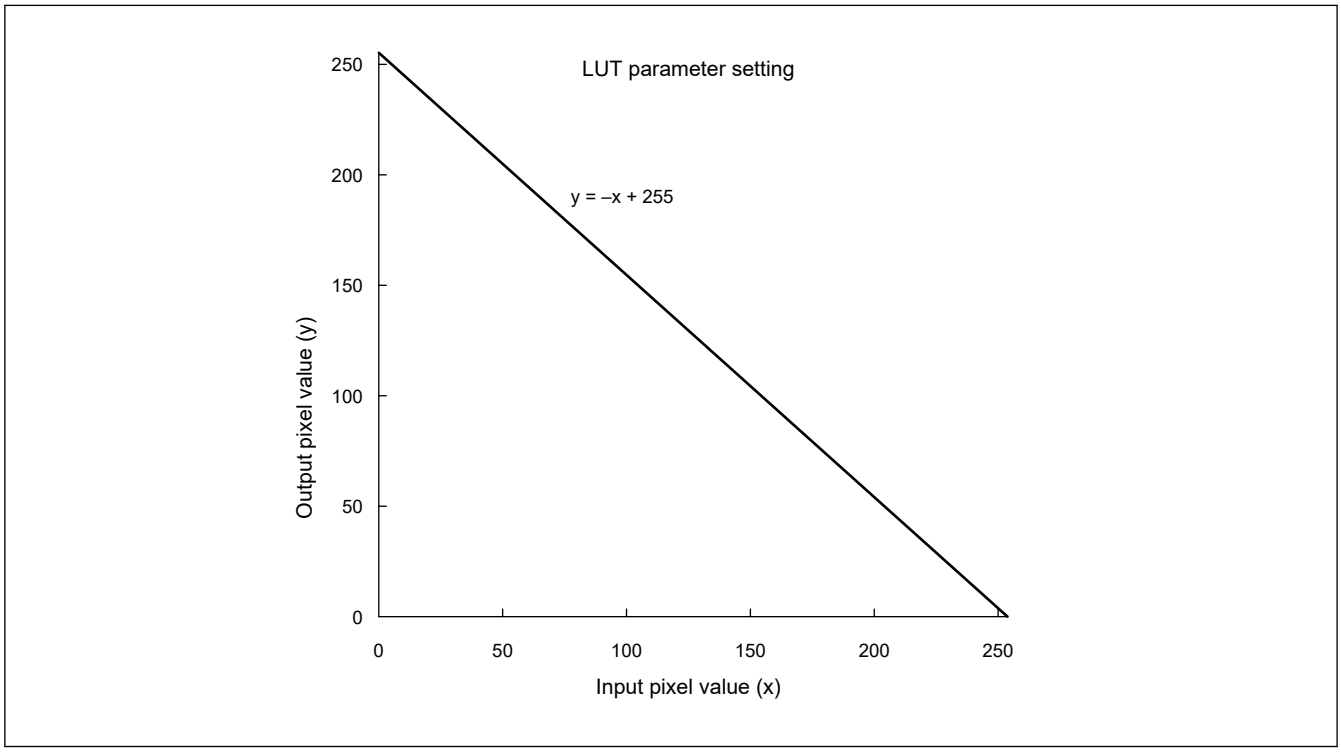


Figure 54.26 Setting example of negative-positive conversion

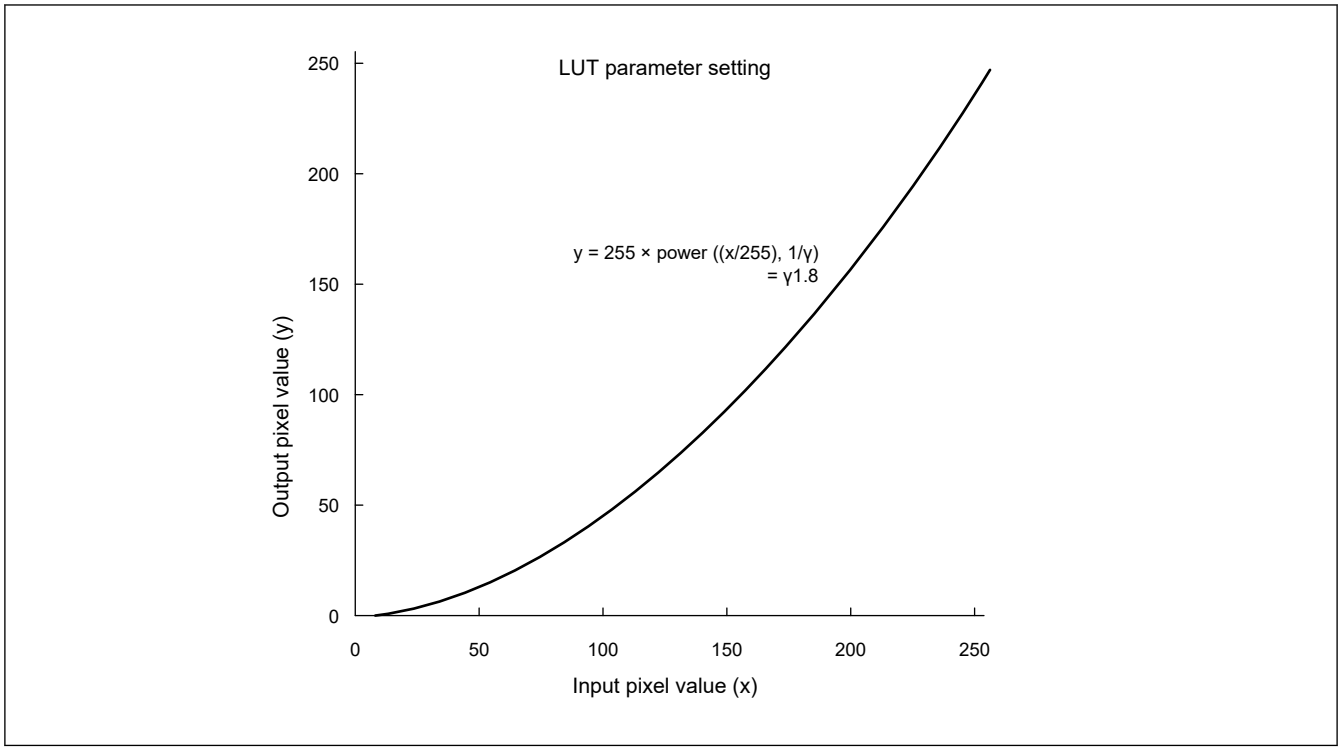


Figure 54.27 Setting example of γ correction

54.4.10 BRS Control Registers

54.4.10.1 VI6_BRS_INCTRL : BRS Input Control Register

Base address: VSPD = 0x920E_0000

Offset address: 0x3900

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	NRM	—	—	—	—	—	—	—	—	—	—	D1ON	D0ON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ODE1	DITH1[2:0]			ODE0	DITH0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	DITH0[2:0]	Dithering of CH0 Input to BRS These bits specify the number of colors for pixels after dithering (color reduction) when dithering (color reduction) for pixel information is enabled through the D0ON bit. When dithering (color reduction) for pixel information is disabled, set these bits to 0. 0 0 0: Dithering of BRSin0 input image is disabled. 0 0 1: Dithering of BRSin0 input image at 18 bpp (RGB666: 260000 colors) 0 1 0: Dithering of BRSin0 input image at 16 bpp (RGB565: 65535 colors) 0 1 1: Dithering of BRSin0 input image at 15 bpp (RGB555: 32768 colors) 1 0 0: Dithering of BRSin0 input image at 12 bpp (RGB444: 4096 colors) 1 0 1: Dithering of BRSin0 input image at 8 bpp (RGB332: 256 colors) Others: Setting prohibited.	R/W

Bit	Symbol	Function	R/W
3	ODE0	Ordered Dither (mode A) of CH0 Input to BRS Enable/Disable Ordered dither is available only for 18 bpp. So, when ODE0 bit is equal to 1, set DITH0 = 1. When ODE0 bit is equal to 0, BRSin0 dither method is specified by D0ON in the register. Ordered dither (mode A) is recommended rather than ordered dither (mode B) in case of 18 bpp. 0: Ordered dither (mode A) of BRSin0 is disabled. 1: Ordered dither (mode A) of BRSin0 is enabled.	R/W
6:4	DITH1[2:0]	Dithering of CH1 Input to BRS These bits specify how to perform dithering of the CH1 input to the BRS. The setting method is the same as that for the DITH0 bits. Read the description of the DITH0 bits with BRSin1 and D1ON replacing BRSin0 and D0ON, respectively.	R/W
7	ODE1	Ordered Dither (mode A) of CH1 Input to BRS Enable/Disable Ordered dither is available only for 18 bpp. So, when ODE1 bit is equal to 1, set DITH1 = 1. When ODE1 bit is equal to 0, BRSin1 dither method is specified by D1ON in the register. Ordered dither (mode A) is recommended rather than ordered dither (mode B) in case of 18 bpp. 0: Ordered dither (mode A) of BRSin1 is disabled. 1: Ordered dither (mode A) of BRSin1 is enabled.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	D0ON	Ordered dither (mode B) Enable of BRS Input 0 Enables or disables dithering (color reduction) of BRS input 0 (BRSin0 in Figure 54.28). When ODE0 in this register is set to 1, set this bit to 0. 0: Dithering (mode B) of BRSin0 is disabled. 1: Dithering (mode B) of BRSin0 is enabled.	R/W
17	D1ON	Ordered dither (mode B) Enable of BRS Input 1 Enables or disables dithering (color reduction) of BRS input 1 (BRSin1 in Figure 54.28). When ODE1 in this register is set to 1, set this bit to 0. 0: Dithering (mode B) of BRSin1 is disabled. 1: Dithering (mode B) of BRSin1 is enabled.	R/W
27:18	—	These bits are read as 0. The write value should be 0.	R/W
28	NRM	Color Data Normalization Enables or disables division by the α value of the color data in BRS blending operation. This is used when converting the RGB color data format to which the α value is multiplied (pre-multiplied color) into the RGB color data format to which the α value is not multiplied (non pre-multiplied color). Do not use this for the YCbCr format. 0: Divider (DIV unit in Figure 54.28) does not divide the color value by α. 1: Divider (DIV unit in Figure 54.28) divides the color value by α.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Figure 54.28 shows the configuration of the BRS. For the BRS inputs, there are two inputs from the DPR and one internal input as a virtual RPF. BRSin0 to BRSin1 are input ports that have the target node values shown in Figure 54.22, and they can be connected to any module on the DPR. The same color space (YCbCr or RGB) has to be used for the two inputs from the DPR to the BRS.

The virtual RPF inside the BRS is an input unit not connected to the DPR. It is called the "virtual RPF" because it outputs images internally created by the BRS. Starting of the virtual RPF is controlled by VI6_WPFn_SRCRPF.VIR_ACT2, and the single-color data created at the virtual RPF can be used for blending or raster operation (ROP) with data from the other input units BRSin0 to BRSin1. The color space for the single color to be set for the virtual RPF needs to match the color space of the two inputs from the DPR to the BRS. For this setting method, see section 54.4.10.4. VI6_BRS_VIRRPF_COL : Color Information Register of BRS Input Virtual RPF.

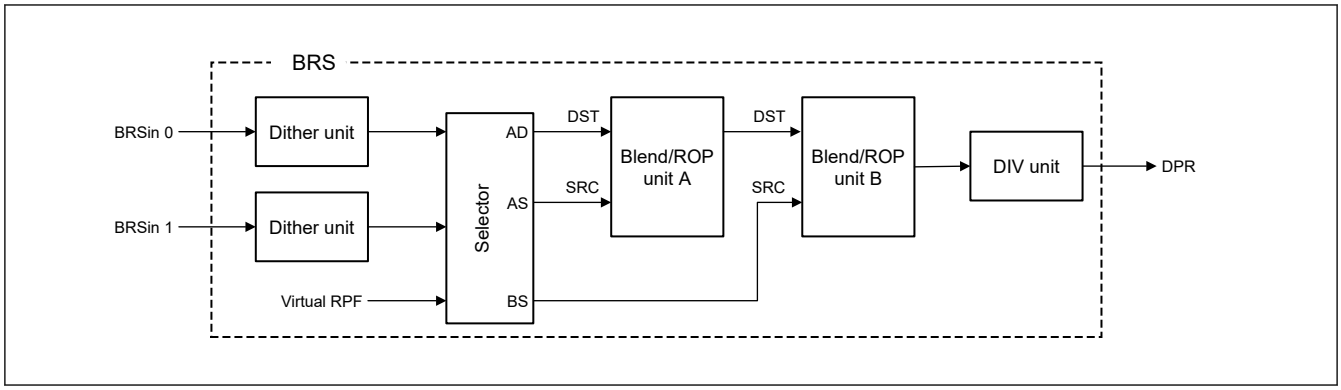


Figure 54.28 BRS configuration

The selector in Figure 54.28 is used to select the SRC and DST inputs to blending/ROP units A to B from BRSin0 to BRSin1 which are inputs from the DPR and the virtual RPF. The SRC and DST input sources for blending/ROP units A to B are either uniquely determined based on the configuration shown in Figure 54.28 or selected as desired by registers. The input sources that can be arbitrarily selected by registers are AD, AS, and BS, which correspond to the registers shown in Table 54.31.

Table 54.31 Correspondence between selector output destinations and register bits for BRS

Selector output	Output destination	Register bits
AD	Blending/ROP unit A - DST	VI6_BRSA_CTRL.DSTSEL
AS	Blending/ROP unit A - SRC	VI6_BRSA_CTRL.SRCSEL

54.4.10.2 VI6_BRS_VIRRPF_SIZE : Size Register of BRS Input Virtual RPF

Base address: VSPD = 0x920E_0000

Offset address: 0x3904

Bit position:	31			28												16			12												0
Bit field:	—	—	—	VIR_HSIZE[12:0]												—	—	—	VIR_VSIZE[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	VIR_VSIZE[12:0]	Virtual RPF Vertical Size*1 These bits set the vertical size of an image from the virtual RPF shown in Figure 54.28.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	VIR_HSIZE[12:0]	Virtual RPF Horizontal Size*1 These bits set the horizontal size of an image from the virtual RPF shown in Figure 54.28.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. A value from 1 to 8190 can be specified.

The virtual RPF has only a function to output a fixed α value and a fixed pixel value. The virtual RPF can internally create a single-color image without accessing external memory via the MAU. Same as images from the other BRS input ports, a sublayer can be blended on an image created in this manner with the image used as the background (master layer). In turn, when using the image as a sublayer, it can be drawn on the master layer as a window.

54.4.10.3 VI6_BRS_VIRRPF_LOC : Display Location Register of BRS Input Virtual RPF

Base address: VSPD = 0x920E_0000

Offset address: 0x3908

Bit position:	31			28												16			12												0
Bit field:	—	—	—	HCOORD[12:0]												—	—	—	VCOORD[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	VCOORD[12:0]	Vertical Coordinate of Virtual RPF Location on Master Layer*1 These bits specify the vertical coordinate of where to locate the top-edge pixel of the virtual RPF's layer, with the top-edge pixel of the master layer set at coordinate 0.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	HCOORD[12:0]	Horizontal Coordinate of Virtual RPF Location on Master Layer*1 These bits specify the horizontal coordinate of where to locate the left-edge pixel of the virtual RPF's layer, with the left-edge pixel of the master layer set at coordinate 0.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This setting should be made in pixel units. A value from 0 to 8189 can be specified. When the virtual RPF is specified as the master layer by VI6_WPF0_SRCRPF.VIR_ACT2, set these bits to 0.

54.4.10.4 VI6_BRS_VIRRPF_COL : Color Information Register of BRS Input Virtual RPF

Base address: VSPD = 0x920E_0000

Offset address: 0x390C

Bit position:	31							24	23							16	15							8	7							0
Bit field:	COL_A[7:0]								COL_RCR[7:0]								COL_GY[7:0]								COL_BCB[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	COL_BCB[7:0]	Fixed B/Cb of Virtual RPF*1 These bits set the fixed B/Cb value when the virtual RPF creates a virtual plane.	R/W
15:8	COL_GY[7:0]	Fixed G/Y of Virtual RPF*1 These bits set the fixed G/Y value when the virtual RPF creates a virtual plane.	R/W
23:16	COL_RCR[7:0]	Fixed R/Cr of Virtual RPF*1 These bits set the fixed R/Cr value when the virtual RPF creates a virtual plane.	R/W
31:24	COL_A[7:0]	Fixed α of Virtual RPF*1 These bits set the fixed α value when the virtual RPF creates a virtual plane.	R/W

Note 1. A value from 0 to 255 can be specified.

The transparency information and color information of the single color that is created by the virtual RPF are set in the bits of this register. As described earlier, the color information is set for the YCbCr or RGB color space. The color space to be set in this register depends on the register settings of the environment and other modules to which the BRS is connected by the DPR. Two cases can be considered. Since the α value (COL_A) is transparency information and irrelevant to the concept of color space, the same setting is made for either the YCbCr or RGB color space.

[Case 1: When an input other than the virtual RPF is used]

When the source RPF is connected to any one of the BRS input ports (BRSin0 to BRSin1) other than the virtual RPF and valid data is being supplied, the same color space data as the color space for the BRS inputs should be set in this register as the color space for the virtual RPF's color information. This is based on the restriction of "all BRS inputs must have the same color space", as described in section 54.4.8.1. Concept of DPR Settings or section 54.4.10.1. VI6_BRS_INCTRL : BRS Input Control Register.

[Case 2: When only the virtual RPF is used]

When only the virtual RPF is used as the source RPF of WPFn, RPFn is not connected to the BRS, as shown in Figure 54.29. Thus, there is no color space for another input that the color space for the virtual RPF has to follow, as in case 1. This means that the color space of the data output by the BRS is determined by the WPF setting.

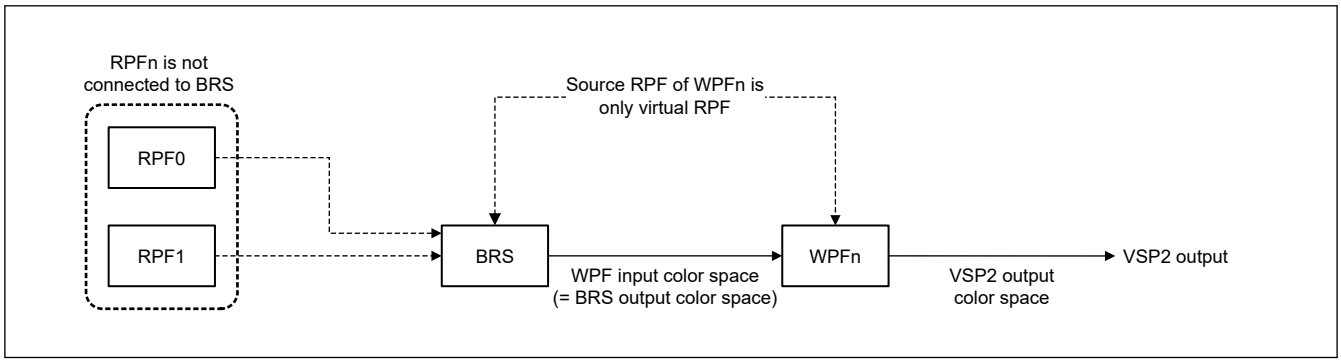


Figure 54.29 Relationship between DPR connection and color space when only virtual RPF is used

Figure 54.29, the output color space of the VSPD (= output color space of WPFn) is determined by VI6_WPFn_OUTFMT.WRFMT. When bit 6 in VI6_WPFn_OUTFMT.WRFMT (WRFMT[6]) is 0, the color space is RGB, while when it is 1, the color space is YCbCr. Next, the WPF input color space (= BRS output color space) is determined by the relationship between the WPF output color space and VI6_WPFn_OUTFMT.CSC. When VI6_WPFn_OUTFMT.CSC is 0, the WPF output color space and WPF input color space (= BRS output color space) are the same. When VI6_WPFn_OUTFMT.CSC is 1, the WPF output color space and WPF input color space (= BRS output color space) are the opposite. This relationship is summarized in Table 54.32.

The color space for the virtual RPF's color information should be set in this register according to the "BRS output color space" shown in Table 54.32.

Table 54.32 Relationship between WPF output color space and BRS output color space

VI6_WPFn_OUTFMT register bit settings				BRS output color space (= WPF input color space)
Bit 6 in WRFMT		CSC		
0	(WPF output is RGB)	0	(YCbCr to RGB conversion is disabled)	RGB
0	(WPF output is RGB)	1	(YCbCr to RGB conversion is enabled)	YCbCr
1	(WPF output is YCbCr)	0	(RGB to YCbCr conversion is disabled)	YCbCr
1	(WPF output is YCbCr)	1	(RGB to YCbCr conversion is enabled)	RGB

54.4.10.5 VI6_BRSA_CTRL : BRS Control Register A

Base address: VSPD = 0x920E_0000

Offset address: 0x3910

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RBC	—	—	—	—	—	—	—	—	DSTSEL[2:0]			—	SRCSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CROP[3:0]				AROP[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	AROP[3:0]	α Data ROP Operator These bits select the ROP operator of the α data in blending/ROP unit A. Select the opcode for ROP operation from Table 54.33.	R/W

Bit	Symbol	Function	R/W
7:4	CROP[3:0]	Color Data ROP Operator These bits select the ROP operator of the color data in blending/ROP unit A. Select the opcode for ROP operation from Table 54.33.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
18:16	SRCSEL[2:0]	Input Selection for SRC Side of Blending/ROP Unit A These bits select the input for the SRC side of blending/ROP unit A shown in Figure 54.28. These bits specify the connection between the BRS input port and the SRC separately from the setting of connections between other modules and the BRS input port through the DPR. 0 0 0: BRS input 0 (BRSin0) is input to SRC. 0 0 1: BRS input 1 (BRSin1) is input to SRC. 1 0 0: Virtual RPF is input to SRC. Others: Setting prohibited.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	DSTSEL[2:0]	Input Selection for DST Side of Blending/ROP Unit A These bits select the input for the DST side of blending/ROP unit A shown in Figure 54.28. These bits specify the connection between the BRS input port and the DST separately from the setting of connections between other modules and the BRS input port through the DPR. 0 0 0: BRS input 0 (BRSin0) is input to DST. 0 0 1: BRS input 1 (BRSin1) is input to DST. 1 0 0: Virtual RPF is input to DST. Others: Setting prohibited.	R/W
30:23	—	These bits are read as 0. The write value should be 0.	R/W
31	RBC	Operation Type of Blending/ROP Unit A Specifies the operation type for blending/ROP unit A shown in Figure 54.28. 0: ROP (raster operation) 1: Blending operation	R/W

Table 54.33 ROP operator of blending/ROP unit m (m = A, B)

Opcode	Operator
0000b	NOP(D)
0001b	AND(S & D)
0010b	AND_REVERSE(S & ~D)
0011b	COPY(S)
0100b	AND_INVERTED(~S & D)
0101b	CLEAR(0)
0110b	XOR(S ^ D)
0111b	OR(S | D)
1000b	NOR(~(S | D))
1001b	EQUIV(~(S ^ D))
1010b	INVERT(~D)
1011b	OR_REVERSE(S | ~D)
1100b	COPY_INVERTED(~S)
1101b	OR_INVERTED(~S | D)
1110b	NAND(~(S & D))
1111b	SET(all 1)

54.4.10.6 VI6_BRSA_BLD : BRS Blend Control Register A

Base address: VSPD = 0x920E_0000

Offset address: 0x3914

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CBES	CCMDX[2:0]			—	CCMDY[2:0]			ABES	ACMDX[2:0]			—	ACMDY[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	COEFX[7:0]								COEFY[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	COEFY[7:0]	Fixed α Value 1*1 These bits specify fixed α value 1 used when the CCMDY or ACMDY bits are set to 100b.	R/W
15:8	COEFX[7:0]	Fixed α Value 0*1 These bits specify fixed α value 0 used when the CCMDX or ACMDX bits are set to 100b.	R/W
18:16	ACMDY[2:0]	α Creation Coefficient Y These bits specify α creation coefficient Y used in the α creation expression determined by the ABES bit. 0 0 0: (α creation coefficient Y) = (DST α data) 0 0 1: (α creation coefficient Y) = 255 - (DST α data) 0 1 0: (α creation coefficient Y) = (SRC α data) 0 1 1: (α creation coefficient Y) = 255 - (SRC α data) 1 0 0: (α creation coefficient Y) = Fixed α value 1 (COEFY setting) Others: Setting prohibited.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	ACMDX[2:0]	α Creation Coefficient X These bits specify α creation coefficient X used in the α creation expression determined by the ABES bit. 0 0 0: (α creation coefficient X) = (DST α data) 0 0 1: (α creation coefficient X) = 255 - (DST α data) 0 1 0: (α creation coefficient X) = (SRC α data) 0 1 1: (α creation coefficient X) = 255 - (SRC α data) 1 0 0: (α creation coefficient X) = Fixed α value 0 (COEFX setting) Others: Setting prohibited.	R/W
23	ABES	Blending α Creation Expression Specifies the expression for creating α data after blending by blending/ROP unit A. α creation coefficients are specified by the ACMDX and ACMDY bits. 0: ACMDX * (DST α data) + ACMDY * (SRC α data) 1: ACMDX * (DST α data) - ACMDY * (SRC α data)	R/W
26:24	CCMDY[2:0]	Blending Coefficient Y Selection These bits specify coefficient Y used in the blending expression determined by the CBES bit. 0 0 0: DST α data is used as blending coefficient Y 0 0 1: 255 - (DST α data) is used as blending coefficient Y 0 1 0: SRC α data is used as blending coefficient Y 0 1 1: 255 - (SRC α data) is used as blending coefficient Y 1 0 0: Fixed α value 1 (COEFY setting)	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
30:28	CCMDX[2:0]	Blending Coefficient X Selection These bits specify coefficient X used in the blending expression determined by the CBES bit. 0 0 0: DST α data is used as blending coefficient X 0 0 1: 255 - (DST α data) is used as blending coefficient X 0 1 0: SRC α data is used as blending coefficient X 0 1 1: 255 - (SRC α data) is used as blending coefficient X 1 0 0: Fixed α value 0 (COEFX setting)	R/W
31	CBES	Blending Expression Selection Selects the blending expression of the color data in the BRS (VI6_BRSA_CTRL.RBC = 1). Blending coefficients are specified by the CCMDX and CCMDY bits. 0: CCMDX * (DST color data) + CCMDY * (SRC color data) 1: CCMDX * (DST color data) - CCMDY * (SRC color data)	R/W

Note 1. A value from 0x00 to 0xFF can be specified.

54.4.10.7 VI6_BRSB_CTRL : BRS Control Register B

Base address: VSPD = 0x920E_0000

Offset address: 0x3918

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RBC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CROP[3:0]				AROP[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	AROP[3:0]	α Data ROP Operator These bits select the ROP operator of the α data in blending/ROP unit B. Select the opcode for ROP operation from Table 54.33.	R/W
7:4	CROP[3:0]	Color Data ROP Operator These bits select the ROP operator of the color data in blending/ROP unit B. Select the opcode for ROP operation from Table 54.33.	R/W
30:8	—	These bits are read as 0. The write value should be 0.	R/W
31	RBC	Operation Type of Blending/ROP Unit B Specifies the operation type for blending/ROP unit B shown in Figure 54.28. 0: ROP (raster operation) 1: Blending operation	R/W

54.4.10.8 VI6_BRSB_BLD : BRS Blend Control Register B

Base address: VSPD = 0x920E_0000

Offset address: 0x391C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CBES	CCMDX[2:0]			—	CCMDY[2:0]			ABES	ACMDX[2:0]			—	ACMDY[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	COEFX[7:0]								COEFY[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	COEFY[7:0]	Fixed α Value 1[*1] These bits specify fixed α value 1 used when the CCMDY or ACMDY bits are set to 100b.	R/W
15:8	COEFX[7:0]	Fixed α Value 0[*1] These bits specify fixed α value 0 used when the CCMDX or ACMDX bits are set to 100b.	R/W
18:16	ACMDY[2:0]	α Creation Coefficient Y These bits specify α creation coefficient Y used in the α creation expression determined by the ABES bit. 0 0 0: (α creation coefficient Y) = (DST α data) 0 0 1: (α creation coefficient Y) = 255 - (DST α data) 0 1 0: (α creation coefficient Y) = (SRC α data) 0 1 1: (α creation coefficient Y) = 255 - (SRC α data) 1 0 0: (α creation coefficient Y) = Fixed α value 1 (COEFY setting) Others: Setting prohibited.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	ACMDX[2:0]	α Creation Coefficient X These bits specify α creation coefficient X used in the α creation expression determined by the ABES bit. 0 0 0: (α creation coefficient X) = (DST α data) 0 0 1: (α creation coefficient X) = 255 - (DST α data) 0 1 0: (α creation coefficient X) = (SRC α data) 0 1 1: (α creation coefficient X) = 255 - (SRC α data) 1 0 0: (α creation coefficient X) = Fixed α value 0 (COEFX setting) Others: Setting prohibited.	R/W
23	ABES	Blending α Creation Expression Specifies the expression for creating α data after blending by blending/ROP unit B. α creation coefficients are specified by the ACMDX and ACMDY bits. 0: ACMDX * (DST α data) + ACMDY * (SRC α data) 1: ACMDX * (DST α data) - ACMDY * (SRC α data)	R/W
26:24	CCMDY[2:0]	Blending Coefficient Y Selection These bits specify coefficient Y used in the blending expression determined by the CBES bit. 0 0 0: DST α data is used as blending coefficient Y. 0 0 1: 255 - (DST α data) is used as blending coefficient Y. 0 1 0: SRC α data is used as blending coefficient Y. 0 1 1: 255 - (SRC α data) is used as blending coefficient Y. 1 0 0: Fixed α value 1 (COEFY setting).	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	CCMDX[2:0]	Blending Coefficient X Selection These bits specify coefficient X used in the blending expression determined by the CBES bit. 0 0 0: DST α data is used as blending coefficient X. 0 0 1: 255 - (DST α data) is used as blending coefficient X. 0 1 0: SRC α data is used as blending coefficient X. 0 1 1: 255 - (SRC α data) is used as blending coefficient X. 1 0 0: Fixed α value 0 (COEFX setting).	R/W
31	CBES	Blending Expression Selection Selects the blending expression of the color data in the BRS (VI6_BRSB_CTRL.RBC = 1). Blending coefficients are specified by the CCMDX and CCMDY bits. 0: CCMDX * (DST color data) + CCMDY * (SRC color data) 1: CCMDX * (DST color data) - CCMDY * (SRC color data)	R/W

Note 1. A value from 0x00 to 0xFF can be specified.

54.4.11 LIF Control Registers

54.4.11.1 VI6_LIF0_CTRL : LIF0 Control Register

Base address: VSPD = 0x920E_0000

Offset address: 0x3B00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	OBTH[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PADL	—	—	—	CFMT	—	—	REQSEL	LIF_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LIF_EN	Enable/Disable of Data Output to External Display Module Enables or disables data output from the LIF to the external display module (DU). 0: Data output to the external display module is disabled. 1: Data output to the external display module is enabled.	R/W
1	REQSEL	External Display Module Selection 0: This value is setting prohibited when 1 is set to LIF_EN. 1: DU is selected as the destination external display module.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	CFMT	Chroma Format*1 This bit selects the output format from the LIF module to the display module. When RGB format is used, this bit shall be set to 0. 0: YCbCr444 or RGB Format 1: YCbCr422 Format	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	PADL	Enable/Disable of padding with dummy lines to output-image to the DU 0: Padding with dummy lines to output image to DU is disabled. 1: Padding with dummy lines to output image to DU is enabled.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
27:16	OBTH[11:0]	Buffer Threshold for Start Ready Notification to Display Module Always set these bits to 1500 when LIF_EN is set to 1.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The DU cannot receive YCbCr422 format. Therefore, when CFMT is 1, YCbCr444 data which has information contents equal to YCbCr422 is output to DU as shown in Figure 54.30.

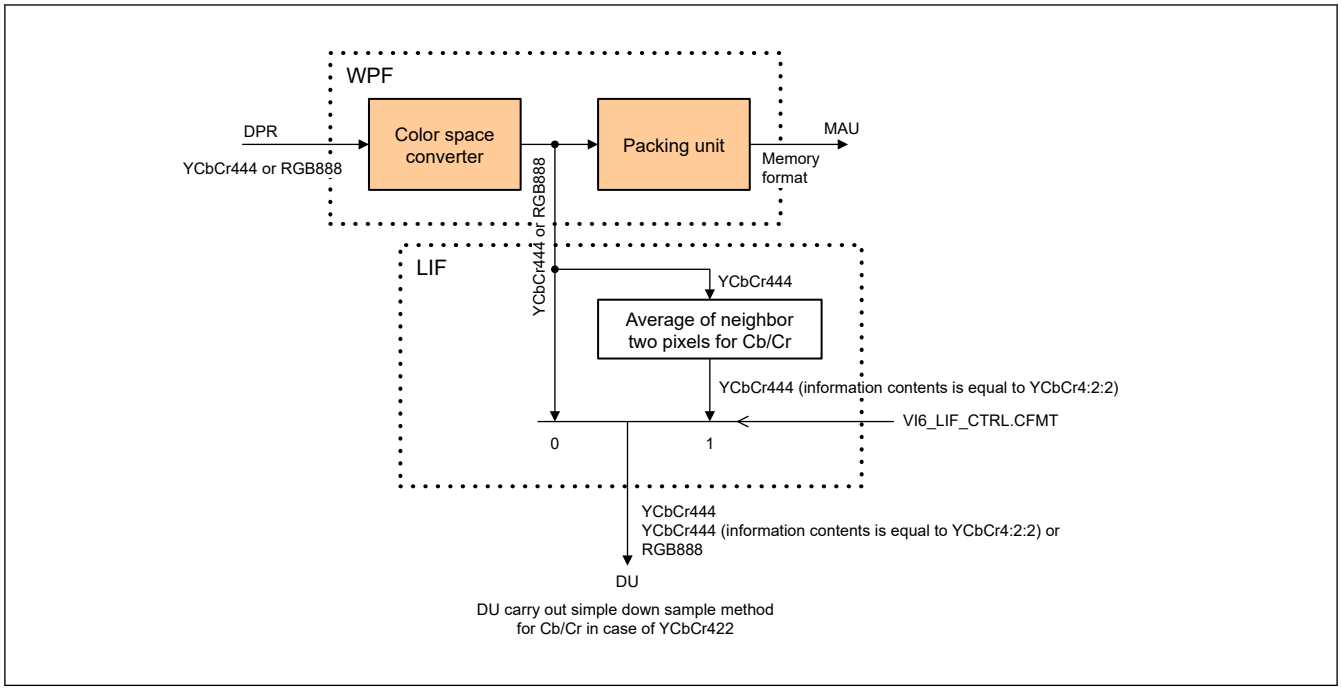


Figure 54.30 Data flow between LIF and DU

54.4.11.2 VI6_LIF0_CSBTH : LIF0 Clock Stop Buffer Control Register

Base address: VSPD = 0x920E_0000

Offset address: 0x3B04

Bit position:	31				27											16				11											0
Bit field:	—	—	—	—	HBTH[11:0]											—	—	—	—	LBTH[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	LBTH[11:0]	Buffer Threshold for Clock Start in Dynamic Clock Control Set LBTH[11:0] = 0x000 (fixed value)	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
27:16	HBTH[11:0]	Buffer Threshold for Clock Stop in Dynamic Clock Control Set HBTH[11:0] = 0x000 (fixed value)	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

54.4.11.3 VI6_LIF0_LBA : LIF0 Buffer Attribute Register

Base address: VSPD = 0x920E_0000

Offset address: 0x3B0C

Bit position:	31				27											16				11											0
Bit field:	LBA0	—	—	—	LBA1[11:0]											—	—	—	—	LBA2[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	LBA2[11:0]	LIF Buffer Attribute Register 2 These bits are internal status for purpose of H/W debugging.	R/W

Bit	Symbol	Function	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
27:16	LBA1[11:0]	LIF Buffer Attribute Register 1 Always set these bits to 1536 when LIF_EN is set to 1.	R/W
30:28	—	These bits are read as 0. The write value should be 0.	R/W
31	LBA0	LIF Buffer Attribute Register 0 Always set this bit to 1 when LIF_EN is set to 1.	R/W

54.4.11.4 VI6_LIF0_PADLN_CYC : LIF0 Padding Line Cycle Register

Base address: VSPD = 0x920E_0000

Offset address: 0x3B30

Bit position:	31																								5						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PADLN_CYC[5:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	PADLN_CYC[5:0]	Cycle of padding pattern*1 These bits specify the minimum number of cycles of the padding pattern of valid lines and dummy lines. For example, if PADLN_CYC[5:0] is 4, lower 4 bits of PADLN_PATTERN[3:0] indicates padding pattern.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. A value from 1 to 32 can be specified.

54.4.11.5 VI6_LIF0_PADLN_PT : LIF0 Padding Line Pattern Register

Base address: VSPD = 0x920E_0000

Offset address: 0x3B34

Bit position:	31																														0
Bit field:	PADLN_PATTERN[31:0]																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PADLN_PATTERN[31:0]	Pattern of padding with dummy lines These bits specify the padding pattern of valid lines and dummy lines. In each bit, 0b indicates an active line, 1b indicates a dummy line. It starts from the lower bit (Bit 0). First line should be a valid line (Always set PADLN_PATTERN[0] to 0).	R/W

54.4.11.6 VI6_LIF0_PADLN_VAL : LIF0 Padding Line Value Register

Base address: VSPD = 0x920E_0000

Offset address: 0x3B38

Bit position:	31								24	23							16	15							8	7						0
Bit field:	PADA[7:0]									PADR[7:0]								PADG[7:0]								PADB[7:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PADB[7:0]	Padding data of B/Cb for dummy line	R/W

Bit	Symbol	Function	R/W
15:8	PADG[7:0]	Padding data of G/Y for dummy line	R/W
23:16	PADR[7:0]	Padding data of R/Cr for dummy line	R/W
31:24	PADA[7:0]	Padding data of EDC code for dummy line	R/W

54.4.11.7 VI6_LIF0_PADLN_SIZE : LIF0 Padding Line Size Register

Base address: VSPD = 0x920E_0000

Offset address: 0x3B3C

Bit position:	31																		12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PADLN_VSIZE[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	PADLN_VSIZE[12:0]	Vertical Size of LIF output in case of enabling Padding line When VI6_LIF0_CTRL.PADL is 1, dummy lines are included in the output data from LIF, so set the LIF output VSIZE including dummy lines to these bits. If the number of output lines does not reach the value of PADLN_VSIZE[12:0] even if all valid lines are output according to padding pattern, dummy lines are inserted until the number of output lines reaches PADLN_VSIZE[12:0]. ● Output v size of LIF is same with input v size to LIF when VI6_LIF0_CTRL.PADL is 0. ● Output v size of LIF is PADLN_VSIZE[12:0] when VI6_LIF0_CTRL.PADL is 1.	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

54.5 DU Register Descriptions

54.5.1 DU_MCR0 : DU Module Control Register 0

Base address: DU = 0x920C_0000

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PB_CLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DI_EN	—	—	—	—	—	—	—	DPI_OE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPI_OE	Display parallel interface output enable*1 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	DI_EN	Display enable 0: Disable 1: Enable	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	PB_CLR	Clear PBUF pointers*1 Writing 1 to this parameter starts clearing the PBUF pointers. After clearing the PBUF pointers is completed, DU_MSR0.ST_PB_WINIT and DU_MSR0.ST_PB_RINIT are read as 1, this parameter must be written back to 0.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This parameter must be changed while VSPD is not running and DU_MSR0.ST_DIF_BSY is 0.

54.5.2 DU_MSR0 : DU Module Status Register 0

Base address: DU = 0x920C_0000

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	ST_PB_RINIT	ST_PB_RUF	ST_PB_REMPTY	—	ST_PB_WINIT	—	ST_PB_WFULL
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ST_DI_BSY	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0.	R
8	ST_DI_BSY	Display I/F status 0: Idle 1: Busy	R
15:9	—	These bits are read as 0.	R
16	ST_PB_WFULL	PBUF write full status 0: Not full 1: Full	R
17	—	This bit is read as 0.	R
18	ST_PB_WINIT	PBUF FIFO write pointer cleared status 0: Not cleared 1: Cleared (initial value)	R
19	—	This bit is read as 0.	R
20	ST_PB_REMPTY	PBUF read empty status 0: Not empty 1: Empty	R
21	ST_PB_RUF	PBUF read underflow status*1 0: Not occur PBUF underflow 1: Occur PBUF underflow	R
22	ST_PB_RINIT	PBUF FIFO read pointer cleared status 0: Not cleared 1: Cleared (initial value)	R
31:23	—	These bits are read as 0.	R

Note 1. This parameter clears when PBUF pointers are cleared.

54.5.3 DU_MSR1 : DU Module Status Register 1

Base address: DU = 0x920C_0000

Offset address: 0x0008

Bit position:	31			28												16			12												0
Bit field:	—	—	—	UF_VACT[12:0]													—	—	—	UF_HACT[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	UF_HACT[12:0]	Hactive counter when the PBUF underflow occurs*1 This parameter is value latched the down counter starting from HACTIVE.	R
15:13	—	These bits are read as 0.	R
28:16	UF_VACT[12:0]	Vactive counter when the PBUF underflow occurs*1 This parameter is value latched the down counter starting from VACTIVE.	R
31:29	—	These bits are read as 0.	R

Note 1. This parameter is cleared when DU_MSR0.ST_PB_RUF is cleared.

54.5.4 DU_IMR0 : DU Interrupt Mask Register 0

Base address: DU = 0x920C_0000

Offset address: 0x000C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IM_PB_RUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IM_PB_RUF	Mask PBUF read underflow interrupt 0: Not mask interrupt 1: Mask interrupt	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

54.5.5 DU_DITR0 : DU Display I/F Timing Register 0

Base address: DU = 0x920C_0000

Offset address: 0x0010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSPOL	VSPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DEMD[1:0]		—	—	—	—	—	—	—	DPI_CLKMD
Value after reset:	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPI_CLKMD	Display parallel interface clock mode 0: Through output 1: Inversion output	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DEMD[1:0]	de output mode 0 0: Fixed to Low 0 1: Reserved (Fixed to High) 1 0: Reserved (Low active) 1 1: Data enable (High active)	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	VSPOL	vsync polarity 0: Low active 1: High active	R/W
17	HSPOL	hsync polarity 0: Low active 1: High active	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

54.5.6 DU_DITR1 : DU Display I/F Timing Register 1

Base address: DU = 0x920C_0000

Offset address: 0x0014

Bit position:	31			28												16				11											0
Bit field:	—	—	—	VACTIVE[12:0]												—	—	—	—	VSA[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	VSA[11:0]	The number of lines in the Vsync period This parameter should be set from 0 to 4095, and DU_DITR1.VSA + DU_DITR2.VBP should be set more than 0. If set 0, vsync is not asserted to DU_DITR0.VSPOL.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
28:16	VACTIVE[12:0]	The number of lines in the Vactive period This parameter should be set from 1 to 8190.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Figure 54.31 shows the relationship between the parameters and the video output timings.

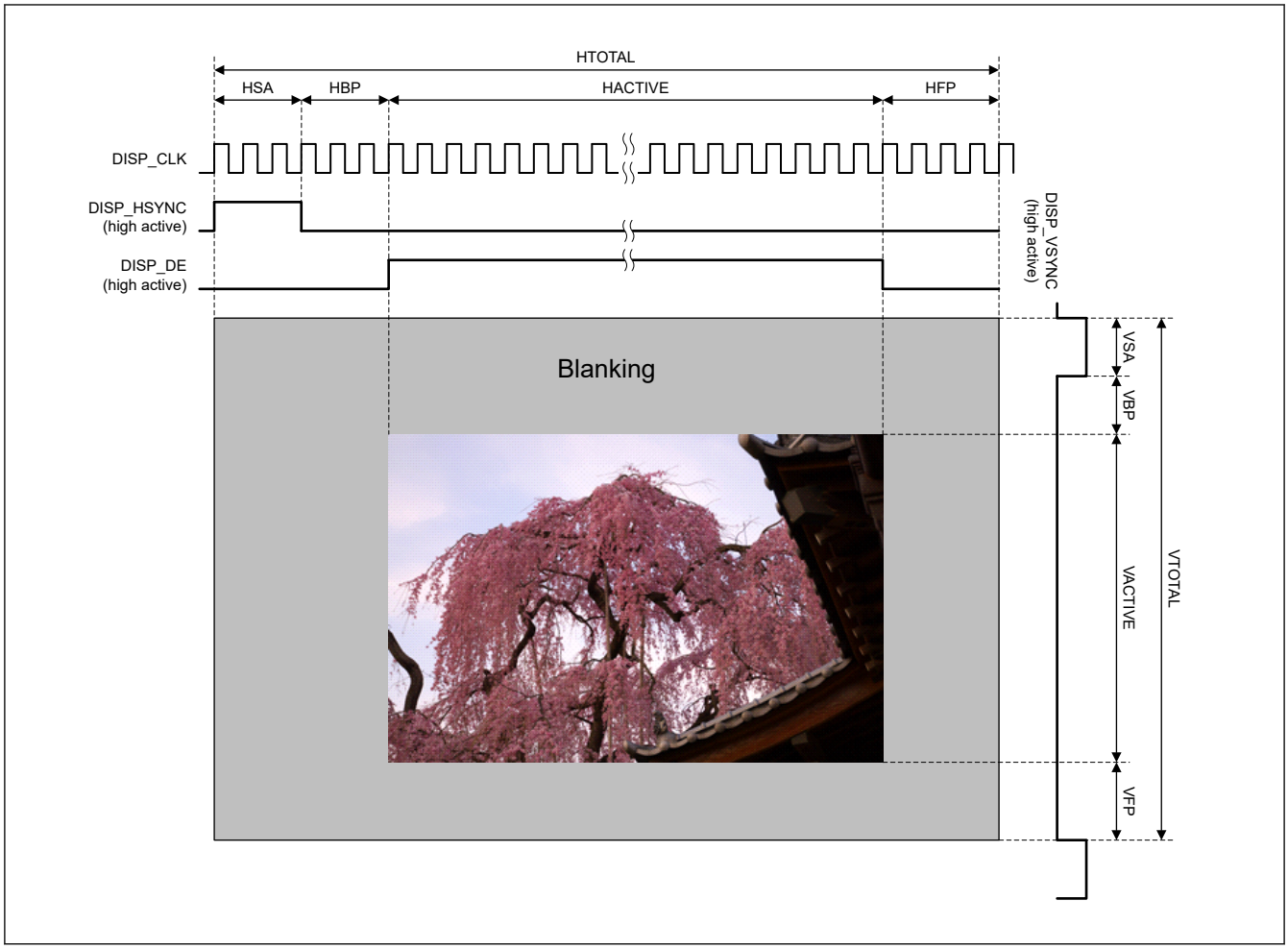


Figure 54.31 Video output timings

54.5.7 DU_DITR2 : DU Display I/F Timing Register 2

Base address: DU = 0x920C_0000

Offset address: 0x0018

Bit position:	31			28												16			12												0
Bit field:	—	—	—	VFP[12:0]												—	—	—	VBP[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	VBP[12:0]	The number of lines in the Vback period This parameter should be set from 0 to 8191, and DU_DITR1.VSA + DU_DITR2.VBP should be set more than 0.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	VFP[12:0]	The number of lines in the Vfront period This parameter should be set from 0 to 8191.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

54.5.8 DU_DITR3 : DU Display I/F Timing Register 3

Base address: DU = 0x920C_0000

Offset address: 0x001C

Bit position:	31			28												16										11										0
Bit field:	—	—	—	HACTIVE[12:0]												—	—	—	—	HSA[11:0]																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	HSA[11:0]	The number of cycles in the Hsync period This parameter should be set from 0 to 4095, and DU_DITR3.HSA + DU_DITR4.HBP + DU_DITR4.HFP + DU_DITR5.HSFT (= Hblank) should be set 3 or more. If set 0, hsync is not asserted to DU_DITR0.HSPOL.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
28:16	HACTIVE[12:0]	The number of cycles (pixels) in the Hactive period This parameter should be set from 1 to 8190.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

54.5.9 DU_DITR4 : DU Display I/F Timing Register 4

Base address: DU = 0x920C_0000

Offset address: 0x0020

Bit position:	31			28												16										12										0
Bit field:	—	—	—	HFP[12:0]												—	—	—	HBP[12:0]																	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	HBP[12:0]	The number of cycles in the Hback period This parameter should be set from 0 to 8191, and DU_DITR3.HSA + DU_DITR4.HBP + DU_DITR4.HFP + DU_DITR5.HSFT (= Hblank) should be set 3 or more.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	HFP[12:0]	The number of cycles in the Hfront period This parameter should be set from 1 to 8191, and DU_DITR3.HSA + DU_DITR4.HBP + DU_DITR4.HFP + DU_DITR5.HSFT (= Hblank) should be set 3 or more.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

54.5.10 DU_MCR1 : DU Module Control Register 1

Base address: DU = 0x920C_0000

Offset address: 0x0040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PB_AUTOCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OPMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OPMD[1:0]	Operation mode*1 Specify behavior of LIFC and PBUF during Vfront and Hfront period. 0 0: After frame end, LIFC wait, and PBUF pointers are cleared if DU_MCR1.PB_AUTOCLR is set to 1. 0 1: After line end, LIFC wait, and PBUF pointers are cleared if DU_MCR1.PB_AUTOCLR is set to 1. 1 0: Reserved 1 1: LIFC always request to VSPD. PBUF pointers are not cleared until DU_MCR0.PB_CLR is set to 1.	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	PB_AUTOCLR	PBUF pointers auto clear enable Specify timing to clear the PBUF pointers. 0: Not clear until DU_MCR0.PB_CLR is asserted. 1: Clear during blanking. (according to DU_MCR1.OPMD)	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This parameter must be changed while VSPD is not running.

54.5.11 DU_PBCR0 : DU PBUF Control Register 0

Base address: DU = 0x920C_0000

Offset address: 0x004C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	PB_DEP[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	PB_DEP[4:0]	Valid PBUF depth This parameter should be set 0x1F. This parameter should be set ($2^n - 1$).	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

54.5.12 DU_PBCR1 : DU PBUF Control Register 1

Base address: DU = 0x920C_0000

Offset address: 0x0050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PB_RUFOP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PB_RUFOP	Specify behavior when the PBUF underflow occurs 0: Continue outputting invalid data 1: Output DU_PBCR2.PB_RUFDAT	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

54.5.13 DU_PBCR2 : DU PBUF Control Register 2

Base address: DU = 0x920C_0000

Offset address: 0x0054

Bit position:	31								23																				0
Bit field:	—	—	—	—	—	—	—	—	PB_RUFDAT[23:0]																				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	PB_RUFDAT[23:0]	Read data when the PBUF underflow occurs.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

54.6 Operation

54.6.1 Operation Control Setting

(a) To start image process

FCPVD registers do not need to be set when starting an image process. Therefore, see section 54.6.3. Concept of VSPD Operation Starting and Stopping for the guidance of starting an image process.

(b) To stop image process immediately

Do following steps to stop immediately an image process:

1. Set the DU_MCR0.DI_EN bit to 1 so that DU can stop to output.
2. Set the VI6_WPF0_IRQ_ENB.FREE bit to 1 so that frame end interruption can be issued.
3. Set the VI6_SRESET.SRST0 bit to 1. VSPD will invoke termination process immediately.
4. Wait frame end interruption from VSPD or until the VI6_WPF0_IRQ_STA.FRE bit is set to 1. After it occurs, VSPD has finished its processing.
5. Wait until the DU_MSR0.ST_DI_BSY bit becomes 0. After it occurs, DU has stopped outputting.
6. Set the VI6_WPF0_IRQ_STA.FRE and VI6_DISP0_IRQ_STA.DST bits to 0. VSPD will clear interrupts.
7. Set the DU_MCR0.PB_CLR bit to 1 so that PBUF can be cleared.
8. Wait until the DU_MSR0.ST_PB_WINIT and DU_MSR0.ST_PB_RINIT bits become 1.
9. Set the DU_MCR0.PB_CLR bit to 0.

With this procedure, VSPD can stop its process quickly, but the output frame of the last frame is corrupted.

54.6.2 Registers to Set Fixed Value

Set fixed value to following registers in any case.

[1] Always set the VI6_CLK_DCSWT register to 0x00000808.

[2] Always set the VI6_DL_CTRL.AR_WAIT[7:0] bits to 256 in case of using display list.

[3] Always set the VI6_DL_EXT_CTRLn.POLINT[5:0] bits to 2 in case of using extended display list of WPFn.

[4] Always set the VI6_DL_EXT_CTRLn.DLPRI bit to 1 in case of using extended display list of WPFn.

[5] Always set the VI6_DL_EXT_CTRLn.EXPRI bit to 0 in case of using extended display list of WPFn.

[6] Always set the VI6_DPR_WPFn_FPORCH register (n = 0, 1) to 0x00000500 in case of using WPFn.

[7] Always set the VI6_LIF0_CTRL.OBTH[11:0] bits to 1500 in case of using LIF (VI6_LIF0_CTRL.LIF_EN = 1).

[8] Always set the VI6_LIF0_LBA register to 0x86000000 in case of using LIF0 (VI6_LIF0_CTRL.LIF_EN = 1).

[9] Always set the VI6_LIFn_CTRL.REQSEL bit to 1 in case of using LIFn (VI6_LIFn_CTRL.LIF_EN = 1).

[10] Always set the VI6_LIFn_CSBTH register to 0 in case of using LIFn (VI6_LIFn_CTRL.LIF_EN = 1).

[11] Always set the DU_PBCR0 register to 0x1F in case of using DU (DU_MCR0.DI_EN = 1).

54.6.3 Concept of VSPD Operation Starting and Stopping

The VSPD provides one channel of image processing. Each channel is started by setting the corresponding start register. Here, starting a processing channel means starting one of WPF0, which are output modules of the VSPD. Use the start registers shown in Table 54.34 to start WPF modules.

After a WPF module is started and specified processing is completed, the corresponding channel stops operation and notifies the end of processing through an end interrupt. End interrupts are generated through the end interrupt source registers shown in Table 54.34; clearing a source register cancels the corresponding interrupt signal. Each of the operating status registers shown in the table indicates the busy state after the corresponding channel is started through the start register until processing is completed and operation stops. Figure 54.33 shows these operation timings.

Table 54.34 Target module and corresponding registers for starting and stopping operation

Target module	Start register	End interrupt source register	Operating status register
WPF0	VI6_CMD0.STRCMD	VI6_WPF0_IRQ_STA. FRE	VI6_STATUS.SYS0_ACT

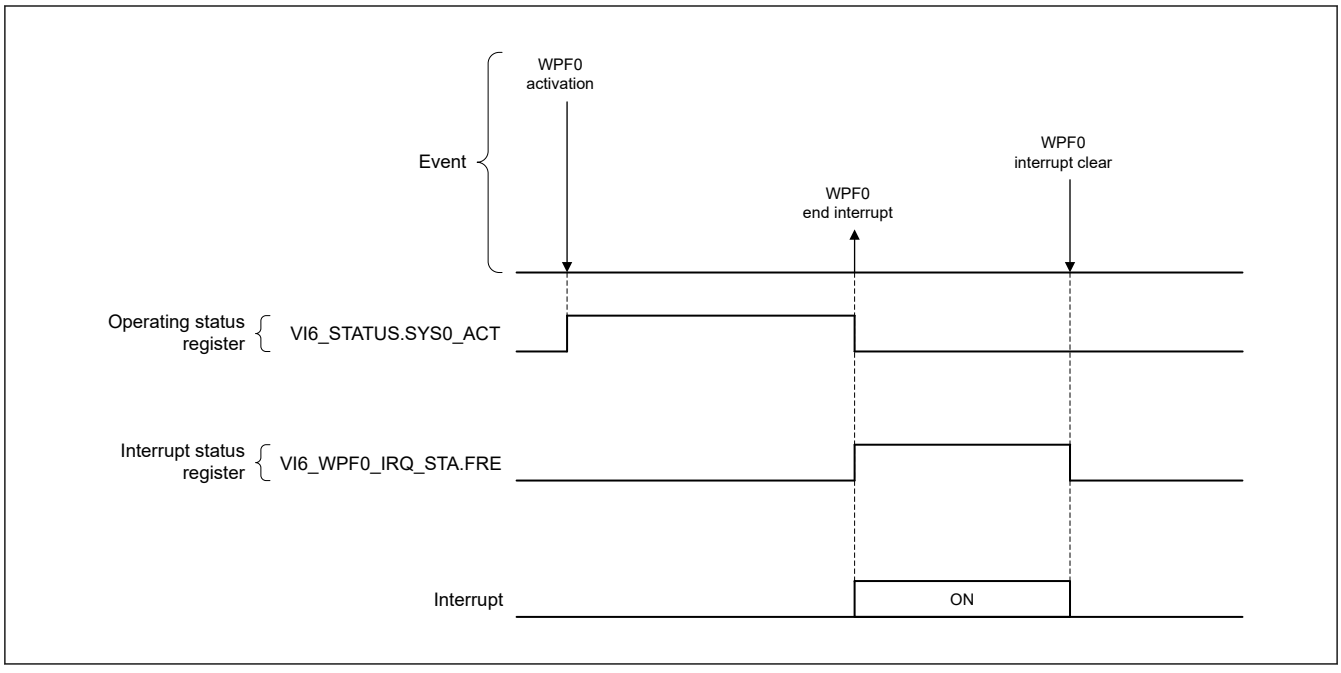


Figure 54.32 VSPD startup and status of each register and interrupt

The following describes the operating states (operating or stopped) of VSPD internal modules. As described in section 54.4. VSPD Register Descriptions, the VSPD has several image processing modules and the connections between modules are determined by the DPR. Accordingly, the operating state of a module is the same as that of the target WPF for that module. For example, when the target WPF for the BRS is WPF0, the BRS operating state is the same as that of WPF0; that is, the BRS operating state is indicated by the VI6_STATUS.SYS0_ACT as shown in Table 54.34 and the status change timing is shown as VI6_STATUS. SYS0_ACT in Figure 54.32. Likewise, the operating states of all modules connected to WPF0 is indicated by VI6_STATUS. SYS0_ACT.

Connections should be changed through the DPR-related registers (described later) while all modules to be affected by any change in connections are stopped. If connections through the DPR are changed during operation, the VSPD will hang.

Figure 54.33 shows relation between start reservation and operating status. When start is reserved while VSPD is operating, the reservation is accepted after VSPD status is moved from "operating" into "idle" as shown in Figure 54.33.

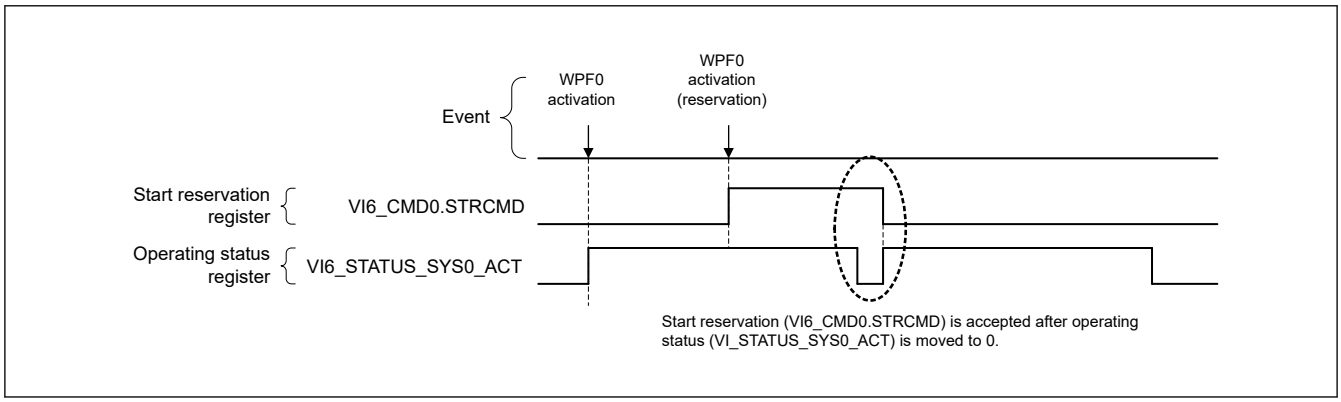


Figure 54.33 VSPD start reservation and operating status

54.6.4 Display List

54.6.4.1 Function Description

The VSPD provides the display list function. As a display list, the VSPD automatically downloads the register settings except for the control registers (section 54.4.4. General Control Registers and section 54.4.5. Display List Control Registers) from external memory and stores the settings in the VSPD registers. This function is advantageous in that the interrupt processing or register setting modification processing can be executed without CPU intervention during multiple-frame processing because the register settings used for VSPD processing are prepared in advance in external memory such as SDRAM.

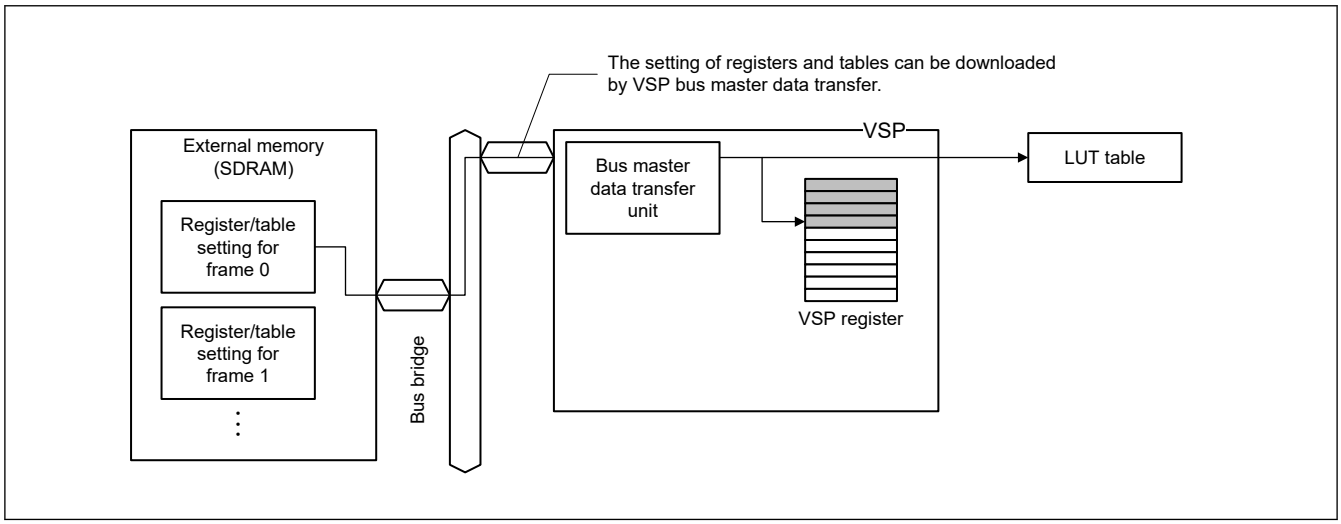


Figure 54.34 Concept of display list

To use display lists, specify the external memory addresses to the display list control registers described in section 54.4.5. Display List Control Registers. The register settings or various types of information should be stored in external memory in the format described in section 54.6.4.2. Normal Display List Mode. Figure 54.35 shows the difference between VSPD operation through normal register settings and through display lists.

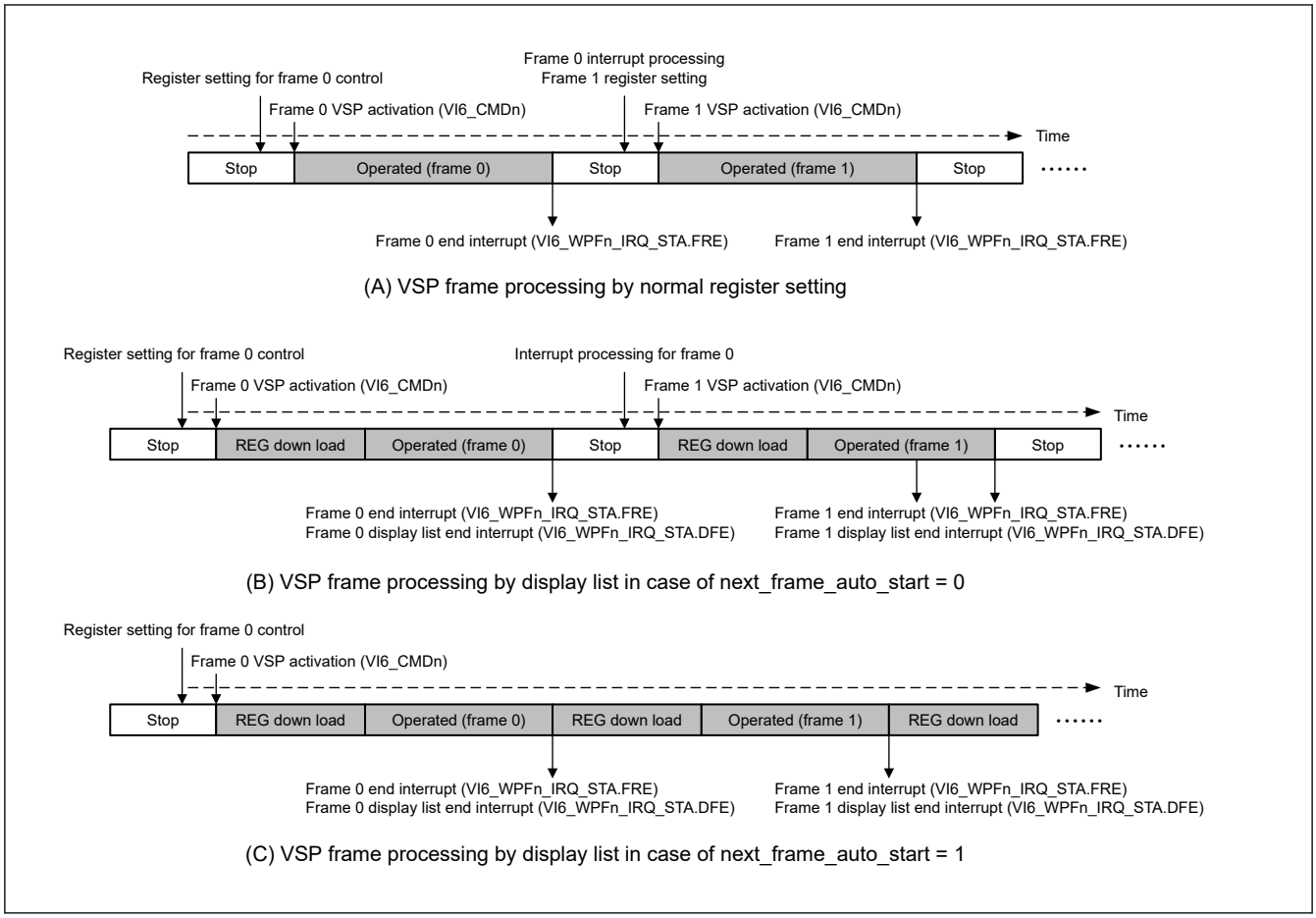


Figure 54.35 Comparison of VSPD operation between normal register settings and display lists

As shown in Figure 54.35 (A), in the VSPD processing through normal register settings, all registers should be set up before the VSPD is started for each frame. After the VSPD processing is completed, the VSPD outputs a frame end interrupt (VI6_WPFn_IRQ_STA.FRE). This method requires a certain amount of time for register settings or interrupt processing by the CPU between frames. In contrast, when display lists are used, the VSPD automatically downloads register settings from external memory as shown in Figure 54.35 (B) and (C), which reduces the load on the CPU between frames.

Figure 54.35 (B) shows the display list usage where the VSPD stops at the end of every frame; only the VSPD start processing for each frame is done by the CPU. This is suitable for the cases when the CPU controls synchronization of frame processing in frame buffer management or when the amount of register values or table data to be set in the VSPD is large. In the case shown in Figure 54.35 (C), as soon as the frame processing ends, the VSPD automatically begins next frame operation and starts downloading new register settings. This is the fastest operation using display lists.

Table 54.35 shows the modes of the display list and the supported functions for each mode. The detail of each mode is described in the following sections.

Table 54.35 Display list mode and supported functions

Mode	Extended display list	Continuous frames
Normal Display List Mode	Supported	Controlled by "next frame auto start" in the header of the display list.
Header-less Display List Mode	Not Supported	Controlled by VI6_DL_CTRL.CFM0 bit.

54.6.4.2 Normal Display List Mode

The VSPD display lists include control information as well as simple register settings in order to control multiple-frame processing in an optimum way for each application. Figure 54.36 shows the display list structure.

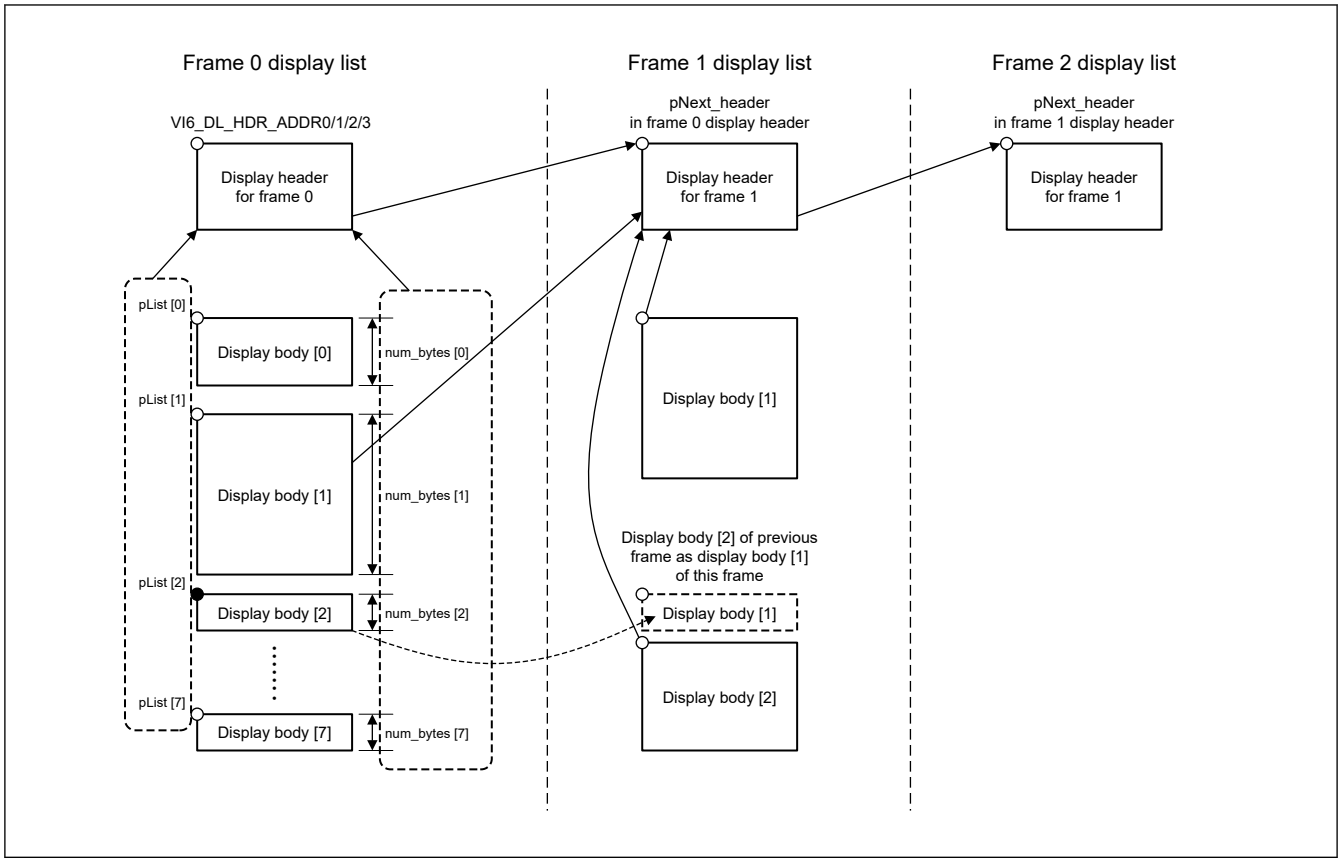


Figure 54.36 Structure and concept of VSPD display list

A VSPD display list consists of two sections; a header section for storing various information and control flags and a body section for storing register and table settings. A combination of these two sections is defined as a display list for a frame. The register and table settings can be divided and stored in up to eight separate bodies allocated in memory. Therefore, the separate bodies storing the register settings for one frame (for example, frame 0) can have non-sequential start addresses; that is, the bodies for one frame can be allocated to areas distant from each other in memory. To gather these bodies and configure the register settings for one frame, a header is used. A display header stores the number of bodies linked with the header and the start address and size of each body.

The VSPD analyzes the header to gather register and table settings stored in separate memory areas and reconfigures the complete register settings.

The addresses of display headers should be specified in the VSPD registers described in section 54.4.5. Display List Control Registers. When activated in a mode that uses display lists, the VSPD downloads display headers from the addresses specified in VI6_DL_HDR_ADDR0 (number 0 correspond to the WPF channel index numbers), analyzes the numbers of bodies and the address and data size of each body, downloads the bodies, and completes register and table settings. After display list downloading is completed, the VSPD becomes ready for frame image processing; the VSPD then starts the actual frame processing.

After processing of a frame ends, the VSPD proceeds to the next frame processing. Here, there are two modes for starting the next frame processing as shown in Figure 54.35. In one mode, the VSPD stops operation and waits for the next activation by the CPU in the same way as when display lists are not used. In this mode, the address used for downloading the display header of the next frame is kept by the internal hardware. Therefore, if valid address information is not stored in the display header for the previous frame, a correct value should be specified in VI6_DL_HDR_ADDR0 while the VSPD is stopped. When the VSPD is started after a correct value is specified, the VSPD starts next frame processing with the same procedure for the previous frame. In contrast to this mode, which stops the VSPD after the end of one-frame processing, there is another mode for automatically starting next frame processing. In automatic start mode, the VSPD downloads the next display header as soon as the previous frame processing ends. After downloading ends, the VSPD starts image processing. The information regarding mode selection, that is, whether to automatically start next frame processing, should be stored in the display header downloaded for the previous frame.

In the automatic start mode, the VSPD continues processing until the display header for a frame specifies that the next frame should not be started automatically. To stop processing during automatic execution, use a software reset (VI6_SRESET).

To strictly define the display list format described above, the following shows the grammatical structure of a display list using pseudo-code. First, to simplify the description of the display list format in the following pages, Table 54.36 defines a function. Function zero bits (num bits) generates a string of one-bit 0s for the number of bits specified by the parameter for the function. By using this function, Table 54.37 defines the header section format of a display list and Table 54.38 defines the body section format and Table 54.39 defines the extended display list body section format.

Data order of Display List Header Section, Display List Body Section and Extended Display List Body Section are assumed as the data is stored in SDRAM by big endian. Table 54.40 shows an example of data order. If data order is not same as the definition, data order within 8-byte unit can be adjusted by setting VI6_DL_SWAP (see section 54.4.5.3. VI6_DL_SWAP0 : Display List-0 Data Swapping Register.

Table 54.36 Definition of a function for simple description

Syntax	Bit count
zero_bits (num_bits)	
{	
for (i = 0; i < num_bits; i++) {	
zero_bit	1
}	
}	

Bit String	Contents
zero_bit	zero_bit indicates a 1-bit integer having a value of 0.

Table 54.37 Format of display list header section (1 of 2)

Syntax	Bit count
display_header () /* Fixed length */	
{	
zero_bits (29)	
num_list_minus1	3
for (i = 0; i < 8; i++) {	
zero_bits (15)	
num_bytes [i]	17
pList [i]	32
}	
pNext_header	32
zero_bits (23)	
wait_wup	1
zero_bits (3)	
ignore_upd_dl	1
zero_bits (2)	
current_frame_int_enable	1
next_frame_auto_start /* 76 bytes from the beginning of this header */	1
if (VI6_DL_EXT_CTRL.EXT) {	
zero_bits (32) /* padding zero 4 bytes for alignment */	

Table 54.37 Format of display list header section (2 of 2)

Syntax	Bit count
zero_bits (6)	
pre_ext_dl_exec	1
post_ext_dl_exec	1
zero_bits (8)	
pre_ext_dl_num_cmd	16
pre_ext_dl_pList	32
zero_bits (16)	
post_ext_dl_num_cmd	16
post_ext_dl_pList /* 96 bytes from the beginning of this header */	32
}	
}	

Bit string	Contents
num_list_minus1	Specifies the value obtained by subtracting 1 from the total number of display list bodies linked with the display header. For example, when this bit field is set to 0, this display list uses one body.
num_bytes [i]	Specifies the number of bytes in the i-th display list body (indicated by index i). Be sure to specify a multiple of eight bytes. For the bodies that are not defined in num_list_minus1 (for example, i = 5 to 7 when num_list_minus1 is set to 4), set this bit field to 0.
pList [i]	Specifies the start address of the i-th display list body (indicated by index i). Be sure to specify an address aligned with an 8-byte boundary (the lower-order three bits are 0). For the bodies that are not defined in num_list_minus1 (for example, i = 5 to 7 when num_list_minus1 is set to 4), set this bit field to 0.
pNext_header	Specifies the address of the display list header for the next frame. After display list downloading ends, the VSPD keeps its value in the internal memory and uses it in the next display list header downloading. Be sure to specify an address aligned with an 8-byte boundary (the lower-order three bits are 0).
wait_wup	When this bit is 1, VSPD starts reading image data from external memory after WUP (Wake Up) signal from any channel of VIN asserts. This bit is available for WPF0.
ignore_upd_dl	If this bit is set to 1, new display list pointed by VI6_DL_HDR_ADDR0 is not downloaded in VSPD H/W for the next frame, even if S/W sets VI6_DL_HDR_ADDR0 during the frame. New display list is downloaded in VSPD H/W at next frame of the processing when this bit is 0. This bit is available for WPF0.
current_frame_int_enable	This is a flag that indicates whether to set the display list end interrupt source (VI6_WPFn_IRQ_STA.DFE) to 1 when the current frame processing ends. If this flag is set to 0, the display list end interrupt source (VI6_WPFn_IRQ_STA.DFE) is not set to 1 when one-frame processing by this display header ends. In this state, even if the display list end interrupt is enabled (VI6_WPFn_IRQ_ENB.DFEE is set to 1), no interrupt will be generated. If this flag is set to 1, the display list end interrupt source (VI6_WPFn_IRQ_STA.DFE) is set to 1 when one-frame processing by this display header ends. In this state, if the display list end interrupt is enabled (VI6_WPFn_IRQ_ENB.DFEE is set to 1), the VSPD generates an interrupt.
next_frame_auto_start	Enables or disables automatic start of next frame processing when one-frame processing by this display header ends. If this bit is set to 1, the VSPD starts next frame processing as soon as one-frame processing by this display header ends, and starts downloading the next frame display header from the pNext_header address specified in this display header. If this bit is set to 0, the VSPD stops operation when one-frame processing by this display header ends. In this case, start the VSPD through VI6_CMDn to process the next frame.
pre_ext_dl_exec	Enables execution of the extended display list for frame preprocessing when VI6_DL_EXT_CTRL.EXT is 1. If this bit is set to 1, the VSPD executes the extended display list for frame preprocessing. The VSPD does not execute it if this bit is set to 0. This bit setting is ignored when VI6_DL_EXT_CTRL.EXT is 0.
post_ext_dl_exec	Set this bit to 0.
pre_ext_dl_num_cmd	Specifies the number of commands in the extended display list body section for frame preprocessing when VI6_DL_EXT_CTRL.EXT is 1. The number of commands that can be specified is 1, and a command is 16 bytes. When pre_ext_dl_exec is set to 0, the extended display list for frame preprocessing is not executed; set this bit to 0. This bit setting is ignored when VI6_DL_EXT_CTRL.EXT is 0.

Bit string	Contents
pre_ext_dl_pList	Specifies the start address of the area where the extended display list body section for frame preprocessing is stored when VI6_DL_EXT_CTRL.EXT is 1. Be sure to specify an address aligned with a 16-byte boundary (lower-order four bits are 0). When pre_ext_dl_exec is set to 0, the extended display list for frame preprocessing is not executed; set this bit to 0. This bit setting is ignored when VI6_DL_EXT_CTRL.EXT is 0.
post_ext_dl_num_cmd	Set this bit to 0.
post_ext_dl_pList	Set this bit to 0.

Table 54.38 Format of display list body section

Syntax	Bit count
display_list (num_bytes) /* Variable length (num_bytes) */	
{	
for (i = 0; i < num_bytes; i += 8) {	
set_address	32
set_data	32
}	
}	

Bit string	Contents
set_address	Specifies the address where the value specified by set_data is to be stored. The upper 17-bit of set_address in Display List Body Section should be set to 0, since the register space of VSPD is 32 KB (0x0000 - 0x7FFC).
set_data	Specifies the value to be stored in the address specified by set_address. Specify a value to be set in a register.

Table 54.39 Format of extended display list body section

Syntax	Bit count
ext_dl_display_list (num_llw) /* Variable length (pre/post_ext_dl_num_bytes) */	
{	
for (i = 0; i < num_llw; i += 2) {	
ext_dl_cmd	64
ext_dl_data	64
}	
}	

Table 54.40 Data order of display list body (1 of 2)

Address	Syntax
pList	set_address [0] (bit 31-24)
pList + 1	set_address [0] (bit 23-16)
pList + 2	set_address [0] (bit 15-8)
pList + 3	set_address [0] (bit 7-0)
pList + 4	set_data [0] (bit 31-24)
pList + 5	set_data [0] (bit 23-16)
pList + 6	set_data [0] (bit 15-8)
pList + 7	set_data [0] (bit 7-0)
pList + 8	set_address [1] (bit 31-24)
pList + 9	set_address [1] (bit 23-16)
pList + 10	set_address [1] (bit 15-8)

Table 54.40 Data order of display list body (2 of 2)

Address	Syntax
pList + 11	set_address [1] (bit 7-0)
pList + 12	set_data [1] (bit 31-24)
pList + 13	set_data [1] (bit 23-16)
pList + 14	set_data [1] (bit 15-8)
pList + 15	set_data [1] (bit 7-0)
~~	~~~

54.6.4.3 Header-less Display List Mode

The header-less display list does not have the display header listed in Table 54.37, and it has the simplest structure that has only single body. The extended display list function is not available in case of the header-less display list mode. Set the VI6_DL_CTRL.NH0 to 1 (see section 54.4.5.1. VI6_DL_CTRL : Display List Control Register) to use the header-less display list. The start address downloaded by the header-less display list should be set to VI6_DL_HDR_ADDR0. And the size of the display body which is originally defined in the display header should be set to VI6_DL_BODY_SIZE0. The header-less display list is available only in WPF0.

54.6.4.4 Restrictions on Display List Usage

Access to the general control registers and display list control registers through a display list is prohibited. When using display lists, be sure to observe the following restrictions on register access by the CPU.

1. Do not execute write access to the same register (same address) from the CPU and through a display list at the same time. If such a conflict occurs, correct operation of the VSPD is not guaranteed.
2. Do not execute write access to the same LUT lookup table from the CPU and through a display list at the same time. If such a conflict occurs, correct operation of the VSPD is not guaranteed. Here, the same lookup table means the address space having the same space name shown in Table 54.41.
3. When read access by the CPU and write access through a display list to the same register (same address) occur at the same time, the read value returned to the CPU is not guaranteed.
4. When read access by the CPU and write access through a display list to the same LUT lookup table occur at the same time, the read value returned to the CPU is not guaranteed. Here, the same lookup table means the address space having the same space name shown in Table 54.41.
5. For other restrictions on values and timing of register setting through display lists, see the restrictions on normal register settings described in section 54.4.3. Restrictions on Access to Registers and Lookup Tables.
6. Manipulation and setting of the registers described in section 54.4.4. General Control Registers and section 54.4.5. Display List Control Registers through a display list is prohibited.
7. The extended display list function enabled by VI6_DL_EXT_CTRL.EXT is to control the registers of the ICB connected with the VSPD. Do not use extended display lists for any other purpose.

54.6.5 Interrupt Processing

54.6.5.1 LCDC_VSPD_INT

When a WPF module generates an internal source that should be notified, an interrupt signal is output. As internal sources are generated in WPF independently, the VSPD has the following registers to control interrupts in WPF.

- See section 54.4.4.12. VI6_WPF0_IRQ_ENB : WPF0 Interrupt Enable Register
- See section 54.4.4.13. VI6_WPF0_IRQ_STA : WPF0 Interrupt Status Register

CAUTION

To use interrupts, enable them through WPF interrupt enable registers. If an interrupt source register has already been set to 1 for some reason, an unintended interrupt will occur as soon as the corresponding interrupt enable register is set as enabled. To avoid this, before enabling interrupts through WPF interrupt enable registers, be sure to clear all WPF interrupt sources to be enabled to 0. Be careful about this procedure when setting up registers before starting the VSPD.

54.6.5.2 LCDC_DU_INT

When the PBUF will detect FIFO underflow, an interrupt signal is output. The DU has the following registers to control interrupt.

- See section 54.5.2. DU_MSR0 : DU Module Status Register 0
- See section 54.5.4. DU_IMR0 : DU Interrupt Mask Register 0

CAUTION

To use interrupts, enable them through DU Interrupt Mask Register. If an interrupt is asserted, the PBUF is underflowing and invalid display data is being output. When an interrupt occurs, the interrupt can only be cleared by DU_MCR0.PB_CLR register.

If DU_MCR1.PB_AUTOCLR is set a value other than 0, an interrupt is cleared according to DU_MCR1.OPMD register.

54.6.6 Lookup Table Settings

54.6.6.1 LUT

For a single entry to the LUT space (see Table 54.41) of the VSPD, the LUT data is set by a write access in the format shown in Figure 54.37.

The entry address of each space is (start address of the space) + (entry number counting from the base point 0 × 0x4). For example, the address of entry 7 to the LUT space is 0x920E_7000 + 7 × 0x4 = 0x920E_701C.

Table 54.41 shows the spaces for which entries can be made. Note that if the module that references that space is operating, write accesses to the relevant space are prohibited. When a read access is made to the relevant space during operation of the referencing module, undefined values will be read out.

The operating/stopped state of each module in Table 54.41 is the operating state of the WPF to which each module is connected. Determine whether the module is operating or stopped using each WPF operating status bit in the VI6_STATUS register.

Table 54.41 LUT space addresses

Space name	Space addresses	Entry count	Module that references the space in the left column
VI6_LUT_TBL	0x920E_7000 to 0x920E_73FC	256	LUT

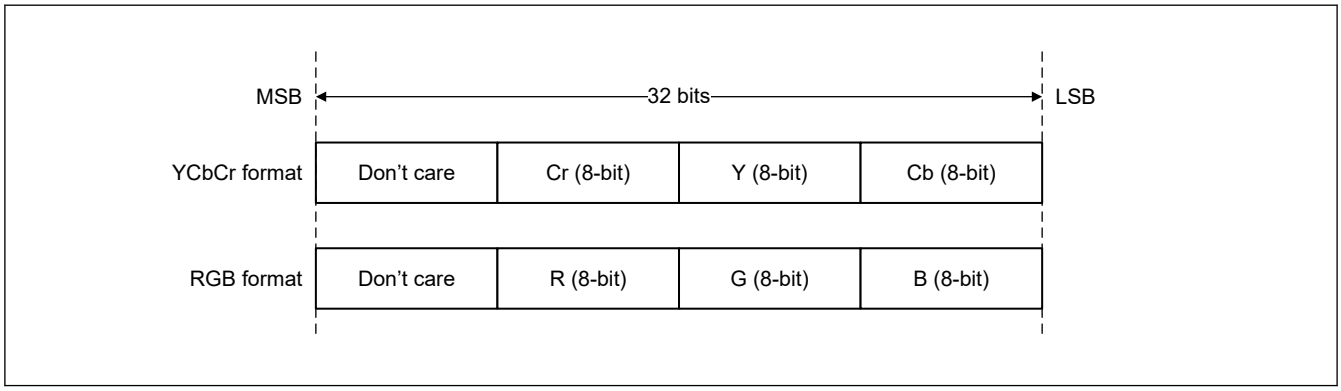


Figure 54.37 VI6_LUT_TBL formats

54.6.7 Linked with DU

54.6.7.1 Operation flow of VSPD and DU

This section shows operation flow of VSPD by using Normal Display List Mode. For details of Normal Display List Mode, see section 54.6.4.2. Normal Display List Mode.

Figure 54.38 to Figure 54.40 shows a procedure (register setting flow) to display image data to display panel.

Figure 54.38 shows setting flow to start VSPD/LIF0 linked with DU. Figure 54.39 shows setting flow to update display image. Figure 54.40 shows setting flow to stop VSPD/LIF0 linked with DU.

Using display list with enabling next_frame_auto_start shown in section 54.6.4. Display List is required to take the flow from Figure 54.38 to Figure 54.40.

NOTE

If display sync operation ([1-7] step in Figure 54.38) is set before activating the VSPD, invalid data is displayed until image data read from external memory is ready to be transferred to DU. In this case, image data from external memory is displayed late by one vsync or more.

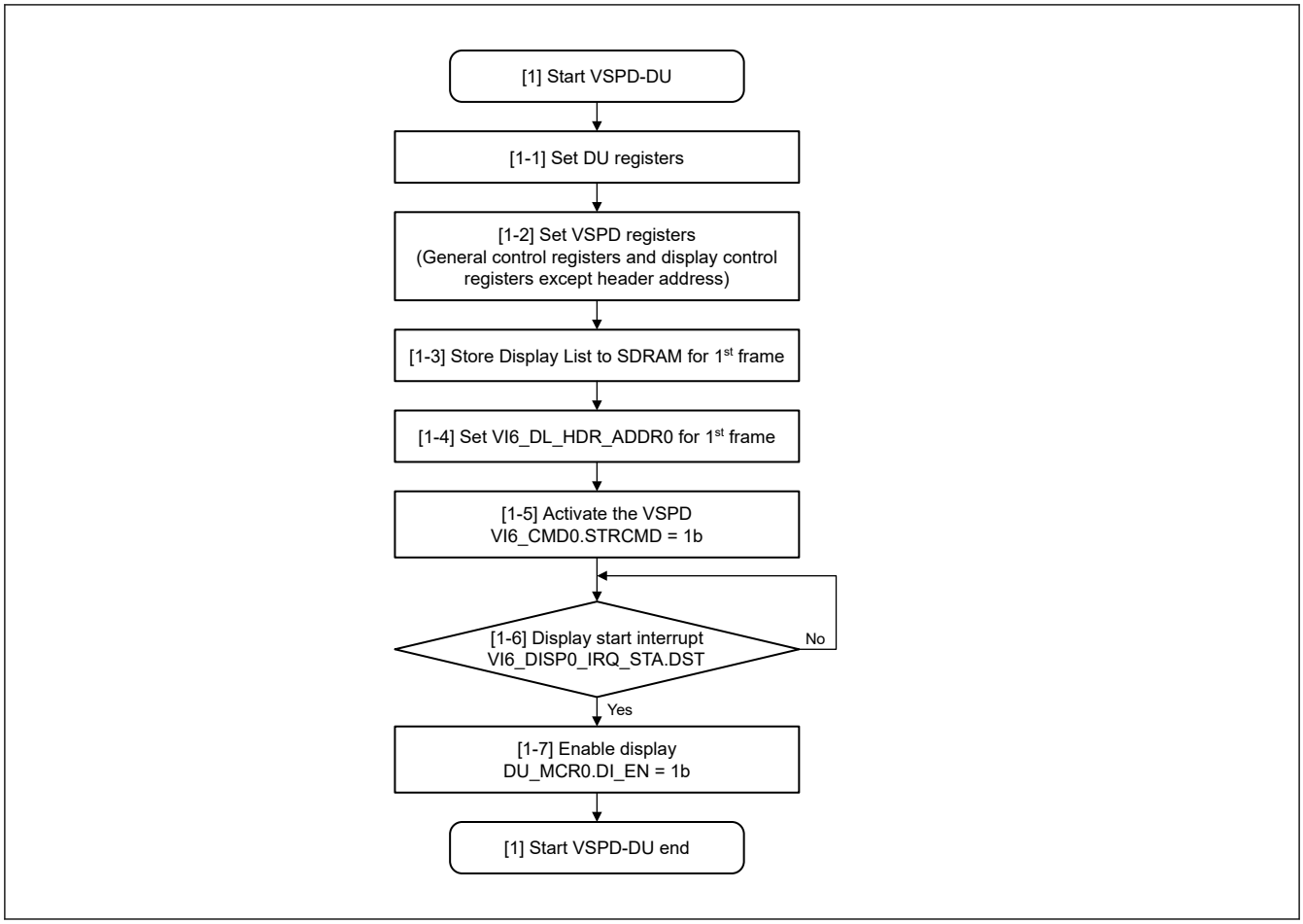


Figure 54.38 Setting flow to start VSPD linked with DU

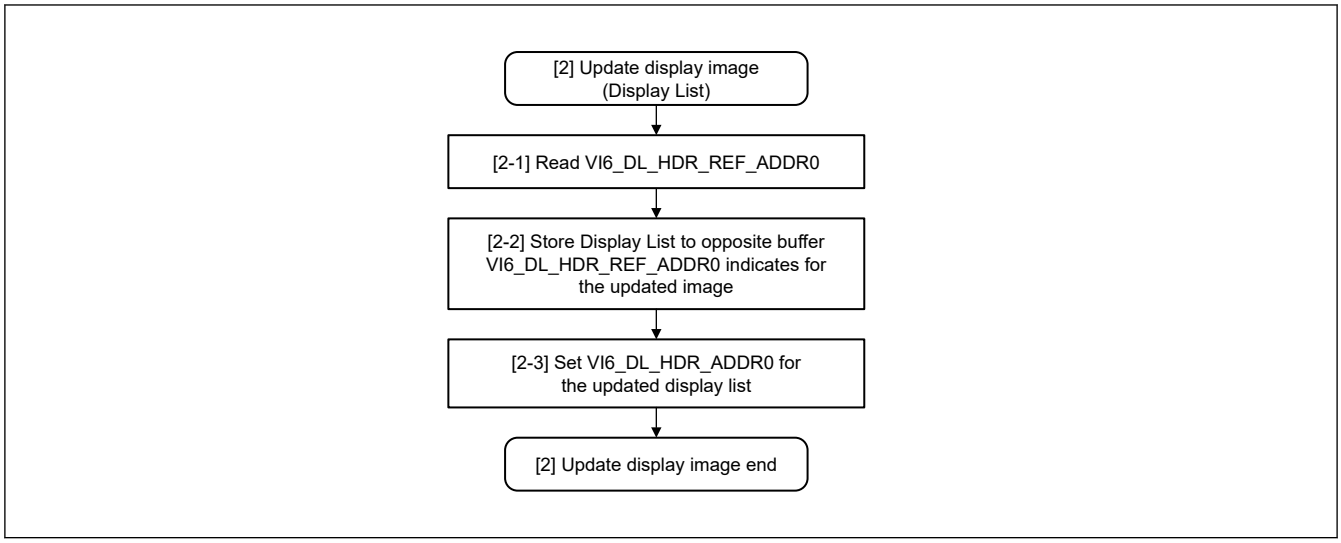
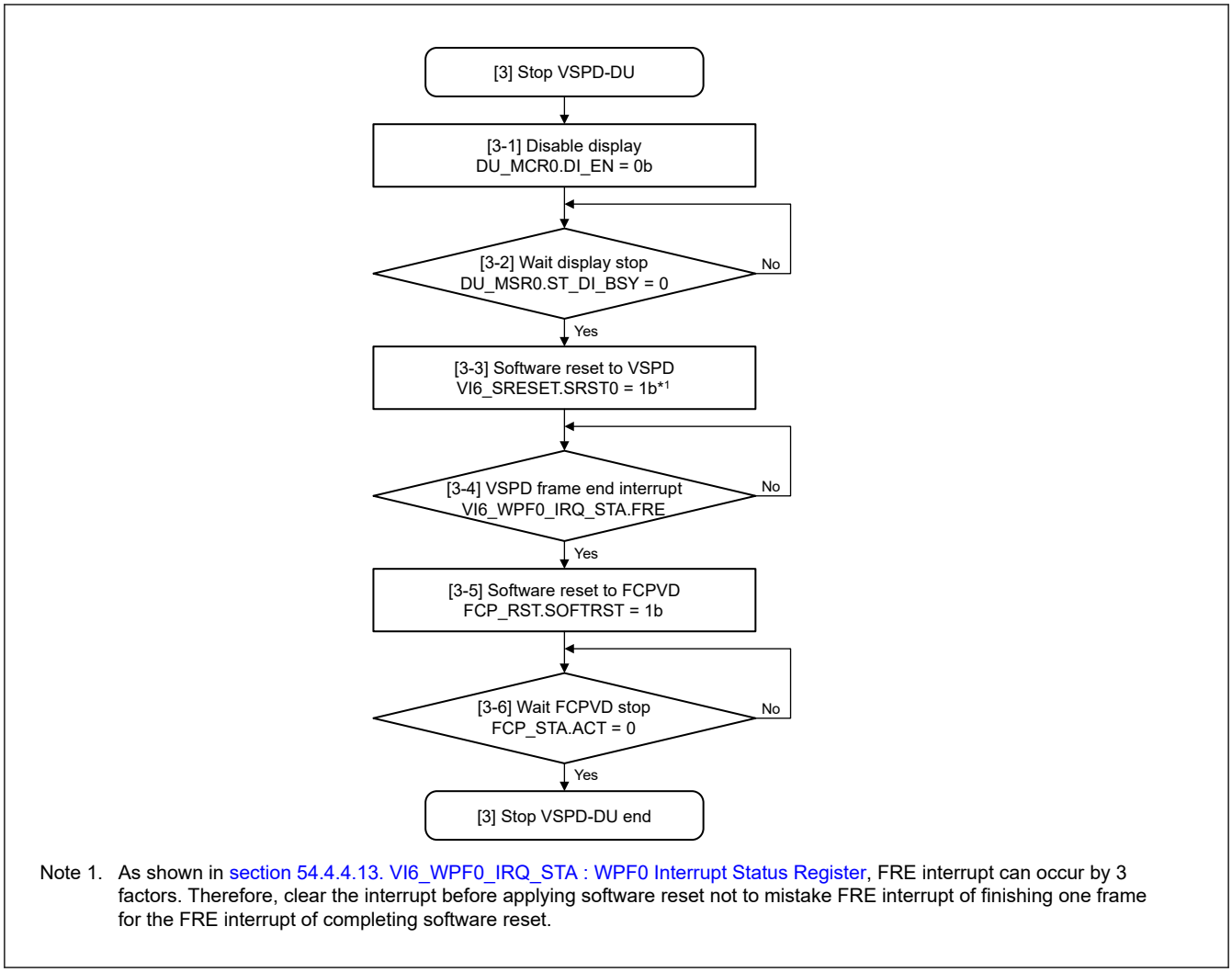


Figure 54.39 Setting flow to update display image



Note 1. As shown in section 54.4.4.13. VI6_WPF0_IRQ_STA : WPF0 Interrupt Status Register, FRE interrupt can occur by 3 factors. Therefore, clear the interrupt before applying software reset not to mistake FRE interrupt of finishing one frame for the FRE interrupt of completing software reset.

Figure 54.40 Setting flow to stop VSPD linked with DU

It is possible to update display list at any time (arbitrary timing) as shown in Figure 54.39. Do not overwrite display list in external memory pointed by VI6_DL_HDR_REF_ADDR. It is because VSPD is reading the display list pointed by VI6_DL_HDR_REF_ADDR or will be reading the display list soon.

Detail explanation is shown below:

As shown in section 54.4.5.6. VI6_DL_HDR_REF_ADDR0 : Display List-0 Header Reference Address Register, when VSPD is reading display list from external memory (Period [A] in Figure 54.41), VI6_DL_HDR_REF_ADDRn indicates header address of the display list referred by VSPD. When VSPD is not reading display list from external memory (Period [B] in Figure 54.41), VI6_DL_HDR_REF_ADDRn indicates the value of VI6_DL_HDR_ADDRn, and there is possibility that the display list is being read soon by VSPD at next frame start timing. Therefore, keep display list in external memory pointed by VI6_DL_HDR_REF_ADDR. Details of software sequence to update display list and VSPD-H/W behavior are shown below.

1. Store latest display list in external memory area different from the area pointed by VI6_DL_HDR_REF_ADDR, because the area pointed by VI6_DL_HDR_REF_ADDR is being read or is being read soon by VSPD.
2. Set header address of new display list into VI6_DL_HDR_ADDRn.
3. VSPD reads display list from the external memory area pointed by VI6_DL_HDR_ADDRn at frame start timing.

See also section 54.4.5.2. VI6_DL_HDR_ADDR0 : Display List-0 Header Address Register and section 54.4.5.6. VI6_DL_HDR_REF_ADDR0 : Display List-0 Header Reference Address Register.

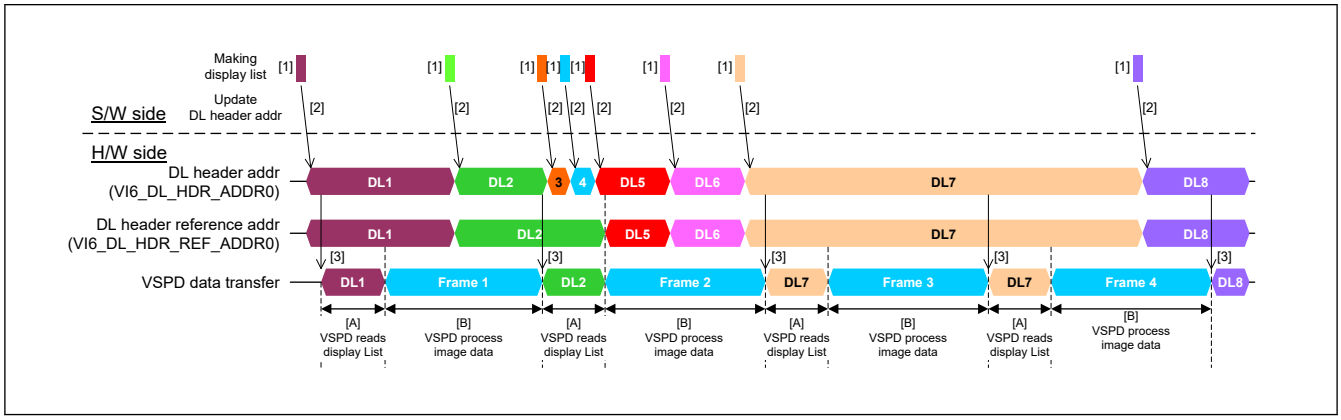
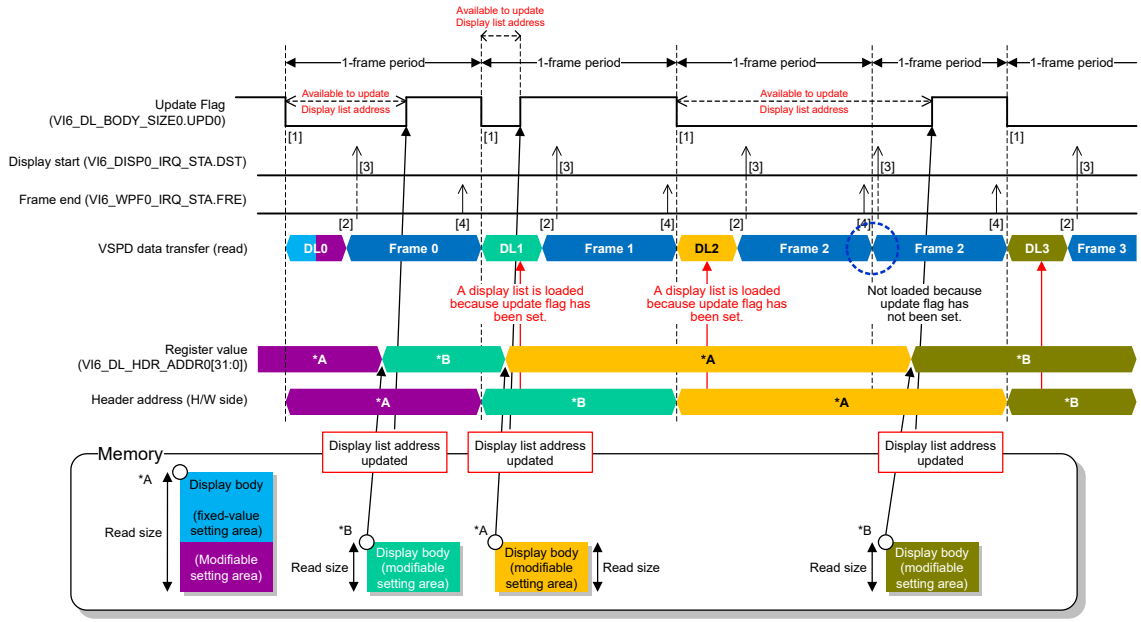


Figure 54.41 Updating display list at arbitrary timing

54.6.7.2 Controlling Two Register Planes using Display Lists

This section shows operation flow of VSPD by using Header-less Display List Mode. Start procedure and stop procedure are same with Normal Display List Mode in section 54.6.7.1. Operation flow of VSPD and DU except for the point that VI6_DL_BODY_SIZE0 should be set before step [1-5] in start procedure shown in Figure 54.40. The procedure to update display list is different with Normal Display List Mode, and is described later. Figure 54.42 shows the control of two register planes using header-less display lists (see section 54.6.4.3. Header-less Display List Mode) and its timing. In the description hereafter, the use of header-less display lists is always assumed and they are simply called display lists.



Display list plane A: "Fixed-value settings" and "modifiable settings" are stored as a set in contiguous area A.
Display list plane B: Only the "modifiable settings" are stored in area B, which is separate from display list A.

To switch settings for every frame, modify two display list planes alternately. After every modification, update the display list address register (VI6_DL_HDR_ADDR0[31:0]) and writing 1 to update flag (VI6_DL_BODY_SIZE0.UPD0) in the VSPD.
Update flag (VI6_DL_BODY_SIZE0.UPD0) is negated automatically after Header Address is loaded into H/W side.

Figure 54.42 Controlling two register planes using a display list

The VSPD downloads a display list immediately after activation. In the start frame, download the display list that contains all necessary settings. From the next frame on, only the necessary register or table values should be specified in a display list.

When the update flag of VI6_DL_BODY_SIZE0.UPD0 is set to 1, VSPD downloads a new display list at the start of the next frame. When the update flag of VI6_DL_BODY_SIZE0.UPD0 is set to 0, the register settings acquired from the display list previously downloaded are retained and used for the next operation without downloading a new display list.

Order of each VSPD-H/W Events mentioned as [1], [2], [3], and [4] in Figure 54.42 are shown below:

[1] VI6_DL_BODY_SIZE0.BS0 and VI6_DL_HDR_ADDR0 are downloaded in H/W side. And Update Flag (VI6_DL_BODY_SIZE0.UPD0) is negated automatically.

[2] Reading display list from external memory into VSPD-H/W is finished through master access.

[3] Display start interrupt status (VI6_DISP_IRQ_STA.DST) asserts.

[4] Frame end interrupt status (VI6_WPF0_IRQ_STA.FRE) asserts.

Do not overwrite the display list in external memory for a period from "[4] VI6_WPF0_IRQ_STA.FRE" to next "[3] VI6_DISP_IRQ_STA.DST", because VSPD is reading the display list for the period.

54.7 Usage Notes

54.7.1 Assignment in Memory Space

Make sure that VSPD memory space shall be mapped to Non-Cache region.

54.7.2 Input Image Size

Table 54.42 is a list of input size specifications.

Table 54.42 List of input size specifications

Module	Min. input size	Max. input size	Restriction on setting unit
RPF	1 (horizontal) × 1 (vertical) pixel	1920 (horizontal) × 1080 (vertical) pixels	YCbCr422: 2-pixel units horizontally and 1-pixel units vertically. YCbCr420: 2-pixel units both horizontally and vertically. Other formats: 1-pixel units both horizontally and vertically. 1. When the 1-bpp alpha plane*1 is input, the size can always be specified in 8-pixel units both horizontally and vertically regardless of the input format. 2. These restrictions including note 1 are applied to the following. VI6_RPFn_SRC_BSIZE VI6_RPFn_SRC_ESIZE
LUT	1 (horizontal) × 1 (vertical) pixel	1920 (horizontal) × 1080 (vertical) pixels	1-pixel units both horizontally and vertically.
BRS	1 (horizontal) × 1 (vertical) pixel	1920 (horizontal) × 1080 (vertical) pixels	1-pixel units both horizontally and vertically.
LIF	1 (horizontal) × 1 (vertical) pixel	1920 (horizontal) × 1080 (vertical) pixels	1-pixel units both horizontally and vertically.
WPF	1 (horizontal) × 1 (vertical) pixel	1920 (horizontal) × 1080 (vertical) pixels	YCbCr422: 2-pixel units horizontally and 1-pixel units vertically. YCbCr420: 2-pixel units both horizontally and vertically. Other formats: 1-pixel units both horizontally and vertically. 1. This restriction on the WPF only applies to the WPF output size. The WPF input size should be specified in 1-pixel units.

Note 1. When VI6_RPFn_ALPH_SEL.ASEL is set to 011b.

The most important restriction shown in Table 54.42 is the setting unit. Make appropriate register settings so that the size of the image input to each module does comply with setting units and does not exceed the limits shown in Table 54.42.

54.7.3 Output Image Size

The size of the output from each WPF is determined by the results of processing in the modules connected with the DPR. As shown in Figure 54.1, the data input to the VSPD is sent to the WPF output modules through the RPF and the modules connected with the DPR. When there is no processing that changes the image size through this data path, the WPF output size is the same as the RPF input size. Table 54.43 is a list of the processing that changes image size.

Table 54.43 Image processing that changes image size

Module	Function*1	Related register	Size of output from module
WPF	Input size clipping	VI6_WPFn_HSZCLIP VI6_WPFn_VSZCLIP	When this function is disabled, the input size and the output size are the same. When this function is enabled, the output is in the following size. Horizontal output size: VI6_WPFn_HSZCLIP.HCL_SIZE × 2 setting Vertical output size: VI6_WPFn_VSZCLIP.VCL_SIZE × 2 setting
LIF	Padding line	VI6_LIFn_CTRL	When padding with dummy lines is disabled, output size is same with Input size. When padding with dummy lines is enabled: Horizontal output size: input size Vertical output size: VI6_LIF0_PADLN_SIZE setting

Note: See section 54.4.1. Notational Conventions for Registers and Bit Fields for explanation of register bit field.
Note 1. For details of each function, see the descriptions of the related registers.

The input image size can be changed only with the modules and functions shown in Table 54.43. With the other modules and functions, the input size and output size are the same. Accordingly, after module connections with the DPR are determined, the VSPD output size can also be determined through the following steps.

1. The image size (VI6_RPFn_SRC_ESIZE) read from the external memory and sent to the DPR by the RPFn is the initial value.
2. When the module connected with the DPR is not a module (function) shown in Table 54.43, the output size from the module is the same as the input size; the image size does not need to be updated.
3. When a module connected with the DPR is a module (function) shown in Table 54.43, the output image size should be updated to the size shown in the table, which should be used as the input image size for the module connected behind it.
4. When the final image size at the WPFn is determined, this size is the VSPD output size for that WPFn path.

Figure 54.43 shows how to determine the image size in a sample DPR connection through the above steps. The related conditions that determine the image size are also shown.

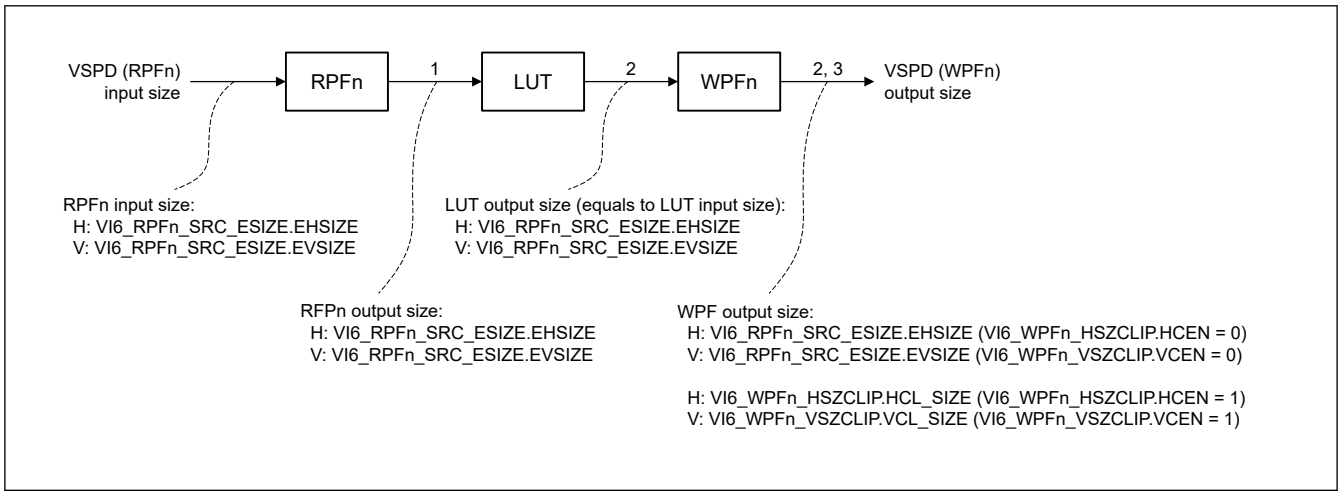


Figure 54.43 Input/output size for each module in a sample DPR connection

Make appropriate register settings in each module so that the VSPD output image size determined as shown in the figure does not violate the restrictions shown in Table 54.43.

54.7.4 Restriction

When the data output from this unit is written back to the same memory area where the input data for this unit has been read, this unit has the following restrictions.

1. The access order and format on the frame memory are the same between the input pixels and output pixels.
2. Specifying a larger output image size than the input image size, either vertically or horizontally or both vertically and horizontally, is prohibited.
3. When the YCbCr4:2:0 format is input, operation between color components is prohibited.

These restrictions are summarized in Table 54.44. See the descriptions of the registers related to each restriction. In the table, RPFm indicates the RPFn that inputs the master layer, and WPFwb indicates the WPFn that writes back the output image to the source image area for the master layer.

Table 54.44 Restrictions on use when output data is written back to input data area

No.	Restriction	Related registers
Restriction 1	The RPFm input format and the WPFwb output format should be the same.	VI6_RPFn_INFMT.RDFMT VI6_RPFn_INFMT.VIR
	The RPFm source image storing address and the WPFwb destination address should be the same.	VI6_RPFn_SRCM_ADDR_*
	The RPFm source picture memory stride and the WPFwb destination memory stride should be the same.	VI6_RPFn_SRCM_PSTRIDE VI6_RPFn_SRCM_ASTRIDE
	The RPFm and WPFwb data swapping settings should be the same.	VI6_WPFn_DSWAP
Restriction 2	The RPFm basic read size and extended read size should be the same.	VI6_RPFn_SRC_BSIZE VI6_RPFn_SRC_ESIZE
	Color space conversion is prohibited in RPFm and WPFwb.	VI6_RPFn_INFMT.CSC VI6_WPFn_OUTFMT.CSC
Restriction 3*1	NOP should be specified for IROP operation.	VI6_RPFn_ALPH_SEL.IROP
	Color keying is prohibited.	VI6_RPFn_CKEY_CTRL.CV

Note 1. When the input format is not YCbCr4:2:0, restriction 3 is not applied.

55. SD/MMC Host Interface (SDHI)

This section describes the functions of the SD/MMC host interface (SDHI).

CAUTION

Development of the SD host--related products needs the conclusion of the following agreement.

- "SD Host/Ancillary Product License Agreement (SD HALA)"

55.1 Overview

55.1.1 Features

- SD memory/IO card interface (1-bit/4-bit SD bus)
- SD, SDHC, and SDXC SD memory card access supported
- Default, high-speed, UHS-I/SDR12, SDR25, SDR50, SDR104, and DDR50 transfer modes supported
- SD clock (SD_CLK) frequency = PCLKAM (200 MHz)/2^n (n = 0 to 9)
- Error check function: CRC7 (for command/response), CRC16 (for data)
- Interrupt request: 2
- Card detect function
- Write protect supported
- MMC interface (1-/4-/8-bit MMC bus)

  Note: Channel 0 only support.
- e-MMC device access supported
- Backward-compatible, high-speed (SDR/DDR), and HS200 transfer modes supported
- High-priority interrupt (HPI) supported

55.1.2 Block Diagram

Figure 55.1 shows a block diagram of the SD/MMC host interface.

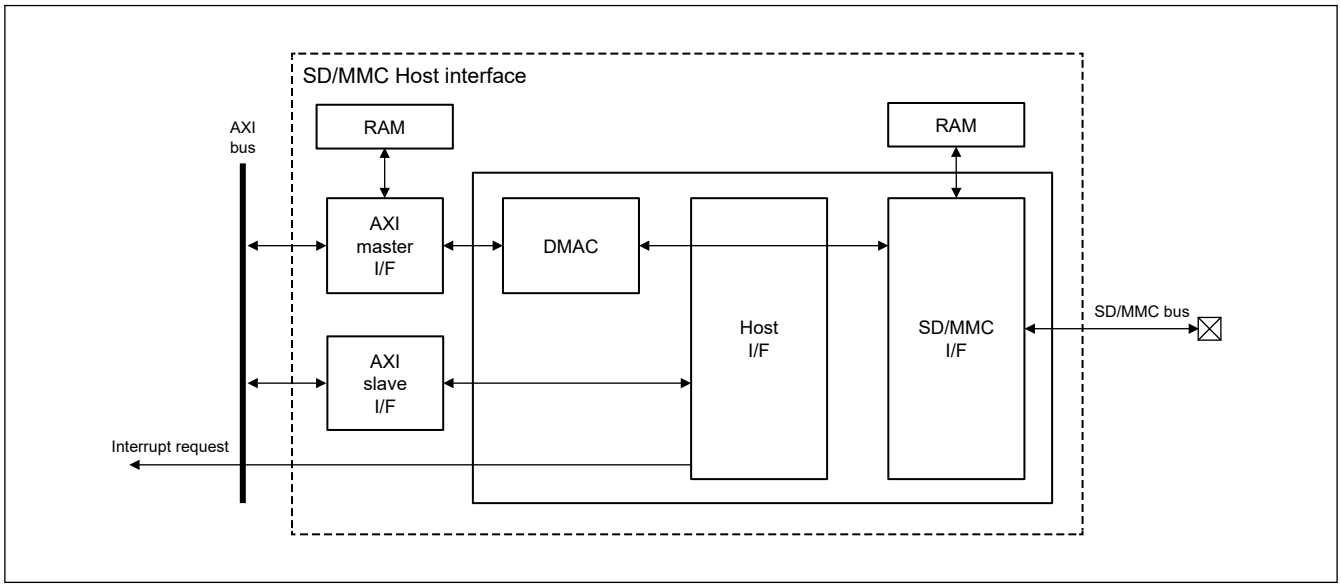


Figure 55.1 Block diagram of SD/MMC host interface

55.1.3 External Pins

Table 55.1 lists the input and output pins used by the interface. The operating voltage on the pins of the SD/MMC host interface is 3.3 V or 1.8 V, except for some supplemental pins.

Table 55.1 Input/output pins of the SDHI

Name	I/O	Function	Operating voltage
SDm_CLK*1	O	SD/MMC clock output	1.8 V/3.3 V
SDm_CMD	I/O	SD/MMC command output, response input	1.8 V/3.3 V
SDm_DATA0*1	I/O	SD/MMC Data 0 [bit 0]	1.8 V/3.3 V
SDm_DATA1*1	I/O	SD/MMC Data 1 [bit 1], SDIO interrupt	1.8 V/3.3 V
SDm_DATA2*1	I/O	SD/MMC Data 2 [bit 2], read wait	1.8 V/3.3 V
SDm_DATA3*1	I/O	SD/MMC Data 3 [bit 3], Card detection	1.8 V/3.3 V
SD0_DATA4*1	I/O	SD/MMC Data 4 [bit 4]	1.8 V/3.3 V
SD0_DATA5*1	I/O	SD/MMC Data 5 [bit 5]	1.8 V/3.3 V
SD0_DATA6*1	I/O	SD/MMC Data 5 [bit 6]	1.8 V/3.3 V
SD0_DATA7*1	I/O	SD/MMC Data 7 [bit 7]	1.8 V/3.3 V
SDm_CD*1	I	SD card detection*2	3.3 V
SDm_WP*1	I	SD write protection*2	3.3 V
SD0_RST#	O	MMC reset	1.8 V/3.3 V
SDm_PWEN	O	SD power enable	3.3 V
SDm_IOVS	O	SD voltage select	3.3 V

Note: m = 0, 1
Note 1. In this document, these pins are referred as SDCLK, SDDAT0, SDDAT1, …, SDDAT7, ISDCD, and ISDWP in order.
Note 2. Fix to high level when not in use.

55.2 Register Map

Table 55.2 SDHI register map (1 of 2)

Address	Register symbol	Register name	Write protection
0x9208_0000 + 0x1_0000 × m (m = 0, 1)	SD_CMD	Command Type Register	—
0x9208_0010 + 0x1_0000 × m (m = 0, 1)	SD_ARG	Command Argument Register	—
0x9208_0018 + 0x1_0000 × m (m = 0, 1)	SD_ARG1	Command Argument Register 1	—
0x9208_0020 + 0x1_0000 × m (m = 0, 1)	SD_STOP	Data Stop Register	—
0x9208_0028 + 0x1_0000 × m (m = 0, 1)	SD_SECCNT	Block Count Register	—
0x9208_0030 + 0x1_0000 × m (m = 0, 1)	SD_RSP10	SD Card Response 10 Register	—
0x9208_0038 + 0x1_0000 × m (m = 0, 1)	SD_RSP1	SD Card Response 1 Register	—
0x9208_0040 + 0x1_0000 × m (m = 0, 1)	SD_RSP32	SD Card Response 32 Register	—
0x9208_0048 + 0x1_0000 × m (m = 0, 1)	SD_RSP3	SD Card Response 3 Register	—
0x9208_0050 + 0x1_0000 × m (m = 0, 1)	SD_RSP54	SD Card Response 54 Register	—
0x9208_0058 + 0x1_0000 × m (m = 0, 1)	SD_RSP5	SD Card Response 5 Register	—
0x9208_0060 + 0x1_0000 × m (m = 0, 1)	SD_RSP76	SD Card Response 76 Register	—
0x9208_0068 + 0x1_0000 × m (m = 0, 1)	SD_RSP7	SD Card Response 7 Register	—
0x9208_0070 + 0x1_0000 × m (m = 0, 1)	SD_INFO1	SD Card Interrupt Flag Register 1	—
0x9208_0078 + 0x1_0000 × m (m = 0, 1)	SD_INFO2	SD Card Interrupt Flag Register 2	—
0x9208_0080 + 0x1_0000 × m (m = 0, 1)	SD_INFO1_MASK	SD_INFO1 Interrupt Mask Register	—

Table 55.2 SDHI register map (2 of 2)

Address	Register symbol	Register name	Write protection
0x9208_0088 + 0x1_0000 × m (m = 0, 1)	SD_INFO2_MASK	SD_INFO2 Interrupt Mask Register	—
0x9208_0090 + 0x1_0000 × m (m = 0, 1)	SD_CLK_CTRL	SD Clock Control Register	—
0x9208_0098 + 0x1_0000 × m (m = 0, 1)	SD_SIZE	Transfer Data Length Register	—
0x9208_00A0 + 0x1_0000 × m (m = 0, 1)	SD_OPTION	SD Card Access Control Option Register	—
0x9208_00B0 + 0x1_0000 × m (m = 0, 1)	SD_ERR_STS1	Error Status Register 1	—
0x9208_00B8 + 0x1_0000 × m (m = 0, 1)	SD_ERR_STS2	SD Error Status Register 2	—
0x9208_00C0 + 0x1_0000 × m (m = 0, 1)	SD_BUF0	SD Buffer Read/Write Register	—
0x9208_00D0 + 0x1_0000 × m (m = 0, 1)	SDIO_MODE	SDIO Mode Control Register	—
0x9208_00D8 + 0x1_0000 × m (m = 0, 1)	SDIO_INFO1	SDIO Interrupt Flag Register	—
0x9208_00E0 + 0x1_0000 × m (m = 0, 1)	SDIO_INFO1_MASK	SDIO_INFO1 Interrupt Mask Register	—
0x9208_0360 + 0x1_0000 × m (m = 0, 1)	CC_EXT_MODE	DMA Mode Enable Register	—
0x9208_0380 + 0x1_0000 × m (m = 0, 1)	SOFT_RST	Software Reset Register	—
0x9208_0388 + 0x1_0000 × m (m = 0, 1)	VERSION	Version Register	—
0x9208_0390 + 0x1_0000 × m (m = 0, 1)	HOST_MODE	Host Interface Mode Setting Register	—
0x9208_0398 + 0x1_0000 × m (m = 0, 1)	SDIF_MODE	SD Interface Mode Setting Register	—
0x9208_03C8 + 0x1_0000 × m (m = 0, 1)	SD_STATUS	SD Status Register	—
0x9208_0820 + 0x1_0000 × m (m = 0, 1)	DM_CM_DTRAN_MODE	DMAC Transfer Mode Register	—
0x9208_0828 + 0x1_0000 × m (m = 0, 1)	DM_CM_DTRAN_CTRL	DMAC Transfer Control Register	—
0x9208_0830 + 0x1_0000 × m (m = 0, 1)	DM_CM_RST	DMAC Reset Register	—
0x9208_0840 + 0x1_0000 × m (m = 0, 1)	DM_CM_INFO1	DMAC Interrupt Register 1	—
0x9208_0848 + 0x1_0000 × m (m = 0, 1)	DM_CM_INFO1_MASK	DM_CM_INFO1 Interrupt Mask Register	—
0x9208_0850 + 0x1_0000 × m (m = 0, 1)	DM_CM_INFO2	DMAC Interrupt Register 2	—
0x9208_0858 + 0x1_0000 × m (m = 0, 1)	DM_CM_INFO2_MASK	DM_CM_INFO2 Interrupt Mask Register	—
0x9208_0880 + 0x1_0000 × m (m = 0, 1)	DM_DTRAN_ADDR	DMAC Transfer Address Register	—
0x9208_1000 + 0x1_0000 × m (m = 0, 1)	SCC_DTCNTL	Initial Setting Register	—
0x9208_1008 + 0x1_0000 × m (m = 0, 1)	SCC_TAPSET	Sampling Clock Position Setting Register	—
0x9208_1010 + 0x1_0000 × m (m = 0, 1)	SCC_DT2FF	Hardware Adjustment Register 1	—
0x9208_1018 + 0x1_0000 × m (m = 0, 1)	SCC_CKSEL	Sampling Clock Selection Register	—
0x9208_1020 + 0x1_0000 × m (m = 0, 1)	SCC_RVSCNTL	Sampling Clock Position Correction Register	—
0x9208_1028 + 0x1_0000 × m (m = 0, 1)	SCC_RVSREQ	Sampling Clock Position Correction Request Register	—
0x9208_1030 + 0x1_0000 × m (m = 0, 1)	SCC_SMPCMP	Sampling Data Comparison Register	—
0x9208_1038 + 0x1_0000 × m (m = 0, 1)	SCC_TMPPORT	Hardware Adjustment Register 2	—

Note 1. All the registers except SCC registers are accessible by 16-bit, 32-bit, or 64-bit. SCC registers are accessible by 32-bit or 64-bit.

Table 55.3 SDHI related system control register

Unit	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
0	—	MSTPCRM.MSTPCRM12	SLVACCCTL4.SDHI0_SL
1	—	MSTPCRM.MSTPCRM13	SLVACCCTL4.SDHI1_SL

55.3 Register Description

55.3.1 SD_CMD : Command Type Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MD7MD6[1:0]		MD5	MD4	MD3	MD2MD0[2:0]			C1C0[1:0]		CF45CF40[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CF45CF40[5:0]	Command Index These bits specify command format[45:40] (command index). [Examples] CMD6: SD_CMD[7:0] = 0x06 CMD18: SD_CMD[7:0] = 0x12 ACMD13: SD_CMD[7:0] = 0x4D	R/W
7:6	C1C0[1:0]	Command Type 0 0: CMD 0 1: ACMD 1 0: Setting prohibited	R/W
10:8	MD2MD0[2:0]	Mode/Response Type Some commands cannot be used in normal mode. For details, see section 55.5.14. Example of SD_CMD Register Setting to select mode/response type. 0 0 0: Normal mode The response type and the transfer mode are selected by SD_CMD[7:0], and the SD_CMD[15:11] setting is disabled. 0 0 1: Setting prohibited 0 1 0: Setting prohibited 0 1 1: Extended mode/No response 1 0 0: Extended mode/R1, R5, R6, or R7 response from the SD card 1 0 1: Extended mode/R1b response from the SD card 1 1 0: Extended mode/R2 response from the SD card 1 1 1: Extended mode/R3 or R4 response from the SD card	R/W
11	MD3	Data Mode (Command Type) 0: Command without data transfer (bc, bcr, ac) 1: Command with data transfer (adtc)	R/W
12	MD4	Write/Read Mode (enabled when the command with data is handled) 0: Write (SD/MMC host interface → SD card) 1: Read (SD/MMC host interface ← SD card)	R/W
13	MD5	Single/Multiple Block Transfer (enabled when the command with data is handled) 0: Single block transfer 1: Multi block transfer	R/W
15:14	MD7MD6[1:0]	Multiple Block Transfer Mode (enabled at multiple block transfer) 0 0: CMD12 is automatically issued at multiple block transfer. 0 1: CMD12 is not automatically issued at multiple block transfer. 1 0: Setting prohibited 1 1: Setting prohibited	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The SD_CMD register is used to select the command type and response type. The command sequence is started by writing to SD_CMD.

For details on the SD_CMD setting, refer to section 55.5.14. Example of SD_CMD Register Setting.

Do not write SD_CMD register when CBSY bit in SD_INFO2 register is 1.

55.3.2 Command Argument Register

Command arguments for SD cards are set in the SD command argument registers (SD_ARG). Set the command arguments before writing to SD_CMD.

Note that the argument of CMD12 within command sequences is 0x00000000 regardless of the setting of SD_ARG.

55.3.2.1 SD_ARG : Command Argument Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0010

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Set command format[39:8] (argument).	R/W

55.3.2.2 SD_ARG1 : Command Argument Register 1

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0018

Bit position:	31															15													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	Set command format[39:24] (argument).	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Mirror of the Upper 32 bits of Command Argument Register

55.3.3 SD_STOP : Data Stop Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HPIMODE	HPICMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0*1	0*1	0*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SEC	—	—	—	—	—	—	—	STP
Value after reset:	0	0	0	0	0	0	0	0*1	0	0	0	0	0	0	0	0*1

Bit	Symbol	Function	R/W
0	STP	Stop ● When STP is set to 1 during multiple block transfer, CMD12 is issued to halt the transfer through the SD/MMC host interface. However, if a command sequence is halted because of a communications error or timeout, CMD12 is not issued. Although continued buffer access is possible even after STP has been set to 1, the buffer access error bit (ERR5 or ERR4) in SD_INFO2 will be set accordingly. ● When STP has been set to 1 during transfer for single block write, the access end flag is set when SD_BUF becomes empty, and CMD12 is not issued. If SD_BUF does contain data, the access end flag is set on completion of reception of the busy state without CMD12 having been issued. ● When STP has been set to 1 during transfer for single block read, the access end flag is set immediately after setting of the STP bit and CMD12 is not issued. ● When STP is set to 1 during reception of the busy state after an R1b response, the access end flag is set on completion of reception of the busy state without CMD12 having been issued. ● When STP is set to 1 after a command sequence has been completed, CMD12 is not issued and the access end flag is not set. ● Set STP to 1 after the response end flag has been set. ● Set STP to 0 after the response end flag has been set.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	SEC	Block Count Enable*2 Set SEC to 1 at multiple block transfer. When SD_CMD is set as follows to start the command sequence while SEC is set to 1, CMD12 is automatically issued to stop multiple block transfer with the number of blocks which is set to SD_SECCNT. 1. CMD18 or CMD25 in normal mode (SD_CMD[10:8] = 000b) 2. SD_CMD[15:13] = 001b in extended mode (CMD12 is automatically issued, multiple block transfer) When the command sequence is halted because of a communications error or timeout, CMD12 is not automatically issued. 0: Disables SD_SECCNT setting value. 1: Enables SD_SECCNT setting value.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	HPICMD	HPI Command Issue When HPICMD is set to 1 while HPIMODE is 1, the HPI command (CMD12) is issued. This bit is cleared to 0 when reception of the response to CMD12 is completed. The timing with which this bit is set to 1 is as follows. ● After reception of the response to CMD12 that was issued by setting the STP bit to 1 has been completed during the CMD6/CMD38 or CMD25 sequence. ● After reception of the response to CMD24/CMD25 has been completed. After HPCMD is set to 1, do not write 0 to this bit while the CBSY bit in SD_INFO2 is 1. Do not set this bit to 1 when the CBSY bit in SD_INFO2 is 0.	R/W*3
17	HPIMODE	HPI Mode Enable 0: Disables HPI mode. 1: Enables HPI mode.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 2. Do not change the value of this bit when the CBSY bit in SD_INFO2 is set to 1.
Note 3. Only effective when 1 is written.

The SD_STOP register is used to enable or disable block counting at multiple block transfer, and to control the issuing of CMD12 within command sequences.

55.3.4 SD_SECCNT : Block Count Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0028

Bit position:	31																												0
Bit field:	CNT31CNT0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CNT31CNT0[31:0]	Number of Transfer Blocks*1 When 0x00000001 is set, the number of transfer blocks is 1. ⋮ When 0x0000FFFF is set, the number of transfer blocks is 65535. ⋮ When 0xFFFFFFFF is set, the number of transfer blocks is 4294967295. Do not set this register to 0x00000000 if multiple blocks are to be transferred.	R/W

Note 1. Do not change the value of these bits when the CBSY bit in SD_INFO2 is set to 1.

The SD_SECCNT register is used to specify the number of transfer blocks at multiple block transfer.

55.3.5 SD Card Response Registers

The SD card response registers (SD_RSP) hold the response from the SD card.

55.3.5.1 SD_RSP10 : SD Card Response 10 Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0030

Bit position:	63																												32
Bit field:	R71R40[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31																												0
Bit field:	R39R8[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:32	R71R40[31:0]	Hold the response from the SD card	R
31:0	R39R8[31:0]		

55.3.5.2 SD_RSP1 : SD Card Response 1 Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0038

Bit position:	31															15													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	R39R24[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	R39R24[15:0]	Hold the response from the SD card	R

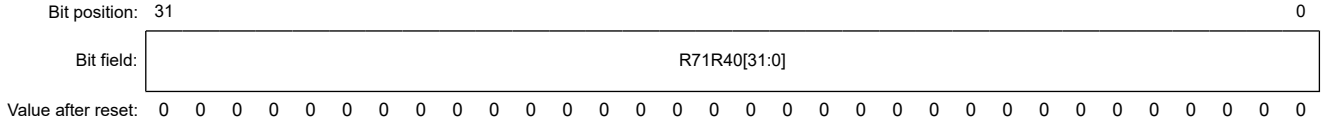
Bit	Symbol	Function	R/W
31:16	—	These bits are read as 0.	R

Mirror of the [31:16] bits of SD Card Response 10 Register

55.3.5.3 SD_RSP32 : SD Card Response 32 Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0040



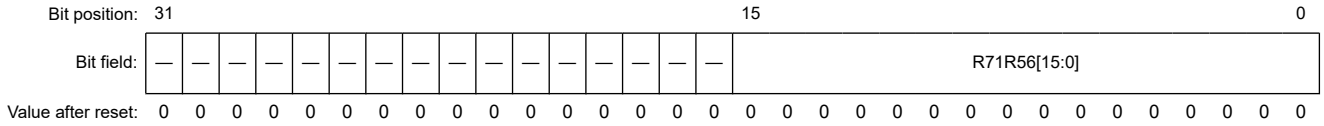
Bit	Symbol	Function	R/W
31:0	R71R40[31:0]	Hold the response from the SD card	R

Mirror of the Upper 32 bits of SD Card Response 10 Register

55.3.5.4 SD_RSP3 : SD Card Response 3 Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0048



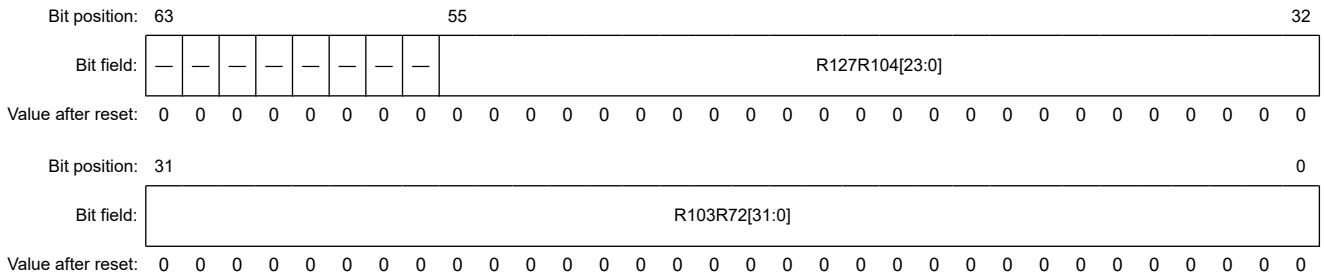
Bit	Symbol	Function	R/W
15:0	R71R56[15:0]	Hold the response from the SD card	R
31:16	—	These bits are read as 0.	R

Mirror of the [31:16] bits of SD_RSP32 Register

55.3.5.5 SD_RSP54 : SD Card Response 54 Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0050



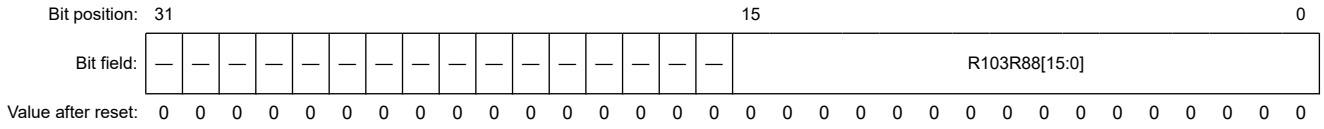
Bit	Symbol	Function	R/W
31:0	R103R72[31:0]	Hold the response from the SD card	R
55:32	R127R104[23:0]		

Bit	Symbol	Function	R/W
63:56	—	These bits are read as 0.	R

55.3.5.6 SD_RSP5 : SD Card Response 5 Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0058



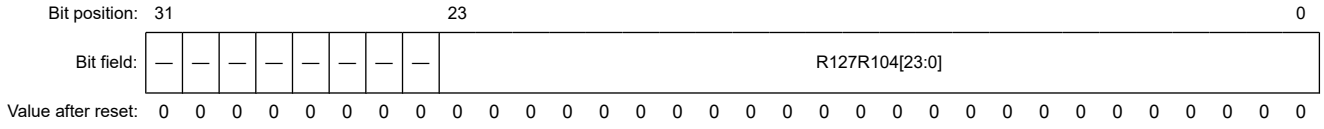
Bit	Symbol	Function	R/W
15:0	R103R88[15:0]	Hold the response from the SD card	R
31:16	—	These bits are read as 0.	R

Mirror of the [31:16] bits of SD_RSP54 Register

55.3.5.7 SD_RSP76 : SD Card Response 76 Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0060



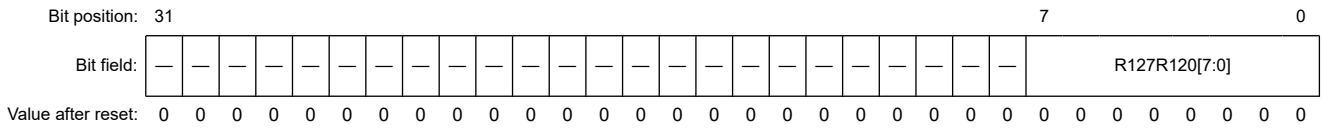
Bit	Symbol	Function	R/W
23:0	R127R104[23:0]	Hold the response from the SD card	R
31:24	—	These bits are read as 0.	R

Mirror of the Upper 32 bits of SD_RSP54 Register

55.3.5.8 SD_RSP7 : SD Card Response 7 Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0068



Bit	Symbol	Function	R/W
7:0	R127R120[7:0]	Hold the response from the SD card	R
31:8	—	These bits are read as 0.	R

Mirror of the [31:16] bits of SD_RSP76 Register

Table 55.4 lists the response types and corresponding SD_RSP registers.

Table 55.4 Response types and corresponding SD_RSP registers

Response types	SD_RSP registers
R1, R1b[39:8]	SD_RSP10 SD_RSP54[*1]
R2[127:8]	SD_RSP54 and SD_RSP10
R3[39:8]	SD_RSP10
R4[39:8]	SD_RSP10
R5[39:8]	SD_RSP10
R6[39:8]	SD_RSP10
R7[39:8]	SD_RSP10

Note 1. The response to CMD18 and to CMD25 is stored in both R[39:8] and R[103:72].
This makes it possible to confirm the response to CMD18 and CMD25 by reading R[103:72] even if the response to automatic CMD12 is stored in R[39:8].

55.3.6 SD_INFO1 : SD Card Interrupt Flag Register 1

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HPIRES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0[*2]

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	INFO10	INFO9	INFO8	INFO7	—	INFO5	INFO4	INFO3	INFO2	—	INFO0
Value after reset:	0	0	0	0	0	x	0	0	x	0	x	0	0	0[*2]	0	0[*2]

Bit	Symbol	Function	R/W
0	INFO0	Response End [Setting conditions] 1. When the reception of the response is completed 2. When the transmission of the command not requiring a response is completed 3. When receiving busy reception after R1b response 4. When reception of the response to CMD52 that was issued by setting the C52PUB bit to 1 is completed in the case of transfer for multiple block read 5. When reception of the response to CMD52 that was issued by setting the C52PUB bit to 1 is completed in the case of transfer for multiple block write In addition to the above conditions, this bit is set when a command sequence is halted because of a communications error or timeout. [Clearing condition] When 0 is written to INFO0 When issuing a command without data, the command sequence ends when the response end is set to 1.	R/W[*1]
1	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
2	INFO2	Access End [Setting conditions] 1. When read access to the buffer is completed in the case of transfer for single block read 2. When read access to the buffer for the last block of data is completed in the case of transfer for multiple block read 3. When read access to the buffer and reception of the response to CMD12 are completed in the case of transfer for multiple block read with automatic issuing of CMD12 4. When reception of the busy state after reception of the CRC status is completed in the case of transfer for single block write 5. When reception of the busy state after reception of the CRC status of the last block of data is completed in the case of transfer for multiple block write 6. When reception of the response busy state for CMD12 is completed in the case of transfer for multiple block write with automatic issuing of CMD12 7. When reception of the response to CMD12 that was issued by setting the STP bit to 1 is completed in the case of transfer for multiple block read 8. When reception of the response busy state for CMD12 that was issued by setting the STP bit to 1 is completed in the case of transfer for multiple block write 9. When reception of the response to CMD52 that was issued by setting the IOABT bit to 1 is completed in the case of transfer for multiple block read 10. When reception of the response to CMD52 that was issued by setting the IOABT bit to 1 is completed in the case of transfer for multiple block write In addition to the above conditions, this bit is set when a command sequence is halted because of a communications error or timeout. [Clearing condition] When 0 is written to INFO2 When the access end bit is set to 1, the command sequence is terminated.	R/W*1
3	INFO3	ISDCD Card Removal [Setting condition] After a change in ISDCD from 0 to 1, Mcycle has elapsed with ISDCD held at 1. [Clearing condition] When 0 is written to INFO3 Mcycle is set by bits 3 to 0 in SD_OPTION.	R/W*1
4	INFO4	ISDCD Card Insertion [Setting condition] After a change in ISDCD from 1 to 0, Mcycle has elapsed with ISDCD held at 0. [Clearing condition] When 0 is written to INFO4 Mcycle is set by bits 3 to 0 in SD_OPTION.	R/W*1
5	INFO5	Indicates the ISDCD state. Mcycle is set by bits 3 to 0 in SD_OPTION. 0: Indicates that Mcycle has elapsed with ISDCD held at 1. 1: Indicates that Mcycle has elapsed with ISDCD held at 0.	R
6	—	This bit is read as 0. The write value should be 0.	R/W
7	INFO7	Write Protect Indicates the ISDWP state. 0: ISDWP is set to 1. 1: ISDWP is set to 0.	R
8	INFO8	SDDAT3 Card Removal [Setting condition] After a change in SDDAT3 from 1 to 0, two cycles of 2 PCLKAM have elapsed with SDDAT3 held at 0. [Clearing condition] When 0 is written to INFO8	R/W*1
9	INFO9	SDDAT3 Card Insertion [Setting condition] After a change in SDDAT3 from 0 to 1, two cycles of 2 PCLKAM have elapsed with SDDAT3 held at 1. [Clearing condition] When 0 is written to INFO9	R/W*1
10	INFO10	Indicates the SDDAT3 state. 0: SDDAT3 is set to 0. 1: SDDAT3 is set to 1.	R
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	HPIRES	Response Reception Completion [Setting condition] When reception of the response to CMD12 that was issued by setting the STP bit to 1 is completed during the CMD6/CMD38 or CMD25 sequence in HPI mode. [Clearing condition] When 0 is written to HPIRES	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only effective when 0 is written.
Note 2. The value is initialized by a reset and also in the case of a reset by the SDRST bit in SOFT_RST.

The SD_INFO1 register indicates the response end and access end in the command sequence. This register also indicates the card detect/write protect state.

For CMD12 and CMD52 (SDIO abort) at multiple block transfer, INFO0 is not set but only INFO2 is set.

Even if the command sequence is halted because of a communications error or timeout, INFO0 or INFO2 is set.

INFO10, INFO9, and INFO8 change depending on the SDDAT3 state after a reset is released and continue to change in 4-bit transfer mode.

To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

55.3.7 SD_INFO2 : SD Card Interrupt Flag Register 2

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0078

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ILA	CBSY	SCLK DIVEN	—	—	—	BWE	BRE	DAT0	ERR6	ERR5	ERR4	ERR3	ERR2	ERR1	ERR0
Value after reset:	0*2	0*2	1*2	0	0	0	0*2	0*2	x	0*2	0*2	0*2	0*2	0*2	0*2	0*2

Bit	Symbol	Function	R/W
0	ERR0	CMD Error [Setting conditions] 1. The command index of the transmitted command differing from the command index of the received response 2. The command index of a command issued within a command sequence*5 differing from the command index of the received response [Clearing condition] When 0 is written to ERR0 The command sequence is halted by the CMD error.*4	R/W*1
1	ERR1	CRC Error [Setting conditions] 1. When an error occurs in the CRC status (i.e. the received CRC status was not 010b) 2. When a CRC error occurs in the read data 3. When a CRC error occurs in the response 4. A CRC error in the response to a command issued within a command sequence*5 [Clearing condition] When 0 is written to ERR1 The command sequence is halted by the CRC error.*4	R/W*1

Bit	Symbol	Function	R/W
2	ERR2	END Error [Setting conditions] 1. When an error occurs in the response length (and the end bit has not been detected) 2. When an error occurs in the read data length (and the end bit has not been detected among the valid bits) 3. When an error occurs in the CRC status length (and the end bit has not been detected) 4. An error in the length of a response to a command issued within a command sequence*5 (i.e. the end bit has not been detected) [Clearing condition] When 0 is written to ERR2 The command sequence is halted by the End error.*4	R/W*1
3	ERR3	Data Timeout (except response timeout) [Setting conditions] 1. After the reception of the R1b response, the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle. 2. After the reception of the CRC status, the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle. 3. After write data, the CRC status is not received even after Ncycle has elapsed. 4. After the read command, read data is not received even after Ncycle has elapsed. 5. After CMD12 has been issued within a command sequence, the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle. 6. After the reception of read data, read data for the next block are not received even after Ncycle has elapsed. 7. After release of the read wait state, read data for the next block are not received even after Ncycle has elapsed. [Clearing condition] When 0 is written to ERR3 Ncycle is set by bits 7 to 4 in SD_OPTION. The command sequence is halted by the data timeout.	R/W*1
4	ERR4	SD_BUF Illegal Write Access [Setting conditions] 1. When data is written to SD_BUF0 while it is not in the data read/write command state 2. When data is written to SD_BUF0 while SD_BUF is full 3. When data is written to SD_BUF0 while an error occurs in the CRC status or CRC status length 4. When data is written to SD_BUF0 while the interface remains in a busy state for at least Ncycle after the CRC status [Clearing condition] When 0 is written to ERR4 Ncycle is set by bits 7 to 4 in SD_OPTION.	R/W*1
5	ERR5	SD_BUF Illegal Read Access [Setting conditions] 1. When SD_BUF is empty while SD_BUF0 is read 2. When data with a CRC error or END error is read from SD_BUF0 [Clearing condition] When 0 is written to ERR5	R/W*1
6	ERR6	Response Timeout [Setting condition] When a response is not received even after 640 cycles of SDCLK have elapsed (including a response to a command issued within a command sequence*5) [Clearing condition] When 0 is written to ERR6 The command sequence is halted by a response timeout.*4	R/W*1
7	DAT0	SDDAT0 Indicates the SDDAT0 state If the data timeout (ERR3) is set but the response timeout (ERR6) is not set after the Erase command has been issued, the end of the Erase sequence (DAT0 = 1) is confirmed by polling DAT0. If a communications error or timeout occurs during a write sequence, the DAT0 bit may retain the value 0. While the SD clock (SDCLK) is stopped, the DAT0 bit retains the value before the clock is stopped. 0: SDDAT0 is set to 0. 1: SDDAT0 is set to 1.	R

Bit	Symbol	Function	R/W
8	BRE	SD_BUF Read Enable [Setting conditions] 1. When data set in SD_SIZE is stored in SD_BUF at single block transfer 2. When data set in SD_SIZE is stored in either bank 1 or bank 2 of SD_BUF at multiple block transfer [Clearing conditions] 1. When 0 is written to BRE 2. Reading of a block of data from SD_BUF by DMA transfer When data is read from SD_BUF0 by the CPU, clear BRE and then read the amount of data specified by SD_SIZE.*3 Even if a CRC error or an END error occurs while block data is read, data is stored in SD_BUF and BRE is set. 0: Data cannot be read from SD_BUF0. 1: Data can be read from SD_BUF0.	R/W*1
9	BWE	SD_BUF Write Enable [Setting conditions] 1. When SD_BUF is empty at single block transfer 2. When either bank 1 or bank 2 of SD_BUF is empty at multiple block transfer [Clearing conditions] 1. When 0 is written to BWE 2. Writing of a block of data to SD_BUF by DMA transfer When data is written to SD_BUF0 by the CPU, clear BWE and then write the amount of data specified by SD_SIZE.*3 0: Data cannot be written in SD_BUF0. 1: Data can be written in SD_BUF0.	R/W*1
10	—	This bit is read as 0. The write value should be 0.	R/W
11	—	This bit is read as 0. The write value should be 1.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	SCLKDIVEN	When a command sequence is started by writing to SD_CMD, the CBSY bit is set to 1 and, at the same time, the SCLKDIVEN bit is set to 0. The SCLKDIVEN bit is set to 1 after 8 cycles of SDCLK have elapsed after setting of the CBSY bit to 0 due to completion of the command sequence. 0: The SD bus (CMD, DAT) is busy. Do not attempt to write to the SD_CLK_CTRL register. 1: The SD bus (CMD, DAT) is not busy.	R
14	CBSY	Command Type Register Busy 0: A command sequence has been completed. 1: A command sequence is being executed.	R
15	ILA	Illegal Access Error [Setting conditions] 1. Writing of data to SD_CMD within a command sequence (CBSY = 1) 2. When SD_CMD[11] = 1 (command with data transfer) and SD_CMD[7:0] = 0x0C (CMD12) are set in SD_CMD [Clearing condition] When 0 is written to ILA	R/W*1
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only effective when 0 is written.
Note 2. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 3. When the WMODE bit in HOST_MODE is 0, the single byte from the fraction of a full 16-bit unit is regarded as excess data due to an odd value for the number of bytes setting in SD_SIZE. When the WMODE bit in HOST_MODE is 1, the single byte or three bytes from the fraction of a full 64-bit unit are regarded as excess data due to an odd value for the number of bytes setting in SD_SIZE, or the two bytes from the fraction of a full 64-bit unit are regarded as excess data due if the value for the number of bytes setting in SD_SIZE is even but is not on a four-byte boundary.
Note 4. After the C52PUB bit in SDIO_MODE has been set to 1, if a communications error or timeout for response occurs in response to the CMD52 that is issued, since the command sequence has not been completed, complete the sequence with error processing as in usage examples in Figure 55.18 under section 55.5.8. IO_RW_EXTENDED (CMD53/Multiple Block Read) or in Figure 55.21 under section 55.5.9. IO_RW_EXTENDED (CMD53/Multiple Block Write).
Note 5. "Command issued within a command sequence" refers to CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

The SD_INFO2 register indicates the access status of the SD buffer (SD_BUF) and SD card. To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

55.3.8 SD_INFO1_MASK : SD_INFO1 Interrupt Mask Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IMASK16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	IMASK9	IMASK8	—	—	—	IMASK4	IMASK3	IMASK2	—	IMASK0
Value after reset:	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	1

Bit	Symbol	Function	R/W
0	IMASK0	INFO0 interrupt masked	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	IMASK2	INFO2 interrupt masked	R/W
3	IMASK3	INFO3 interrupt masked	R/W
4	IMASK4	INFO4 interrupt masked	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	IMASK8	INFO8 interrupt masked	R/W
9	IMASK9	INFO9 interrupt masked	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	IMASK16	HPIRES interrupt masked	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The SD_INFO1_MASK register is used to enable or disable the SD_INFO1 interrupt. When 0 is set in SD_INFO1_MASK while the corresponding flag in SD_INFO1 is set, an interrupt occurs.

55.3.9 SD_INFO2_MASK : SD_INFO2 Interrupt Mask Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0088

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IMASK	—	—	—	—	—	BMASK1	BMASK0	—	EMASK6	EMASK5	EMASK4	EMASK3	EMASK2	EMASK1	EMASK0
Value after reset:	1	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	EMASK0	ERR0 interrupt masked	R/W
1	EMASK1	ERR1 interrupt masked	R/W
2	EMASK2	ERR2 interrupt masked	R/W
3	EMASK3	ERR3 interrupt masked	R/W
4	EMASK4	ERR4 interrupt masked	R/W

Bit	Symbol	Function	R/W
5	EMASK5	ERR5 interrupt masked	R/W
6	EMASK6	ERR6 interrupt masked	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	BMASK0	BRE interrupt masked	R/W
9	BMASK1	BWE interrupt masked	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	—	This bit is read as 1. The write value should be 1.	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	IMASK	ILA interrupt masked	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The SD_INFO2_MASK register is used to enable or disable the SD_INFO2 interrupt. When 0 is set in SD_INFO2_MASK while the corresponding flag in SD_INFO2 is set, an interrupt occurs.

55.3.10 SD_CLK_CTRL : SD Clock Control Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SDCLKOFFEN	SCLKEN	DIV7DIV0[7:0]							
Value after reset:	0	0	0	0	0	0*1	0	0*1	0	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DIV7DIV0[7:0]	SD Clock (SDCLK) In addition, in the case of data transfer in DDR mode (DDR bit in SDIF_MODE = 1), do not set DIV[7:0] to 0xFF. 0x80: PCLKAM/512 0x40: PCLKAM/256 0x20: PCLKAM/128 0x10: PCLKAM/64 0x08: PCLKAM/32 0x04: PCLKAM/16 0x02: PCLKAM/8 0x01: PCLKAM/4 0x00: PCLKAM/2 0xFF: PCLKAM Others: Setting prohibited	R/W*2
8	SCLKEN	SD Clock (SDCLK) Output Control Enable 0: SD clock (SDCLK) output is disabled. The SDCLK signal is fixed 0. 1: SD clock (SDCLK) output is enabled.	R/W

Bit	Symbol	Function	R/W
9	SDCLKOFFEN	SD Clock (SDCLK) Output Automatic Control Enable This function of automatic control for SD clock (SDCLK) output causes SDCLK output only within a command sequence. The timing with which SDCLK output starts and stops is as follows. SDCLK output starts after writing to SD_CMD. SDCLK output stops when 8 cycles of SDCLK have elapsed after the end of the command sequence. In addition, SDCLK is fixed to 0 while SCLKEN of SD_CLK_CTRL is 0, regardless of the value of this bit. 0: Automatic control for SD clock (SDCLK) output is disabled. 1: Automatic control for SD clock (SDCLK) output is enabled.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 2. Writing to SD_CLK_CTRL is impossible when the CBSY bit in SD_INFO2 is 1.

The SD_CLK_CTRL register is used to control the SD clock (SDCLK) output and to set the frequency. Set SCLKEN to 1 before writing to SD_CMD to issue a command. Do not write to SD_CLK_CTRL while the SCLKDIVEN bit in SD_INFO2 is set to 0.

CAUTION

Notes on when setting the SD clock (SDCLK) to PCLKAM (DIV[7:0] = 0xFF)

When changing the setting of bits DIV[7:0] to 0xFF, or from 0xFF to other setting, perform the following processing before writing to SD_CMD.

1. Set the SCLKEN bit to 0 by writing to SD_CLK_CTRL. (Do not change the setting of bits other than SCLKEN at this time.)
2. Change the setting of bits DIV[7:0] by writing to SD_CLK_CTRL. (Do not change the setting of bits other than DIV[7:0] at this time. The SCLKEN bit should retain the value 0.)
3. Set the SCLKEN bit to 1 by writing to SD_CLK_CTRL. (Do not change the setting of bits other than SCLKEN at this time.)

Also when changing the setting of bits DIV[7:0] to 0xFF after having set the SDRST bit in SOFT_RST to 0 and then changed it to 1, perform this processing before writing to SD_CMD.

55.3.11 SD_SIZE : Transfer Data Length Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0098

Bit position:	31																					9								0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LEN9LEN0[9:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	LEN9LEN0[9:0]	Transfer Data Size*1 These bits specify a size between 1 and 512 bytes for single block transfer. In cases of multiple block transfer with automatic issuing of CMD12 (CMD18 and CMD25), the only specifiable transfer data size is 512 bytes. Furthermore, in cases of multiple block transfer without automatic issuing of CMD12, as well as 512 bytes, 32, 64, 128, and 256 bytes are specifiable. However, in the reading of 32, 64, 128, and 256 bytes for the transfer of multiple blocks, this is restricted to multiple block transfer by CMD53. Additionally, if a command accompanies data transfer, do not set these bits to 0. Do not specify a data size larger than 512 bytes.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not change the values of these bits when the CBSY bit in SD_INFO2 is 1.

The SD_SIZE register is used to specify the transfer data size.

55.3.12 SD_OPTION : SD Card Access Control Option Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x00A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WIDTH	—	WIDTH8	—	—	—	EXTOP	TOUTMASK	TOP27TOP24[3:0]				CTOP24CTOP21[3:0]			
Value after reset:	0*1	1	0*1	0	0	0	0*1	0*1	1*1	1*1	1*1	0*1	1*1	1*1	1*1	0*1

Bit	Symbol	Function	R/W
3:0	CTOP24CTOP21[3:0]	Card Detect Time Counter 0x0: 2^{13} SDCLK 0x1: 2^{14} SDCLK ⋮ 0xD: 2^{26} SDCLK 0xE: 2^{27} SDCLK 0xF: Setting prohibited	R/W
7:4	TOP27TOP24[3:0]	Timeout Counter*2 0x0: 2^{10} SDCLK 0x1: 2^{11} SDCLK ⋮ 0xD: 2^{23} SDCLK 0xE: 2^{24} SDCLK 0xF: Setting prohibited	R/W
8	TOUTMASK	Timeout Mask If a timeout occurs while it is disabled, perform a software reset to terminate a command sequence. 0: Enables timeout 1: Disables timeout (The ERR6 and ERR3 bits in SD_INFO2 and the E6 to E0 bits in SD_ERR_STS2 are not set.)	R/W
9	EXTOP	Timeout Mode Select 0: The TOP27TOP24[3:0] bits specify the timeout count from 2^{10} SDCLK to 2^{24} SDCLK. 1: The TOP27TOP24[3:0] bits specify the timeout count from 2^{11} SDCLK to 2^{25} SDCLK.	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
13	WIDTH8	Bus width*2 See the description of the WIDTH bit.	R/W
14	—	This bit is read as 1. The write value should be 1.	R/W
15	WIDTH	Bus width*2 {WIDTH, WIDTH8} = 01: 8-bit width {WIDTH, WIDTH8} = 00: 4-bit width {WIDTH, WIDTH8} = 10, 11: 1-bit width In the case of data transfer in DDR mode (DDR bit in SDIF_MODE = 1), do not set this bit to 1. In the case of writing of one-byte block, 8-bit width cannot be specified for the bus width. Change the bus width to 4 bits or 1 bit before writing one-byte block.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 2. Do not change the values of these bits when the CBSY bit in SD_INFO2 is 1.

The SD_OPTION register is used to set the bus width and timeout counter.

55.3.13 SD_ERR_STS1 : SD Error Status Register 1

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x00B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	E14E12[2:0]			E11	E10	E9	E8	—	—	E5	E4	E3	E2	E1	E0
Value after reset:	x	0*1	1*1	0*1	0*1	0*1	0*1	0*1	0	0	0*1	0*1	0*1	0*1	0*1	0*1

Bit	Symbol	Function	R/W
0	E0	Set to 1 when an error occurs in the command index of a response (other than a response to a command issued within a command sequence*2).	R
1	E1	Set to 1 when an error occurs in the command index of the response to a command issued within a command sequence*2. In cases where CMD12 is issued by setting a command index in SD_CMD, this is Indicated in E0.	R
2	E2	Set to 1 when an error occurs in the response length (other than a response to a command issued within a command sequence*2).	R
3	E3	Set to 1 when an error occurs in the response length to a command issued within a command sequence*2. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E2.	R
4	E4	Set to 1 when an error occurs in the read data length (and the end bit has not been detected among the valid bits).	R
5	E5	Set to 1 when an error occurs in the CRC status length (and the end bit has not been detected).	R
7:6	—	These bits are read as 0.	R
8	E8	Set to 1 when a CRC error occurs in a response (other than a response to a command issued within a command sequence*2).	R
9	E9	Set to 1 when a CRC error occurs in the response to a command issued within a command sequence*2. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E8.	R
10	E10	Set to 1 when a CRC error occurs in the read data.	R
11	E11	Set to 1 when an error occurs in the CRC status.	R
14:12	E14E12[2:0]	These bits hold the CRC status. (normal: 010b)	R
31:15	—	The read values are undefined.	R

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.

Note 2. "Command issued within a command sequence" refers to CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

The SD_ERR_STS1 register indicates the CRC status, CRC error, End error, and CMD error.

55.3.14 SD_ERR_STS2 : SD Error Status Register 2

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x00B8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	E6	E5	E4	E3	E2	E1	E0
Value after reset:	0	0	0	0	0	0	0	0	0	0*1	0*1	0*1	0*1	0*1	0*1	0*1

Bit	Symbol	Function	R/W
0	E0	Set to 1 when the response (other than a response to a command issued within a command sequence*2) is not received even after 640 cycles of SDCLK have elapsed.	R
1	E1	Set to 1 when the response to a command issued within a command sequence*2 is not received even after 640 cycles of SDCLK have elapsed. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E0.	R
2	E2	Set to 1 when the interface remains in a busy state for at least Ncycle after R1b response.	R
3	E3	Set to 1 when the interface remains in a busy state for at least Ncycle after CMD12 has been issued within a command sequence. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E2.	R
4	E4	Set to 1 when read data is not received even after Ncycle has elapsed after the command has been read. Set to 1 when read data for the next block are not received even after Ncycle has elapsed after the reception of read data. Set to 1 when read data for the next block are not received even after Ncycle has elapsed after release of the read wait state.	R
5	E5	Set to 1 when the CRC status is not received even after Ncycle has elapsed after data writing.	R
6	E6	Set to 1 when the interface remains in a busy state for at least Ncycle after the CRC status.	R
31:7	—	These bits are read as 0.	R

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.

Note 2. "Command issued within a command sequence" refers to CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

The SD_ERR_STS2 register indicates the timeout state. Ncycle is set by bits 7 to 4 in SD_OPTION.

55.3.15 SD_BUF0 : SD Buffer Read/Write Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x00C0

Bit position:	63	32
Bit field:		
Value after reset:	x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x	

Bit position:	31	0
Bit field:		
Value after reset:	x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x	

Bit	Symbol	Function	R/W
63:0	n/a	When writing to the SD card, the write data is written to this register. When reading from the SD card, the read data is read from this register. This register is internally connected to two 512-byte buffers (SD_BUF). When both buffers are not empty at multiple block read, suspend data reception by stopping the SD clock. When either buffer becomes empty, restart data reception by starting supply of the SD clock.	R/W

Note: When using the DMAC, the bus width should be fixed at 64 bits.

55.3.16 SDIO_MODE : SDIO Mode Control Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x00D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	C52PUB	IOABT	—	—	—	—	—	RWREQ	—	IOMOD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IOMOD	SDIO Mode[*1] 0: Disables the SD/MMC host interface to receive SDIO interrupts from the SDIO card 1: Enables the SD/MMC host interface to receive SDIO interrupts from the SDIO card	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	RWREQ	Read Wait Request When RWREQ is set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks. [Releasing the read wait state] 1. The read wait state is released when RWREQ is cleared to 0 in the read wait state. 2. When IOABT is set to 1 in the read wait state, RWREQ is automatically cleared to 0 after CMD52 has been issued, and then the read wait state is released. 3. When C52PUB and RWREQ are set to 1 simultaneously in the CMD53 (multiple block) read sequence, the read wait state is not automatically released. Therefore, after the CMD52 response is received, clear RWREQ. (Be sure to set RWREQ and C52PUB simultaneously.) When RWREQ is set to 1 while the last block in the CMD53 (multiple block) read sequence is transferred, the read wait state is not entered and RWREQ is automatically cleared to 0 by setting access end. Set RWREQ to 1 after the response end flag has been set.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	IOABT	SDIO Abort ● When IOABT is set to 1 in the CMD53 (multiple block) sequence, the CMD53 sequence is halted and CMD52 is issued. However, if a command sequence is halted because of a communications error or timeout, CMD52 is not issued. Although continued buffer access is possible even after IOABT has been set to 1, the buffer access error bit (ERR5 or ERR4) in SD_INFO2 will be set accordingly. Set SD_ARG before setting IOABT to 1. ● When IOABT has been set to 1 during transfer for single block write, the access end flag is set when SD_BUF becomes empty, and CMD52 is not issued. If SD_BUF does contain data, the access end flag is set on completion of reception of the busy state without CMD52 having been issued. ● When IOABT has been set to 1 during transfer for single block read, the access end flag is set immediately after setting of IOABT and CMD52 is not issued. ● When IOABT is set to 1 during reception of the busy state after an R1b response, the access end flag is set on completion of reception of the busy state without CMD52 having been issued. ● When IOABT is set to 1 after a command sequence has been completed, CMD52 is not issued and the access end flag is not set. ● Set IOABT to 1 after the response end flag has been set. Set IOABT to 0 after the access end flag has been set.	R/W
9	C52PUB	SDIO None Abort ● When C52PUB is set to 1 in the CMD53 (multiple block) write sequence, CMD52 is automatically issued between blocks if SD_BUF becomes empty. C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag has been set to 1. ● When C52PUB and RWREQ are set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks and CMD52 is automatically issued. C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag has been set to 1. ● If C52PUB is set to 1 in the CMD53 (multiple block) read sequence, be sure to set RWREQ to 1 as well as C52PUB. ● Set SD_ARG before setting C52PUB to 1. ● Set C52PUB to 1 after the response end flag has been set.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not change the value of this bit when the CBSY bit in SD_INFO2 is set to 1.

The SDIO_MODE register controls the CMD52 issuance and the read wait state at multiple block transfer, and the reception of SDIO interrupt. C52PUB and IOABT should not be set to 1 simultaneously.

55.3.17 SDIO_INFO1 : SDIO Interrupt Flag Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x00D8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EXWT	EXPUB52	—	—	—	—	—	—	—	—	—	—	—	—	—	IOIRQ
Value after reset:	0*2	0*2	0	0	0	0	0	0	0	0	0	0	0	0*2	0*2	0*2

Bit	Symbol	Function	R/W
0	IOIRQ	[Setting condition] When an SDIO interrupt from the SDIO card is received while IOMOD in SDIO_MODE is set to 1. [Clearing condition] When 0 is written to IOIRQ*3	R/W*1
2:1	—	The read values are undefined. The write value should be 1.	R/W
13:3	—	These bits are read as 0. The write value should be 0.	R/W
14	EXPUB52	[Setting conditions] 1. While the last block in the CMD53 (multiple block) sequence is transferred, C52PUB in SDIO_MODE is set to 1. 2. While C52PUB is set to 1 in the CMD53 (multiple block) write sequence, the last block is transferred. [Clearing condition] When 0 is written to EXPUB52	R/W*1
15	EXWT	[Setting condition] While the last block in the CMD53 (multiple block) read sequence is transferred, RWREQ in SDIO_MODE is set to 1. [Clearing condition] When 0 is written to EXWT	R/W*1
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only effective when 0 is written.
Note 2. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 3. Before clearing this bit, access the SDIO card to negate the SDIO interrupt signal from the SDIO card. If the interrupt signal is not negated, this bit may be set again.

The SDIO_INFO1 register indicates the status regarding to the SDIO card access. To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

55.3.18 SDIO_INFO1_MASK : SDIO_INFO1 Interrupt Mask Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x00E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MEXWT	MEXPUB52	—	—	—	—	—	—	—	—	—	—	—	—	—	IOMSK
Value after reset:	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	IOMSK	IOIRQ interrupt masked	R/W
2:1	—	These bits are read as 1. The write value should be 1.	R/W
13:3	—	These bits are read as 0. The write value should be 0.	R/W
14	MEXPUB52	EXPUB52 interrupt masked	R/W
15	MEXWT	EXWT interrupt masked	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The SDIO_INFO1_MASK register enables or disables the SD_INFO1 interrupt. When 0 is set in SDIO_INFO1_MASK while the corresponding flag in SD_INFO1 is set, an interrupt occurs.

55.3.19 CC_EXT_MODE : DMA Mode Enable Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0360

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMASDRW	—
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	DMASDRW	SD_BUF Read/Write DMA Transfer*1 0: The SD_BUF read/write DMA transfer is disabled. 1: The SD_BUF read/write DMA transfer is enabled.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	—	This bit is read as 1. The write value should be 1.	R/W
11:5	—	These bits are read as 0. The write value should be 0.	R/W
12	—	This bit is read as 1. The write value should be 1.	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not change the value of this bit when the CBSY bit in SD_INFO2 is set to 1.

The CC_EXT_MODE register enables the DMA transfer.

55.3.20 SOFT_RST : Software Reset Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0380

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	SDRST	Software Reset of SD Interface Unit 0: Reset 1: Reset released	R/W
2:1	—	These bits are read as 1. The write value should be 1.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The SOFT_RST register sets a software reset. Also use this register to check that release from the reset state has been completed before attempting to use the SD/MMC host interface and before attempting access to the other registers.

55.3.21 VERSION : Version Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0388

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	UR3UR0[3:0]				IP7IP0[7:0]							
Value after reset:	1	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
7:0	IP7IP0[7:0]	Version of introductory IP	R
11:8	UR3UR0[3:0]	Version of Renesas' IP	R
13:12	—	These bits are read as 0.	R
15:14	—	These bits are read as 1.	R
31:16	—	These bits are read as 0.	R

The VERSION register indicates the version of the SD/MMC host interface.

55.3.22 HOST_MODE : Host Interface Mode Setting Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0390

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BUSWIDTH	—	—	—	—	—	—	ENDIAN	WMODE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WMODE	Width for Access to SD_BUF*1 *2 Read or write access to SD_BUF0 can be performed with the specified width for access. 0: 64-bit access 1: 16-bit or 32-bit access	R/W
1	ENDIAN	SD_BUF0 data swap	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	BUSWIDTH	Width for Access to SD_BUF*1 *2 Read or write access to SD_BUF0 can be performed with the specified width for access. This bit is enabled while the WMODE bit is set to 1. 0: 16-bit access 1: 32-bit access	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not change the value of this bit when the CBSY bit in SD_INFO2 is set to 1.
Note 2. When using the built-in DMAC of this module, fix the bus width to 64 bits.

The HOST_MODE register selects the width for access to the data bus.

55.3.23 SDIF_MODE : SD Interface Mode Setting Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0398

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	NOCHKCR	—	—	—	—	—	—	—	DDR
Value after reset:	0	0	0	0	0	0	0*1	0*1	0	0	0	0	0	0	0	0*1

Bit	Symbol	Function	R/W
0	DDR	DDR Mode Select*2 Set this bit to 0 when the SD clock division ratio is specified as 1:1 (bits DIV[7:0] in SD_CLK_CTRL are set to 0xFF). 0: Normal mode (default, high speed, or SDR) 1: DDR mode	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	NOCHKCR	CRC Check Mask (test command for MMC supported) Enables or disables checking of the CRC16 and CRC status. 0: Enables the CRC check 1: Disables the CRC check (the CRC16 value is ignored at read, and the CRC status is not detected at write)	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.
Note 2. Do not change the value of this bit when the CBSY bit in SD_INFO2 is set to 1.

The SDIF_MODE register specifies DDR mode.

55.3.24 SD_STATUS : SD Status Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x03C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SD_IOVS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SD_RST	SD_PWEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	SD_PWEN	Controls the output value on the SDm_PWEN pin 0: The output value on the SDm_PWEN pin is 0. 1: The output value on the SDm_PWEN pin is 1.	R/W
1	SD_RST	Controls the output value on the SD0_RST# pin*1 0: The output value on the SD0_RST# pin is 0. 1: The output value on the SD0_RST# pin is 1.	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	SD_IOVS	Controls the output value on the SDm_IOVS pin 0: The output value on the SDm_IOVS pin is 0. 1: The output value on the SDm_IOVS pin is 1.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Channel 0 only. Reserved for channel 1.

The effective bit of SD_STATUS register controls the output value on the SD0_RST#, SDm_PWEN, and SDm_IOVS pins.

55.3.25 DM_CM_DTRAN_MODE : DMAC Transfer Mode Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0820

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CH_NUM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	BUS_WIDTH[1:0]		—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
5:4	BUS_WIDTH[1:0]	Bus width selector 1 1: 64-bit Others: Setting prohibited	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CH_NUM[1:0]	DMAC channel selector 0 0: SD downstream 0 1: SD upstream Others: Setting prohibited	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The DM_CM_DTRAN_MODE register sets the operation mode of the DMAC of this module.

55.3.26 DM_CM_DTRAN_CTRL : DMAC Transfer Control Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0828

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DM_START
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DM_START	DMAC Start Writing 1 to this bit starts DMAC operation. This bit is automatically cleared when DMA transfer is started.	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The DM_CM_DTRAN_CTRL register controls DMAC operation of the module.

55.3.27 DM_CM_RST : DMAC Reset Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0830

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DTRANRST1	DTRANRST0	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 1. The write value should be 1.	R/W
8	DTRANRST0	Software reset of DMAC channel 0 of the module 0: Reset 1: Reset released	R/W
9	DTRANRST1	Software reset of DMAC channel 1 of the module 0: Reset 1: Reset released	R/W
31:10	—	These bits are read as 1. The write value should be 1.	R/W

Note: Make sure there is no communication to the SD/MMC device before applying a software reset using this register.

55.3.28 DM_CM_INFO1 : DMAC Interrupt Register 1

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0840

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	DTRANEND1	—	—	—	DTRANEND0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0	0	0*2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	DTRANEND0	DMAC Channel 0 Transfer End [Setting conditions] 1. When transfer of DMAC channel 0 is completed 2. When an error occurs on DMAC channel 0 [Clearing condition] When 0 is written to DTRANEND0	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	DTRANEND1	DMAC Channel 1 Transfer End [Setting conditions] 1. When transfer of DMAC channel 1 is completed 2. When an error occurs on DMAC channel 1 [Clearing condition] When 0 is written to DTRANEND1	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value is applied at a reset and when the DTRANRST1 bit in DM_CM_RST is 0.
Note 2. The initial value is applied at a reset and when the DTRANRST0 bit in DM_CM_RST is 0.

The DM_CM_INFO1 register indicates the status of the DMAC of the module and the sequencer. To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

55.3.29 DM_CM_INFO1_MASK : DM_CM_INFO1 Interrupt Mask Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0848

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	DTRANEND1_MASK	—	—	—	DTRANEND0_MASK
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 1. The write value should be 1.	R/W
16	DTRANEND0_MASK	DTRANEND0 interrupt masked	R/W
19:17	—	These bits are read as 1. The write value should be 1.	R/W
20	DTRANEND1_MASK	DTRANEND1 interrupt masked	R/W
31:21	—	These bits are read as 1. The write value should be 1.	R/W

The DM_CM_INFO1_MASK register enables or disables the DM_CM_INFO1 interrupt. When 0 is set in DM_CM_INFO1_MASK while the corresponding flag in DM_CM_INFO1 is set, an interrupt occurs.

55.3.30 DM_CM_INFO2 : DMAC Interrupt Register 2

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0850

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTRANERR1	DTRANERR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*1	0*2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	DTRANERR0	DMAC Channel 0 Error [Setting condition] When an error occurs on DMAC channel 0 [Clearing condition] When 0 is written to DTRANERR0	R/W
17	DTRANERR1	DMAC Channel 1 Error [Setting condition] When an error occurs on DMAC channel 1 [Clearing condition] When 0 is written to DTRANERR1	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value is applied at a reset and when the DTRANRST1 bit in DM_CM_RST is 0.
Note 2. The initial value is applied at a reset and when the DTRANRST0 bit in DM_CM_RST is 0.

The DM_CM_INFO2 register indicates the status of the DMAC of the module and the sequencer. To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

55.3.31 DM_CM_INFO2_MASK : DM_CM_INFO2 Interrupt Mask Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0858

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTRANERR1_MASK	DTRANERR0_MASK
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 1. The write value should be 1.	R/W
16	DTRANERR0_MASK	DTRANERR0 interrupt masked	R/W
17	DTRANERR1_MASK	DTRANERR1 interrupt masked	R/W
31:18	—	These bits are read as 1. The write value should be 1.	R/W

The DM_CM_INFO2_MASK register enables or disables the DM_CM_INFO2 interrupt. When 0 is set in DM_CM_INFO2_MASK while the corresponding flag in DM_CM_INFO2 is set, an interrupt occurs.

55.3.32 DM_DTRAN_ADDR : DMAC Transfer Address Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0880

Bit position:	31																											3		0
Bit field:	DADDR[28:0]																											—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
31:3	DADDR[28:0]	Destination address/Source address (8 byte unit) Note that the value of DM_DTRAN_ADDR + transfer data length is less than or equal to 232.	R/W

The DM_DTRAN_ADDR register sets the transfer destination and source addresses for the DMAC of the module.

55.3.33 SCC_DTCNTL : Initial Setting Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	TAPNUM7TAPNUM0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TAPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TAPEN	SCC Sampling Clock Operation Enable 0: SCC sampling clock operation is disabled. 1: SCC sampling clock operation is enabled.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TAPNUM7TAPNUM0[7:0]	Set these bits to 0x08.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

55.3.34 SCC_TAPSET : Sampling Clock Position Setting Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1008

Bit position:	31																											7		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TAPSET7TAPSET0[7:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TAPSET7TAPSET0[7:0]	SCC Sampling Clock Position Set the tuning result in the range from 0 to TAPNUM − 1.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

55.3.35 SCC_DT2FF : Hardware Adjustment Register 1

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DT2NE[7:0]								DT2NS[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DT2NS[7:0]	Hardware Adjustment 1 This is a setting register for adjusting the timing inside the IP when using Tuning. When using Tuning, set 0x07 to these bits, make the setting before setting 1 in SCC_CKSEL[0].DTSEL.	R/W
15:8	DT2NE[7:0]	Hardware Adjustment 2 This is a setting register for adjusting the timing inside the IP when using Tuning. When using Tuning, set 0x02 to these bits, make the setting before setting 1 in SCC_CKSEL[0].DTSEL.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

This register makes a setting that SD_DATA, which has been fetched by the sampling clock at each TAP position, is used in the appropriate timing.

55.3.36 SCC_CKSEL : Sampling Clock Selection Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTSEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DTSEL	Sampling Clock Selection ● For SDR104 or HS200, set DTSEL to 1. DIV[7:0] in the SD_CLK_CTRL register to 0xFF (1:1 mode). ● When this bit is switched, stop the SD clock output from the SD/MMC host interface (set SCLKEN in SD_CLK_CTRL to 0). 0: An SCC sampling clock is not used (for other than SDR104 and HS200). 1: An SCC sampling clock is used (for SDR104 or HS200).	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

55.3.37 SCC_RVSCNTL : Sampling Clock Position Correction Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TAPSEL[7:0]								—	—	—	—	—	—	—	RVSEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RVSEN	SCC Sampling Clock Position Correction Enable When RVSEN is set to 1 after tuning has been performed, this module corrects the SCC sampling clock position each time of a command sequence of the SD/MMC host interface. However, when RVSERR is 1, this module does not correct the SCC sampling clock position. While tuning is being performed, set RVSEN to 0. 0: SCC sampling clock position correction is disabled. 1: SCC sampling clock position correction is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	TAPSEL[7:0]	SCC Sampling Clock Position Display Displays the SCC sampling clock position selected by hardware. After RVSEN has been set to 1, the value may differ from that of TAPSET.	R
31:16	—	These bits are read as 0. The write value should be 0.	R/W

55.3.38 SCC_RVSREQ : Sampling Clock Position Correction Request Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	RVSERR	REQTAPUP	REQTAPDWN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	REQTAPDWN	SCC Sampling Clock Position Negative Direction Correction Request ● If this bit is set to 1 after a command sequence, write 0 to this bit and rewrite TAPSET in the negative direction (when TAPSEL = 0, set TAPNUM − 1 to TAPSET). ● When RVSEN is 1, this bit is disabled (this bit is not set to 1). ● Writing 1 to this bit is disabled and writing 0 to this bit is only enabled. 0: There is no correction request. 1: There is a correction request.	R/W

Bit	Symbol	Function	R/W
1	REQTAPUP	SCC Sampling Clock Position Positive Direction Correction Request ● If this bit is set to 1 after a command sequence, write 0 to this bit and rewrite TAPSET in the positive direction (when TAPSEL = TAPNUM − 1, set 0 to TAPSET). ● When RVSEN is 1, this bit is disabled (this bit is not set to 1). ● Writing 1 to this bit is disabled and writing 0 to this bit is only enabled. 0: There is no correction request. 1: There is a correction request.	R/W
2	RVSERR	SCC Sampling Clock Position Correction Error ● If this bit is set to 1 after a command sequence, write 0 to this bit and perform tuning again. ● Ignore this bit while tuning is being performed. ● Writing 1 to this bit is disabled and writing 0 to this bit is only enabled. 0: There is no correction error. 1: There is a correction error.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

55.3.39 SCC_SMPCMP : Sampling Data Comparison Register

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1030

Bit position:	31							24							16							8							0
Bit field:	—	—	—	—	—	—	—	CMPNGU[8:0]								—	—	—	—	—	—	—	CMPNGD[8:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	CMPNGD[8:0]	Comparison of sampling data with the after TAP Clock*1 Bit 0-7 is the comparison result of data 0-7. Bit 8 is the comparison result of CMD. 0: Match 1: Mismatch	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
24:16	CMPNGU[8:0]	Comparison of sampling data with the previous TAP Clock*1 Bit 16-23 is the comparison result of data 0-7. Bit 24 is the comparison result of CMD. 0: Match 1: Mismatch	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. [Clear conditions]
The start of the command sequence
Write to SCC_TAPSET register

Data comparison register indicates the result of the comparison of the sampling data. The subject of the comparison is the before and the behind TAP.

55.3.40 SCC_TMPPORT : Hardware Adjustment Register 2

Base address: SDHIm = 0x9208_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1038

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMPOUT[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TMPOUT[15:0]	Hardware adjustment 3 When using the delay tuning mechanism, set the following values. Set 0x0000 to these bits. When operating with other than SDR104/HS200, this register adjusts the clock delay of the Flip Flop that latches the received data from the outside of the chip. For this product, set 0x0010 when transferring at 3.3 V, and set 0x0001 when transferring at 1.8 V.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

This register makes a setting that SD_DATA, which has been fetched by the sampling clock at each TAP position, is used in the appropriate timing.

55.4 Operation

55.4.1 SD Interface

55.4.1.1 SD Data Format

When data is read from the SD card, the procedure is as follows.

1. The SD/MMC host interface receives data from the SD card via the SDDAT signal. (SDDAT signal: see Figure 55.2, Figure 55.3, and Figure 55.5.)
2. The receive data is stored in SD_BUF of the SD/MMC host interface. (SD_BUF store data: see Figure 55.7)
3. The data stored in SD_BUF is read from SD_BUF0. (Reading from SD_BUF0: see Table 55.5)

When data is written to the SD card, the above procedure will be reversed.

When accessing SD_BUF0, caution should be taken for the transfer order in SDDAT and the store order in SD_BUF. In addition, data stored in SD_BUF0 can be replaced in bytes with the EXT_SWAP. (See Figure 55.7)

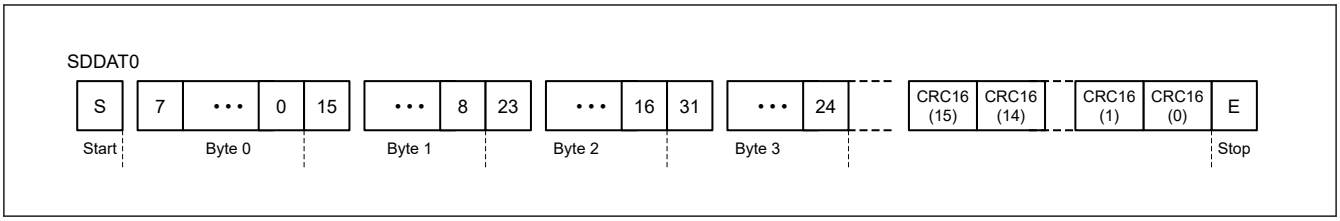


Figure 55.2 SDDAT in 1-bit width mode

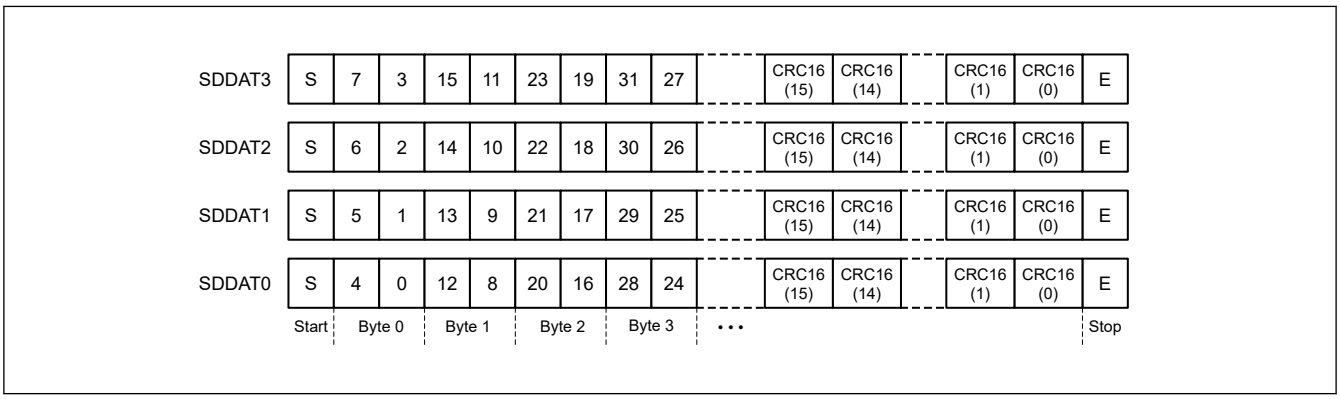


Figure 55.3 SDDAT in 4-bit width mode

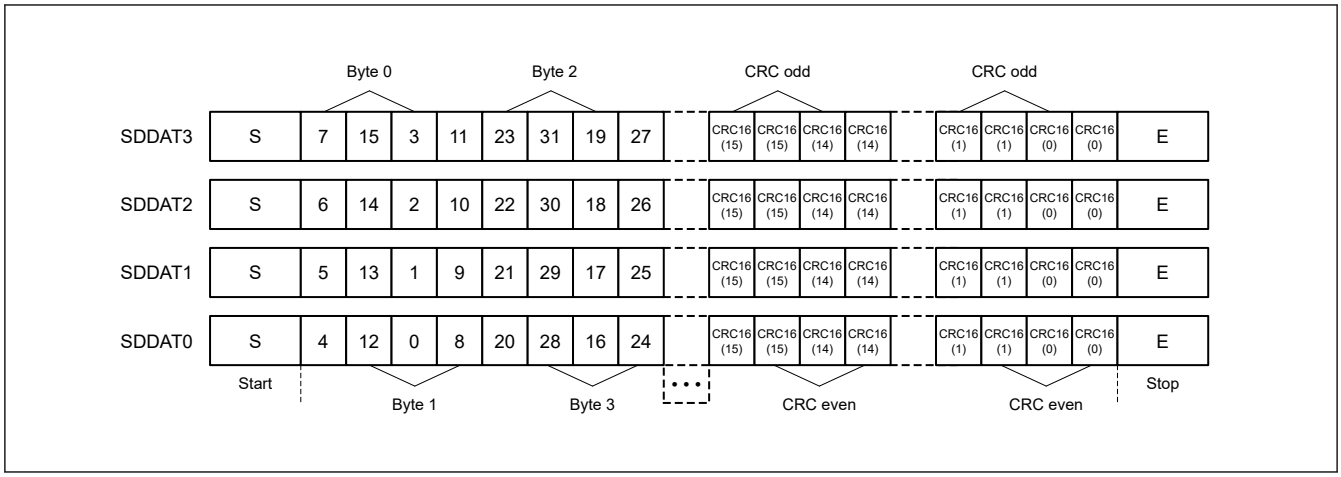


Figure 55.4 SDDAT in 4-bit width DDR mode

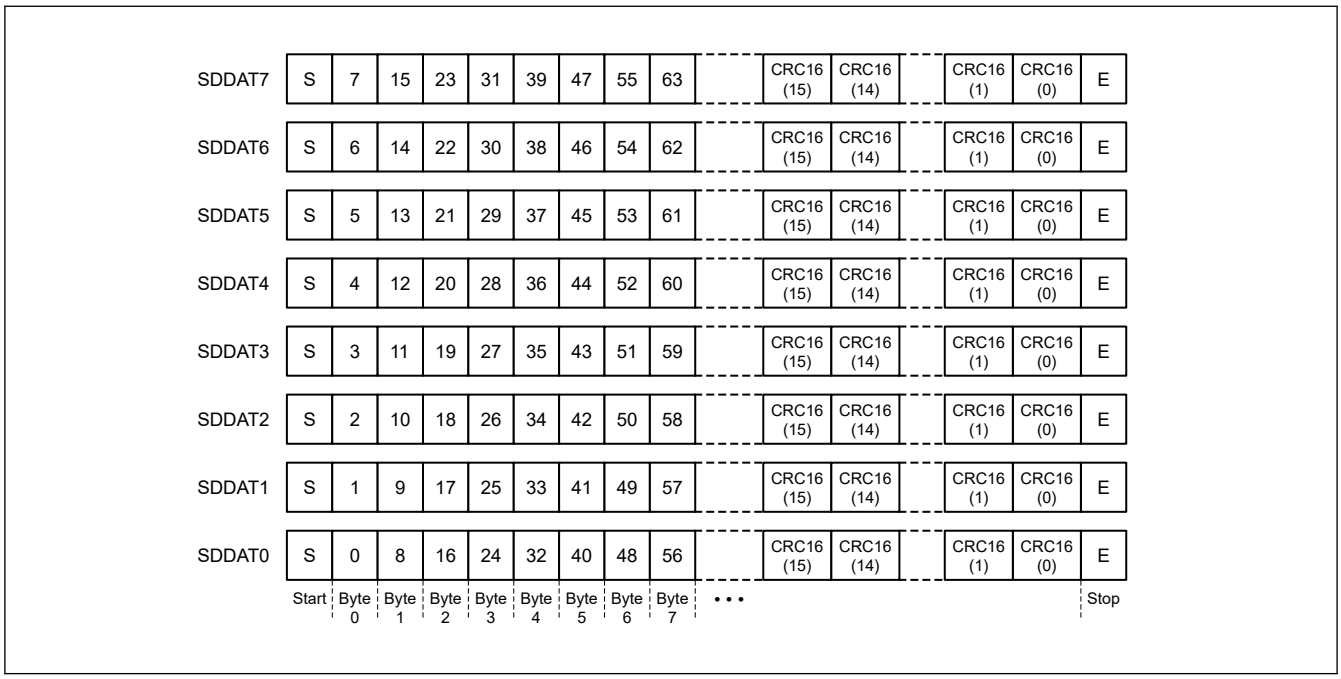


Figure 55.5 SDDAT in 8-bit width mode

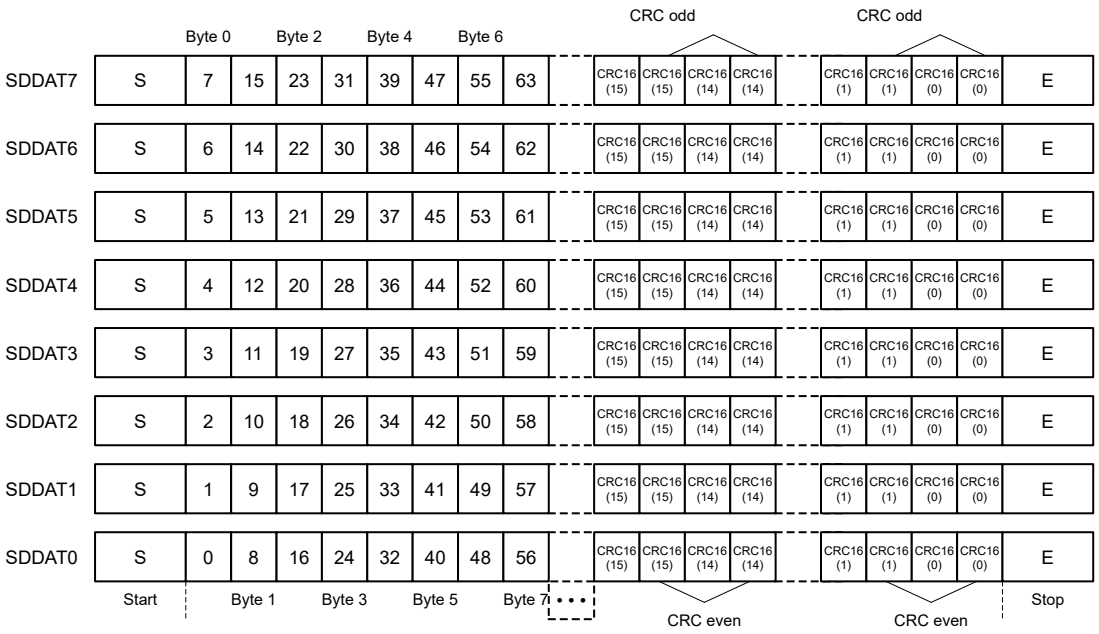


Figure 55.6 SDDAT in 8-bit width DDR mode

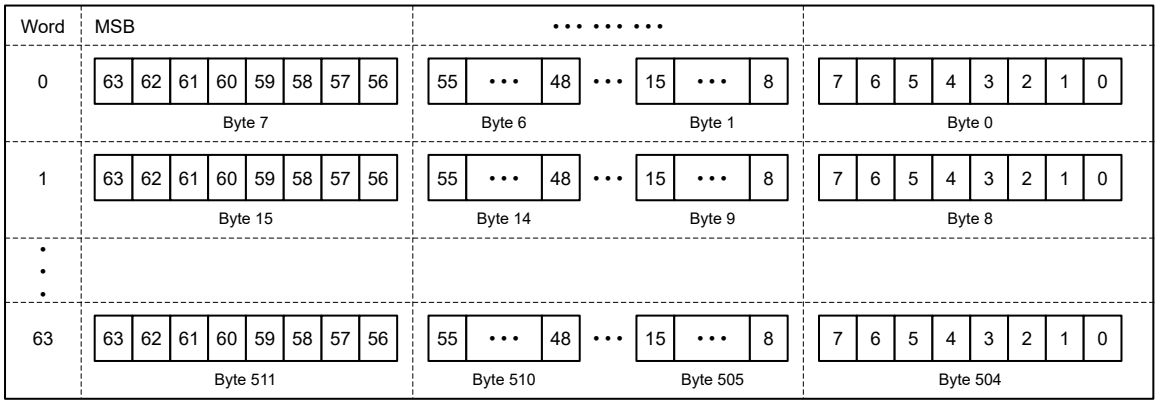


Figure 55.7 SD_BUF store data

Table 55.5 Reading from SD_BUF0 (1 of 2)

WMODE*1	BUSWIDTH*1	ENDIAN*1	Read Data*2
0	0	0	0x0123456789ABCDEF
0	0	1	0xEFCDAB8967452301
1	1	0	0x89ABCDEF (1st) 0x01234567 (2nd)
1	1	1	0xEFCDAB89 (1st) 0x67452301 (2nd)
1	0	0	0xCDEF (1st) 0x89AB (2nd) 0x4567 (3rd) 0x0123 (4th)

Table 55.5 Reading from SD_BUF0 (2 of 2)

WMODE*1	BUSWIDTH*1	ENDIAN*1	Read Data*2
1	0	1	0xEFCD (1st) 0xAB89 (2nd) 0x6745 (3rd) 0x2301 (4th)

Note 1. The name of a bit in HOST_MODE
Note 2. When the data stored in SD_BUF is 0x0123456789ABCDEF

55.4.1.2 Bus Signal Voltage Switch

Change the electric potential of the bus signal in the following procedure after checking that the SD card supports 1.8 V.

1. Issuing CMD11
   Perform command sequence processing of CMD11.
2. Stopping the SD clock (a)
   Set the SCLKEN bit in the SD_CLK_CTRL to 0 to stop*1 the output of the SD clock. When the SDCLKOFFEN bit in the SD_CLK_CTRL register is 1, the SDCLKOFFEN bit is also set to 0.

   Note 1. When the SDCLKOFFEN bit in the SD_CLK_CTRL register is 1, the SD clock has automatically been stopped.
3. Checking the value of SDDAT
   Check that the DAT0 bit in the SD_INFO2 register is 0.
4. Changing the supply voltage of the host device
   Change the voltage which is supplied through the power supply pin of the given channel (VCC1833_6 for channel 0 or VCC1833_7 for channel 1) from 3.3 V to 1.8 V by corresponding SDm_IOVS pin.
5. Starting supply of the SD clock (b)
   After the SD clock has been stopped ((a) above) and 5 ms or more has elapsed, set the SCLKEN bit in the SD_CLK_CTRL register to 1 and allow the output of the SD clock. The SDCLKOFFEN bit must be 0.
6. Checking the value of SDDAT
   After supplying the SD clock has been started ((b) above) and 1 ms or more has elapsed, check that the DAT0 bit in the SD_INFO2 register is 1. It is possible to set the SDCLKOFFEN bit in the SD_CLK_CTRL register to 1 and allow SD Clock (SDCLK) Output Automatic Control Enable.

55.4.2 Card Detect/Write Protect

55.4.2.1 Card Detect

The SD/MMC host interface has two types of card detect functions as described in the following.

(1) Card detect with ISDCD

Figure 55.8 shows the timing chart of card detect using ISDCD. ISDCD is connected to the card socket and pulled up on the host device. The resistance of the pull-up resistor is decided by the specification of the SD host device.

[Card insertion]

ISDCD is pulled down when a card is inserted. At this time, if ISDCD has been pulled down for the Mcycle period (set in SD_OPTION), INFO4 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)

[Card removal]

ISDCD is pulled up when a card is removed. At this time, if ISDCD has been pulled up for the Mcycle period (set in SD_OPTION), INFO3 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)

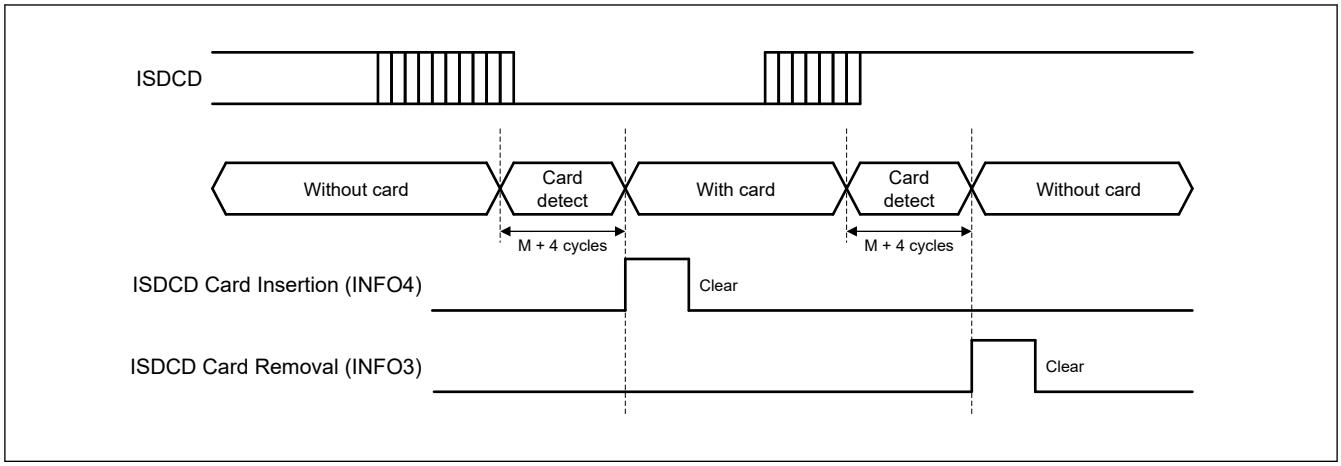


Figure 55.8 Example of card detect with ISDCD

(2) SD card detect with SDDAT3

Figure 55.9 shows the timing chart when the SD card is detected using SDDAT3. In addition, SDDAT3 is pulled down on the host device. The resistance of the pull-down resistor is decided by the specification of the SD host device.

[Card insertion]

When an SD card is inserted, SDDAT3 is pulled up. Accordingly, INFO9 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)

[Card removal]

When an SD card is removed, SDDAT3 is pulled down. Accordingly, INFO8 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)

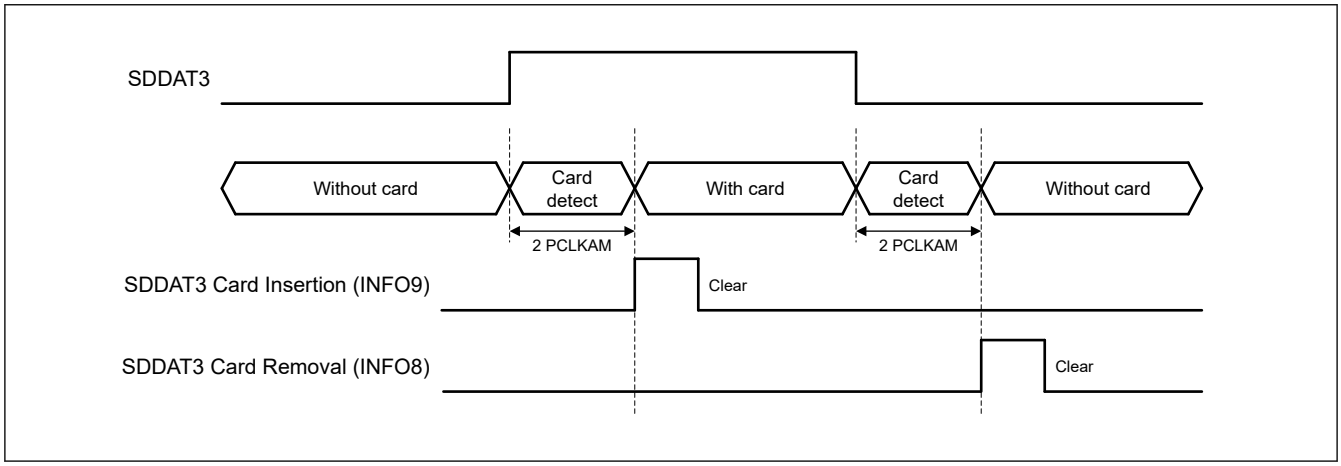


Figure 55.9 SD card detect with SDDAT3

55.4.2.2 Write Protect

The SD/MMC host interface has two types of write protect functions.

(1) Write protect with ISDWP

ISDWP is connected to the card socket, and pulled up or pulled down by the card insertion. The selection of pulling up or pulling down and the resistance value is decided by the specifications of the SD host device. As the ISDWP state is reflected to INFO7 in SD_INFO1, the write protect is decided after the SD card is inserted.

(2) Write protect with command

The card's internal write protection and the card lock/unlock operation are realized by the command.

55.4.3 Interrupt Request

The SD/MMC host interface has the interrupt requests shown in Table 55.6 shows the relationship between the interrupt flag registers and the interrupt mask registers. When a bit in an interrupt mask register is set to 0, an interrupt occurs by setting the corresponding bit in the interrupt flag register to 1.

To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

Table 55.6 Interrupt request

Interrupt request	Interrupt flag register		Interrupt mask register	
	Register name	Bit name	Register name	Bit name
Card access interrupt*1	SD_INFO1	INFO2	SD_INFO1_MASK	IMASK2
		INFO0		IMASK0
	SD_INFO2	ILA	SD_INFO2_MASK	IMASK
		BWE		BMASK1
		BRE		BMASK0
		ERR6		EMASK6
		ERR5		EMASK5
		ERR4		EMASK4
		ERR3		EMASK3
		ERR2		EMASK2
		ERR1		EMASK1
		ERR0		EMASK0
SDIO access interrupt*2	SDIO_INFO1	EXWT	SDIO_INFO1_MASK	MEXWT
		EXPUB52		MEXPUB52
		IOIRQ		IOMSK
Card detect interrupt*1	SD_INFO1	INFO9	SD_INFO1_MASK	IMASK9
		INFO8		IMASK8
		INFO4		IMASK4
		INFO3		IMASK3
DMAC interrupt*1	DM_CM_INFO1	DTRANEND1	DM_CM_INFO1_MASK	DTRANEND1_MASK
		DTRANEND0		DTRANEND0_MASK
		SEQEND		SEQEND_MASK
	DM_CM_INFO2	DTRANERR1	DM_CM_INFO2_MASK	DTARERR1_MASK
		DTRANERR0		DTARERR0_MASK
		SEQERR		SEQERR_MASK

Note 1. Interrupt signal "OXMNIRQ" is asserted.
Note 2. Interrupt signals, both "OXMNIRQ" and "OXASIOIRQ", are asserted.

55.4.4 Communications Errors and Timeouts

Table 55.7 and Table 55.8 show the relationships between the SD card interrupt flag register and SD error status register for communications errors and timeouts, respectively. When a bit in the SD card interrupt flag register is set to 1, the corresponding bit in the SD error status register is set to 1. The values of the SD error status register are cleared by writing to SD_CMD or writing 0 to the SDRST bit in SOFT_RST.

Table 55.7 Communications errors

Communication error	SD interrupt flag register		SD error status register		Description
	Register name	Bit name	Register name	Bit name	
END error	SD_INFO2	ERR2	SD_ERR_STS1	E5	When an error occurs in the CRC status length
				E4	When an error occurs in read data length
				E3	When an error occurs in the response length to a command issued within a command sequence
				E2	When an error occurs in the response length (other than a response to a command issued within a command sequence)
CRC error		ERR1		E11	When an error occurs in the CRC status
				E10	When a CRC error occurs in the read data
				E9	When a CRC error occurs in the response to a command issued within a command sequence
				E8	When a CRC error occurs in the response (other than a response to a command issued within a command sequence)
CMD error		ERR0		E1	The command index of the transmitted command differed from the command index of the received response (for a command issued within a command sequence)
				E0	The command index of the transmitted command differed from the command index of the received response (for a command issued other than within a command sequence)

Table 55.8 Timeouts

Timeout	SD interrupt flag register		SD error status register		Description
	Register name	Bit name	Register name	Bit name	
Response timeout	SD_INFO2	ERR6	SD_ERR_STS2	E1	When the response to a command issued within a command sequence is not received even after 640 cycles of SDCLK have elapsed
				E0	When the response (other than a response to a command issued within a command sequence) is not received even after 640 cycles of SDCLK have elapsed
Data timeout (other than response timeout)		ERR3		E6	When the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle[*1] after the CRC status
				E5	When the CRC status is not received even after Ncycle[*1] has elapsed after data writing
				E4	When read data is not received even after Ncycle[*1] has elapsed after read command
					When read data for the next block are not received even after Ncycle[*1] has elapsed after the reception of read data
					When read data for the next block are not received even after Ncycle[*1] has elapsed after release of the read wait state
				E3	When the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle[*1] after CMD12 has been issued within a command sequence
				E2	When the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle[*1] after R1b response

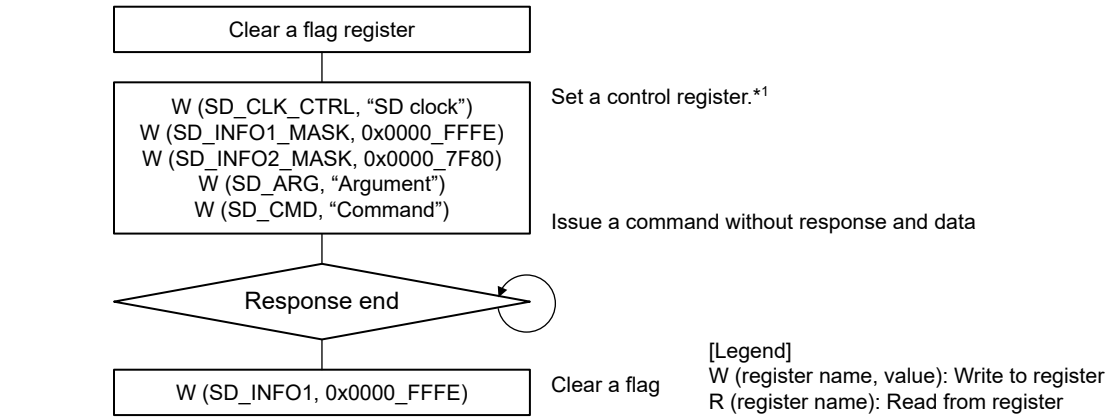
Note 1. Ncycle is set by bit 7 to bit 4 in SD_OPTION.

55.5 Usage Example

55.5.1 Command without Data Transfer

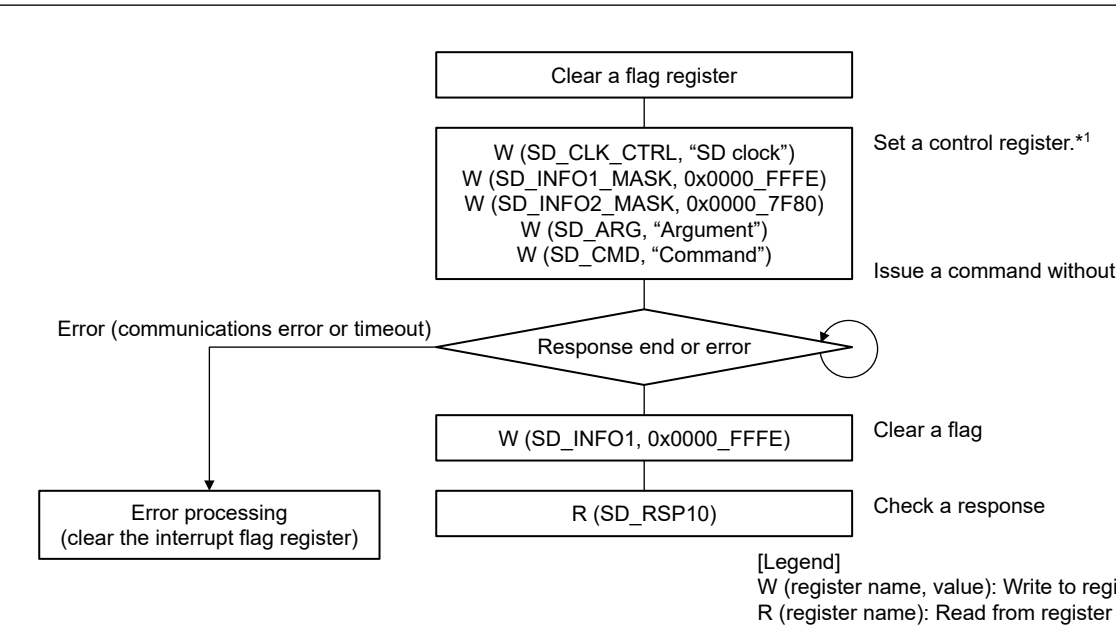
55.5.1.1 Flowchart

Figure 55.10 and Figure 55.11 show flowchart examples.



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.10 Flow example of command without response and data



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.11 Flow example of command without data

55.5.1.2 Operation for Command without Data Transfer

The legend below is used for description of register read/write.

— W (register name, value): Write to register

— R (register name): Read from register

The operation is described below.

(1) Command without Response and Data

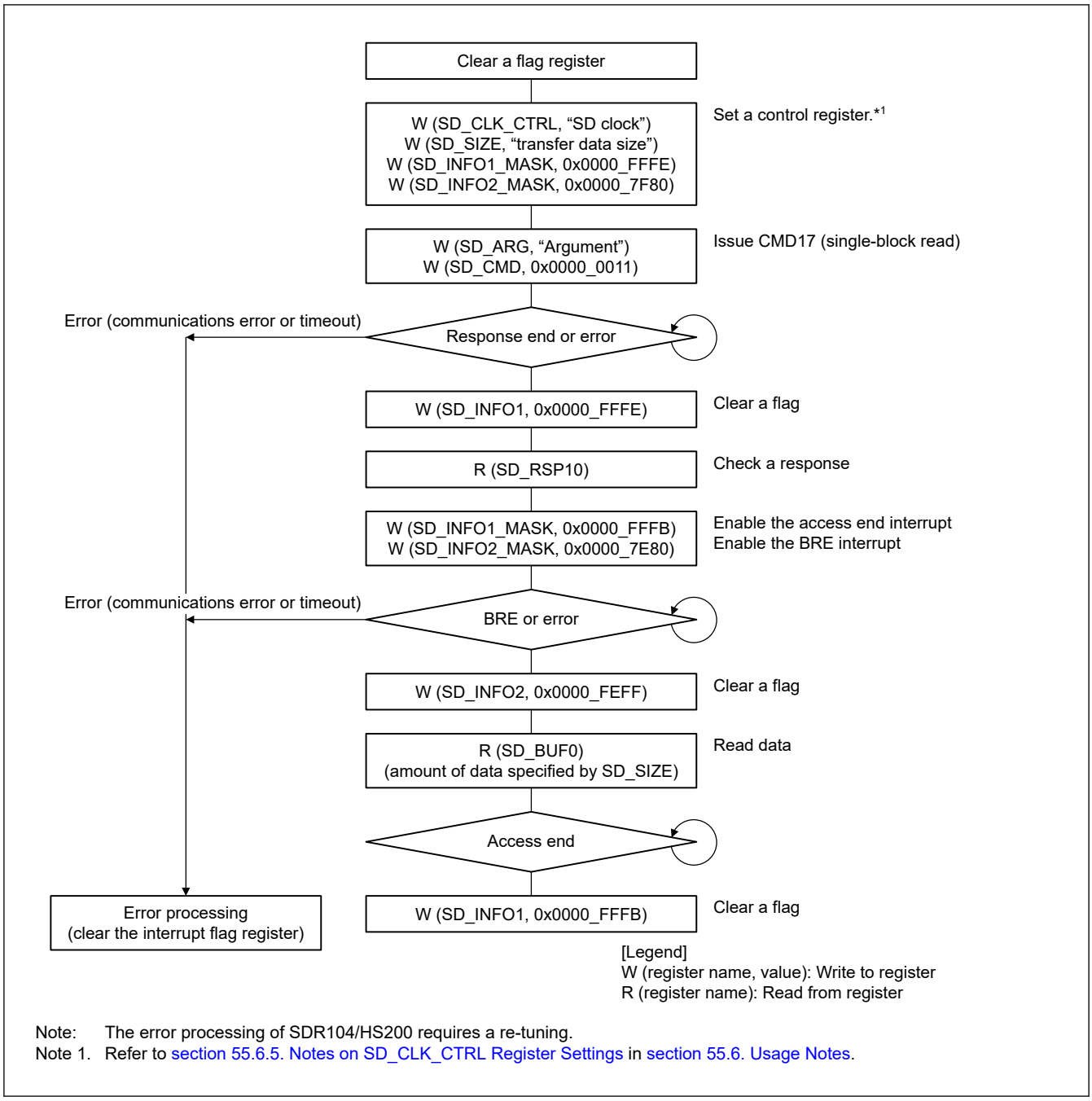
1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), interrupt mask, and so on. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)
3. Command issue
   Set CMD Argument in SD_ARG and write to SD_CMD.
   Accordingly, CMD is issued, and the operation is started.
4. Flag clear
   When transmission of a command is completed, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt.
   Clear INFO0 to 0.

(2) Command without Data

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), interrupt mask, and so on. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)
3. Command issue
   Set CMD Argument in SD_ARG and write to SD_CMD.
   Accordingly, CMD is issued, and the operation is started.
4. Flag clear
   When a response is received, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0.
5. Read a response from SD_RSP10.
   Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

55.5.2 Single Block Read

55.5.2.1 Flowchart



Note: The error processing of SDR104/HS200 requires a re-tuning.
Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.12 Single block read flowchart example

55.5.2.2 Operation for Single Block Read

The operation of the single block read is described below.

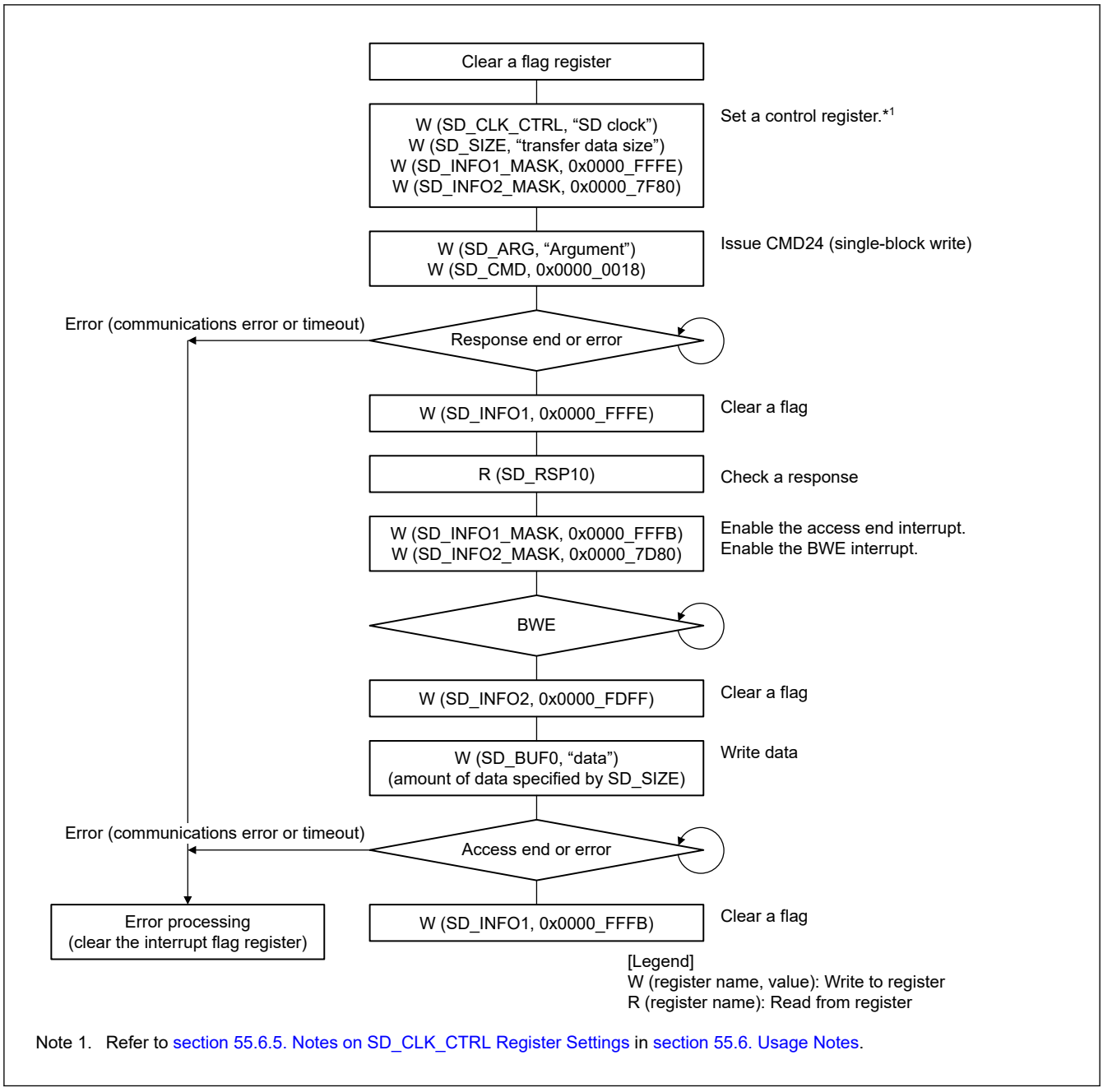
1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)

3. Command issue (CMD17)
   Set CMD17 Argument in SD_ARG and write 0x0000 0011 to SD_CMD.
   Accordingly, CMD17 is issued, and the single block read operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP10.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. Furthermore, this causes CMD12 and CMD52 to not be issued. If the command sequence is halted, the INFO2 bit (access end) in SD_INFO1 is set to 1 to generate an interrupt.
5. Data receive from SD card and data read
   Write 0x0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000 7E80 to SD_INFO2_MASK to enable the BRE interrupt. When the data receive from the SD card is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified by SD_SIZE from SD_BUF0.
6. Operation complete
   When the data read from SD_BUF0 is completed, INFO2 (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO2 to 0 to end the single block read operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

55.5.3 Single Block Write

55.5.3.1 Flowchart



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.13 Single block write flowchart example

55.5.3.2 Operation for Single Block Write

The operation of the single block write is described below.

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
3. Command issue (CMD24)

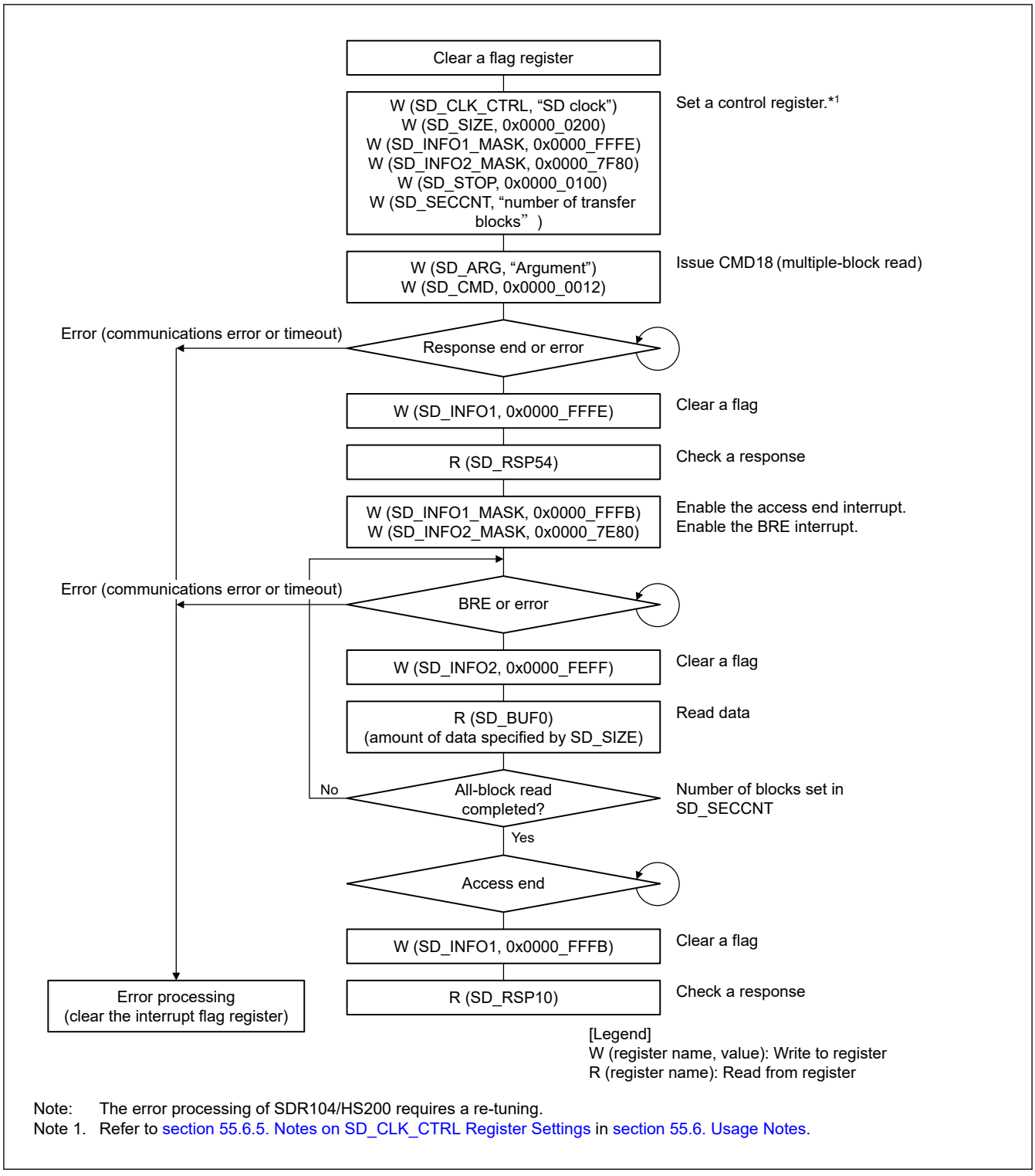
Set CMD24 Argument in SD_ARG and write 0x0000 0018 to SD_CMD.
Accordingly, CMD24 is issued, and the single block write operation is started.

4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP10.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. Furthermore, this causes CMD12 and CMD52 to not be issued. If the command sequence is halted, the INFO2 bit (access end) in SD_INFO1 is set to 1 to generate an interrupt.
5. Data write and data transmit to SD card
   Write 0x0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card.
   However, a communications error or timeout may be generated if data are being transmitted after writing to SD_BUF0.
6. Operation complete
   When the CRC status and busy state are received from the SD card, INFO2 (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the INFO2 bit to 0 to end the single block write operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

55.5.4 Multiple Block Read

55.5.4.1 Flowchart



Note: The error processing of SDR104/HS200 requires a re-tuning.
Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.14 Multiple block read flowchart example

55.5.4.2 Operation for Multiple Block Read

The operation of the multiple block read is described below.

1. Flag register clear

First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

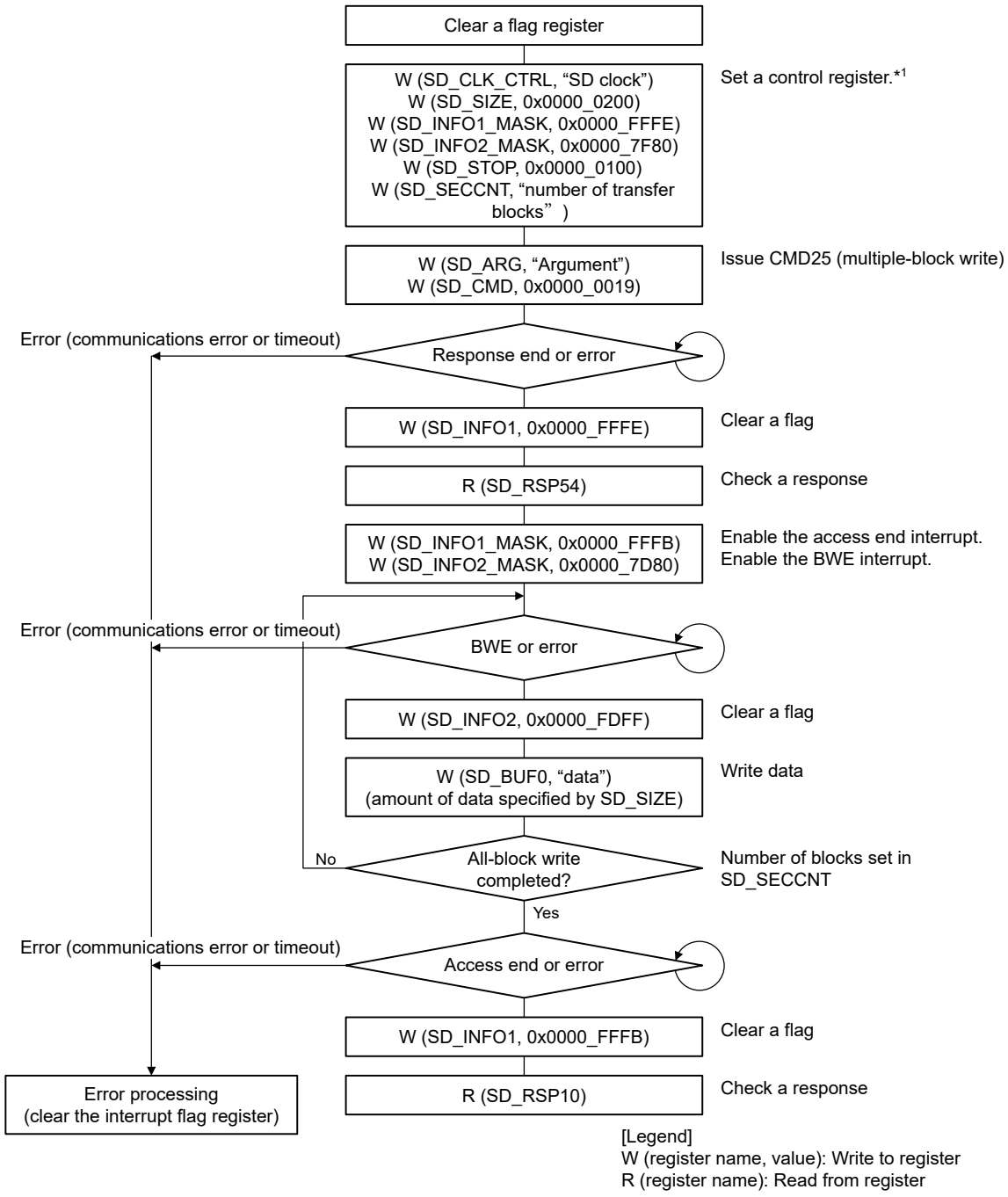
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
   Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.
3. Command issue (CMD18)
   Set CMD18 Argument in SD_ARG and write 0x0000 0012 to SD_CMD.
   Accordingly, CMD18 is issued, and the multiple block read operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP54.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit (access end) in SD_INFO1 to 1 when reception of the response has been completed. Clear the INFO2 bit to 0 and read the response.
5. Data receive from SD card and data read
   Write 0x0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000 7E80 to SD_INFO2_MASK to enable the BRE interrupt. When one-block data receive from the SD card is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified by SD_SIZE from SD_BUF0. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communications error or timeout may be generated if data are being received while reading of SD_BUF0 is in progress. CMD12 is automatically issued to stop multi-block transfer with the number of blocks which is set to SD_SECCNT and the response is received. At this point, CMD12 Argument is automatically set to 0x0000 0000.
6. Operation complete
   When all-block data read and the CMD12 response receive are completed, INFO2 (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO2 to 0 to read the response. This is the end of multiple block read operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

55.5.5 Multiple Block Write (when Using Internal Timer)

55.5.5.1 Flowchart

Figure 55.15 shows the flowchart when using an internal timer.



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.15 Multiple block write flowchart example (when using internal timer)

55.5.5.2 Operation for Multiple Block Write (when Using Internal Timer)

The operation of the multiple block write is described below.

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
   Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

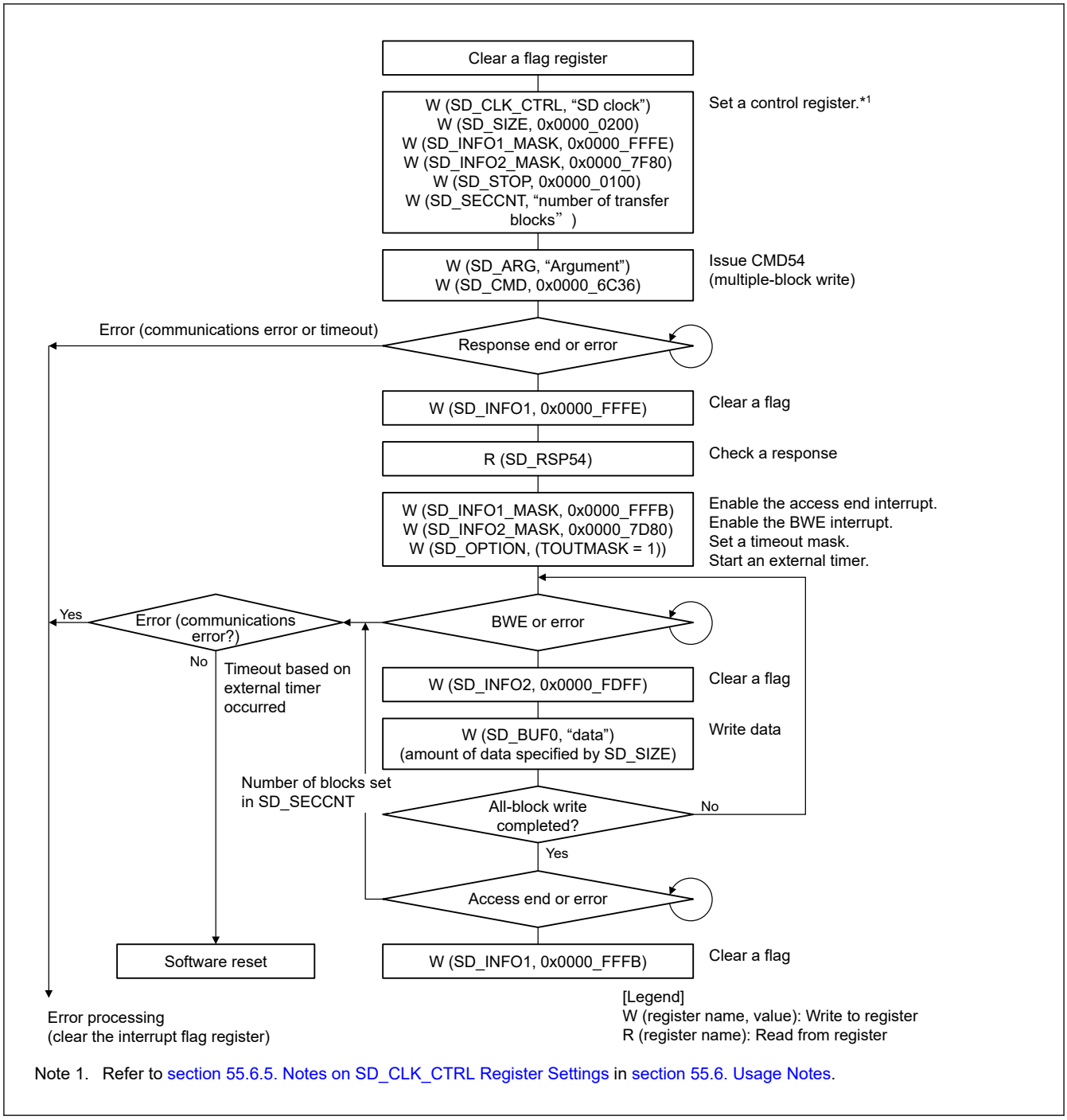
3. Command issue (CMD25)
   Set CMD25 Argument in SD_ARG and write 0x0000 0019 to SD_CMD.
   Accordingly, CMD25 is issued, and the multiple block write operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP54.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit (access end) bit in SD_INFO1 to 1 when reception of the response has been completed. Clear the INFO2 bit to 0 and read the response.
5. Data write and data transmit to SD card
   Write 0x0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card. Doing this repeats transfer of the number of blocks set in SD_SECCNT.
   However, a communications error or timeout may be generated if data are being transmitted while writing to SD_BUF0 is in progress. CMD12 is automatically issued to stop multi-block transfer with the number of blocks which is set to SD_SECCNT and the response is received. At this point, CMD12 Argument is automatically set to 0x0000 0000.
6. Operation complete
   When all-block data transmit and the CRC status receive are completed, the INFO2 bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the INFO2 bit to 0 to read the response. This is the end of multiple block write operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

55.5.6 Multiple Block Write (when Using External Timer)

55.5.6.1 Flowchart

The flowchart when using an external timer instead of an internal timer of this module is shown below.



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.16 Multiple block write flowchart example (when using external timer)

55.5.6.2 Operation for Multiple Block Write (when Using External Timer)

The operation of the multiple block write is described below.

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
   Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.
3. Command issue (CMD54)
   Set CMD54 Argument in SD_ARG and write 0x0000 6C36 to SD_CMD.

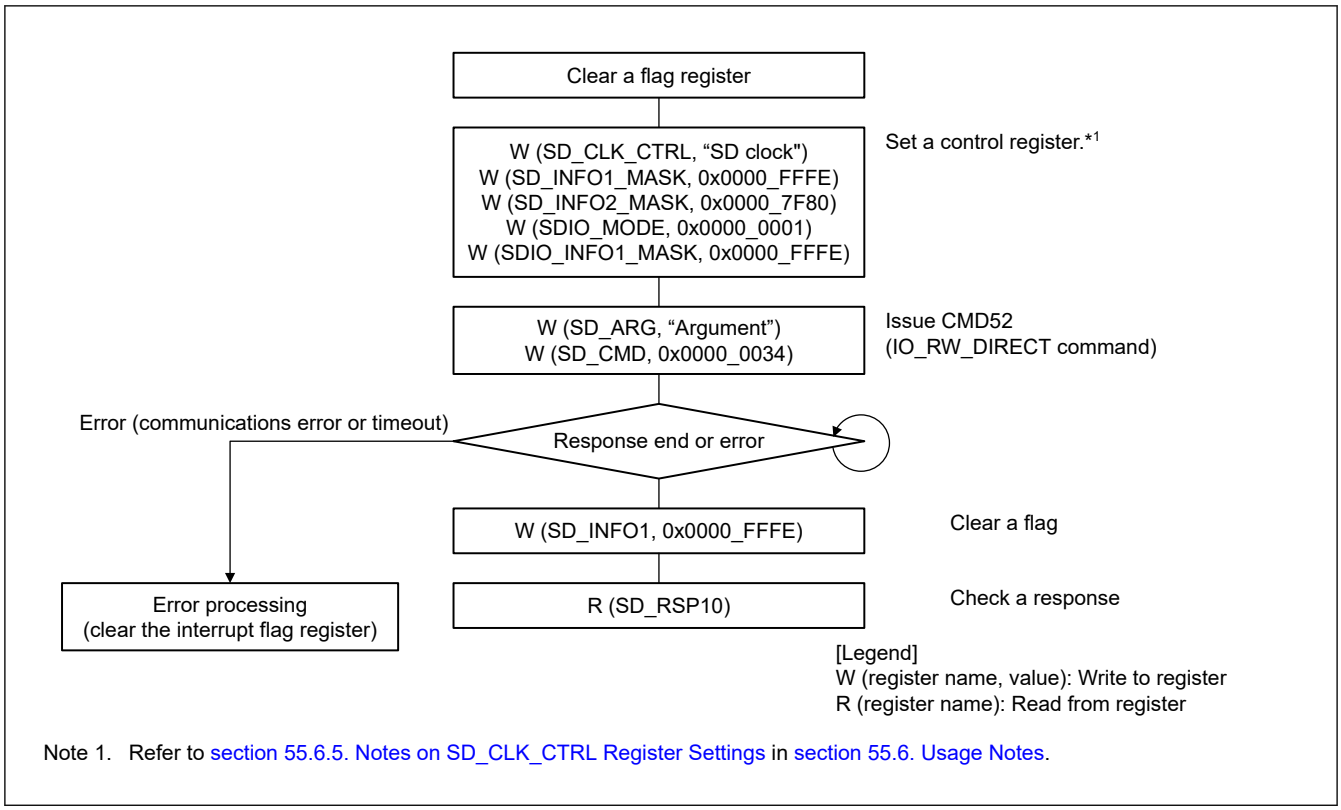
Accordingly, CMD54 is issued, and the multiple block write operation is started.

4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP54.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit (access end) bit in SD_INFO1 to 1 when reception of the response has been completed. Clear the INFO2 bit to 0 and read the response.
5. Data write and data transmit to SD card
   Write 0x0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. Set the TOUTMASK bit in SD_OPTION to disable timeout and start an external timer.
   When SD_BUF0 is ready for the data to be written, the BWE bit in the SD_INFO2 register is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communications error may be generated if data are being transmitted while writing to SD_BUF0 is in progress.
6. Operation complete
   When all-block data transmit and the CRC status receive are completed, the INFO2 bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the INFO2 bit to 0 to read the response. This is the end of multiple block write operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs at response reception (a communications error or timeout) or at data transmission. Perform a software reset if a timeout occurs at data transmission based on an external timer.

55.5.7 IO_RW_DIRECT Commend (CMD52)

55.5.7.1 Flowchart



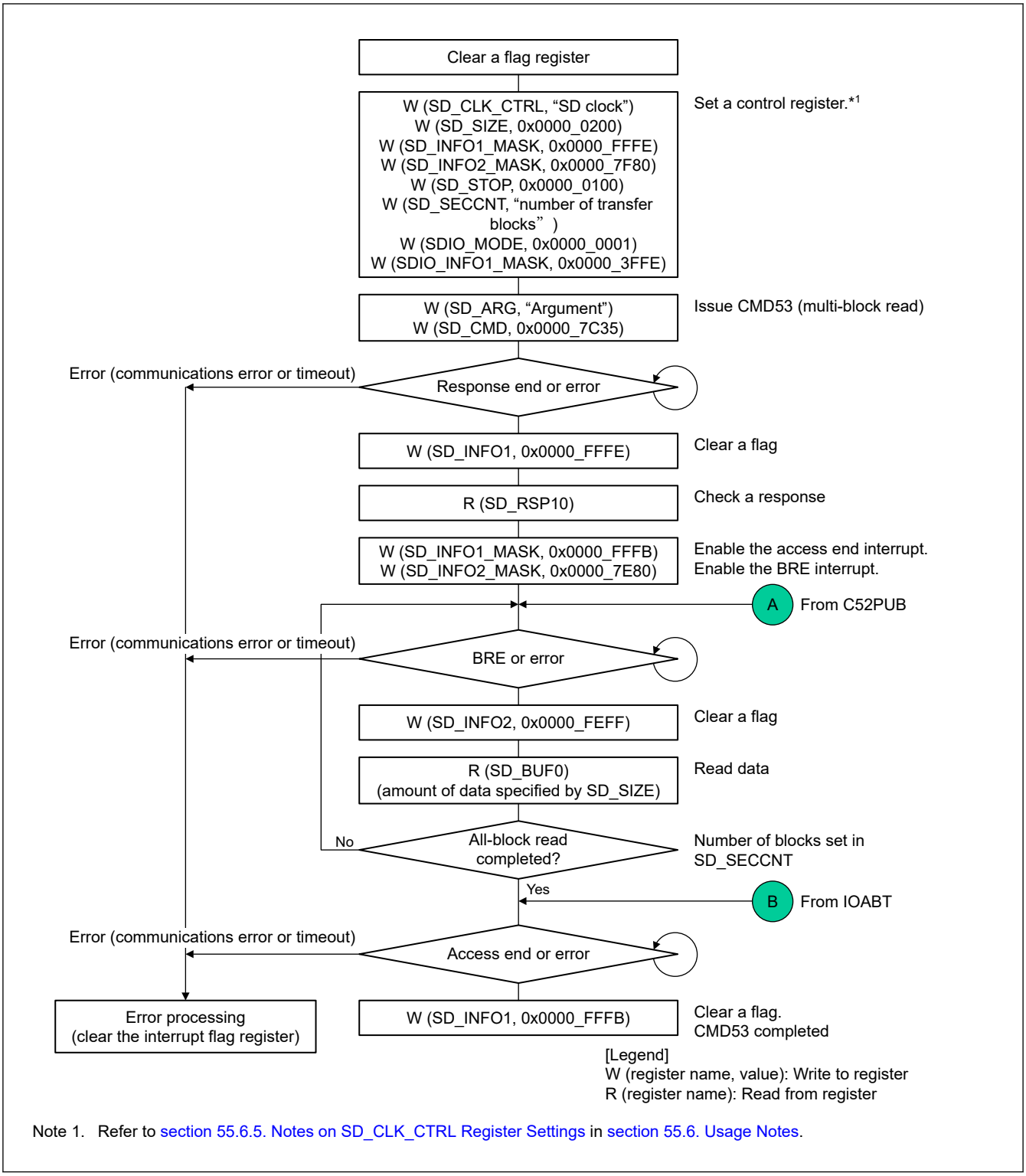
Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.17 IO_RW_DIRECT command (CMD52) flowchart example

55.5.8 IO_RW_EXTENDED (CMD53/Multiple Block Read)

55.5.8.1 Flowchart

Figure 55.18 shows a flowchart example for CMD53 (multiple block read).



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.18 CMD53 (Multiple Block Read) flowchart example

Figure 55.19 shows a flowchart example when CMD52 (SDIO abort) is issued at CMD53 (multiple block read).

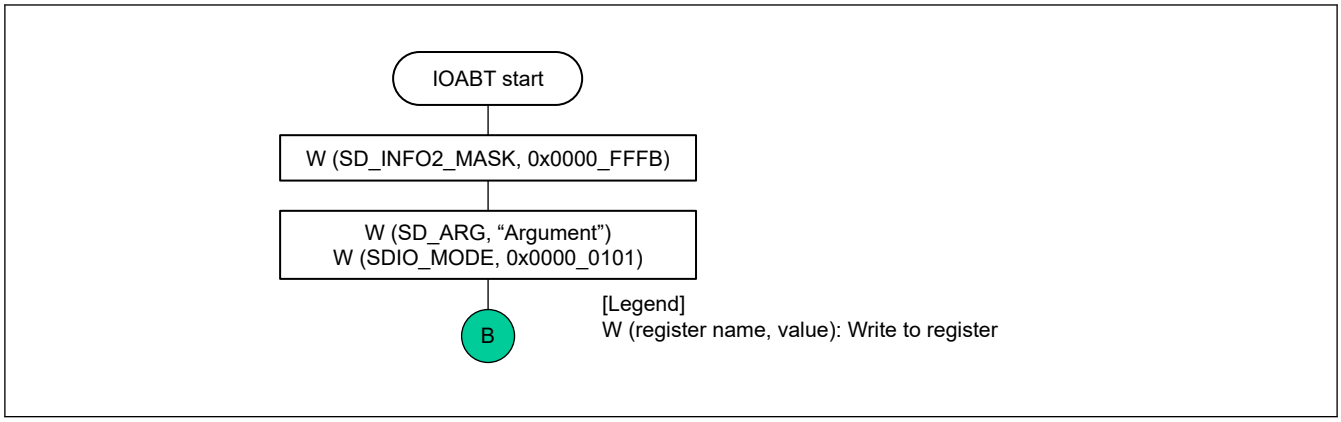


Figure 55.19 Flowchart example when CMD52 (SDIO Abort) is Issued at CMD53 (Multiple Block Read)

Figure 55.20 shows a flowchart example when CMD52 (SDIO none abort) is issued at CMD53 (multiple block read) while the SD/MMC host interface is in the read wait state.

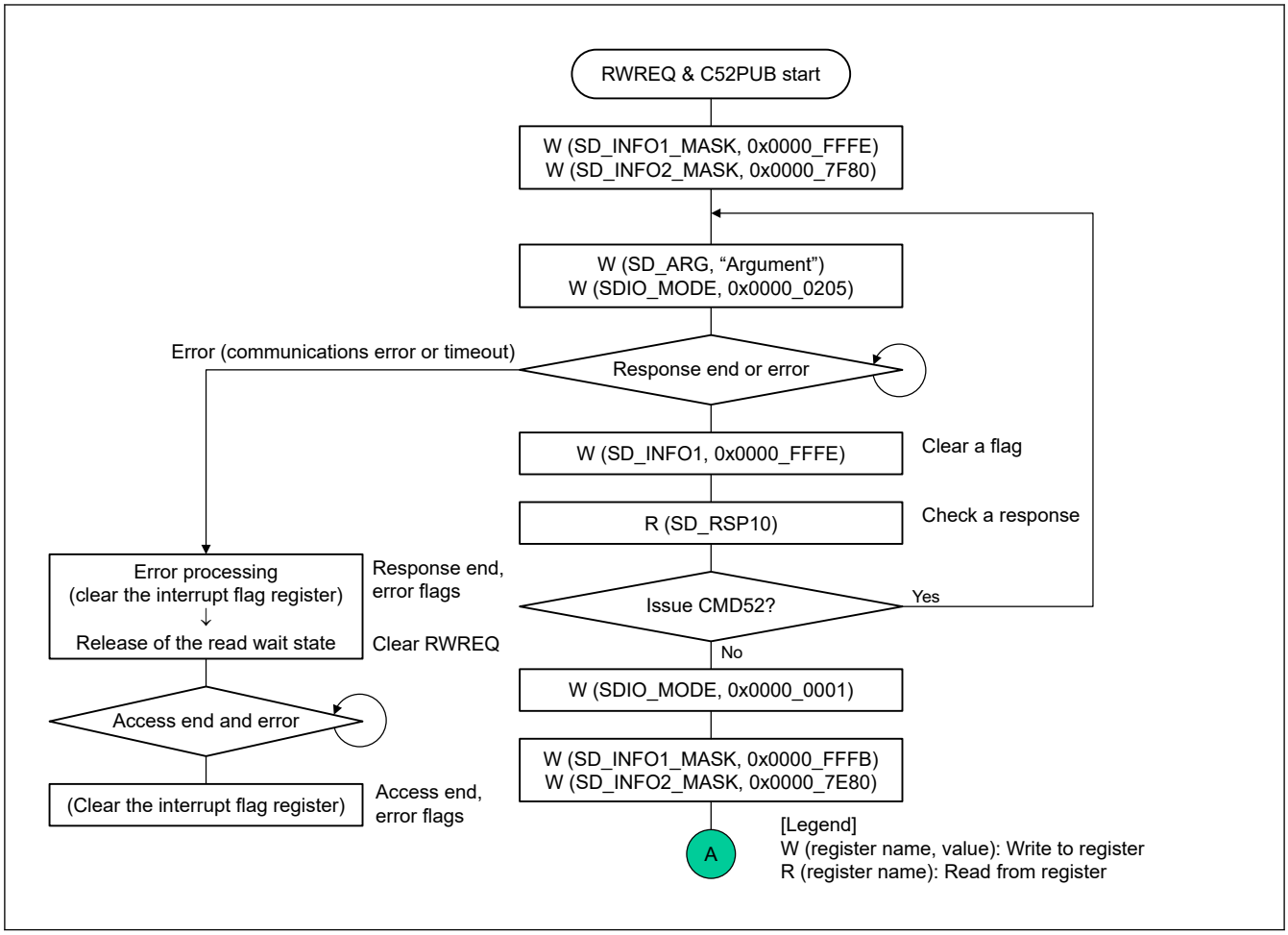
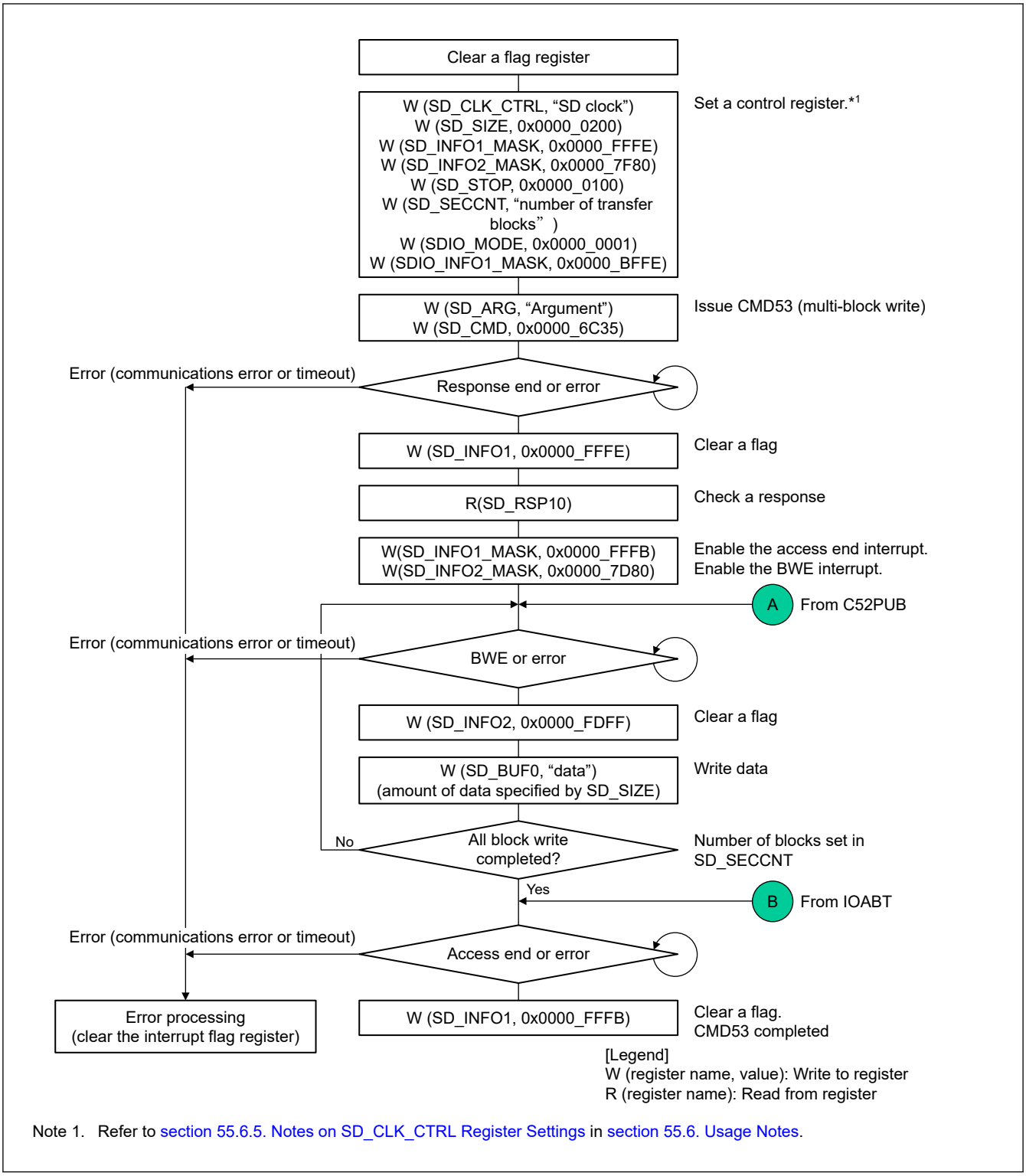


Figure 55.20 Flowchart example when CMD52 (SDIO None Abort) is Issued after Read Wait State is entered at CMD53 (Multi Block Read)

55.5.9 IO_RW_EXTENDED (CMD53/Multiple Block Write)

55.5.9.1 Flowchart

Figure 55.21 shows a flowchart example for CMD53 (multiple block write).



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.21 CMD53 (Multiple Block Write) flowchart example

Figure 55.22 shows a flowchart example when CMD52 (SDIO abort) is issued at CMD53 (multiple block write).

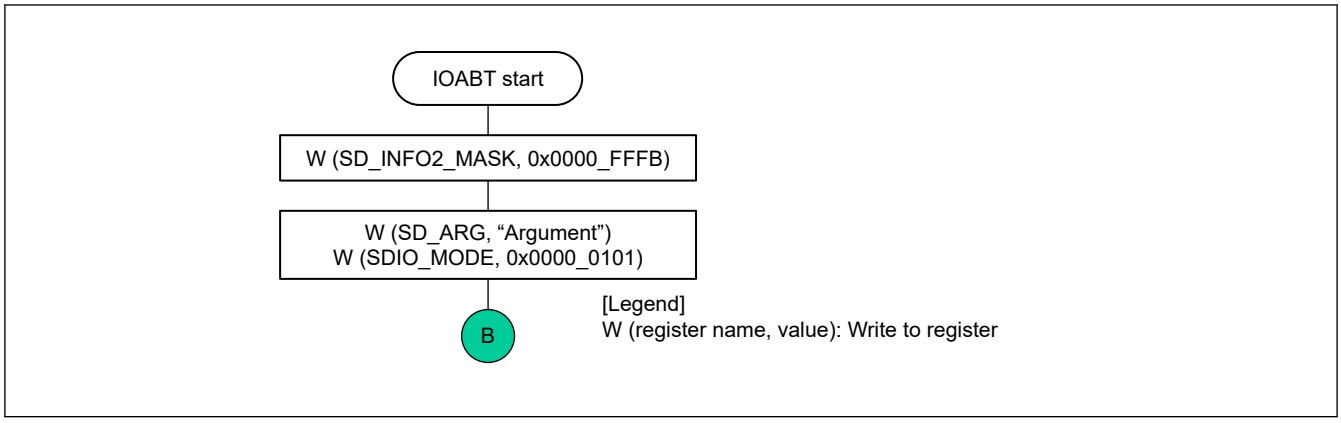


Figure 55.22 Flowchart example when CMD52 (SDIO Abort) is issued at CMD53 (Multiple Block Write)

Figure 55.23 shows a flowchart example when CMD52 (SDIO none abort) is issued at CMD53 (multiple block write).

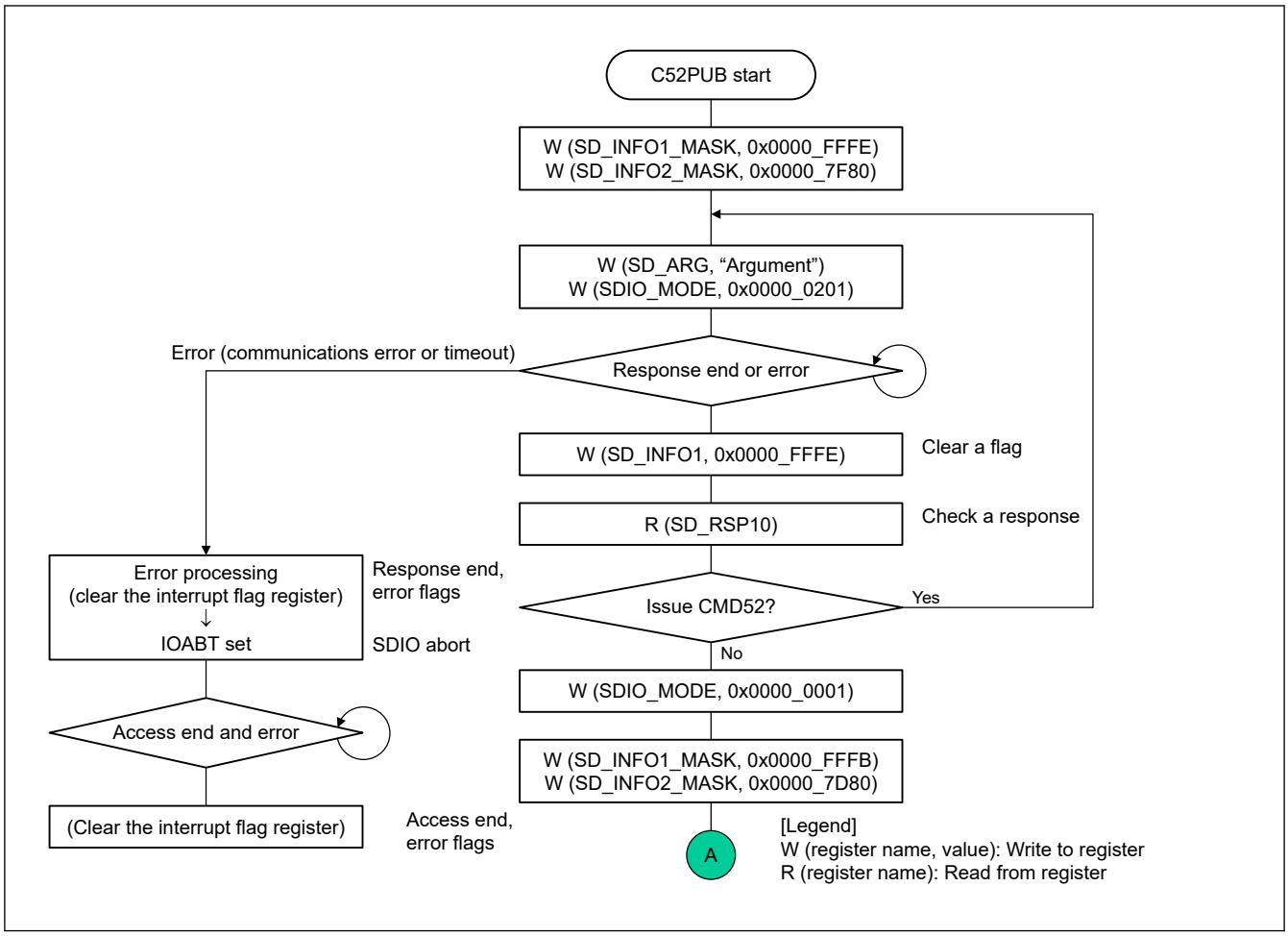
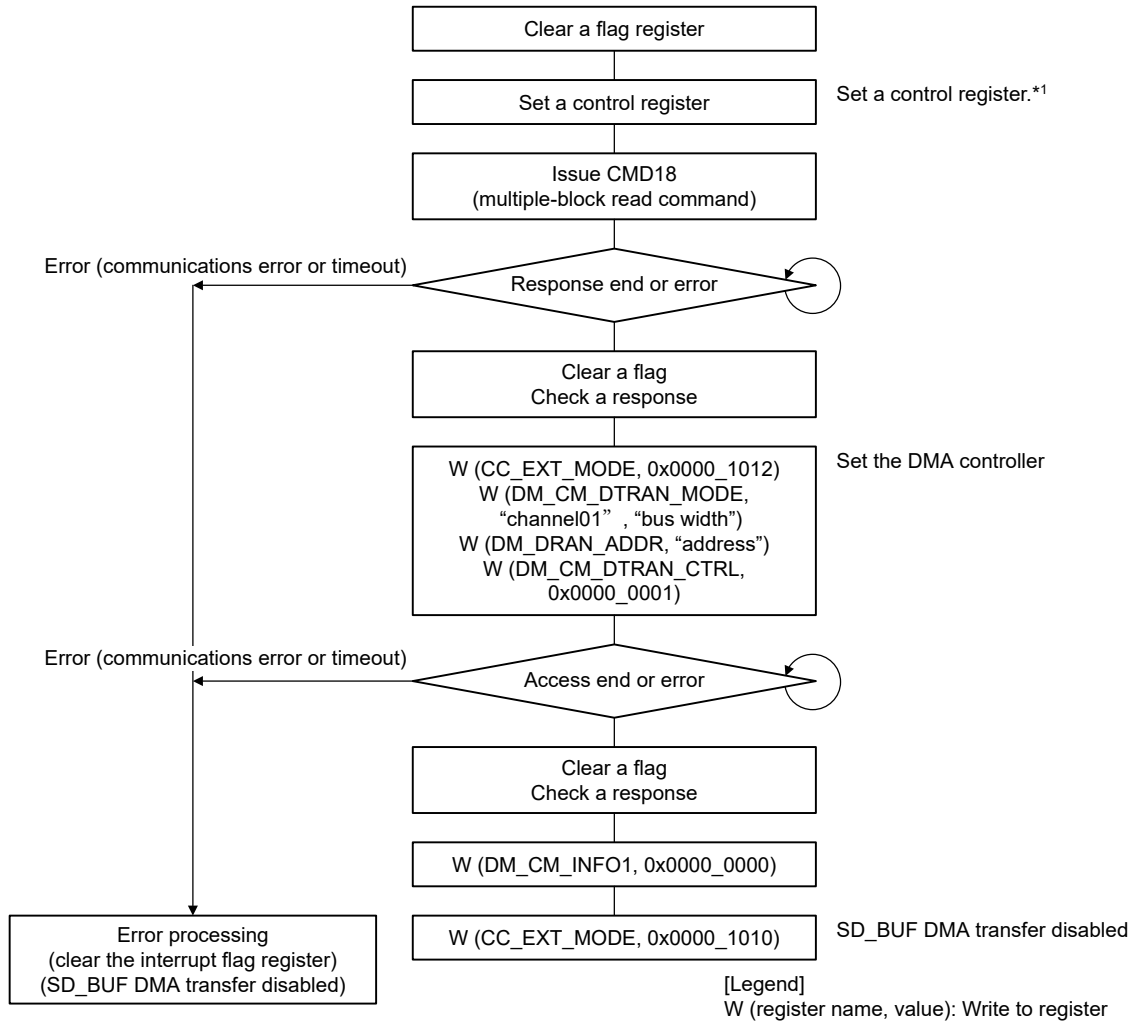


Figure 55.23 Flowchart example when CMD52 (SDIO None Abort) is issued at CMD53 (Multiple Block Write)

55.5.10 DMA Transfer

55.5.10.1 SD_BUF DMA transfer

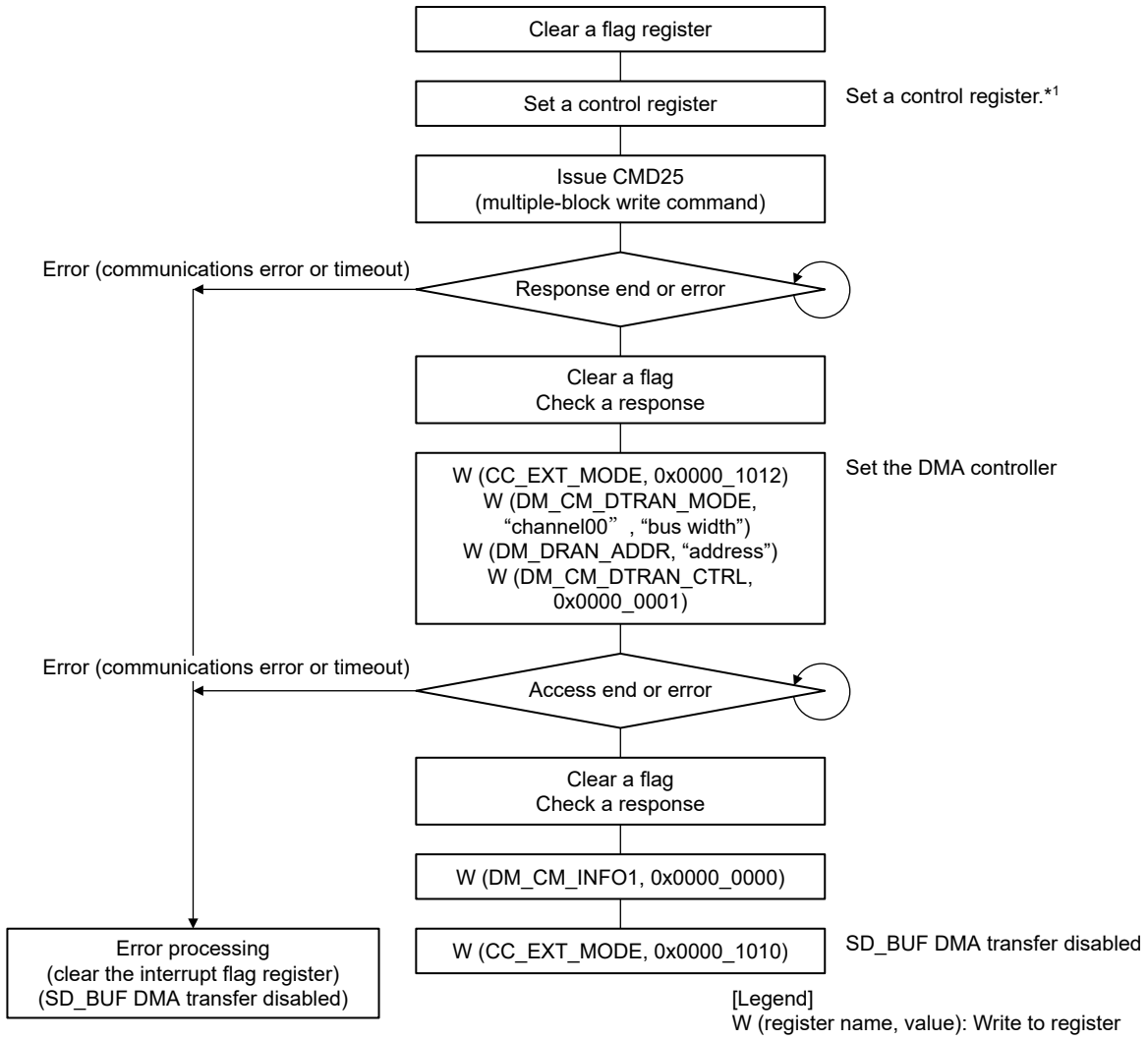
Figure 55.24 shows a flowchart example for SD_BUF DMA read when CMD18 (multiple block read) is issued.



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.24 SD_BUF DMA read flowchart example

Figure 55.25 shows a flowchart example for SD_BUF DMA write when CMD25 (multiple block write) is issued.

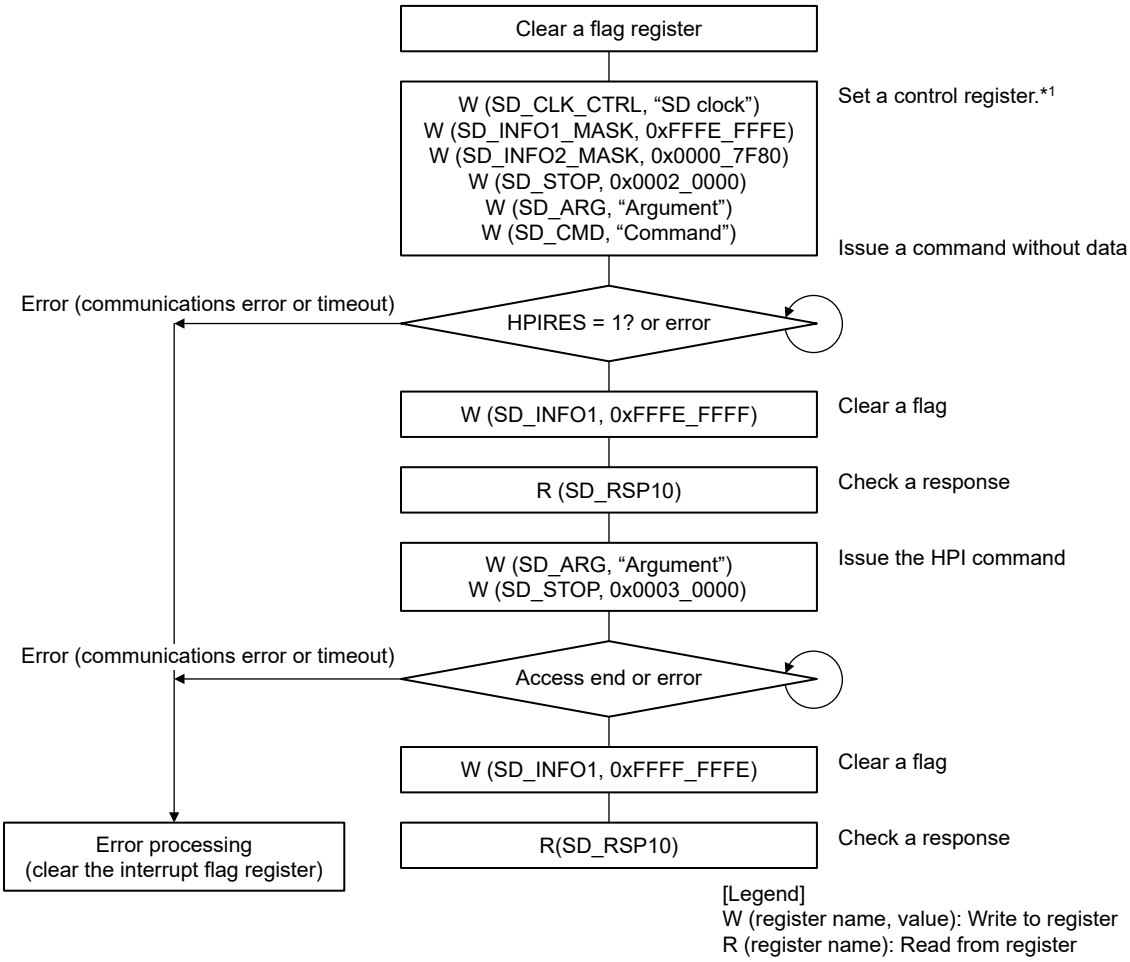


Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.25 SD_BUF DMA write flowchart example

55.5.11 High-Priority Interrupt (without Data Transfer)

55.5.11.1 Flowchart



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.26 Example of the High-Priority Interrupt (without Data Transfer) flowchart

55.5.11.2 Operation for High-Priority Interrupt without Data Transfer

The operation of the high-priority interrupt (HPI) without data transfer is described below.

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), HPI enable, interrupt mask, and so on. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)
3. Command issue
   Set the CMD Argument in SD_ARG and write to SD_CMD.
   Accordingly, CMD is issued, and the operation is started.
4. Flag clear
   On receiving the response, the HPIRES bit in SD_INFO1 is set to 1 to generate an interrupt. Clear the HPIRES bit to 0.
5. Read the response from SD_RSP10.
6. HPI command issue

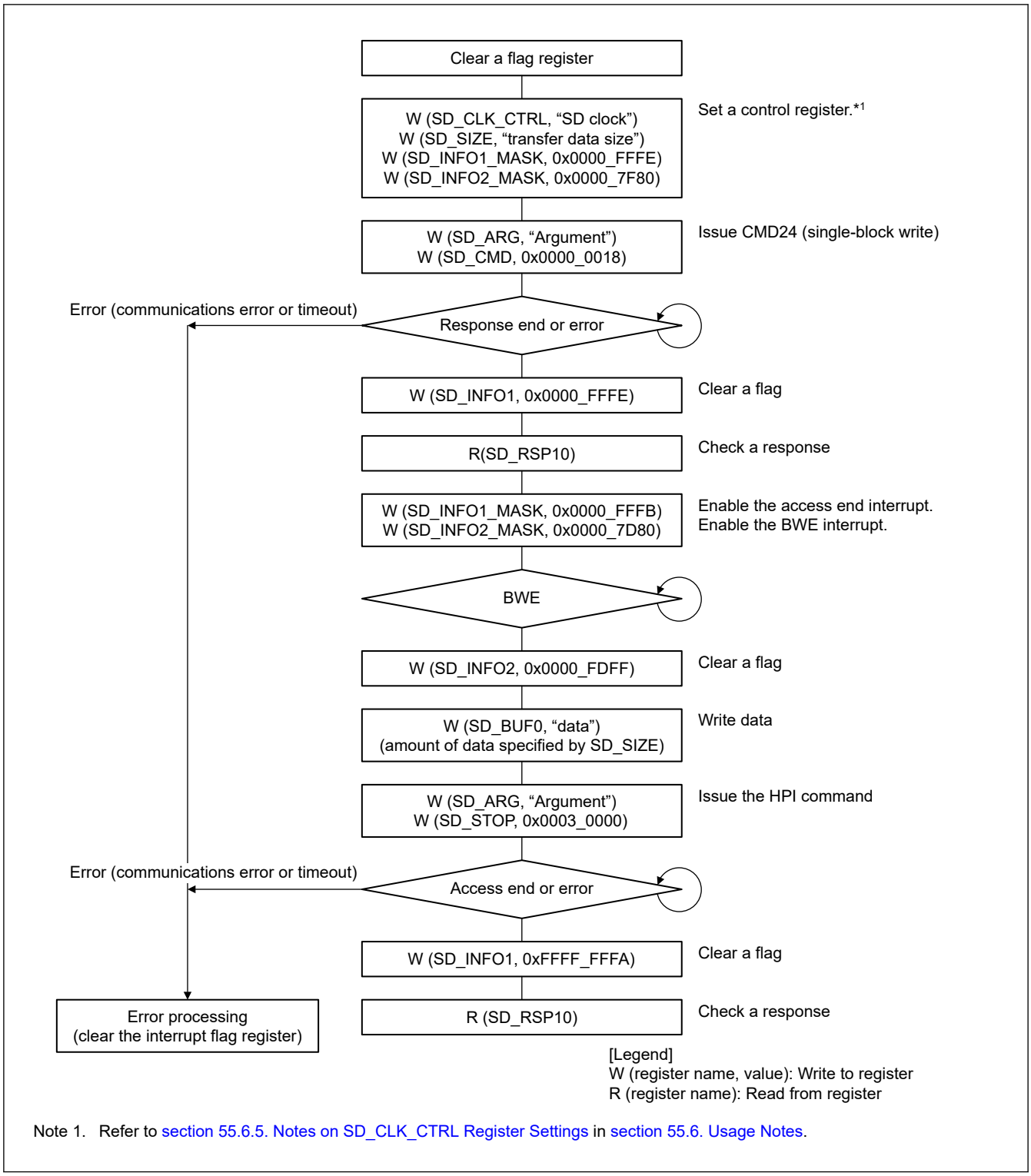
Set the HPI command argument in SD_ARG and set the HPIMODE and HPICMD bits in SD_STOP to 1.

7. Operation complete
   When reception of the response to the HPI command is completed and the busy state is released, the INFO0 bit in SD_INFO1 is set to 1 to generate an interrupt. Clear the INFO0 bit to 0 to read the response from SD_RSP10. This is the end of HPI operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

55.5.12 High-Priority Interrupt (at Single Block Write)

55.5.12.1 Flowchart



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.27 Example of the High-Priority Interrupt (at Single Block Write) flowchart

55.5.12.2 Operation for High-Priority Interrupt at Single Block Write

The operation of the high-priority interrupt (HPI) at single block write is described below.

1. Flag register clear

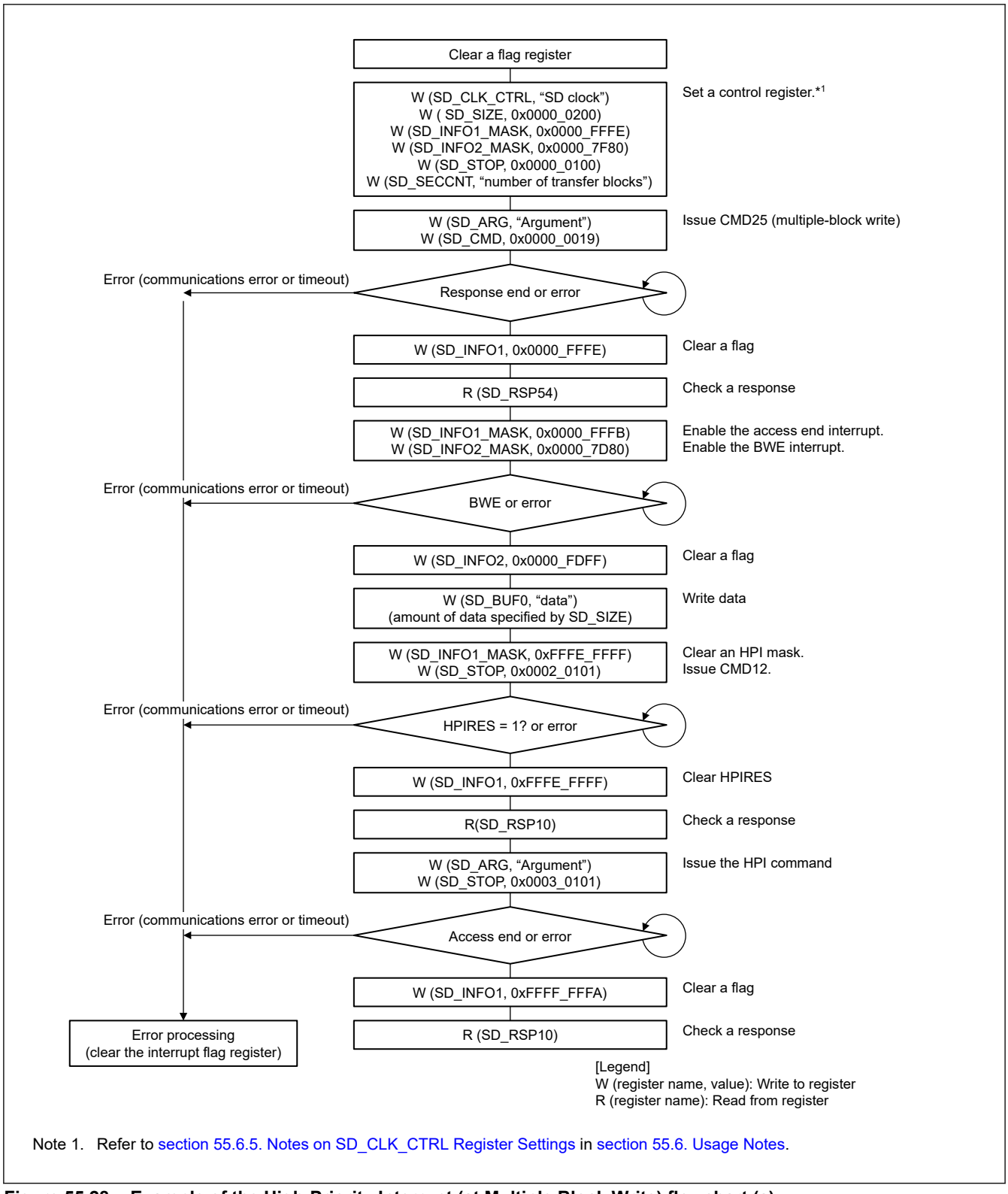
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
3. Command issue (CMD24)
   Set CMD24 Argument in SD_ARG and write 0x0000 0018 to SD_CMD.
   Accordingly, CMD24 is issued, and the single block write operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP10.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. Furthermore, this causes CMD12 and CMD52 to not be issued. If the command sequence is halted, the INFO2 bit (access end) in SD_INFO1 is set to 1 to generate an interrupt.
5. Data write and data transmit to SD card
   Write 0x0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card.
6. HPI command issue
   Set the HPI command argument in SD_ARG and set the HPIMODE and HPICMD bits in SD_STOP to 1.
7. Operation complete
   When reception of the response to the HPI command is completed and the busy state is released, the INFO2 and INFO0 bits in SD_INFO1 are set to 1 to generate an interrupt. Clear the INFO2 and INFO0 bits to 0 to read the response from SD_RSP10. This is the end of HPI operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

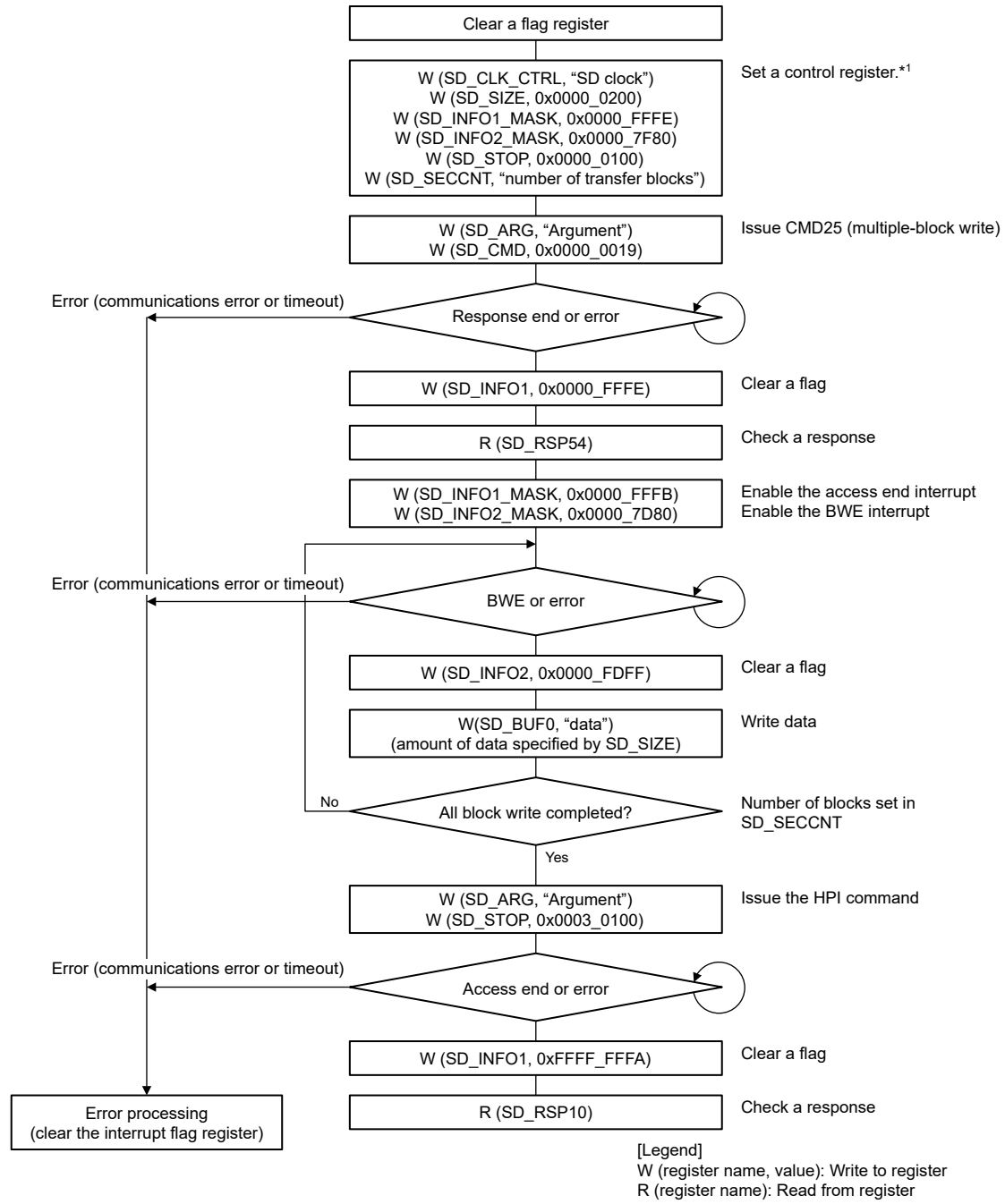
55.5.13 High-Priority Interrupt (at Multiple Block Write)

55.5.13.1 Flowchart



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.28 Example of the High-Priority Interrupt (at Multiple Block Write) flowchart (a)



Note 1. Refer to section 55.6.5. Notes on SD_CLK_CTRL Register Settings in section 55.6. Usage Notes.

Figure 55.29 Example of the High-Priority Interrupt (at Multiple Block Write) flowchart (b)

55.5.13.2 Operation for High-Priority Interrupt at Multiple Block Transfer

The operation of the high-priority interrupt (HPI) at the multiple block write is described below.

(1) When not all the data has been written to SD_BUF

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set

Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

3. Command issue (CMD25)
Set CMD25 Argument in SD_ARG and write 0x0000 0019 to SD_CMD.
Accordingly, CMD25 is issued, and the multiple block write operation is started.

4. Response check
On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP54.
If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit (access end) bit in SD_INFO1 to 1 when reception of the response has been completed. Clear the INFO2 bit to 0 and read the response.

5. Data write and data transmit to SD card
Write 0x0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card.
Then, the CRC status and busy state are received from the SD card.
However, a communications error or timeout may be generated if data are being transmitted while writing to SD_BUF0 is in progress.

6. Command issue (CMD12)
Write 0xFFFE FFFF to SD_INFO1_MASK to enable the HPIRES interrupt. Write 0x0002 0101 to SD_STOP, which causes CMD12 to be issued.

7. Response check
When the response is received, the HPIRES bit in SD_INFO1 is set to 1 to generate an interrupt. Clear the HPIRES bit to 0 to read the response from SD_RSP10.

8. HPI command issue (CMD25)
Set the HPI command argument in SD_ARG and set the HPIMODE and HPICMD bits in SD_STOP to 1.

9. Operation complete
When reception of the response to the HPI command is completed and the busy state is released, the INFO2 and INFO0 bits in SD_INFO1 are set to 1 to generate an interrupt. Clear the INFO2 and INFO0 bits to 0 to read the response from SD_RSP10. This is the end of HPI operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

(2) When all the data has been written to SD_BUF

1. Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

2. Control register set
Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

3. Command issue (CMD25)
Set CMD25 Argument in SD_ARG and write 0x0000 0019 to SD_CMD.
Accordingly, CMD25 is issued, and the multiple block write operation is started.

4. Response check
On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP54.
If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit

(access end) bit in SD_INFO1 to 1 when reception of the response has been completed. Clear the INFO2 bit to 0 and read the response.

5. Data write and data transmit to SD card
   Write 0x0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card.
   Then, the CRC status and busy state are received from the SD card. Doing this repeats transfer of the number of blocks set in SD_SECCNT.
   However, a communications error or timeout may be generated if data are being transmitted while writing to SD_BUF0 is in progress.
6. HPI command issue
   Set the HPI command argument in SD_ARG and set the HPIMODE and HPICMD bits in SD_STOP to 1.
7. Operation complete
   When reception of the response to the HPI command is completed and the busy state is released, the INFO2 and INFO0 bits in SD_INFO1 are set to 1 to generate an interrupt. Clear the INFO2 and INFO0 bits to 0 to read the response from SD_RSP10. This is the end of HPI operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

55.5.14 Example of SD_CMD Register Setting

Table 55.9 lists the example of SD_CMD (SD interface) register setting.

Table 55.9 Example of SD_CMD register setting (SD) (1 of 2)

Type	Command	Example of SD_CMD register setting	Remark
CMD	CMD0	0x0000 0000	
	CMD2	0x0000 0002	
	CMD3	0x0000 0003	
	CMD4	0x0000 0004	
	CMD5	0x0000 0705 or 0x0000 0005	
	CMD6	0x0000 1C06 or 0x0000 0006	
	CMD7	0x0000 0007	When the card is placed in the deselected state, the response timeout flag will be set since there is no response.
	CMD8	0x0000 0408 or 0x0000 0008	
	CMD9	0x0000 0009	
	CMD10	0x0000 000A	
	CMD11	0x0000 040B or 0x0000 000B	
	CMD12	0x0000 000C	
	CMD13	0x0000 000D	
	CMD15	0x0000 000F	
	CMD16	0x0000 0010	
	CMD17	0x0000 0011	
	CMD18	0x0000 0012	With auto CMD12 enabled (other than in SDR104 mode)
		0x0000 7C12	With auto CMD12 disabled (only in SDR104 mode)
	CMD19	0x0000 1C13 or 0x0000 0013	Prohibited in DDR50 mode.
	CMD20	0x0000 0514 or 0x0000 0014	
	CMD23	0x0000 0417 or 0x0000 0017	

Table 55.9 Example of SD_CMD register setting (SD) (2 of 2)

Type	Command	Example of SD_CMD register setting	Remark
CMD	CMD24	0x0000 0018	
	CMD25	0x0000 0019	With auto CMD12 enabled (other than in SDR104 mode)
		0x0000 6C19	With auto CMD12 disabled (only in SDR104 mode)
	CMD27	0x0000 001B	
	CMD28	0x0000 001C	
	CMD29	0x0000 001D	
	CMD30	0x0000 001E	
	CMD32	0x0000 0020	
	CMD33	0x0000 0021	
	CMD38	0x0000 0026	
	CMD42	0x0000 002A	
	CMD48	0x0000 1C30	
	CMD49	0x0000 0C31	
	CMD52	0x0000 0434 or 0x0000 0034	
	CMD53	0x0000 1C35	Single read
		0x0000 0C35	Single write
		0x0000 7C35	Multiple read
		0x0000 6C35	Multiple write
		0x0000 0035	The value on the left can be set irrespective of whether single or multi. However, the CF39 bit in SD_ARG must be set as follows. Read: 0 Write: 1
	CMD55	0x0000 0037	
	CMD56	0x0000 0038	
	CMD58	0x0000 7C3A	
	CMD59	0x0000 6C3B	
ACMD	ACMD6	0x0000 0046	
	ACMD13	0x0000 004D	
	ACMD22	0x0000 0056	
	ACMD23	0x0000 0057	
	ACMD41	0x0000 0069	
	ACMD42	0x0000 006A	
	ACMD51	0x0000 0073	

Table 55.10 lists the example of SD_CMD (MMC interface) register setting.

Table 55.10 Example of SD_CMD register setting (MMC) (1 of 2)

Type	Command	Example of SD_CMD Register Setting	Remark
CMD	CMD0	0x0000 0000	
	CMD1	0x0000 0701	
	CMD2	0x0000 0002	
	CMD3	0x0000 0003	

Table 55.10 Example of SD_CMD register setting (MMC) (2 of 2)

Type	Command	Example of SD_CMD Register Setting	Remark
CMD	CMD4	0x0000 0004	
	CMD5	0x0000 0505	
	CMD6	0x0000 0506	In the response busy state
		0x0000 0406	Not in the response busy state
	CMD7	0x0000 0007	When the card is placed in the deselected state, the response timeout flag will be set since there is no response.
	CMD8	0x0000 1C08	
	CMD9	0x0000 0009	
	CMD10	0x0000 000A	
	CMD12	0x0000 000C	
	CMD13	0x0000 000D	
	CMD14	0x0000 1C0E	SDIF_MODE must be set to 0x0100 (with CRC16 disabled).
	CMD15	0x0000 000F	
	CMD16	0x0000 0010	
	CMD17	0x0000 0011	
	CMD18	0x0000 7C12	Pre-defined
	CMD19	0x0000 0C13	SDIF_MODE must be set to 0x0100 (with CRC16 disabled).
	CMD21	0x0000 1C15	Setting prohibited in DDR mode
	CMD23	0x0000 0017	
	CMD24	0x0000 0018	
	CMD25	0x0000 6C19	Pre-defined
	CMD26	0x0000 0C1A	
	CMD27	0x0000 001B	
	CMD28	0x0000 001C	
	CMD29	0x0000 001D	
	CMD30	0x0000 001E	
	CMD31	0x0000 1C1F	
	CMD35	0x0000 0423	
	CMD36	0x0000 0424	
	CMD38	0x0000 0026	
	CMD39	0x0000 0427	
	CMD40	0x0000 0428	
	CMD42	0x0000 002A	
	CMD49	0x0000 0C31	
	CMD53	0x0000 7C35	
	CMD54	0x0000 6C36	
	CMD55	0x0000 0037	
	CMD56	0x0000 0038	

55.6 Usage Notes

55.6.1 SD_BUF Illegal Write Access

When writing data to SD_BUF0 after the single block write or multi block write command is issued, the data of the size specified by SD_SIZE must be written to.

If the data of the size which exceeds the size specified by SD_SIZE is written to, the ERR4 bit in SD_INFO2 is set to 1. In addition, the data written to SD_BUF0 may not be transmitted and it causes the SCLKDIVEN bit in SD_INFO2 to hold the value of 0. In such cases, clearing the SDRST bit in SOFT_RST to 0 and then restoring its value to 1 clears the SCLKDIVEN bit to 1.

However, for the single byte (in the case of 16- or 32-bit access) or three bytes (in the case of 32-bit access) when the number of bytes setting in SD_SIZE is odd, or the fraction of bytes when the number of bytes setting in SD_SIZE is even (in the case of 32-bit access), since the portion of dummy data writing is regarded as excess data and ignored, it is not within the scope of the above description (the fraction of bytes: the two bytes that are not in a four-byte unit).

55.6.2 Block Number Limitation for Multiple Block Read

When performing a multiple block read of one or two blocks, depending on the timing with which the response register is read, the response value may not be read properly. This must be avoided by either of the following countermeasures.

1. When receiving one or two blocks of data, use single block reading.
2. Read the response to CMD18 from SD_RSP54.

[Mechanism of incorrect reading]

Figure 55.30 shows the processing flows of SD/MMC host interface (hardware) operation and software operation when a multiple block read is performed on two blocks. As shown in the incorrect operation of Figure 55.30, when an interrupt is generated on reception of the CMD18 response and the timing with which the SD card response register (SD_RSP10) is read by the interrupt is delayed, the data during the CMD12 response reception or the CMD12 response may be read. In the case of a multiple block read of three or more blocks, CMD12 is not issued until the block of data has been read, so this problem does not arise. Furthermore, in the case of a multiple block write, since the CMD25 response is read before the block of data is sent, the problem does not arise.

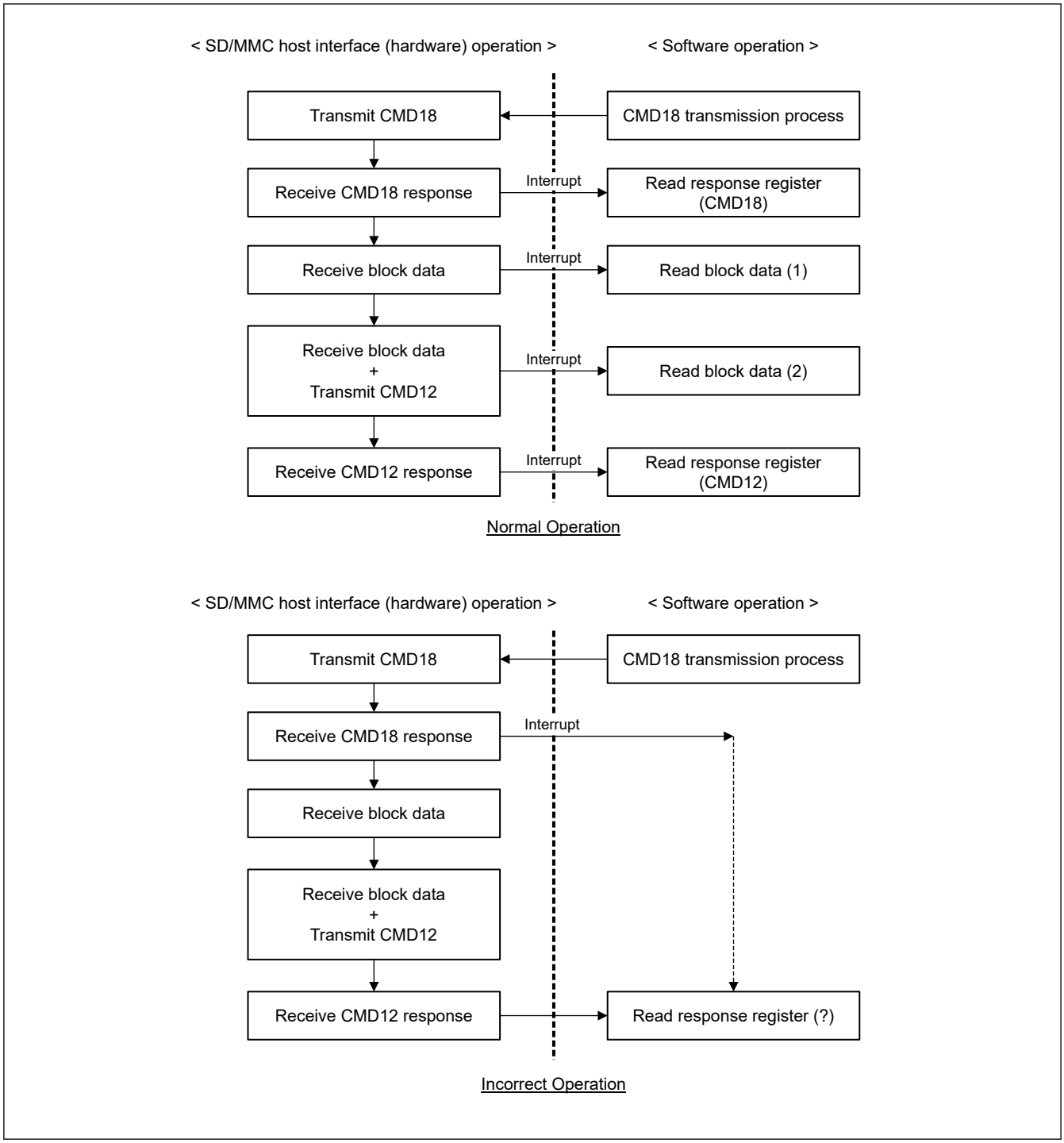


Figure 55.30 Flowcharts for multiple block read operation (two blocks)

55.6.3 Automatic Control of SDCLK Output

In the SD Card standard, 74 cycles of SDCLK must be output before initialization of the card. For this reason, use automatic control of SDCLK output after 74 cycles of SDCLK have been output. Furthermore, if automatic control of SDCLK output was in use, SDCLK output is stopped on completion of the sequence for a communications error or timeout. Thus, in cases where state transitions within the SD card are necessary and so on after completion of the sequence, release automatic control of SDCLK output and restart supply of SDCLK to the SD card.

55.6.4 Control of the C52PUB Setting for Multiple Block Write

If the C52PUB bit in SDIO_MODE is set to 1 during a sequence of multiple block write due to CMD53, CMD52 is not issued until SD_BUF becomes empty. For this reason, set the C52PUB bit after suspending writing to SD_BUF by following the appropriate procedure below.

- When DMA transfer is not in use

1. Before setting the C52PUB bit, suspend writing to SD_BUF by making the setting in SD_INFO2 to disable BWE interrupts.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the INFO0 interrupt processing in SD_INFO1 due to the issuing of CMD52 has been completed, restart writing to SD_BUF by making the setting in SD_INFO2 to enable BWE interrupts.

- When DMA transfer is in use

1. Every time DMA transfer of the value set in SD_SIZE × n blocks (where n = 1, 2, ...) proceeds, suspend writing to SD_BUF by DMA transfer before the C52PUB bit is set.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the INFO0 interrupt processing in SD_INFO1 due to the issuing of CMD52 has been completed, restart writing to SD_BUF by DMA transfer.

55.6.5 Notes on SD_CLK_CTRL Register Settings

When the SCLKDIVEN bit in SD_INFO2 is 0, SD_CLK_CTRL cannot be written to. Before writing to SD_CLK_CTRL, be sure to check that the SCLKDIVEN bit in SD_INFO2 is 1.

55.6.6 Restrictions on Specifications

1. The SDIO suspend/resume is not supported.
2. The SPI bus is not supported.
3. The shared bus and 8-bit SD bus for embedded SDIO are not supported.
4. The stream transfer for MMC cards is not supported.
5. Open ended multiple block transfer for MMC is not supported.

55.6.7 STP Bit Setting during Multiple Block Read

During execution of multiple block read with automatic CMD12 execution by setting the SEC bit in SD_STOP to 1, even if the STP bit in SD_STOP is set to 1 to forcibly stop the execution, the command sequence may not stop depending on the timing of setting the STP bit.

To avoid this, when setting the STP bit in SD_STOP to 1 during multiple block transfer, clear the SEC bit in SD_STOP to 0 at the same time. (Even when the SCLKDIVEN bit in SD_INFO2 is 0, change the SEC bit from 1 to 0.)

When the command sequence has not stopped because the SEC bit was not cleared to 0, the command sequence can be stopped by clearing the SDRST bit in SOFT_RST to 0.

When forcibly terminating the CMD53 multiple block transfer through the IOABT bit in SDIO_MODE, be sure to leave the SEC bit in SD_STOP as 1.

55.7 Sampling Clock Controller (SCC)

55.7.1 Features

This module controls a sampling clock (hereafter referred to as the SCC sampling clock) that is used for SD UHS-I/SDR104 and MMC HS200. When this module is used with the LSI, SD UHS-I/SDR104 and MMC HS200 can be supported.

55.7.2 SCC Block Diagram

Figure 55.31 shows a block diagram of the sampling clock controller.

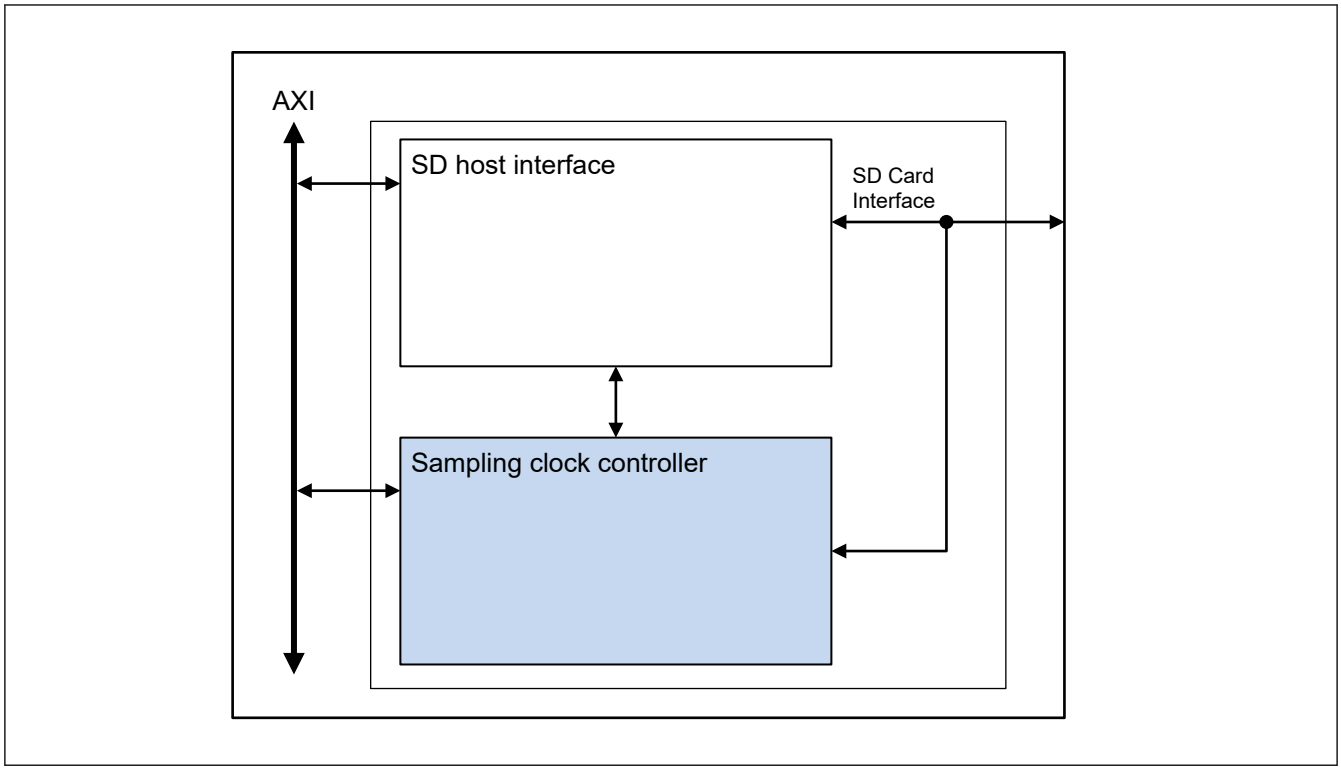


Figure 55.31 Block diagram of the sampling clock controller

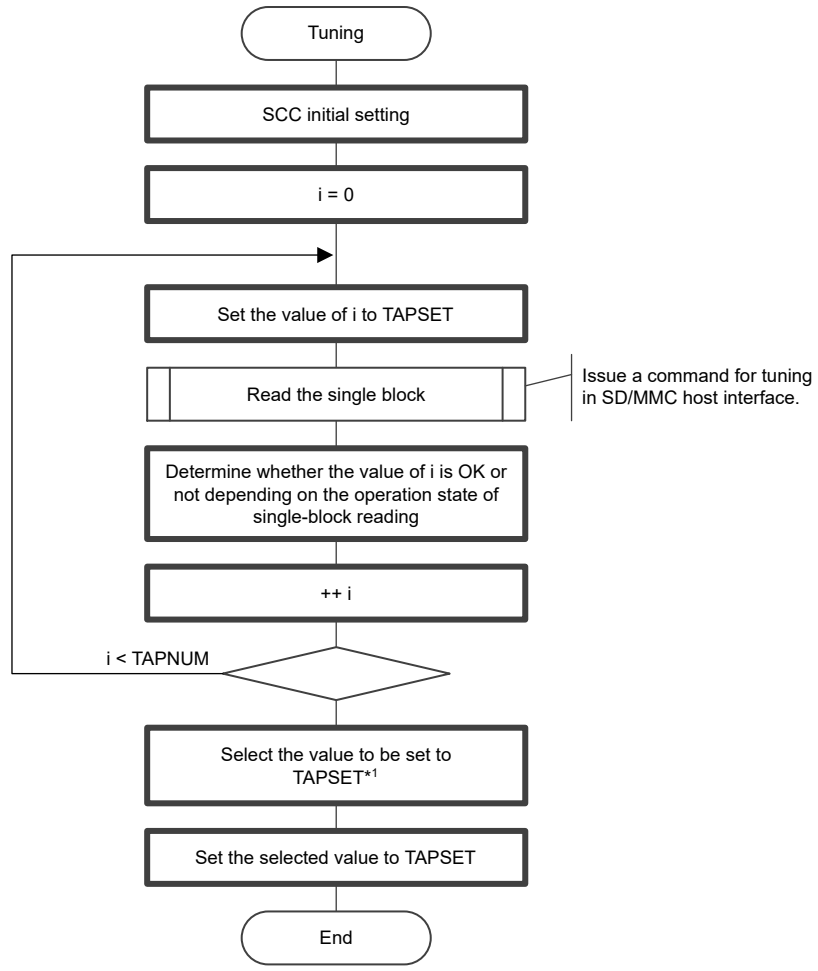
55.8 Usage Example of SCC

55.8.1 Tuning

SCC is tuned by using operation of single-block reading.

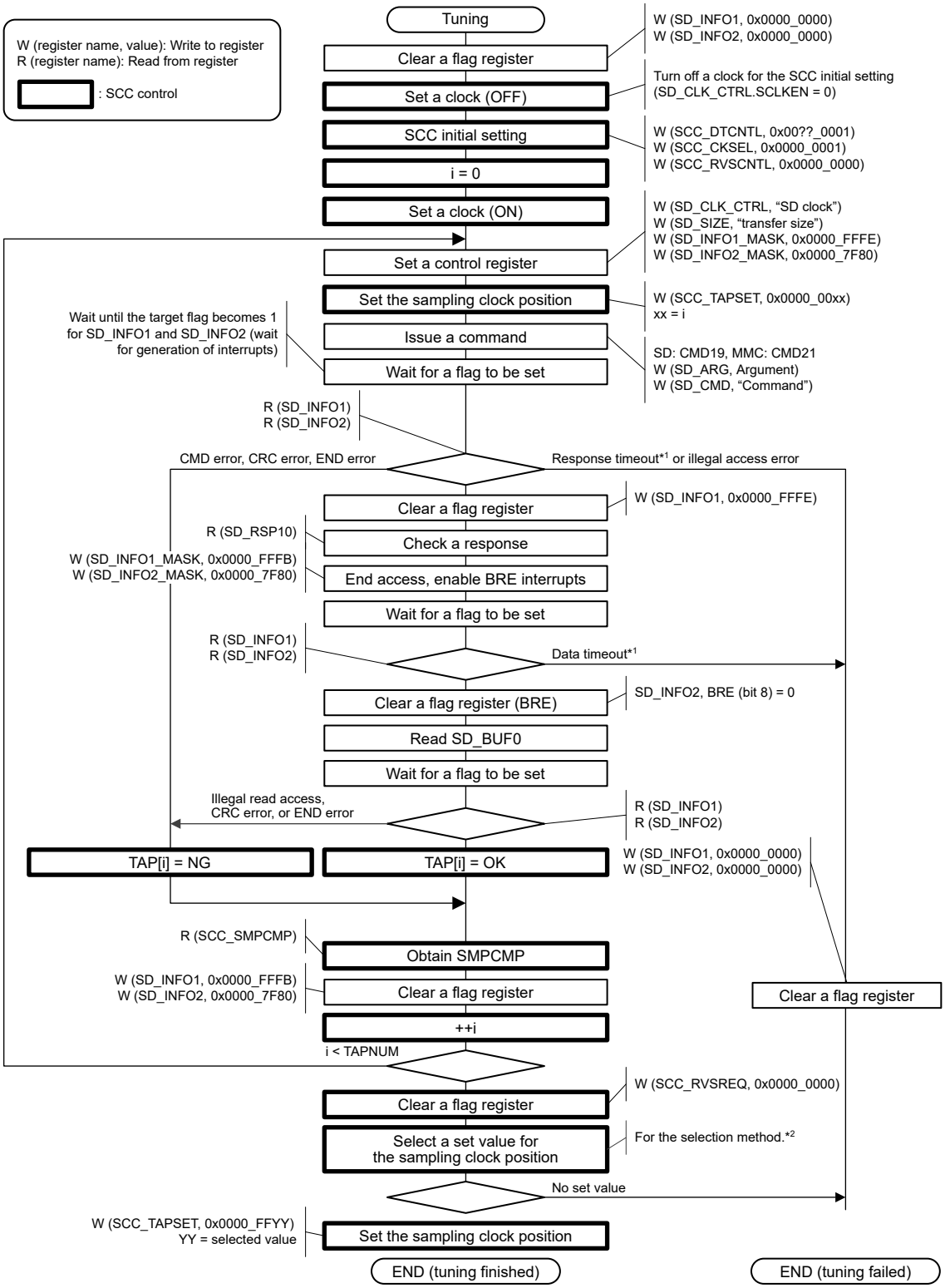
As shown in Figure 55.32, check whether the single-block read command normally ends when the sampling clock position is changed from 0 to TAPNUM − 1 and save the result. After checking, confirm that there exists the range which has three or more continuous normal ends (OK). Then, the median value within the continuous range is determined as the final adjustment value.

Figure 55.33 and Table 55.11 show the detailed tuning flow and the method how to select the sampling clock position (example when TAPNUM = 8).



Note 1. See the example shown in Table 55.11.

Figure 55.32 Example of tuning flow (outline)



Note 1. If all of the TAP[i] has become "NG".
Note 2. See Table 55.11.

Figure 55.33 Example of tuning flow (detailed)

Table 55.11 Example of the method how to select the sampling clock position (when TAPNUM = 8)

Item	i	Case 1	Case 2	Case 3	Case 4	Case 5*1
TAP[i]	0	NG	OK	NG	OK	OK
	1	OK	OK	NG	NG	OK
	2	OK	NG	OK	NG	OK
	3	OK (←)	NG	OK	NG	OK
	4	OK	NG	NG	NG	OK
	5	OK	OK	NG	OK	OK
	6	NG	OK	NG	OK (←)	OK
Max. value →	7	NG	OK (←)	NG	OK	OK
	(0)	NG	OK	NG	OK	OK
	(1)	OK	OK	NG	NG	OK
	(2)	OK	NG	OK	NG	OK
	(3)	OK	NG	OK	NG	OK
	(4)	OK	NG	NG	NG	OK
	(5)	OK	OK	NG	OK	OK
	(6)	NG	OK	NG	OK	OK
	(7)	NG	OK	NG	OK	OK
Selected value		i = 3	i = 7	Fail	i = 6 or 7	i = 0 to 7

Note: (←): Example of selection, (x): repeated display of index x of TAP[x]
(a) The sampling clock position is selected by considering a margin in the range which has three or more continuous 'TAP[i] = OK'.
(b) The sampling clock position is repeated from 0 after the maximum value (TAPNUM − 1). In case 2 above, that position is continued in the order of 5→6→7→0→1.

Note 1. If all of the TAP [i] is OK, the sampling clock position is selected by identifying the change point of data.
Change point of the data can be found in the value of SCC_SMPCMP register. Usage example is section 55.8.3. Change Point of the Input Data.

55.8.2 Sampling Clock Position Correction after Tuning

After tuning, correction of the sampling clock position may be required when a command is issued.

There are manual and automatic correction methods. After a command sequence, if the CMD, CRC, END error or time out occurs or the correction error occurs, tuning will be performed again. The following shows examples of manual and automatic correction methods.

55.8.2.1 Manual Correction of the Sampling Clock Position

Figure 55.34 shows the flow of manual correction of the sampling clock position. Table 55.12 shows set values determined when correction is required (when TAPNUM = 8).

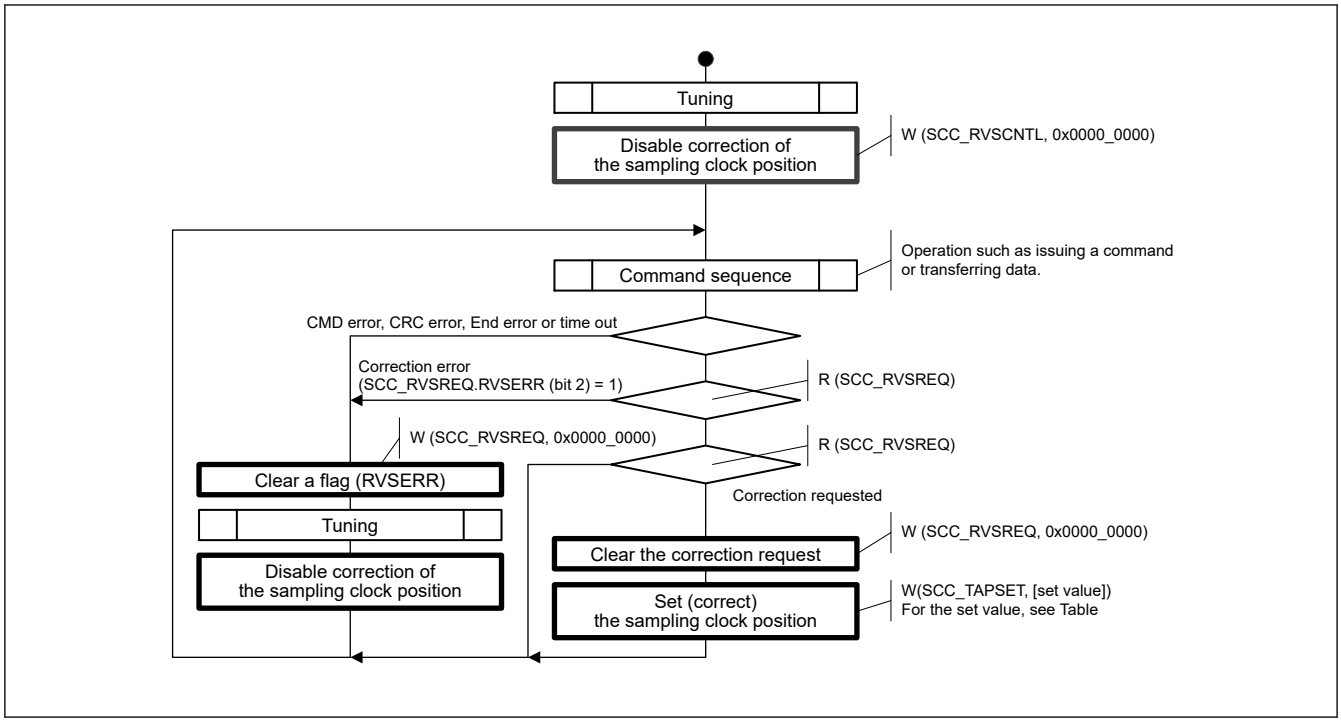


Figure 55.34 Flow of manual correction of the sampling clock position (example)

Table 55.12 Set values for TAPSET when correction is required (when TAPNUM = 8)

No.	Current value of TAPSET	Value set to TAPSET when REQTAPUP = 1	Value set to TAPSET when REQTAPDWN = 1
1	0	1	7
2	1	2	0
3	2	3	1
4	3	4	2
5	4	5	3
6	5	6	4
7	6	7	5
8	7	0	6

Note: As is the case in the tuning selection method, the sampling clock position is 0 after the maximum value (TAPNUM − 1).

55.8.2.2 Automatic Correction of the Sampling Clock Position

Figure 55.35 shows the flow of automatic correction of the sampling clock position.

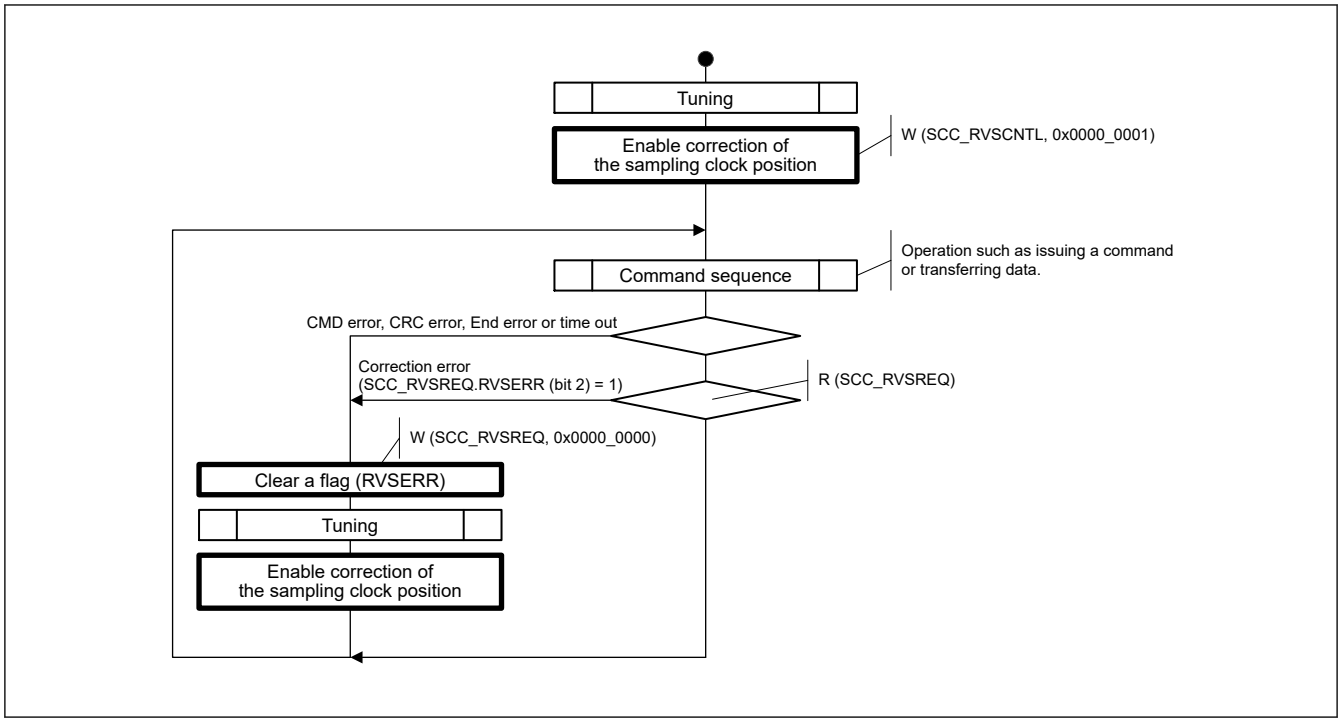
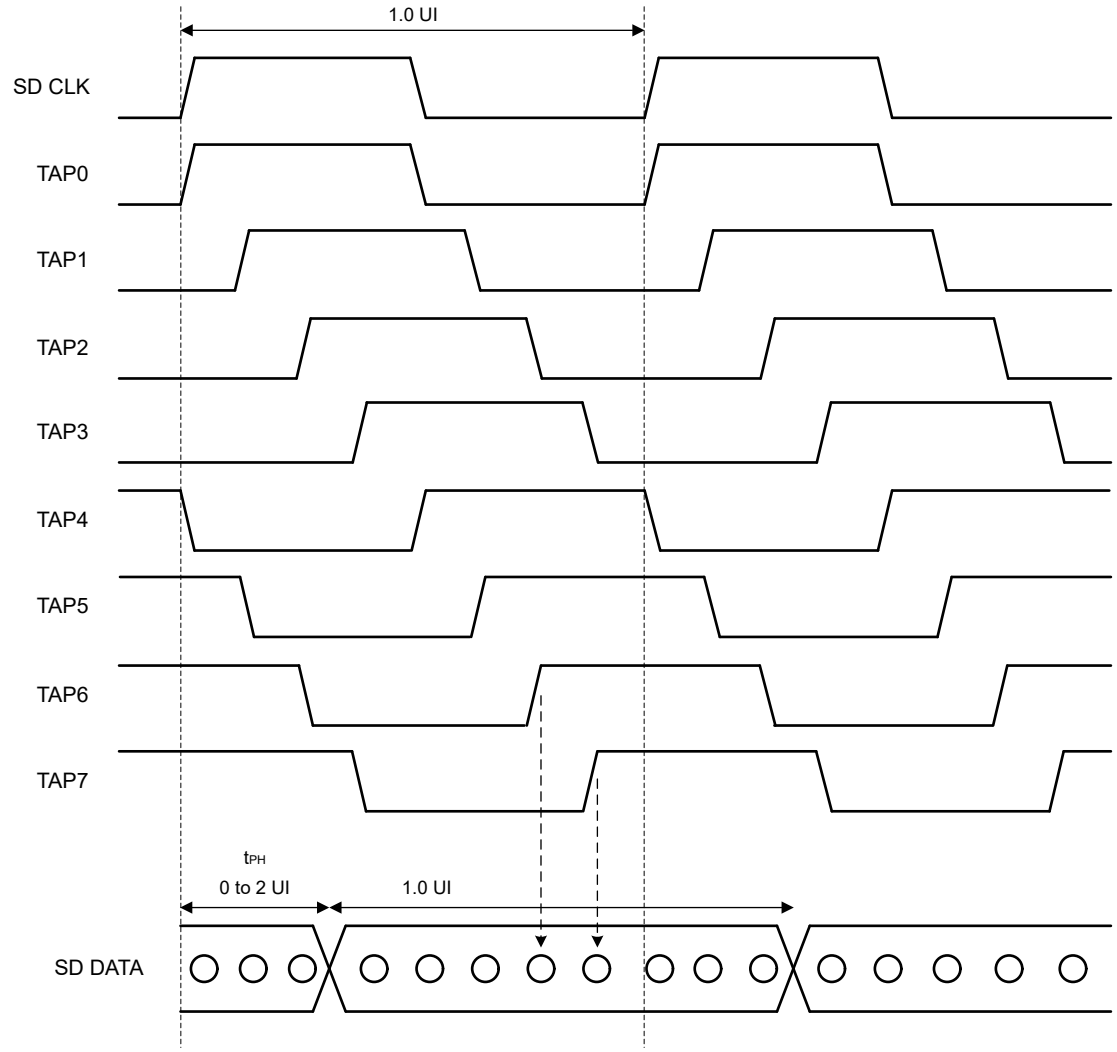


Figure 55.35 Flow of automatic correction of the sampling clock position (example)

55.8.3 Change Point of the Input Data

Tuning is capture the data by the TAP clock selected. However, also captures the data by the previous TAP clock and the behind the TAP clock at the same time. This result is reflected in the sampling data comparison register (SCC_SMPCMP). Point of mismatch before and after the selected TAP clock is the changing point of the data. In this example, it is desirable to set as TAP6 or TAP7.



Note: After recognizing the changing point of the input data, need to set the TAP clock.

Figure 55.36 Example, All taps is OK.

Figure 55.36 is shown change point of data between TAP2 and TAP3. Change point of the data can be confirmed by sampling data comparison register (SCC_SMPCMP). When the tuning of TAP2 or TAP3, CMPNGU bit of sampling data comparison register indicates a mismatch. As the width of the input data is 1 (UI), select TAP6 or TAP7 which is the median of next TAP3 from TAP3.

56. PCI Express 3.0 Interface (PCIE)

56.1 Overview

This module has a dual structure so that the device incorporating it can operate as a root complex or endpoint device, and includes a Type 0/1 Configuration Register for this purpose. Furthermore, it has an internal DMA controller. However, limitations imposed by the AXI place restrictions on some specifications of PCI Express functionality.

Table 56.1 PCIE specifications

Item	Specification
Number of units	• 1 unit
Lane/channel configuration	• 1 lane / 2 channels or 2 lanes / 1 channel selectable • 1 lane / 2 channels: Common / Separate reference clock support • 2 lanes / 1 channel: Common reference clock support only
Support Specification	• PCI Express Base Specification 3.1 Compliant
Functions	• PCI Express Gen1 (2.5 [GT/s]) / Gen2 (5.0 [GT/s]) / Gen3 (8.0 [GT/s]) • Root Complex / Endpoint Applications, Type0/1 Configuration Register • Lane implementation: ×1 or ×2 • Support Polarity inversion • Maximum data payload of 256 bytes, Maximum read request size 512 bytes • Not support for Virtual channels (support VC0 only) • Number of outstanding: 8 • Dynamic control of speed/width up/down configuration • Not support for Clock Power Management (not support P1.CPM, P2.CPM) • Power Management (not support ASPM L1-Substate) • Error handling/logging (AER Support) • Replay FIFO with ECC • Internal Memory without Parity • Number of Support Functions: 2
Internal interface	• AMBA® AXI Protocol • Master interface: 128-bit width 1 port, or 2 ports (when multilink is selected) • Slave interface: 128-bit width 1 port, or 2 ports (when multilink is selected) • DMAC – Register control and descriptor control support – 8 channels – Allowable number of requests to be issued: Maximum total 8 for all the channels – PCIe MRd: 8 / ch, PCIe MWr: 1 / ch – AXI Read: 1 / ch, AXI Write: 1 / ch
Module-stop function	• Module-stop state can be set to reduce power consumption.

The initial values of configuration registers for vendor ID, device ID, revision ID, class code, subsystem vendor ID, subsystem ID, and base address register mask are 0. Set appropriate values in the registers before the start of link up.

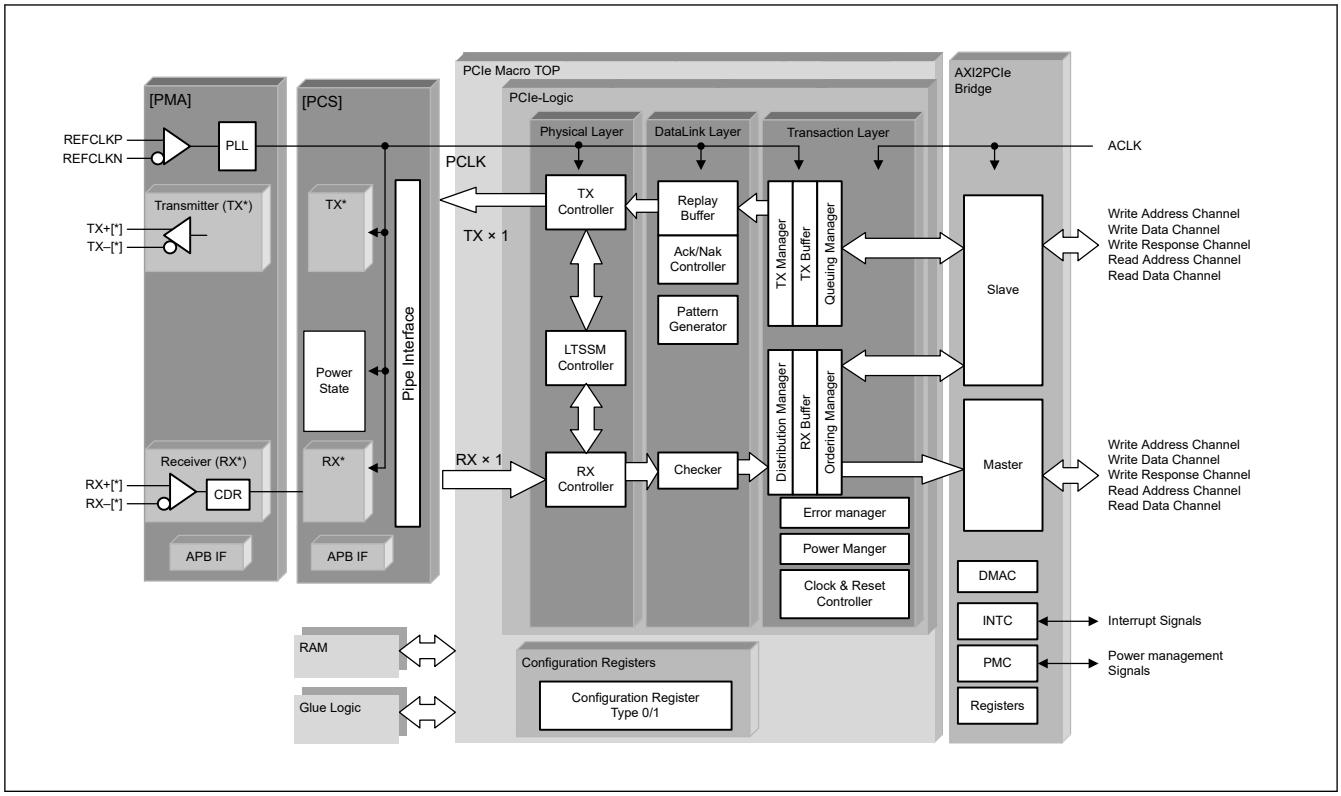


Figure 56.1 PCIE block diagram

Table 56.2 PCIE input/output pins

Pin name	I/O	Function
PCIE_REFCLK_P0	Input	Ch0 Reference clock input (Positive)
PCIE_REFCLK_N0	Input	Ch0 Reference clock input (Negative)
PCIE_REFCLK_P1	Input	Ch1 Reference clock input (Positive)
PCIE_REFCLK_N1	Input	Ch1 Reference clock input (Negative)
PCIE_RXDP_L0	Input	Serial data input 0 (Positive)
PCIE_RXDN_L0	Input	Serial data input 0 (Negative)
PCIE_RXDP_L1	Input	Serial data input 1 (Positive)
PCIE_RXDN_L1	Input	Serial data input 1 (Negative)
PCIE_TXDP_L0	Output	Serial data output 0 (Positive)
PCIE_TXDN_L0	Output	Serial data output 0 (Negative)
PCIE_TXDP_L1	Output	Serial data output 1 (Positive)
PCIE_TXDN_L1	Output	Serial data output 1 (Negative)
PCIE_RSTOUT0B	Output	PCIE Reset output ch0 for Root Complex
PCIE_RSTOUT1B	Output	PCIE Reset output ch1 for Root Complex

Table 56.3 PCIE Interrupt sources (1 of 2)

Name	Interrupt Sources
PCIEn_INTA_RC	INTA interrupt for Root complex
PCIEn_INTB_RC	INTB interrupt for Root complex
PCIEn_INTC_RC	INTC interrupt for Root complex
PCIEn_INTD_RC	INTD interrupt for Root complex
PCIEn_INTMSI_RC	MSI interrupt for Root complex

Table 56.3 PCIE Interrupt sources (2 of 2)

Name	Interrupt Sources
PCIEn_INT_LINK_BANDWIDTH	Link width change interrupt for Root complex
PCIEn_INT_EQUALIZATION_REQUEST	Link Equalization Request interrupt for Root complex
PCIEn_INT_PM_PME	PM_PME Message receive interrupt for Root complex
PCIEn_INT_SERR	Error detect interrupt for Root complex
PCIEn_INT_SERR_COR	Correctable error detect interrupt for Root complex
PCIEn_INT_SERR_FATAL	Fatal error detect for Root complex
PCIEn_INT_SERR_NONFATAL	Non fatal error detect interrupt for Root complex
PCIEn_INT_DMA	Event interrupt of DMA
PCIEn_INT_PCIE_EVT	Event interrupt of PCIE
PCIEn_INT_AXI_ERR	Error detect interrupt of AXI
PCIEn_INT_MSG	Message receive interrupt
PCIEn_INT_ALL	ORed interrupt of all interrupt
PCIEn_TURN_OFF_EVENT	PME_Turn_Off Msg. receive flag output for Endpoint
PCIEn_PMU_POWEROFF	POWEROFF indication on L2
PCIEn_D3_EVENT_F0	Non-D0 State transition request receive output Function #0 for Endpoint
PCIEn_D3_EVENT_F1	Non-D0 State transition request receive output Function #1 for Endpoint
PCIEn_CFG_PMCSR_PME_STATUS_WRITECLEAR_F0	PME_STATUS clear output Function #0 for Endpoint
PCIEn_CFG_PMCSR_PME_STATUS_WRITECLEAR_F1	PME_STATUS clear output Function #1 for Endpoint
PCIEn_FLR_REQ_F0	FLR request Function #0 for Endpoint
PCIEn_FLR_REQ_F1	FLR request Function #1 for Endpoint

Note: n = 0, 1

56.1.1 Multi Link: Configuration of 1 Lane × 1

Figure 56.2 shows the relationship between the modules and lanes used in the configuration of 2 lanes × 1 channel and 1 lane × 2 channels.

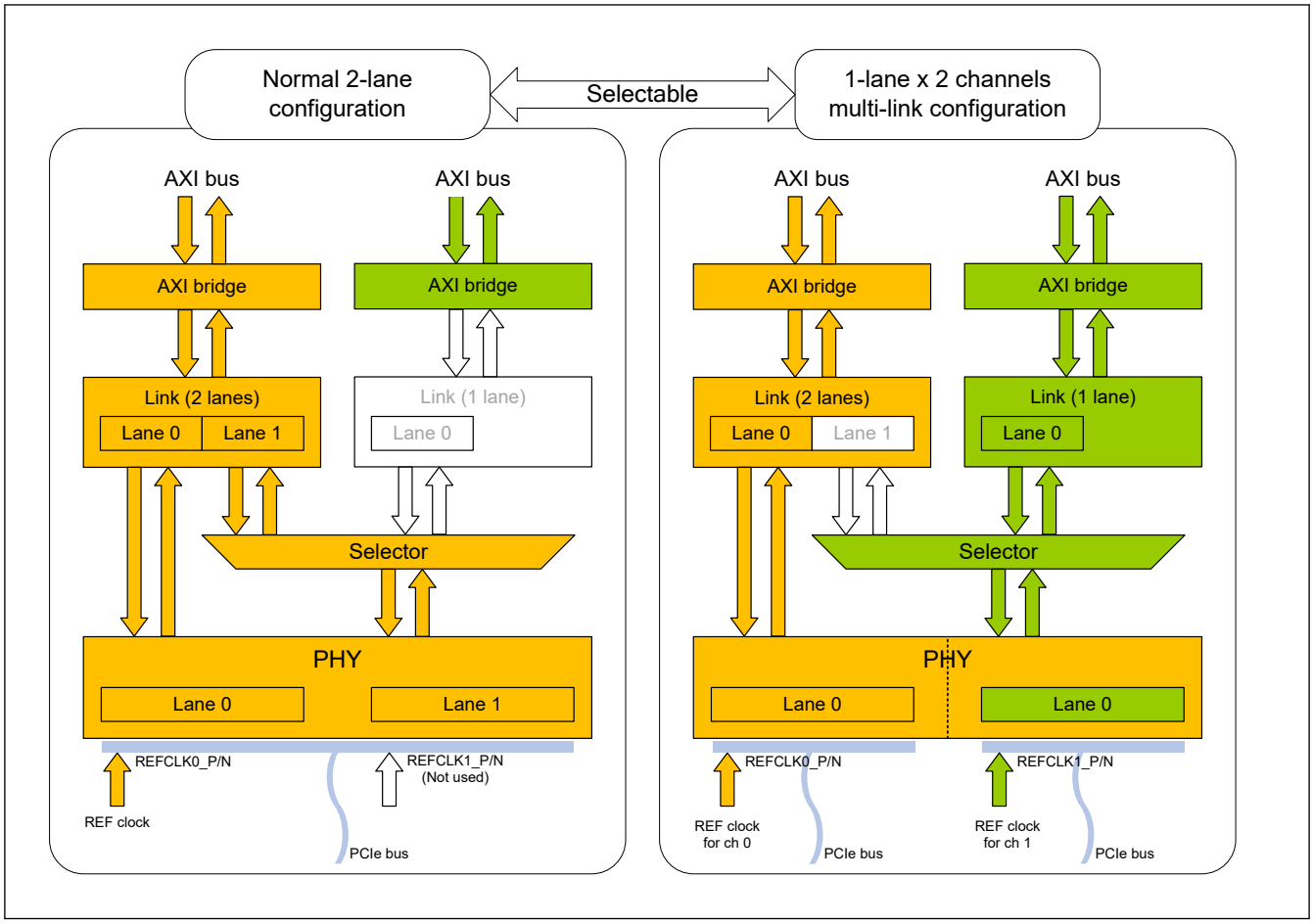


Figure 56.2 Relationship between modules and lanes

In the multi-link (1 lane × 2 channels) configuration, channel 0/1 are independent PCIe cores.

When channels 0/1 are placed in the same AXI area, overlapping the AXI window and PCI window of channel 0/1 is prohibited. In addition, since channel 0/1 are independent PCIe cores, they must be controlled by the system when channel 0/1 are linked for use.

The lane reversal is not supported in the multi-link configuration.

56.1.2 Single Link (2 Lanes × 1 channel) Configuration

The AXI bus for channel 1, which is not used, responds with OKAY and ERROR as usual, and is not deadlocked. ERROR response is generated in case of read access to the AXI Window by incorrect AXI transaction.

Figure 56.3 shows the supported configuration for the lane connection.

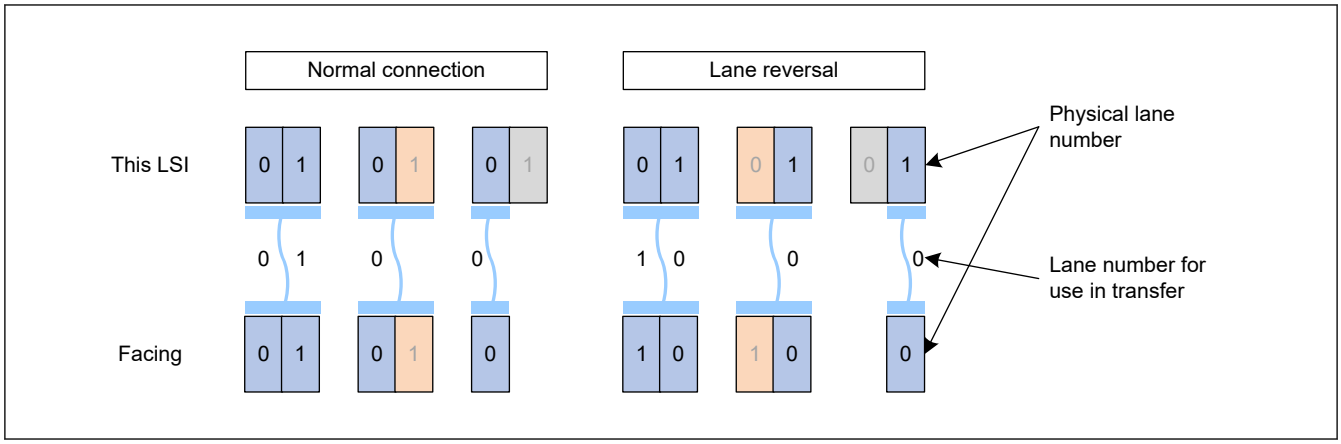


Figure 56.3 Lane connection configuration

56.2 Register Map

56.2.1 Root Complex Mode

Table 56.4 PCIE register map (AXI bridge registers) (1 of 3)

Address	Register symbol	Register name	Write protection
0x9210_0080 + 0x1_0000 × m + 0x4 × n	PCI_RC_REQDATAn	Request Data Register n (n = 0 to 2)	—
0x9210_008C + 0x1_0000 × m	PCI_RC_REQRCVDAT	Request Receive Data Register	—
0x9210_0090 + 0x1_0000 × m	PCI_RC_REQADR1	Request Address Register 1	—
0x9210_0094 + 0x1_0000 × m	PCI_RC_REQADR2	Request Address Register 2	—
0x9210_0098 + 0x1_0000 × m	PCI_RC_REQBE	Request Byte Enable Register	—
0x9210_009C + 0x1_0000 × m	PCI_RC_REQISS	Request Issue Register	—
0x9210_0100 + 0x1_0000 × m	PCI_RC_MSIRCVWADRL	MSI Receive Window Address (Lower) Register	—
0x9210_0104 + 0x1_0000 × m	PCI_RC_MSIRCVWADRU	MSI Receive Window Address (Upper) Register	—
0x9210_0108 + 0x1_0000 × m	PCI_RC_MSIRCVWMSKL	MSI Receive Window Mask (Lower) Register	—
0x9210_010C + 0x1_0000 × m	PCI_RC_MSIRCVWMSKU	MSI Receive Window Mask (Upper) Register	—
0x9210_0110 + 0x1_0000 × m	PCI_RC_PINTRCVIE	PCI INTx Receive Interrupt Enable Register	—
0x9210_0114 + 0x1_0000 × m	PCI_RC_PINTRCVIS	PCI INTx Receive Interrupt Status Register	—
0x9210_0120 + 0x1_0000 × m	PCI_RC_MSGRCVIE	Message Receive Interrupt Enable Register	—
0x9210_0124 + 0x1_0000 × m	PCI_RC_MSGRCVIS	Message Receive Interrupt Status Register	—
0x9210_0130 + 0x1_0000 × m	PCI_RC_MSGCODE	Message Code Register	—
0x9210_0134 + 0x1_0000 × m	PCI_RC_MSGDATA	Message Data Register	—
0x9210_0138 + 0x1_0000 × m	PCI_RC_MSGH3DW	Message Header 3rdDW Register	—
0x9210_013C + 0x1_0000 × m	PCI_RC_MSGH4DW	Message Header 4thDW Register	—
0x9210_0140 + 0x1_0000 × m	PCI_RC_INTTABLE	Interrupt Table Register	—
0x9210_0200 + 0x1_0000 × m	PCI_RC_PEIE0	PCIe Event Interrupt Enable 0 Register	—
0x9210_0204 + 0x1_0000 × m	PCI_RC_PEIS0	PCIe Event Interrupt Status 0 Register	—
0x9210_0208 + 0x1_0000 × m	PCI_RC_PEIE1	PCIe Event Interrupt Enable 1 Register	—
0x9210_020C + 0x1_0000 × m	PCI_RC_PEIS1	PCIe Event Interrupt Status 1 Register	—
0x9210_0210 + 0x1_0000 × m	PCI_RC_AMEIE	AXI Master Error Interrupt Enable Register	—
0x9210_0214 + 0x1_0000 × m	PCI_RC_AMEIS	AXI Master Error Interrupt Status Register	—
0x9210_0220 + 0x1_0000 × m	PCI_RC_ASEIE1	AXI Slave Error Interrupt Enable 1 Register	—

Table 56.4 PCIE register map (AXI bridge registers) (2 of 3)

Address	Register symbol	Register name	Write protection
0x9210_0224 + 0x1_0000 × m	PCI_RC_ASEIS1	AXI Slave Error Interrupt Status 1 Register	—
0x9210_0230 + 0x1_0000 × m	PCI_RC_ASEIS3	AXI Slave Error Interrupt Status 3 Register	—
0x9210_0300 + 0x1_0000 × m	PCI_RC_PERM	Permission Register	—
0x9210_0310 + 0x1_0000 × m	PCI_RC_RESET	Reset Register	—
0x9210_0314 + 0x1_0000 × m	PCI_RC_MSET0	Mode Set 0 Register	—
0x9210_0318 + 0x1_0000 × m	PCI_RC_MSET1	Mode Set 1 Register	—
0x9210_0380 + 0x1_0000 × m	PCI_RC_MSET3	Mode Set 3 Register	—
0x9210_0384 + 0x1_0000 × m	PCI_RC_MSET4	Mode Set 4 Register	—
0x9210_0388 + 0x1_0000 × m	PCI_RC_MSET5	Mode Set 5 Register	—
0x9210_0390 + 0x1_0000 × m	PCI_RC_MSTAT0	Mode Status 0 Register	—
0x9210_0400 + 0x1_0000 × m	PCI_RC_PCMSET1	PCIe Core Mode Set 1 Register	—
0x9210_0404 + 0x1_0000 × m	PCI_RC_PCCTRL1	PCIe Core Control 1 Register	—
0x9210_0408 + 0x1_0000 × m	PCI_RC_PCSTAT1	PCIe Core Status 1 Register	—
0x9210_0410 + 0x1_0000 × m	PCI_RC_PCCTRL2	PCIe Core Control 2 Register	—
0x9210_0414 + 0x1_0000 × m	PCI_RC_PCSTAT2	PCIe Core Status 2 Register	—
0x9210_042C + 0x1_0000 × m	PCI_RC_PCSTAT5	PCIe Core Status 5 Register	—
0x9210_04D0 + 0x1_0000 × m	PCI_RC_DMAINTVEC0	DMA Interrupt Vector 0 Register	—
0x9210_04D4 + 0x1_0000 × m	PCI_RC_DMAINTVEC1	DMA Interrupt Vector 1 Register	—
0x9210_0600 + 0x1_0000 × m + 0x10 × n	PCI_RC_MSIRCVEn	MSI Receive Enable Register n (n = 0 to 15)	—
0x9210_0604 + 0x1_0000 × m + 0x10 × n	PCI_RC_MSIRCVMSGDATAn	MSI Receive Message Data Register n (n = 0 to 15)	—
0x9210_0608 + 0x1_0000 × m + 0x10 × n	PCI_RC_MSIRCVMSKn	MSI Receive Mask Register n (n = 0 to 15)	—
0x9210_060C + 0x1_0000 × m + 0x10 × n	PCI_RC_MSIRCVSTATn	MSI Receive Status Register n (n = 0 to 15)	—
0x9210_0800 + 0x1_0000 × m	PCI_RC_DMACTRL	DMAC Control Register	—
0x9210_0808 + 0x1_0000 × m	PCI_RC_DMAINTE	DMAC Interrupt Enable Register	—
0x9210_080C + 0x1_0000 × m	PCI_RC_DMAINTS	DMAC Interrupt Status Register	—
0x9210_0900 + 0x1_0000 × m + 0x80 × n	PCI_RC_DMACHCTLn	DMAC Channel Control Register n (n = 0 to 7)	—
0x9210_0908 + 0x1_0000 × m + 0x80 × n	PCI_RC_DPSADRLn	Descriptor Start Address (Lower) Register n (n = 0 to 7)	—
0x9210_090C + 0x1_0000 × m + 0x80 × n	PCI_RC_DPSADRUn	Descriptor Start Address (Upper) Register n (n = 0 to 7)	—
0x9210_0910 + 0x1_0000 × m + 0x80 × n	PCI_RC_QUEEn	QUE Entry Register n (n = 0 to 7)	—
0x9210_0920 + 0x1_0000 × m + 0x80 × n	PCI_RC_DMADPCTLn	DMA Descriptor Control (Descriptor 00h) n (n = 0 to 7)	—
0x9210_0924 + 0x1_0000 × m + 0x80 × n	PCI_RC_DMATCTLn	DMA Transaction Control (Descriptor 04h) n (n = 0 to 7)	—
0x9210_0928 + 0x1_0000 × m + 0x80 × n	PCI_RC_DMASIZEn	DMA Size (Descriptor 08h) n (n = 0 to 7)	—
0x9210_0930 + 0x1_0000 × m + 0x80 × n	PCI_RC_DMASLAn	DMA Source Lower Address (Descriptor 10h) n (n = 0 to 7)	—
0x9210_0934 + 0x1_0000 × m + 0x80 × n	PCI_RC_DMASUAn	DMA Source Upper Address (Descriptor 14h) n (n = 0 to 7)	—
0x9210_0938 + 0x1_0000 × m + 0x80 × n	PCI_RC_DMADESTLAn	DMA Destination Lower Address (Descriptor 18h) n (n = 0 to 7)	—
0x9210_093C + 0x1_0000 × m + 0x80 × n	PCI_RC_DMADESTUAn	DMA Destination Upper Address (Descriptor 1Ch) n (n = 0 to 7)	—

Table 56.4 PCIE register map (AXI bridge registers) (3 of 3)

Address	Register symbol	Register name	Write protection
0x9210_0940 + 0x1_0000 × m + 0x80 × n	PCI_RC_DMADPLLPn	DMA Descriptor Lower Link Pointer (Descriptor 20h) n (n = 0 to 7)	—
0x9210_0944 + 0x1_0000 × m + 0x80 × n	PCI_RC_DMADPULPn	DMA Descriptor Upper Link Pointer (Descriptor 24h) n (n = 0 to 7)	—
0x9210_0950 + 0x1_0000 × m + 0x80 × n	PCI_RC_DMARESTSIZEn	DMA Rest Size Register n (n = 0 to 7)	—
0x9210_0960 + 0x1_0000 × m + 0x80 × n	PCI_RC_AREQALn	AXI Request Address (Lower) Register n (n = 0 to 7)	—
0x9210_0964 + 0x1_0000 × m + 0x80 × n	PCI_RC_AREQAUn	AXI Request Address (Upper) Register n (n = 0 to 7)	—
0x9210_0968 + 0x1_0000 × m + 0x80 × n	PCI_RC_PREQALn	PCIe Request Address (Lower) Register n (n = 0 to 7)	—
0x9210_096C + 0x1_0000 × m + 0x80 × n	PCI_RC_PREQAUn	PCIe Request Address (Upper) Register n (n = 0 to 7)	—
0x9210_0970 + 0x1_0000 × m + 0x80 × n	PCI_RC_QUESTAn	QUE Status Register n (n = 0 to 7)	—
0x9210_0978 + 0x1_0000 × m + 0x80 × n	PCI_RC_DMACESTAn	DMAC Error Status Register n (n = 0 to 7)	—
0x9210_1000 + 0x1_0000 × m + 0x20 × n	PCI_RC_AWBASELn	AXI Window Base (Lower) Register n (n = 0 to 7)	—
0x9210_1004 + 0x1_0000 × m + 0x20 × n	PCI_RC_AWBASEUn	AXI Window Base (Upper) Register n (n = 0 to 7)	—
0x9210_1008 + 0x1_0000 × m + 0x20 × n	PCI_RC_AWMASKLn	AXI Window Mask (Lower) Register n (n = 0 to 7)	—
0x9210_100C + 0x1_0000 × m + 0x20 × n	PCI_RC_AWMASKUn	AXI Window Mask (Upper) Register n (n = 0 to 7)	—
0x9210_1010 + 0x1_0000 × m + 0x20 × n	PCI_RC_ADESTLn	AXI Destination (Lower) Register n (n = 0 to 7)	—
0x9210_1014 + 0x1_0000 × m + 0x20 × n	PCI_RC_ADESTUn	AXI Destination (Upper) Register n (n = 0 to 7)	—
0x9210_1100 + 0x1_0000 × m + 0x20 × n	PCI_RC_PWBASELn	PCIe Window Base (Lower) Register n (n = 0 to 7)	—
0x9210_1104 + 0x1_0000 × m + 0x20 × n	PCI_RC_PWBASEUn	PCIe Window Base (Upper) Register n (n = 0 to 7)	—
0x9210_1108 + 0x1_0000 × m + 0x20 × n	PCI_RC_PWMASKLn	PCIe Window Mask (Lower) Register n (n = 0 to 7)	—
0x9210_110C + 0x1_0000 × m + 0x20 × n	PCI_RC_PWMASKUn	PCIe Window Mask (Upper) Register n (n = 0 to 7)	—
0x9210_1110 + 0x1_0000 × m + 0x20 × n	PCI_RC_PDESTLn	PCIe Destination (Lower) Register n (n = 0 to 7)	—
0x9210_1114 + 0x1_0000 × m + 0x20 × n	PCI_RC_PDESTUn	PCIe Destination (Upper) Register n (n = 0 to 7)	—

Note: m = 0 (ch 0), 1 (ch 1)

Table 56.5 PCIE register map (PCI express configuration registers (Type 1)) (1 of 3)

Address	Register symbol	Register name	Write protection
0x9210_6000 + 0x1_0000 × m	PCI_RC_VID	Vendor and Device ID Register	—
0x9210_6004 + 0x1_0000 × m	PCI_RC_COM_STA	Command and Status Register	—
0x9210_6008 + 0x1_0000 × m	PCI_RC_RID_CC	Revision ID and Class Code Register	—
0x9210_600C + 0x1_0000 × m	PCI_RC_CL_HT	Cache Line and Header Type Register	—
0x9210_6010 + 0x1_0000 × m	PCI_RC_BAR0	Base Address Register 0	—
0x9210_6014 + 0x1_0000 × m	PCI_RC_BAR1	Base Address Register 1	—
0x9210_6018 + 0x1_0000 × m	PCI_RC_BNR	Bus Number Register	—
0x9210_601C + 0x1_0000 × m	PCI_RC_IOBL_SS	I/O Base/Limit and Secondary Status Register	—
0x9210_6020 + 0x1_0000 × m	PCI_RC_MEMBL	Memory Base/Limit Register	—
0x9210_6024 + 0x1_0000 × m	PCI_RC_PMBL	Prefetchable Memory Base/Limit Register	—

Table 56.5 PCIE register map (PCI express configuration registers (Type 1)) (2 of 3)

Address	Register symbol	Register name	Write protection
0x9210_6028 + 0x1_0000 × m	PCI_RC_PBUP32	Prefetchable Base Upper 32bits Register	—
0x9210_602C + 0x1_0000 × m	PCI_RC_PLUP32	Prefetchable Limit Upper 32bits Register	—
0x9210_6030 + 0x1_0000 × m	PCI_RC_IOBLUP16	I/O Base/Limit Upper 16bits Register	—
0x9210_6034 + 0x1_0000 × m	PCI_RC_CP	Capability Pointer Register	—
0x9210_603C + 0x1_0000 × m	PCI_RC_BC_INT	Bridge Control and Interrupt Register	—
0x9210_6040 + 0x1_0000 × m	PCI_RC_PMC	PM Capabilities Register	—
0x9210_6044 + 0x1_0000 × m	PCI_RC_PMSC	PM Status/Control Register	—
0x9210_6060 + 0x1_0000 × m	PCI_RC_PCIEC	PCI Express Capability Register	—
0x9210_6064 + 0x1_0000 × m	PCI_RC_DEVC	Device Capabilities Register	—
0x9210_6068 + 0x1_0000 × m	PCI_RC_DEVCS	Device Control/Status Register	—
0x9210_606C + 0x1_0000 × m	PCI_RC_LINKC	Link Capabilities Register	—
0x9210_6070 + 0x1_0000 × m	PCI_RC_LINKCS	Link Control/Status Register	—
0x9210_6074 + 0x1_0000 × m	PCI_RC_SLOTC	Slot Capabilities Register	—
0x9210_6078 + 0x1_0000 × m	PCI_RC_SLOTCS	Slot Control/Status Register	—
0x9210_607C + 0x1_0000 × m	PCI_RC_ROOTCC	Root Control/Capabilities Register	—
0x9210_6080 + 0x1_0000 × m	PCI_RC_ROOTS	Root Status Register	—
0x9210_6084 + 0x1_0000 × m	PCI_RC_DEVC2	Device Capabilities 2 Register	—
0x9210_6088 + 0x1_0000 × m	PCI_RC_DEVCS2	Device Control 2/Status 2 Register	—
0x9210_608C + 0x1_0000 × m	PCI_RC_LINKC2	Link Capabilities 2 Register	—
0x9210_6090 + 0x1_0000 × m	PCI_RC_LINCS2	Link Control 2/Status 2 Register	—
0x9210_60A0 + 0x1_0000 × m	PCI_RC_BARMSK00L	Base Address Register Mask 00 (Lower) Register	—
0x9210_60A4 + 0x1_0000 × m	PCI_RC_BARMSK00U	Base Address Register Mask 00 (Upper) Register	—
0x9210_60C8 + 0x1_0000 × m	PCI_RC_BSIZE00_01	Base Size 00/01 Register	—
0x9210_60D8 + 0x1_0000 × m	PCI_RC_TSUPPORT00_01_02	Type Supported 00/01/02 Register	—
0x9210_6100 + 0x1_0000 × m	PCI_RC_ADVERC	Advanced Error Reporting Capability Register	—
0x9210_6104 + 0x1_0000 × m	PCI_RC_UNCESTS	Uncorrectable Error Status Register	—
0x9210_6108 + 0x1_0000 × m	PCI_RC_UNCEMASK	Uncorrectable Error Mask Register	—
0x9210_610C + 0x1_0000 × m	PCI_RC_UNCESVY	Uncorrectable Error Severity Register	—
0x9210_6110 + 0x1_0000 × m	PCI_RC_CESTS	Correctable Error Status Register	—
0x9210_6114 + 0x1_0000 × m	PCI_RC_CEMASK	Correctable Error Mask Register	—
0x9210_6118 + 0x1_0000 × m	PCI_RC_ADVECC	Advanced Error Capabilities and Control Register	—
0x9210_611C + 0x1_0000 × m	PCI_RC_HLOG0	Header Log Register 0	—
0x9210_6120 + 0x1_0000 × m	PCI_RC_HLOG1	Header Log Register 1	—
0x9210_6124 + 0x1_0000 × m	PCI_RC_HLOG2	Header Log Register 2	—
0x9210_6128 + 0x1_0000 × m	PCI_RC_HLOG3	Header Log Register 3	—
0x9210_612C + 0x1_0000 × m	PCI_RC_ROOTEC	Root Error Command Register	—
0x9210_6130 + 0x1_0000 × m	PCI_RC_ROOTES	Root Error Status Register	—
0x9210_6134 + 0x1_0000 × m	PCI_RC_ERRSI	Error Source Identification Register	—
0x9210_6150 + 0x1_0000 × m	PCI_RC_DEVSNEXTC	Device Serial Number Extended Capability Register	—
0x9210_6154 + 0x1_0000 × m	PCI_RC_SNL	Serial Number Register (Lower DW) Register	—

Table 56.5 PCIE register map (PCI express configuration registers (Type 1)) (3 of 3)

Address	Register symbol	Register name	Write protection
0x9210_6158 + 0x1_0000 × m	PCI_RC_SNU	Serial Number Register (Upper DW) Register	—
0x9210_61B0 + 0x1_0000 × m	PCI_RC_SPEECH	Secondary PCI Express Extended Capability Header Register	—
0x9210_61B4 + 0x1_0000 × m	PCI_RC_LINC3	Link Control 3 Register	—
0x9210_61B8 + 0x1_0000 × m	PCI_RC_LESTA	Lane Error Status Register	—
0x9210_61BC + 0x1_0000 × m	PCI_RC_LEQCTL	Lane Equalization Control Register	—

Note: m = 0 (ch 0), 1 (ch 1)

56.2.2 Endpoint Mode

Table 56.6 PCIE register map (AXI bridge registers) (1 of 3)

Address	Register symbol	Register name	Write protection
0x9210_0080 + 0x1_0000 × m + 0x4 × n	PCI_EP_REQDATAn	Request Data Register n (n = 0 to 2)	—
0x9210_008C + 0x1_0000 × m	PCI_EP_REQRCVDAT	Request Receive Data Register	—
0x9210_0090 + 0x1_0000 × m	PCI_EP_REQADR1	Request Address Register 1	—
0x9210_0094 + 0x1_0000 × m	PCI_EP_REQADR2	Request Address Register 2	—
0x9210_0098 + 0x1_0000 × m	PCI_EP_REQBE	Request Byte Enable Register	—
0x9210_009C + 0x1_0000 × m	PCI_EP_REQISS	Request Issue Register	—
0x9210_0118 + 0x1_0000 × m	PCI_EP_INTXOUTS	PCI INTx Out Status Register	—
0x9210_0120 + 0x1_0000 × m	PCI_EP_MSGRCVIE	Message Receive Interrupt Enable Register	—
0x9210_0124 + 0x1_0000 × m	PCI_EP_MSGRCVIS	Message Receive Interrupt Status Register	—
0x9210_0130 + 0x1_0000 × m	PCI_EP_MSGCODE	Message Code Register	—
0x9210_0134 + 0x1_0000 × m	PCI_EP_MSGDATA	Message Data Register	—
0x9210_0138 + 0x1_0000 × m	PCI_EP_MSGH3DW	Message Header 3rdDW Register	—
0x9210_013C + 0x1_0000 × m	PCI_EP_MSGH4DW	Message Header 4thDW Register	—
0x9210_0140 + 0x1_0000 × m	PCI_EP_INTTABLE	Interrupt Table Register	—
0x9210_0200 + 0x1_0000 × m	PCI_EP_PEIE0	PCIe Event Interrupt Enable 0 Register	—
0x9210_0204 + 0x1_0000 × m	PCI_EP_PEIS0	PCIe Event Interrupt Status 0 Register	—
0x9210_0208 + 0x1_0000 × m	PCI_EP_PEIE1	PCIe Event Interrupt Enable 1 Register	—
0x9210_020C + 0x1_0000 × m	PCI_EP_PEIS1	PCIe Event Interrupt Status 1 Register	—
0x9210_0210 + 0x1_0000 × m	PCI_EP_AMEIE	AXI Master Error Interrupt Enable Register	—
0x9210_0214 + 0x1_0000 × m	PCI_EP_AMEIS	AXI Master Error Interrupt Status Register	—
0x9210_0220 + 0x1_0000 × m	PCI_EP_ASEIE1	AXI Slave Error Interrupt Enable 1 Register	—
0x9210_0224 + 0x1_0000 × m	PCI_EP_ASEIS1	AXI Slave Error Interrupt Status 1 Register	—
0x9210_0230 + 0x1_0000 × m	PCI_EP_ASEIS3	AXI Slave Error Interrupt Status 3 Register	—
0x9210_0240 + 0x1_0000 × m	PCI_EP_PEIE2	PCIe Event Interrupt Enable 2 Register	—
0x9210_0244 + 0x1_0000 × m	PCI_EP_PEIS2	PCIe Event Interrupt Status 2 Register	—
0x9210_0300 + 0x1_0000 × m	PCI_EP_PERM	Permission Register	—
0x9210_0310 + 0x1_0000 × m	PCI_EP_RESET	Reset Register	—
0x9210_0314 + 0x1_0000 × m	PCI_EP_MSET0	Mode Set 0 Register	—
0x9210_0318 + 0x1_0000 × m	PCI_EP_MSET1	Mode Set 1 Register	—
0x9210_0380 + 0x1_0000 × m	PCI_EP_MSET3	Mode Set 3 Register	—

Table 56.6 PCIE register map (AXI bridge registers) (2 of 3)

Address	Register symbol	Register name	Write protection
0x9210_0384 + 0x1_0000 × m	PCI_EP_MSET4	Mode Set 4 Register	—
0x9210_0388 + 0x1_0000 × m	PCI_EP_MSET5	Mode Set 5 Register	—
0x9210_0390 + 0x1_0000 × m	PCI_EP_MSTAT0	Mode Status 0 Register	—
0x9210_0400 + 0x1_0000 × m	PCI_EP_PCMSET1	PCIe Core Mode Set 1 Register	—
0x9210_0404 + 0x1_0000 × m	PCI_EP_PCCTRL1	PCIe Core Control 1 Register	—
0x9210_0408 + 0x1_0000 × m	PCI_EP_PCSTAT1	PCIe Core Status 1 Register	—
0x9210_0410 + 0x1_0000 × m	PCI_EP_PCCTRL2	PCIe Core Control 2 Register	—
0x9210_0414 + 0x1_0000 × m	PCI_EP_PCSTAT2	PCIe Core Status 2 Register	—
0x9210_042C + 0x1_0000 × m	PCI_EP_PCSTAT5	PCIe Core Status 5 Register	—
0x9210_04D0 + 0x1_0000 × m	PCI_EP_DMAINTVEC0	DMA Interrupt Vector 0 Register	—
0x9210_04D4 + 0x1_0000 × m	PCI_EP_DMAINTVEC1	DMA Interrupt Vector 1 Register	—
0x9210_0800 + 0x1_0000 × m	PCI_EP_DMACTRL	DMA Control Register	—
0x9210_0808 + 0x1_0000 × m	PCI_EP_DMAINTE	DMA Interrupt Enable Register	—
0x9210_080C + 0x1_0000 × m	PCI_EP_DMAINTS	DMA Interrupt Status Register	—
0x9210_0900 + 0x1_0000 × m + 0x80 × n	PCI_EP_DMACHCTLn	DMAC Channel Control Register n (n = 0 to 7)	—
0x9210_0908 + 0x1_0000 × m + 0x80 × n	PCI_EP_DPSADRLn	Descriptor Start Address (Lower) Register n (n = 0 to 7)	—
0x9210_090C + 0x1_0000 × m + 0x80 × n	PCI_EP_DPSADRUn	Descriptor Start Address (Upper) Register n (n = 0 to 7)	—
0x9210_0910 + 0x1_0000 × m + 0x80 × n	PCI_EP_QUEEn	QUE Entry Register n (n = 0 to 7)	—
0x9210_0920 + 0x1_0000 × m + 0x80 × n	PCI_EP_DMADPCTLn	DMA Descriptor Control (Descriptor 00h) n (n = 0 to 7)	—
0x9210_0924 + 0x1_0000 × m + 0x80 × n	PCI_EP_DMATCTLn	DMA Transaction Control (Descriptor 04h) n (n = 0 to 7)	—
0x9210_0928 + 0x1_0000 × m + 0x80 × n	PCI_EP_DMASIZEn	DMA Size (Descriptor 08h) n (n = 0 to 7)	—
0x9210_0930 + 0x1_0000 × m + 0x80 × n	PCI_EP_DMASLAn	DMA Source Lower Address (Descriptor 10h) n (n = 0 to 7)	—
0x9210_0934 + 0x1_0000 × m + 0x80 × n	PCI_EP_DMASUAn	DMA Source Upper Address (Descriptor 14h) n (n = 0 to 7)	—
0x9210_0938 + 0x1_0000 × m + 0x80 × n	PCI_EP_DMADESTLAn	DMA Destination Lower Address (Descriptor 18h) n (n = 0 to 7)	—
0x9210_093C + 0x1_0000 × m + 0x80 × n	PCI_EP_DMADESTUAn	DMA Destination Upper Address (Descriptor 1Ch) n (n = 0 to 7)	—
0x9210_0940 + 0x1_0000 × m + 0x80 × n	PCI_EP_DMADPLLPn	DMA Descriptor Lower Link Pointer (Descriptor 20h) n (n = 0 to 7)	—
0x9210_0944 + 0x1_0000 × m + 0x80 × n	PCI_EP_DMADPULPn	DMA Descriptor Upper Link Pointer (Descriptor 24h) n (n = 0 to 7)	—
0x9210_0950 + 0x1_0000 × m + 0x80 × n	PCI_EP_DMARESTSIZEn	DMA Rest Size Register n (n = 0 to 7)	—
0x9210_0960 + 0x1_0000 × m + 0x80 × n	PCI_EP_AREQALn	AXI Request Address (Lower) Register n (n = 0 to 7)	—
0x9210_0964 + 0x1_0000 × m + 0x80 × n	PCI_EP_AREQAUn	AXI Request Address (Upper) Register n (n = 0 to 7)	—
0x9210_0968 + 0x1_0000 × m + 0x80 × n	PCI_EP_PREQALn	PCIe Request Address (Lower) Register n (n = 0 to 7)	—
0x9210_096C + 0x1_0000 × m + 0x80 × n	PCI_EP_PREQAUn	PCIe Request Address (Upper) Register n (n = 0 to 7)	—

Table 56.6 PCIE register map (AXI bridge registers) (3 of 3)

Address	Register symbol	Register name	Write protection
0x9210_0970 + 0x1_0000 × m + 0x80 × n	PCI_EP_QUESTAn	QUE Status Register n (n = 0 to 7)	—
0x9210_0978 + 0x1_0000 × m + 0x80 × n	PCI_EP_DMACESTAn	DMAC Error Status Register n (n = 0 to 7)	—
0x9210_1000 + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_AWBASELn_Fi	AXI Window Base (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—
0x9210_1004 + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_AWBASEUn_Fi	AXI Window Base (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—
0x9210_1008 + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_AWMASKLn_Fi	AXI Window Mask (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—
0x9210_100C + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_AWMASKUn_Fi	AXI Window Mask (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—
0x9210_1010 + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_ADESTLn_Fi	AXI Destination (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—
0x9210_1014 + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_ADESTUn_Fi	AXI Destination (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—
0x9210_1100 + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_PWBASELn_Fi	PCIe Window Base (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—
0x9210_1104 + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_PWBASEUn_Fi	PCIe Window Base (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—
0x9210_1108 + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_PWMASKLn_Fi	PCIe Window Mask (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—
0x9210_110C + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_PWMASKUn_Fi	PCIe Window Mask (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—
0x9210_1110 + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_PDESTLn_Fi	PCIe Destination (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—
0x9210_1114 + 0x1_0000 × m + 0x20 × n + 0x200 × i	PCI_EP_PDESTUn_Fi	PCIe Destination (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)	—

Note: m = 0 (ch 0), 1 (ch 1)

Table 56.7 PCIE register map (PCI express configuration registers (type 0)) (1 of 3)

Address	Register symbol	Register name	Write protection
0x9210_6000 + 0x1_0000 × m + 0x1000 × i	PCI_EP_VID_Fi	Vendor and Device ID Register (Function #i) (i = 0, 1)	—
0x9210_6004 + 0x1_0000 × m + 0x1000 × i	PCI_EP_COM_STA_Fi	Command and Status Register (Function #i) (i = 0, 1)	—
0x9210_6008 + 0x1_0000 × m + 0x1000 × i	PCI_EP_RID_CC_Fi	Revision ID and Class Code Register (Function #i) (i = 0, 1)	—
0x9210_600C + 0x1_0000 × m + 0x1000 × i	PCI_EP_CL_HT	Cache Line and Header Type Register (Function #i) (i = 0, 1)	—
0x9210_6010 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BAR0_Fi	Base Address Register 0 (Function #i) (i = 0, 1)	—
0x9210_6014 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BAR1_Fi	Base Address Register 1 (Function #i) (i = 0, 1)	—
0x9210_6018 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BAR2_Fi	Base Address Register 2 (Function #i) (i = 0, 1)	—
0x9210_601C + 0x1_0000 × m + 0x1000 × i	PCI_EP_BAR3_Fi	Base Address Register 3 (Function #i) (i = 0, 1)	—
0x9210_6020 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BAR4_Fi	Base Address Register 4 (Function #i) (i = 0, 1)	—
0x9210_6024 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BAR5_Fi	Base Address Register 5 (Function #i) (i = 0, 1)	—
0x9210_602C + 0x1_0000 × m + 0x1000 × i	PCI_EP_SUBSID_Fi	Subsystem ID Register (Function #i) (i = 0, 1)	—
0x9210_6034 + 0x1_0000 × m + 0x1000 × i	PCI_EP_CP_Fi	Capabilities Pointer Register (Function #i) (i = 0, 1)	—
0x9210_603C + 0x1_0000 × m + 0x1000 × i	PCI_EP_INT_Fi	Interrupt Register (Function #i) (i = 0, 1)	—

Table 56.7 PCIE register map (PCI express configuration registers (type 0)) (2 of 3)

Address	Register symbol	Register name	Write protection
0x9210_6040 + 0x1_0000 × m + 0x1000 × i	PCI_EP_PMC_Fi	PM Capabilities Register (Function #i) (i = 0, 1)	—
0x9210_6044 + 0x1_0000 × m + 0x1000 × i	PCI_EP_PMSC_Fi	PM Status/Control Register (Function #i) (i = 0, 1)	—
0x9210_6060 + 0x1_0000 × m + 0x1000 × i	PCI_EP_PCIEC_Fi	PCI Express Capability Register (Function #i) (i = 0, 1)	—
0x9210_6064 + 0x1_0000 × m + 0x1000 × i	PCI_EP_DEVC_Fi	Device Capabilities Register (Function #i) (i = 0, 1)	—
0x9210_6068 + 0x1_0000 × m + 0x1000 × i	PCI_EP_DEVCS_Fi	Device Control/Status Register (Function #i) (i = 0, 1)	—
0x9210_606C + 0x1_0000 × m + 0x1000 × i	PCI_EP_LINKC_Fi	Link Capabilities Register (Function #i) (i = 0, 1)	—
0x9210_6070 + 0x1_0000 × m + 0x1000 × i	PCI_EP_LINKCS_Fi	Link Control/Status Register (Function #i) (i = 0, 1)	—
0x9210_6084 + 0x1_0000 × m + 0x1000 × i	PCI_EP_DEVC2_Fi	Device Capabilities 2 Register (Function #i) (i = 0, 1)	—
0x9210_6088 + 0x1_0000 × m + 0x1000 × i	PCI_EP_DEVCS2_Fi	Device Control 2/Status 2 Register (Function #i) (i = 0, 1)	—
0x9210_608C + 0x1_0000 × m + 0x1000 × i	PCI_EP_LINKC2_Fi	Link Capabilities 2 Register (Function #i) (i = 0, 1)	—
0x9210_6090 + 0x1_0000 × m + 0x1000 × i	PCI_EP_LINCS2_Fi	Link Control 2/Status 2 Register (Function #i) (i = 0, 1)	—
0x9210_60A0 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BARMSK00L_Fi	Base Address Register Mask00 (Lower) Register (Function #i) (i = 0, 1)	—
0x9210_60A4 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BARMSK00U_Fi	Base Address Register Mask00 (Upper) Register (Function #i) (i = 0, 1)	—
0x9210_60A8 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BARMSK01L_Fi	Base Address Register Mask01 (Lower) Register (Function #i) (i = 0, 1)	—
0x9210_60AC + 0x1_0000 × m + 0x1000 × i	PCI_EP_BARMSK01U_Fi	Base Address Register Mask01 (Upper) Register (Function #i) (i = 0, 1)	—
0x9210_60B0 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BARMSK02L_Fi	Base Address Register Mask02 (Lower) Register (Function #i) (i = 0, 1)	—
0x9210_60B4 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BARMSK02U_Fi	Base Address Register Mask02 (Upper) Register (Function #i) (i = 0, 1)	—
0x9210_60C8 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BSIZE00_01_Fi	Base Size 00/01 Register (Function #i) (i = 0, 1)	—
0x9210_60CC + 0x1_0000 × m + 0x1000 × i	PCI_EP_BSIZE02_03_Fi	Base Size 02/03 Register (Function #i) (i = 0, 1)	—
0x9210_60D0 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BSIZE04_05_Fi	Base Size 04/05 Register (Function #i) (i = 0, 1)	—
0x9210_60D4 + 0x1_0000 × m + 0x1000 × i	PCI_EP_BSIZE06_Fi	Base Size 06 Register (Function #i) (i = 0, 1)	—
0x9210_60D8 + 0x1_0000 × m + 0x1000 × i	PCI_EP_TSUPPORT00_01_02_Fi	Type Supported 00/01/02 Register (Function #i) (i = 0, 1)	—
0x9210_60E0 + 0x1_0000 × m + 0x1000 × i	PCI_EP_MSICAP_Fi	MSI Capability Register (Function #i) (i = 0, 1)	—
0x9210_60E4 + 0x1_0000 × m + 0x1000 × i	PCI_EP_MSGADR_Fi	Message Address Register (Function #i) (i = 0, 1)	—
0x9210_60E8 + 0x1_0000 × m + 0x1000 × i	PCI_EP_MSGUADR_Fi	Message Upper Address Register (Function #i) (i = 0, 1)	—
0x9210_60EC + 0x1_0000 × m + 0x1000 × i	PCI_EP_MSGDAT_Fi	Message Data Register (Function #i) (i = 0, 1)	—
0x9210_60F0 + 0x1_0000 × m + 0x1000 × i	PCI_EP_MSKBIT_Fi	Mask Bits Register (Function #i) (i = 0, 1)	—
0x9210_60F4 + 0x1_0000 × m + 0x1000 × i	PCI_EP_PENDBIT_Fi	Pending Bits Register (Function #i) (i = 0, 1)	—
0x9210_6100 + 0x1_0000 × m + 0x1000 × i	PCI_EP_ADVERC_Fi	Advanced Error Reporting Capability Register (Function #i) (i = 0, 1)	—
0x9210_6104 + 0x1_0000 × m + 0x1000 × i	PCI_EP_UNCESTS_Fi	Uncorrectable Error Status Register (Function #i) (i = 0, 1)	—

Table 56.7 PCIE register map (PCI express configuration registers (type 0)) (3 of 3)

Address	Register symbol	Register name	Write protection
0x9210_6108 + 0x1_0000 × m + 0x1000 × i	PCI_EP_UNCEMASK_Fi	Uncorrectable Error Mask Register (Function #i) (i = 0, 1)	—
0x9210_610C + 0x1_0000 × m + 0x1000 × i	PCI_EP_UNCESVY_Fi	Uncorrectable Error Severity Register (Function #i) (i = 0, 1)	—
0x9210_6110 + 0x1_0000 × m + 0x1000 × i	PCI_EP_CESTS_Fi	Correctable Error Status Register (Function #i) (i = 0, 1)	—
0x9210_6114 + 0x1_0000 × m + 0x1000 × i	PCI_EP_CEMASK_Fi	Correctable Error Mask Register (Function #i) (i = 0, 1)	—
0x9210_6118 + 0x1_0000 × m + 0x1000 × i	PCI_EP_ADVECC_Fi	Advanced Error Capabilities and Control Register (Function #i) (i = 0, 1)	—
0x9210_611C + 0x1_0000 × m + 0x1000 × i	PCI_EP_HLOG0_Fi	Header Log Register 0 (Function #i) (i = 0, 1)	—
0x9210_6120 + 0x1_0000 × m + 0x1000 × i	PCI_EP_HLOG1_Fi	Header Log Register 1 (Function #i) (i = 0, 1)	—
0x9210_6124 + 0x1_0000 × m + 0x1000 × i	PCI_EP_HLOG2_Fi	Header Log Register 2 (Function #i) (i = 0, 1)	—
0x9210_6128 + 0x1_0000 × m + 0x1000 × i	PCI_EP_HLOG3_Fi	Header Log Register 3 (Function #i) (i = 0, 1)	—
0x9210_6150 + 0x1_0000 × m + 0x1000 × i	PCI_EP_DEVSNEXTC_Fi	Device Serial Number Extended Capability Register (Function #i) (i = 0, 1)	—
0x9210_6154 + 0x1_0000 × m + 0x1000 × i	PCI_EP_SNL_Fi	Serial Number Register (Lower DW) (Function #i) (i = 0, 1)	—
0x9210_6158 + 0x1_0000 × m + 0x1000 × i	PCI_EP_SNU_Fi	Serial Number Register (Upper DW) (Function #i) (i = 0, 1)	—
0x9210_61B0 + 0x1_0000 × m	PCI_EP_SPEECH_F0	Secondary PCI Express Extended Capability Header Register (Function #0)	—
0x9210_61B4 + 0x1_0000 × m	PCI_EP_LINC3_F0	Link Control 3 Register (Function #0)	—
0x9210_61B8 + 0x1_0000 × m	PCI_EP_LESTA_F0	Lane Error Status Register (Function #0)	—
0x9210_61BC + 0x1_0000 × m	PCI_EP_LEQCTI_F0	Lane Equalization Control Register (Function #0)	—

Note: m = 0 (ch 0), 1 (ch 1)

56.2.3 Physical Layer Control/Status

Table 56.8 PCIE register map (physical layer control/status)

Address	Register symbol	Register name	Write protection
0x9210_2000 + 0x10 × n (n = 0 to 26)	PCI_PHY_XCFGDn	XCFGD Setting Register n (n = 0 to 26)	—
0x9210_2400 + 0x10 × n (n = 0 to 15)	PCI_PHY_XCFGA_CMNn	XCFGA_CMN Setting Register n (n = 0 to 15)	—
0x9210_2500 + 0x10 × n (n = 0 to 5)	PCI_PHY_XCFGA_L0n	XCFGA_L0 Setting Register n (n = 0 to 5)	—
0x9210_2560 + 0x10 × n (n = 0 to 5)	PCI_PHY_XCFGA_L1n	XCFGA_L1 Setting Register n (n = 0 to 5)	—

56.2.4 Supplemental Setting

Table 56.9 PCIE register map (supplemental setting) (1 of 2)

Address	Register symbol	Register name	Write protection
0x8029_2000 + 0x30 × m	PCIE_INTX	PCIe INTX Register	—
0x8029_2004 + 0x30 × m	PCIE_MSI1	PCIe MSI1 Register	—
0x8029_2008 + 0x30 × m	PCIE_MSI2	PCIe MSI2 Register	—
0x8029_200C + 0x30 × m	PCIE_MSI3	PCIe MSI3 Register	—
0x8029_2010 + 0x30 × m	PCIE_MSI4	PCIe MSI4 Register	—

Table 56.9 PCIE register map (supplemental setting) (2 of 2)

Address	Register symbol	Register name	Write protection
0x8029_2014 + 0x30 × m	PCIE_MSI5	PCIe MSI5 Register	—
0x8029_2018 + 0x30 × m	PCIE_PME	PCIe PME Register	—
0x8029_201C + 0x30 × m	PCIE_ACK	PCIe ASK Register	—
0x8029_2020 + 0x30 × m	PCIE_MISC	PCIe MISC Register	—
0x8029_2024 + 0x30 × m	PCIE_MODE	PCIe MODE Register	—
0x8029_2060	PCIE_LINKMODE	PCIe Link Mode Register	—

Note: m = 0 (ch 0), 1 (ch 1)

Table 56.10 PCIE related system control register

Registers	Module reset control register	Module stop control register	Slave access control register
Root Complex Mode Endpoint Mode Physical Layer Control/Status	MRCTLM.MRCTLM08	MSTPCRM.MSTPCRM08	SLVACCCTL5.TZCPCIE_SL
Supplemental Setting	—	—	SLVACCCTL5.TZCPCIE_SL

56.3 Register Descriptions (Root Complex Mode)

The registers in this section are availabe when the MODE_PORT bit in PCIE_MODE register is set to 1.

The registers are implemented in both ch0 and ch1, but the registers related to lane 1 are reserved for ch1.

Registers with the HwInit attribute

Some registers with the RO attribute stated in the PCI Express Base Specification are writable at the time of initialization. When writing to these registers, CFG_HWINIT_EN (Permission Register (offset: 0x0300) bit[2]) should be set to 1.

56.3.1 AXI Bridge Register

Registers can be accessed from the AXI only.

56.3.1.1 PCI_RC_REQDATAn : Request Data Register n (n = 0 to 2)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0080 + 0x4 × n

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Request Data Issue various requests to PCIe See Table 56.11.	R/W

Table 56.11 Relationship between transaction type and request data registers

Transaction type	Request Data Register 0	Request Data Register 1	Request Data Register 2
Zero-Length Read Request	Invalid	Invalid	Invalid
Config Write	Invalid	Invalid	Write data
Config Read	Invalid	Invalid	Invalid
Message Request	3rd Header	4th Header	Invalid
Message Request with data payload	3rd Header	4th Header	Message data

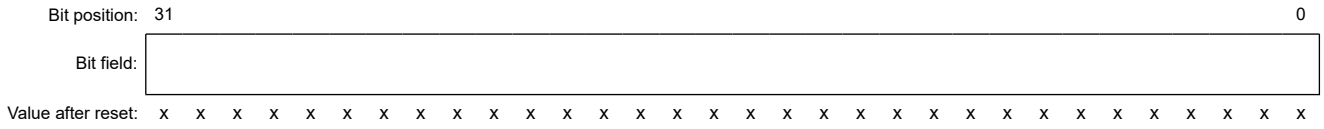
Note: The bits should be set to 0 for the requests indicated as "Invalid".

The PCI_RC_REQDATAn register issues various requests.

56.3.1.2 PCI_RC_REQRCVDAT : Request Receive Data Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x008C



Bit	Symbol	Function	R/W
31:0	n/a	Request Receive Data After issuing a read request, the data read on reception of the completion response are set in these bits. However, these bits are invalid for Zero-Lentgh Read and any kind of various write requests.	R

The PCI_RC_REQRCVDAT register indicates the data read on reception of the completion response after issuing a read request.

56.3.1.3 PCI_RC_REQADR1 : Request Address Register 1

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0090

Bit	Symbol	Function	R/W
31:0	n/a	Request Address Set the Address, etc. when issuing a Request. See Table 56.12.	R/W

The PCI_RC_REQADR1 register issues requests to PCIe.

Table 56.12 Relationship between transaction type and request address register 1

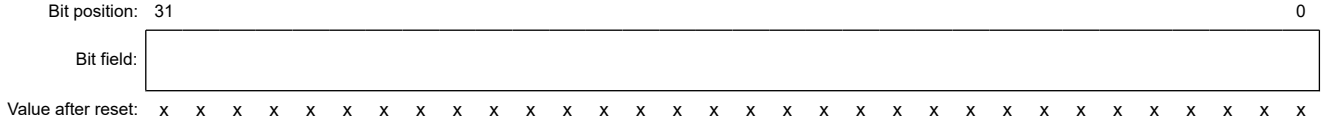
Transaction type	[31:27]	[26:24]	[23:19]	[18:16]	[15:12]	[11:8]	[7:2]	[1:0]
Zero-Length Read Request	Address							Reserved
Config Write	Bus number		Reserved	Function number	Reserved	Ext. Reg. number	Register number	Reserved
Config Read	Bus number		Reserved	Function number	Reserved	Ext. Reg. number	Register number	Reserved
Message Request	Reserved	Routing type	Reserved	Reserved	Reserved	Reserved	Message code	
Message Request with data payload	Reserved	Routing type	Reserved	Reserved	Reserved	Reserved	Message code	

Note: The bits should be set to 0 for the requests indicated as "Reserved".

56.3.1.4 PCI_RC_REQADR2 : Request Address Register 2

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0094



Bit	Symbol	Function	R/W
31:0	n/a	Request Address63:32 See Table 56.13.	R/W

The PCI_RC_REQADR2 register issues requests to PCIe.

Table 56.13 Relationship between transaction type and request address register 2

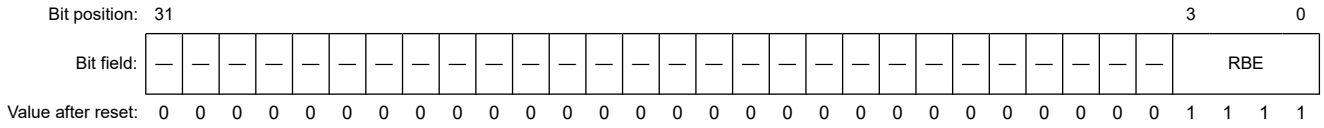
Transaction type	[31:0]
Zero-Length Read Request	Address
Config Write	Invalid
Config Read	Invalid
Message Request	Invalid
Message Request with data payload	Invalid

Note: The bits should be set to 0 for the requests indicated as "Invalid".

56.3.1.5 PCI_RC_REQBE : Request Byte Enable Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0098



Bit	Symbol	Function	R/W
3:0	RBE	Request Byte Enable Specify byte enable when issuing Cfg Request as required. Normally use 0xF. 0: Byte Enable enabled 1: Byte Enable disabled	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_REQBE register specifies the 1st DW byte enable bit of the TLP header when issuing a request to PCIe.

Table 56.14 Request Byte Enable settings (1 of 2)

Transaction type	RBE[3:0]
Zero-Length Read Request	0x0
Config Write	Random (Usually 0xF)
Config Read	Random (Usually 0xF)
Message Request	Invalid (0xF)

Table 56.14 Request Byte Enable settings (2 of 2)

Transaction type	RBE[3:0]
Message Request with data payload	Invalid (0xF)

56.3.1.6 PCI_RC_REQISS : Request Issue Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x009C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	RR	MOR_CD_PERR	MOR_CH_PERR	MOR_EP_ERR	MOR_STATUS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	FUNC[2:0]			TR_TYPE[3:0]				—	—	—	—	—	—	—	RI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RI	Request Issue 0: Write: No operation. Read: Request can be accepted. (Indicates that the processing of the issued Request has ended.) 1: Write: Request issuance. Read: Processing Request. (Indicates that the issued Request is being processed.)	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
11:8	TR_TYPE[3:0]	Set the type of Request. See Table 56.15.	R/W
14:12	FUNC[2:0]	Set the function of Request.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	MOR_STATUS[2:0]	These bits retain the MOR Status of Completion TLP for Non-Posted requests issued. It is not updated at the time of Posted request. 0 0 0: Successful Completion (SC) 0 0 1: Unsupported Request (UR) 0 1 0: Configuration Request Retry Status (CRS) (not supported) 0 1 1: Completion Timeout 1 0 0: Completer Abort (CA) 1 0 1: Unexpected Completion and mismatched type (Lock Completion responds to non-Lock Request) 1 1 0: Reserved 1 1 1: Overrun Completion length	R
19	MOR_EP_ERR	It is set to 1 when a Poisoned Completion TLP for a Non-Posted request issued by this register is received. Normally not used. It is not updated at the time of Posted request.	R
20	MOR_CH_PERR	It is set to 1 when a header error occurs in the Completion TLP for the Non-Posted request issued by this register. Normally not used. It is not updated at the time of Posted request.	R
21	MOR_CD_PERR	It is set to 1 when a data error occurs in the Completion TLP for the Non-Posted request issued by this register. Normally not used. It is not updated at the time of Posted request.	R

Bit	Symbol	Function	R/W
22	RR	Request Rejection Indicates that the stopped or hibernated state of the PCI Express transmit (TX) side was detected and the process was forcibly terminated. 0: Normal state (Request issued) 1: Rejection	R
31:23	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_REQISS register issues requests to PCIe.

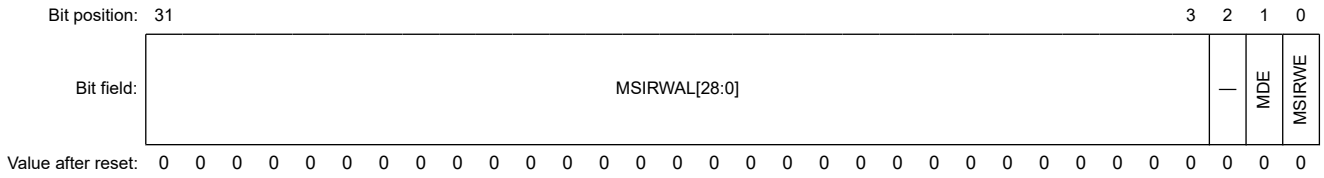
Table 56.15 Type of Request settings

Transaction type	TR_TYPE[11:8]	Posted/Non-posted	Device type	
			Root complex	Endpoint
Zero-Length Read Request	0x0	Non-posted	Issuable	Issuable
Configuration Read Type0	0x4	Non-posted	Issuable	Issuing prohibited
Configuration Write Type0	0x5	Non-posted	Issuable	Issuing prohibited
Configuration Read Type1	0x6	Non-posted	Issuable	Issuing prohibited
Configuration Write Type1	0x7	Non-posted	Issuable	Issuing prohibited
Message Request	0x8	Posted	Issuable	Issuable
Message Request with data payload	0x9	Posted	Issuable	Issuable
	Others	—	Issuing prohibited	Issuing prohibited

56.3.1.7 PCI_RC_MSIRCVWADRL : MSI Receive Window Address (Lower) Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0100



Bit	Symbol	Function	R/W
0	MSIRWE	MSI Receive Window Enable 0: Window is disabled. 1: Window is enabled.	R/W
1	MDE	Message Data Enabled Control whether to use Message Data in MSI reception judgment. The MSI reception judgment result selected by this register is reflected in the INTMSI_RC interrupt. 0: Not used (conventional mode) 1: Used (multiple interrupts)	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
31:3	MSIRWAL[28:0]	MSI Receive Window Address (Lower) Set Start Address[31:3] of MSI receive window. However, it must be aligned with the size set by MSI Receive Window Mask. Even if you set a non-aligned address, the address bit for which the MSI Receive Window Mask is set will be 0. Make sure that MSI Receive Window Enable (MSIRWE) is 0 when changing this register.	R/W

The PCI_RC_MSIRCVWADRL register sets the start address of the MSI memory space to receive.

56.3.1.8 PCI_RC_MSIRCVWADRU : MSI Receive Window Address (Upper) Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0104

Bit position:	31																												0
Bit field:	MSIRWAU[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MSIRWAU[31:0]	MSI Receive Window Address (Upper) Set the Start Address[63:32] of the MSI receive window. However, it must be aligned with the size set by MSI Receive Window Mask. Even if you set a non-aligned address, the address bit for which the MSI Receive Window Mask is set will be 0. Make sure that MSI Receive Window Enable is 0 when changing this register.	R/W

The PCI_RC_MSIRCVWADRU register sets the start address of the MSI memory space to receive.

56.3.1.9 PCI_RC_MSIRCVWMSKL : MSI Receive Window Mask (Lower) Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0108

Bit position:	31																												0
Bit field:	MSIRWML[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
31:0	MSIRWML[31:0]	MSI Receive Window Mask (Lower) 11b fixed. MSI Receive Window Mask [1:0] should always be set to Mask.	R/W

The PCI_RC_MSIRCVWMSKL register specifies the size of the area from the address set in the MSI Receive Window Address bits.

56.3.1.10 PCI_RC_MSIRCVWMSKU : MSI Receive Window Mask (Upper) Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x010C

Bit position:	31	30																											0
Bit field:	—	MSIRWMU[30:0]																											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
30:0	MSIRWMU[30:0]	MSI Receive Window Mask (Upper) Set [62:32] for MSI Receive Window Mask.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_RC_MSIRCVWMSKU register specifies the size of the area from the address set in the MSI Receive Window Address bits.

56.3.1.11 PCI_RC_PINTRCVIE : PCI INTx Receive Interrupt Enable Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0110

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MSIRIE	INTDRIE	INTCRIE	INTBRIE	INTARIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INTARIE	INTA Receive Interrupt Enable Enable INTA_RC. 0: disable 1: enable	R/W
1	INTBRIE	INTB Receive Interrupt Enable Enable INTB_RC. 0: disable 1: enable	R/W
2	INTCRIE	INTC Receive Interrupt Enable Enable INTC_RC. 0: disable 1: enable	R/W
3	INTDRIE	INTD Receive Interrupt Enable Enable INTD_RC. 0: disable 1: enable	R/W
4	MSIRIE	MSI Receive Interrupt Enable Enable INTMSI_RC with MSI reception. 0: disable 1: enable	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_PINTRCVIE register enables INTx_RC interrupts.

56.3.1.12 PCI_RC_PINTRCVIS : PCI INTx Receive Interrupt Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0114

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MSIRIS	INTDRIS	INTCRIS	INTBRIS	INTARIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INTARIS	INTA Receive Interrupt Status It is set when an Assert INTA Message is received, and cleared when a Deassert INTA Message is received.	R/W
1	INTBRIS	INTB Receive Interrupt Status It is set when an Assert INTB Message is received, and cleared when a Deassert INTB Message is received.	R/W
2	INTCRIS	INTC Receive Interrupt Status It is set when an Assert INTC Message is received, and cleared when a Deassert INTC Message is received.	R/W
3	INTDRIS	INTD Receive Interrupt Status It is set when an Assert INTD Message is received, and cleared when a Deassert INTD Message is received.	R/W
4	MSIRIS	MSI Receive Interrupt Status MSI Receive Window Set when Memory Write Req is received from the PCI bus side in the set area. This bit is set when MSI is issued to AXI as a Write Transaction and the Response is returned.	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_PINTRCVIS register indicates the INTx_RC interrupt factor. When Assert_INTx is received in response to a message request from PCIe, the corresponding bit in the PCI_RC_PINTRCVIS register is set and interrupt INTx_RC is asserted. When Deassert_INTx is received in response to a message request, the corresponding bit in the PCI_RC_PINTRCVIS register is cleared and interrupt INTx_RC is deasserted. Although the PCI_RC_PINTRCVIS register can be cleared by writing 1 by software, it is not recommended to clear this bit by the software itself during normal operation for interrupts on the bus for PCIe. The fields of the PCI_RC_PINTRCVIS register are set if each factor is detected, regardless of the setting of the PCI_INTx_Receive_Interrupt_Enable register for RC.

56.3.1.13 PCI_RC_MSGRCVIE : Message Receive Interrupt Enable Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0120

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	MRI	—	—	—	—	PMASNR	PMPMERI	PMETORI	PMETOARI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	PMETOARI	PME_TO_Ack Receive Interrupt Enable control of MSG_INT assertion by PME_TO_Ack reception 0: Disable 1: Enable	R/W
17	PMETORI	PME_Turn_Off Receive Interrupt Enable control of MSG_INT assertion by PME_Turn_Off reception 0: Disable 1: Enable	R/W
18	PMPMERI	PM_PME Receive Interrupt Enable control of MSG_INT assert by PM_PME reception 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
19	PMASNR	PM_Active_State_Nak Receive Enable control of MSG_INT assertion by PM_Active_State_Nak reception 0: Disable 1: Enable	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	MRI	Message Receive Interrupt Enable control of MSG_INT assert by message reception 0: enable 1: disable	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_MSGRCVIE register controls enabling of MSG_INT in response to the reception of message requests other than INTx and error-related messages.

56.3.1.14 PCI_RC_MSGRCVIS : Message Receive Interrupt Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0124

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	MRI	—	—	—	—	PMASNR	PMPMERI	PMETORI	PMETOARI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	PMETOARI	PME_TO_Ack Receive Interrupt Set when receiving a PM_TO_Ack message	R/W
17	PMETORI	PME_Turn_Off Receive Interrupt Disabled due to RC	R/W
18	PMPMERI	PM_PME Receive Interrupt Set when receiving a PM_PME message	R/W
19	PMASNR	PM_Active_State_Nak Receive Disabled due to RC	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	MRI	Message Receive Interrupt Set when receiving a message (does not depend on the type of message)	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_MSGRCVIS register is a status register that indicates the reception of message requests other than INTx and error-related messages. The value of the PCI_RC_MSGRCVIS register is reflected in MSG_INT. Only the message code is used to judge the message type.

The validity of routing and validity of Msg/MsgD selection are not verified. The corresponding message is considered to have been received.

56.3.1.15 PCI_RC_MSGCODE : Message Code Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0130

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MC[7:0]								R[2:0]			—	—	—	—	MP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MP	Message Payload Stores the presence or absence of the data payload of the last received Message. When Power Management Message (PME_TO_Ack Message, PME_Turn_Off Message, PM_PME Message, PM_Active_State_Nak Message) is received, this register is not written. 0: Msg (Without Payload) 1: MsgD (with Payload)	R
4:1	—	These bits are read as 0.	R
7:5	R[2:0]	Routing Stores the Routing of the last received Message.	R
15:8	MC[7:0]	Message Code Stores the Code of the last received Message.	R
31:16	—	These bits are read as 0.	R

The PCI_RC_MSGCODE register stores the code and routing of the last received message.

56.3.1.16 PCI_RC_MSGDATA : Message Data Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0134

Bit position:	31	0
Bit field:		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	n/a	Message Data Stores the 1DW data of the last received Message. It is updated only when MsgD (with Data) is received, and the previous value is retained when Msg (without Data) is received. When Power Management Message (PME_TO_Ack Message, PME_Turn_Off Message, PM_PME Message, PM_Active_State_Nak Message) is received, this register is not written.	R

The PCI_RC_MSGDATA register stores the data of the last received message.

56.3.1.17 PCI_RC_MSGH3DW : Message Header 3rdDW Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0138

Bit position:	31																													0
Bit field:																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Message Header 3rdDW Stores the Header (3rdDW) of the last received Message.	R

The PCI_RC_MSGH3DW register stores the header (the 3rd DW) of the last received message.

56.3.1.18 PCI_RC_MSGH4DW : Message Header 4thDW Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x013C

Bit position:	31																													0
Bit field:																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Message Header 4thDW Registers Stores the Header (4thDW) of the last received Message.	R

The PCI_RC_MSGH4DW register stores the header (the 4th DW) of the last received message.

56.3.1.19 PCI_RC_INTTABLE : Interrupt Table Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0140

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	INT_LINK_EQUALIZATION_REQUEST	—	—	—	—	AXI_ERR_IN	PCIE_EVT_INT	MSG_INT	—	—	—	INTMSI_RC	INTD_RC	INTC_RC	INTB_RC	INTA_RC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INTA_RC	INTA_RC interrupt signal monitor	R
1	INTB_RC	INTB_RC interrupt signal monitor	R
2	INTC_RC	INTC_RC interrupt signal monitor	R
3	INTD_RC	INTD_RC interrupt signal monitor	R

Bit	Symbol	Function	R/W
4	INTMSI_RC	INT_MSI interrupt signal monitor	R
7:5	—	These bits are read as 0.	R
8	MSG_INT	Message interrupt signal monitor	R
9	PCIE_EVT_INT	Event interrupt signal monitor	R
10	AXI_ERR_IN	Error interrupt signal monitor	R
14:11	—	These bits are read as 0.	R
15	INT_LINK_EQUALIZATION_REQUEST	LINK_EQUALIZATION_REQUEST Interrupt signal level monitor	R
23:16	—	The read values are undefined.	R
31:24	—	These bits are read as 0.	R

The PCI_RC_INTTABLE is an index of interrupt factors. The interrupt signal (active high) status of each category can be monitored in a list.

56.3.1.20 PCI_RC_PEIE0 : PCIe Event Interrupt Enable 0 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	UI_LINK_WIDTH_CHANGE_DONE_EN	UI_LINK_SPEED_CHANGE_DONE_EN	RDEN	—	—	—	CA_EN	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	RX_DLLP_PM_ENTER_L23_EN	—	ASPML1REN	DLUDEN	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	—	These bits are read as 0. The write value should be 0.	R/W
9	DLUDEN	DL_UpDown EN Interrupt enable on DL state change 0: Disable 1: Enable	R/W
10	ASPML1REN	ASPM L1 Rejected interrupt enable 0: Disable 1: Enable	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	RX_DLLP_PM_ENTER_L23_EN	RX_DLLP_PM_ENTER_L23 interrupt enable 0: Disable 1: Enable	R/W
23:13	—	These bits are read as 0. The write value should be 0.	R/W
24	CA_EN	CA (Completer Abort) interrupt enable 0: Disable 1: Enable	R/W
27:25	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
28	RDEN	Request complete interrupt enable 0: Disable 1: Enable	R/W
29	UI_LINK_SPEED_CHANGE_DONE_EN	Speed change operation completion interrupt enable 0: Disable 1: Enable	R/W
30	UI_LINK_WIDTH_CHANGE_DONE_EN	Up/Down Configure operation complete interrupt enable 0: Disable 1: Enable	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_RC_PEIE0 register enables interrupts of the various PCI Express event factors. The PCI_RC_PEIE0 register enables writing to the PCIe Event Interrupt Status 0 Register. See the description of this status register for details of the factors.

56.3.1.21 PCI_RC_PEIS0 : PCIe Event Interrupt Status 0 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0204

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	UI_LINK_WIDTH_CHANGE_DONE	UI_LINK_SPEED_CHANGE_DONE	RD	—	—	—	CA	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	RX_DLLP_PM_ENTER_L23	—	ASPML1R	DLUDEN	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	—	These bits are read as 0. The write value should be 0.	R/W
9	DLUDEN	DL_UpDown EN Set to 1 on transition from DL_Down state to DL_Up state, or DL_Up state to DL_Down state. Check the DL_Down/DL_Up status with PCIe Core Status 1 Registers (Offset: 0x408).	R/W
10	ASPML1R	ASPM L1 Rejected Indicates rejected ASPM L1 transition	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	RX_DLLP_PM_ENTER_L23	Indicates transition to L2/L3 State in Power Management control.	R/W
23:13	—	These bits are read as 0. The write value should be 0.	R/W
24	CA	Indicates that the device has responded with CA (Completer Abort).	R/W
27:25	—	These bits are read as 0. The write value should be 0.	R/W
28	RD	Request Done For requests submitted in the Request Issue Registers (Offset: 0x9C): Non-Posted: Indicates that Completion has been received. Posted: Indicates that the request has finished being submitted.	R/W
29	UI_LINK_SPEED_CHANGE_DONE	Indicates completion of Speed Change operation.	R/W

Bit	Symbol	Function	R/W
30	UI_LINK_WIDTH_CHANGE_DONE	Indicates completion of Up/DownConfigure operation.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_RC_PEIS0 register is a status register that indicates each PCI Express event. Set to 1 by the factor in the table. After checking the factor, write 1 to clear it.

56.3.1.22 PCI_RC_PEIE1 : PCIe Event Interrupt Enable 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0208

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXB_PARITY_ERR_EN	ERR_RPC_REPLAYFIFO_PERR_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ERR_REPLAY_UPPER_CORRECTABLE_ERROR_EN	ERR_REPLAY_LOWER_CORRECTABLE_ERROR_EN	—	—	—	—	—	—	ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR_EN	ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR_EN	Enable ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR interrupts. Interrupt notification enable setting when an error of ECC 2-bit or more (Uncorrectable Error) occurs in the lower 64-bit data bus of the Replay FIFO installed in the Data Link Layer 0: Disable 1: Enable	R/W
1	ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR_EN	Enable ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR interrupts. Interrupt notification enable setting when an error of ECC 2-bit or more (Uncorrectable Error) occurs in the upper 64-bit data bus of the Replay FIFO installed in the Data Link Layer 0: Disable 1: Enable	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	ERR_REPLAY_LOWER_CORRECTABLE_ERROR_EN	Enable ERR_REPLAY_LOWER_CORRECTABLE_ERROR interrupts. Interrupt notification enable setting when an ECC 1-bit error (Correctable Error) occurs in the lower 64-bit data bus of the Replay FIFO installed in the Data Link Layer 0: Disable 1: Enable	R/W
9	ERR_REPLAY_UPPER_CORRECTABLE_ERROR_EN	Enable ERR_REPLAY_UPPER_CORRECTABLE_ERROR interrupts. Interrupt notification enable setting when an ECC 1-bit error (Correctable Error) occurs in the upper 64-bit data bus of the Replay FIFO installed in the Data Link Layer 0: Disable 1: Enable	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	ERR_RPC_REPLAY FIFO_PERR_EN	Enable ERR_RPC_REPLAYFIFO_PERR interrupts. Parity error interrupt notification enable setting for Replay FIFO installed in Data Link Layer 0: Disable 1: Enable	R/W
17	TXB_PARITY_ERR_ EN	Enable the TXB_PARITY_ERR interrupt. Parity error interrupt notification enable setting for TX Buffer installed in Transaction Layer 0: Do not allow 1: Allow	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_PEIE1 register enables parity error and ECC error interrupts. When each bit is set to the valid setting, the value of each corresponding status bit of the PCIe Event Interrupt Status 1 Register becomes valid.

56.3.1.23 PCI_RC_PEIS1 : PCIe Event Interrupt Status 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x020C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXB_PARITY_ERR	ERR_RPC_REPLAYFIFO_PERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ERR_REPLAY_UPPER_CORRECTABLE_ERROR	ERR_REPLAY_LOWER_CORRECTABLE_ERROR	—	—	—	—	—	—	ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR	ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR	ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR interrupt Interrupt notification when an ECC 2-bit or more error (Uncorrectable Error) occurs in the lower 64-bit data bus of the Replay FIFO installed in the Data Link Layer	R/W
1	ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR	ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR interrupt Interrupt notification when an ECC 2-bit or more error (Uncorrectable Error) occurs in the upper 64-bit data bus of the Replay FIFO installed in the Data Link Layer	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	ERR_REPLAY_LOWER_CORRECTABLE_ERROR	ERR_REPLAY_LOWER_CORRECTABLE_ERROR interrupt Interrupt notification when an ECC 1-bit error (Correctable Error) occurs in the lower 64-bit data bus of the Replay FIFO installed in the Data Link Layer	R/W
9	ERR_REPLAY_UPPER_CORRECTABLE_ERROR	ERR_REPLAY_UPPER_CORRECTABLE_ERROR interrupt Interrupt notification when an ECC 1-bit error (Correctable Error) occurs in the upper 64-bit data bus of the Replay FIFO installed in the Data Link Layer	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	ERR_RPC_REPLAYFIFO_PERR	ERR_RPC_REPLAYFIFO_PERR interrupt Parity error interrupt notification of Replay FIFO installed in Data Link Layer	R/W
17	TXB_PARITY_ERR	TXB_PARITY_ERR interrupt Parity error interrupt notification of TX Buffer installed in Transaction Layer	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_PEIS1 register is a status register that indicates parity error and ECC error interrupts. After checking the factor, write 1 to the corresponding bit to clear it.

56.3.1.24 PCI_RC_AMEIE : AXI Master Error Interrupt Enable Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0210

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WMSTERRINTEN[3:0]				—	—	—	—	RMSTERRINTEN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	RMSTERRINTEN[3:0]	Read MSTERR INT Enable Each corresponding bit can be turned on/off individually. 0: Disable 1: Enable	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	WMSTERRINTEN[3:0]	Write MSTERR INT Enable Each corresponding bit can be turned on/off individually. 0: Disable 1: Enable	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_AMEIE register enables the AXI master error interrupt.

56.3.1.25 PCI_RC_AMEIS : AXI Master Error Interrupt Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0214

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	WERRID[3:0]				—	—	—	—	RERRID[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WMSTERRINT[3:0]				—	—	—	—	RMSTERRINT[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	RMSTERRINT[3:0]	Read MSTERR INT Indicates that an error was detected in the AXI Master Port. Only the first detected error is saved, and when bit [3:0] is cleared, a new error can be saved. bit 3: Length Error (When the length of the data received on the TER and the data channel do not match.) bit 2: ID Mismatch (When the MARID and MRID values received on the data channel are different.) bit 1: When DECERR is received bit 0: When SLVERR is received	R/W

Bit	Symbol	Function	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	WMSTERRINT[3:0]	Write MSTERR INT Indicates that an error was detected in the AXI Master Port. Only the first detected error is saved, and when bit [11:8] is cleared, a new error can be saved. bit 11: Length Error (When the length of the data sent by the TEF and the data channel do not match.) bit 10: ID Mismatch (When the MBID value received on the MAWID response channel is different.) bit 9: When DECERR is received bit 8: When SLVERR is received	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RERRID[3:0]	Read ERR ID Save the ID of the first DECERR/SLVERR received. A new error ID can be saved when bit [3:0] is cleared. 0x0: normal access Others: Error ID	R
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	WERRID[3:0]	Write ERR ID Save the ID of the first DECERR/SLVERR received. When bit [11:8] is cleared, a new error ID can be saved. 0x0: normal access Others: Error ID	R
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_AMEIS register indicates the AXI master error interrupt status.

56.3.1.26 PCI_RC_ASEIE1 : AXI Slave Error Interrupt Enable 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0220

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WSLVERRINTEN[3:0]				—	—	—	—	—	—	RSLVERRINTEN[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RSLVERRINTEN[1:0]	Read SLVERR INT EN Enable Read SLVERR INT. Each corresponding bit can be turned on/off individually. 0: Disable 1: Enable	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
11:8	WSLVERRINTEN[3:0]	Write SLVERR INT EN Enable Write SLVERR INT. Each corresponding bit can be turned on/off individually. 0: Disable 1: Enable	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_ASEIE1 register enables the AXI slave error interrupt.

56.3.1.27 PCI_RC_ASEIS1 : AXI Slave Error Interrupt Status 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0224

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WSLVERRINT[3:0]				—	—	—	—	—	—	RSLVERRINT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RSLVERRINT[1:0]	Read SLVERR INT Indicates that an unrecoverable error was detected in the AXI Slave Port. (Transaction response will be SLVERR.) bit 1: Burst Type Disabled (When SARBURST is 11b (undefined). When SARBURST is 10b (wrapping) and burst length is other than 2, 4, 8, 16.) bit 0: Data Size Invalid (When SARSIZE is (exceeding AXI Bus width/not supported).) Each bit means: 0: No error detected 1: error detection	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
11:8	WSLVERRINT[3:0]	Write SLVERR INT Indicates that an unrecoverable error was detected in the AXI Slave Port. (Transaction response will be SLVERR.) bit 11: Burst Length Error (When the SAWLEN and the burst length of the data received on the data channel do not match.) bit 10: ID Mismatch (When the SAWID and SWID values received on the data channel are different.) bit 9: Burst Type Disabled (When SAWBURST is 11b (undefined). When SAWBURST is 10b (wrapping) and burst length is other than 2, 4, 8, or 16.) bit 8: Data Size Invalid (When SAWSIZE is between 100b and 111b (Exceeding AXI Bus width/not supported).) Each bit means: 0: No error detected 1: error detection	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_ASEIS1 register indicates the AXI slave error interrupt status.

56.3.1.28 PCI_RC_ASEIS3 : AXI Slave Error Interrupt Status 3 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0230

Bit position:	31																													0
Bit field:																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	ERR ID Save the ID when the first error occurred in the AXI Slave Error Interrupt Status1 register (Offset 0x224). Only the first detected error is saved, and when bit[11:8] and bit[1:0] of the AXI Slave Error Interrupt Status1 register are all cleared, a new error ID can be saved.	R

The PCI_RC_ASEIS3 register indicates the AXI ID at the time of the first AXI slave error.

56.3.1.29 PCI_RC_PERM : Permission Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CFG_HWINIT_EN	PHY_REG_CLK_EN	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	PHY_REG_CLK_EN	Write enable for Physical Layer Control/Status Registers. 0: Disable 1: Enable	R/W
2	CFG_HWINIT_EN	Enable HwInit attribute registers to be rewritten from AXI. 0: Disable 1: Enable	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

56.3.1.30 PCI_RC_RESET : Reset Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0310

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FTD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	RST_PREG_B	RST_OUT_B	RST_PS_B	RST_LOAD_B	RST_CFG_B	RST_RSM_B	RST_GP_B	RST_B
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RST_B	Reset to PCI Express core part inside macro 0: reset 1: normal operation	R/W
1	RST_GP_B	Reset to the PCI Express core part (ACLK domain) inside the macro 0: reset 1: normal operation	R/W
2	RST_RSM_B	POWERGOOD reset of AUX Power (AUX not supported) Reset to Sticky register. Reset to PHY. 0: reset 1: normal operation	R/W

Bit	Symbol	Function	R/W
3	RST_CFG_B	Reset to Configuration Register 0: Reset 1: normal operation	R/W
4	RST_LOAD_B	Reset to Configuration Register Reset to bits not initialized by RST_CFG_B. 0: reset 1: normal operation	R/W
5	RST_PS_B	Reset to the PCI Express core part (PCLK domain) inside the macro 0: reset 1: normal operation	R/W
6	RST_OUT_B	RST_OUT_B output 0: reset 1: normal operation	R/W
7	RST_PREG_B	Not used in this macro. The setting value does not affect macro behavior. 0: reset 1: normal operation	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	FTD0	force to D0 After sending PME_TO_Ack, force PM Control to transition to D0 State. Auto-cleared when PM Control transitions to D0. This bit is normally prohibited as it can create inconsistencies with the power state of the entire system. 0: No operation 1: Transition to D0	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_RESET register resets the PCIe core. Supplied to the internal core. The write value is saved when accessed from the AXI, but a low-level pulse is generated when the PCIe writes 0. However, when the AXI has already written 0, the signal remains at the low level.

Writing 1 from the PCIe will be ignored.

56.3.1.31 PCI_RC_MSET0 : Mode Set 0 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0314

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	AWPROT[2:0]			AWCACHE_L[3:0]				—	—	AWLOCK[1:0]		AWCACHE_D[3:0]			
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ARPROT[2:0]			—	—	—	—	—	—	ARLOCK[1:0]		ARCACHE[3:0]			
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ARCACHE[3:0]	Lock type for PCIe-to-AXI transactions. This bit provides information about the atomic nature of the transfer. 0 0: Normal access 0 1: Exclusive access 1 0: Locked access 1 1: Reserved	R/W

Bit	Symbol	Function	R/W
5:4	ARLOCK[1:0]	Lock type for PCIe-to-AXI transactions. This bit provides information about the atomic nature of the transfer. 0 0: Normal access 0 1: Exclusive access 1 0: Locked access 1 1: Reserved	R/W
11:6	—	These bits are read as 0. The write value should be 0.	R/W
14:12	ARPROT[2:0]	Sets the protection type for PCIe→AXI transactions. This bit indicates whether the transaction's protection level is Normal, Privileged, or Secure, and whether the transaction is a data or instruction access. [2] 1: instruction access / 0: data access [1] 1: non-secure access / 0: secure access [0] 1: privileged access / 0: normal access	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
19:16	AWCACHE_D[3:0]	Indicates the value of MAWCACHE[3:0] to be issued to AXI. This setting is output when issuing an AXI request other than the output condition of AWCACHE_L. *Recommended value is 0001b. [3] 1: write allocatable / 0: not write allocatable [2] 1: Read assignable / 0: Read assignable [1] 1: cacheable / 0: non-cacheable [0] 1: Bufferable / 0: Not Bufferable	R/W
21:20	AWLOCK[1:0]	Lock type for PCIe-to-AXI transactions. This signal provides information about the atomic nature of the transfer. 0 0: Normal access 0 1: Exclusive access 1 0: Locked access 1 1: Reserved	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
27:24	AWCACHE_L[3:0]	Indicates the value of MAWCACHE[3:0] to be issued to AXI. This setting is printed when issuing an AXI request containing the last byte. *Recommended value is 0000b. [3] 1: write allocatable / 0: not write allocatable [2] 1: Read assignable / 0: Read assignable [1] 1: cacheable / 0: non-cacheable [0] 1: Bufferable / 0: Not Bufferable	R/W
30:28	AWPROT[2:0]	Sets the protection type for PCIe→AXI transactions. This bit indicates whether the transaction's protection level is Normal, Privileged, or Secure, and whether the transaction is a data or instruction access. [2] 1: instruction access / 0: data access [1] 1: non-secure access / 0: secure access [0] 1: privileged access / 0: normal access	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_RC_MSET0 register sets AXI mode.

56.3.1.32 PCI_RC_MSET1 : Mode Set 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0318

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AXIMIW[3:0]				AXIMIR[3:0]				AXIMMB[3:0]				—	—	RAMPE	PCIERO
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	0

Bit	Symbol	Function	R/W
0	PCIERO	PCIe Request Order Issue a Read Request to PCIe from the same AXI master without waiting for Completion. Set to 1 if you want to strictly follow the order of Requests to the Completer. 0: Do not wait for Completion. 1: Wait for Completion.	R/W
1	RAMPE	RAM Parity Enable Sets whether or not to check the parity of the internal SRAM. The default value is Enable, but it is ignored by non-Parity macros. 0: RAM parity check disabled 1: RAM parity check enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	AXIMMB[3:0]	AXI Master Max Burst Sets the maximum burst length as an AXI Master operation.	R/W
11:8	AXIMIR[3:0]	AXI Max Issue Read Set the number of read issues that can be issued by AXI Master. Set within the range allowed by Interconnect. 0x0: possible number 1 0x1: possible number 2 ⋮ 0xF: Possible number 16	R/W
15:12	AXIMIW[3:0]	AXI Max Issue Write Set the number of writes that can be issued by AXI Master. Set within the range allowed by Interconnect. 0x0: possible number 1 0x1: possible number 2 ⋮ 0xF: Possible number 16	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_MSET1 register sets AXI mode.

56.3.1.33 PCI_RC_MSET3 : Mode Set 3 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0380

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ASPML1IT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ASPML1IT[7:0]	ASPM L1 Idle Time Sets the idle period for AXI transactions that the macro checks on ASPM L1 transitions. One of the conditions for ASPM L1 transition is that the idle period is confirmed for the number of cycles of the 8 bits set by this bit plus 0xFF to the lower 8 bits. 0x00: 256 [ACLK] 0x01: 512 [ACLK] ⋮ 0xFF: 65536 [ACLK]	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_MSET3 register outputs the setting value as the ASPM L1 Idle Time bit.

56.3.1.34 PCI_RC_MSET4 : Mode Set 4 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0384

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MODE_EQ_RX_EVAL_WAIT[19:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MODE_EQ_RX_EVAL_WAIT[19:0]				—	—	—	—	MODE_8GT_EQTS2	MODE_EQ_PHASE23_ENABLE1	MODE_EQ_AUTONOMOUS1	—	—	UI_EQ_REQTYPE[1:0]		UI_EQ_REQUEST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UI_EQ_REQUEST	Set when sending an Equalization Request with Quiesce Guarantee = 1 in the Recovery.RcvrCfg state. 1 set prohibited except when MODE_QUIESCE_GUARANTEE = 1. After setting 1, hold until confirmation of UI_EQ_DONE = 1. 0: Do not send Equalization Request 1: Send Equalization Request	R/W
2:1	UI_EQ_REQTYPE[1:0]	Operation specification when sending Equalization Request with Quiesce Guarantee = 1 in Recovery.RcvrCfg state 0 0: Equalization Request transmission with Equalization Request Data Rate = 0 (8G) 0 1: Send Equalization Request with Equalization Request Data Rate = 1 (16G) (Setting prohibited) 1 0: Operation of 0 + set Link Equalization Request 8.0 GT/s register 1 1: Set 1 operation + Link Equalization Request 16.0 GT/s register (setting prohibited)	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	MODE_EQ_AUTONOMOUS1	Fixed 0	R/W
6	MODE_EQ_PHASE23_ENABLE1	Fixed 0	R/W
7	MODE_8GT_EQTS2	8GT EQ TS2OS transmission enable/disable setting for Upstream Port Note: Not used with Downsream Port. (fixed to 0) 0: Do not send 1: Send	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
31:12	MODE_EQ_RX_EVAL_WAIT[19:0]	During RxEval (Downstream Port Phase3, Upstream Port Phase2), wait time setting until RxEval is executed when Block Alignment cannot be obtained. Set by the count number of PCLK. Note: It is necessary to set the register value (recommended: equivalent to 1 msec) before starting. Please set from the local CPU.	R/W

56.3.1.35 PCI_RC_MSET5 : Mode Set 5 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0388

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MODE_8GT_EQTS2_PRESET[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MODE_8GT_EQTS2_PRESET[15:0]	Transmitter Preset value of 8GT EQ TS2OS to transmit at Upstream Port (USP). No setting is required if 8GT_EQTS2 is not sent from USP (DSP Preset is not specified). Settings must be made in advance if necessary. 4-bit configuration for each lane. Gen3 supports up to x2 lane configurations. The correspondence between each bit and lane number is as follows. [3:0] Lane #0 [7:4] Lane #1 (Not used in case of ch1) [11:8] Lane #2 (Not used. Fixed to 0x0) [15:12] Lane #3 (Not used. Fixed to 0x0) The above lane number is the default lane number for circuit implementation. Note that this is not a negotiated lane number. Note: Not used with Downsream Port. (fixed to 0x0000_0000)	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

56.3.1.36 PCI_RC_MSTAT0 : Mode Status 0 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0390

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	RX_EQ_REQTYPE[1:0]		RX_EQ_REQUEST	UI_EQ_DONE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UI_EQ_DONE	Set Equalization Request with Quiesce Guarantee=1 in Recovery .RcvrCfg state when sending Related flow: "Set UI_EQ_REQUEST = 1 → Confirm UI_EQ_DONE = 1 → Clear UI_EQ_REQUEST = 0" 0: No Equalization Request sent 1: Sent Equalization Request	R
1	RX_EQ_REQUEST	Status display when receiving 8 consecutive TS2OS with Request Equalization bit (Symbol6, bit7) = 1 in Recovery.RcvrCfg state 0: No Equalization Request received 1: Equalization Request received	R
3:2	RX_EQ_REQTYPE[1:0]	Quiesce Guarantee and Equalization Request Data Rate bit display when receiv ing 8 consecutive TS2OS with Request Equalization bit (Symbol6, bit7) = 1 in Recovery.RcvrCfg state EP (USP) (1) When Request Equalization is received bit 0: Quiesce Guarantee (TS2OS, Symbol6, bit6) (= 1 fixed) bit 1: Equalization Request Data Rate (TS2OS, Symbol6, bit5) (2) When receiving EQTS2OS bit 0: 0 (fixed) bit 1: Equalization Request Data Rate (0: EQTS2, 1: 8GT EQTS2) RC (DSP) bit 0: Quiesce Guarantee (TS2OS, Symbol6, bit6) bit 1: Equalization Request Data Rate (TS2OS, Symbol6, bit5)	R
31:4	—	These bits are read as 0.	R

56.3.1.37 PCI_RC_PCMSET1 : PCIe Core Mode Set 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0400

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	ASPM_L1_INTERVAL_TIME[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MODE_LINK_UPCONFIGURE_CAPABILITY_DISABLED	—	MODE_TXSWING	—	—	—	—	—	—	—	—	MODE_PORT	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	MODE_PORT	Device type setting register. When the bit value of this register is fixed to the initial value 1, the setting of the MODE_PORT bit in PCIE_MODE register becomes valid. When the MODE_PORT bit in PCIE_MODE register is fixed to 1, the setting by this register bit becomes valid. 0: Endpoint 1: Root Complex	R/W
9:2	—	These bits are read as 0. The write value should be 0.	R/W
10	MODE_TXSWING	SerDes serial output amplitude control 0: Full swing mode (default) 1: Half swing mode	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	MODE_LINK_UPCONFIGURE_CAPABILITY_DISABLED	Setting of Link Upconfigure Capability bit of Training Sequence Ordered-set (TS-OS) When connecting with a Gen1 PCIe device, LinkUp may not occur unless this bit is 0. In that case, change it to 0 in F/W. 0: Link Upconfigure Capability bit = 1 setting 1: Link Upconfigure Capability bit = 0 setting (Gen1 x1)	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
27:16	ASPM_L1_INTERVAL_TIME[11:0]	Interval settings for ASPM L1 requests The PCIe Base Spec stipulates that ASPM L1 transition requests must not be accepted continuously within 10 µs, and this field sets the timer value to guard against this. Set so that ACLK cycle x set value is 10 µs or more. At ACLK: 400 [MHz] or higher, this bit setting value should be the value (1/16) of the lower 4 bits of the setting value of the above specifications, and set the clock number as the timer value setting. Settings of this product (Default): When ACLK = 400 MHz(2.5 ns): 4000(d)/16 = FA(h)	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_PCMSET1 register sets the operating mode of the PCI Express core.

56.3.1.38 PCI_RC_PCCTRL1 : PCIe Core Control 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0404

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	BLB_RELAX_ORDERING_EN	—	—	—	—	—	UI_ENTER_L1S	—	RETURN_TO_L0	UI_RC_REJECT_ASPML1	APMASN	UI_ENTER_L2	UI_ENTER_TXL0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MODE_QUIESCE_GUARANTEE	UI_ENTER_TXMODE_SRIS	MODE_EQ_AUTONOMOUS	MODE_EQ_PHASE23_ENABLE	MODE_RESET_EIEOS_INTERVALL0S	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	MODE_RESET_EIEOS_INTERVALL0S	Gen3 features: Reset EIEOS Interval information of bit 2, symbol 6 of TS1OS transmitted in Recovery.Equalization state	R/W
9	MODE_EQ_PHASE23_ENABLE	Gen3 features: Setting whether to execute EQ PHASE2 and EQ PHASE3 in RC mode (MODE_PORT=1). 0: Do not execute EQ PHASE2/3 1: Execute EQ PHASE2/3	R/W
10	MODE_EQ_AUTONOMOUS	Gen3 feature: Autonomous Equalization Basically only changeable during the reset period 0: Do not use Autonomous Mechanism 1: Use Autonomous Mechanisms	R/W
11	UI_ENTER_TXMODE_SRIS	Setting Clock Tolerance Compensation 0: SRNS (default) 1: SRIS (not supported)	R/W
12	MODE_QUIESCE_GUARANTEE	Symbol6 bit6 Quiesce Guarantee control bit of TS2OS 0: Set 0 to TS2OS (default) 1: Set 1 to TS2OS	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	UI_ENTER_TXL0S	TxL0s transition control 0: Do not perf orm ASPM L0s transition (default) 1: Execute ASPM L0s transition when internal conditions are satisfied	R/W
17	UI_ENTER_L2	RC mode L2 transition control Set to 1 when transitioning to L2 state in RC mode. When transitioning to the L2 state, the PCIe core must be reset by controlling the Reset register. When returning, this bit must be cleared to 0 after releasing the reset.	R/W
18	APMASN	Auto PM_Active_State_Nak PM_ActiveState_Nak Message Transmission Mode for ASPM L1 Rejection Set to 1 if you want to reject ASPM L1 in RC. This Bit is automatically cleared when PM_ActiveState_Nak is automatically sent. Note: Autosend only once.	R/W
19	UI_RC_REJECT_ASPML1	ASPM L1 transition rejection control 0: Accept ASPM L1 transition request from EP device (default) 1: Reject ASPM L1 transition request from EP device	R/W

Bit	Symbol	Function	R/W
20	RETURN_TO_L0	RC mode L1, L2 state to L0 state control (usually not used) Cleared automatically after confirming PMU_LINKSTATE[0] =1. 0: Normal operation (default) 1: Start return operation to L0 state when in L1 or L2 state in RC mode	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	UI_ENTER_L1S	L1SubState transition permission setting 0: L1SubState transition disabled (default) 1: L1SubState transition permission (setting prohibited)	R/W
27:23	—	These bits are read as 0. The write value should be 0.	R/W
28	BLB_RELAX_ORDERING_EN	Control of RO bit of Request to be sent 0: RO bit of Request TLP to be sent is always 0 (default) 1: A TLP can be sent with the RO bit of the Request TLP to be sent set to 1.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_PCCTRL1 register controls power management and LTSSM (Link Training Sequence State Machine) state transitions of the PCI Express core.

56.3.1.39 PCI_RC_PCSTAT1 : PCIe Core Status 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0408

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BD	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	LTSSM_STATE[6:0]							PMU_LINKSTATE[3:0]				—	—	STATE_VC0_NEGOTIATION_PENDING	DLDS
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x

Bit	Symbol	Function	R/W
0	DLDS	DL_Down status Indicates whether PCIe Core is in DL_Down or DL_Up state 0: DL_Up Status 1: DL_Down Status	R
1	STATE_VC0_NEGOTIATION_PENDING	Flow Control initialization operation monitor If this bit is 1, do not initiate a transaction from the AXI side. Check that this bit is 0 and DL_Down Status (bit[0]) is 0. 0: Indicates that Flow Control initialization is complete 1: Indicates that Flow Control initialization has not completed	R
2	—	The read value is undefined.	R
3	—	This bit is read as 0.	R
7:4	PMU_LINKSTATE[3:0]	L-state monitor of power management control unit 0x4: L1 state 0x8: L2 state	R

Bit	Symbol	Function	R/W
14:8	LTSSM_STATE[6:0]	Indicates the state of the Link Training & Status State Machine in the PCIe Core Link. The following states are indicated by the upper 5 bits [14:10]. 000xxb: Detect 001xxb: Polling 010xxb: Config 01100b: L0 01101b: L1 0111xb: L2 100xxb: Recovery 101xxb: Disable 110xxb: Loopback	R
16:15	—	These bits are read as 0.	R
17	BD	bme_down Indicates that the PCIe core transmitter is in an unusable state.	R
31:18	—	These bits are read as 0.	R

The PCI_RC_PCSTAT1 register controls power management and LTSSM (Link Training Sequence State Machine) state transitions of the PCI Express core.

56.3.1.40 PCI_RC_PCCTRL2 : PCIe Core Control 2 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0410

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	UI_LINK_WIDTH_CHANGE_ENABLE[7:0]								MODE_NODEEMPHASIS[1:0]		MODE_PRESET_ENABLE[4:0]					UI_LINK_WIDTH_CHANGE_REQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	UI_LINK_SPEED_CHANGE[1:0]		—	—	—	UI_LINK_CHANGE_AUTONOMOUS	—	—	—	UI_LINK_SPEED_CHANGE_REQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UI_LINK_SPEED_CHANGE_REQ	Link Speed change request control Setting this bit to 1 requests to change the Link Speed to the speed set in bit [8] UI_LINK_SPEED_CHANGE field. By asserting it in the L0 state, it transitions to the Recovery state and performs negotiation with the peer device. PCIe Core Status 2 Register (Offset: 0x414) bit [28] Set to 0 after confirming that UI_LINK_SPEED_CHANGE_DONE has been asserted.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	UI_LINK_CHANGE_AUTONOMOUS	Link Width/Speed change reason setting 0: reliability reason (change for reliability, direction of bandwidth reduction) 1: autonomous reason (intentional change)	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
9:8	UI_LINK_SPEED_CHANGE[1:0]	Link speed setting Set the Link Speed you want to change. 0 0: 2.5 GT/s 0 1: 5.0 GT/s 1 0: 8.0 GT/s 1 1: 16.0 GT/s (Setting prohibited)	R/W

Bit	Symbol	Function	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	UI_LINK_WIDTH_CHANGE_REQ	Link Width change request control Setting this bit to 1 issues a request to change the Link Width to the configuration set in bit [31:24] UI_LINK_WIDTHCHANGE_ENABLE field. By asserting it in the L0 state, it transitions from the Recovery state to the Configuration state, and performs negotiation with the other device. Set to 0 after confirming that PCIe Core Status 2 Register (Offset: 0x414) bit [29] UI_LINK_WIDTH_CHANGE_DONE is asserted. Available for Single Link structure. Prohibited to set for Multi Link structure or ch1.	R/W
21:17	MODE_PRESET_ENABLE[4:0]	Reduced Swing mode setting for Gen3 operation [0] 8GT/s Preset P0, 0: disable / 1: enable (default) [1] 8GT/s Preset P2, 0: disable / 1: enable (default) [2] 8GT/s Preset P7, 0: disable / 1: enable (default) [3] 8GT/s Preset P8, 0: disable / 1: enable (default) [4] 8GT/s Preset P10, 0: disable / 1: enable (default)	R/W
23:22	MODE_NODEEMPHASIS[1:0]	No de-emphasis mode setting for Gen1/Gen2 operation [0] Gen1 operation, 0: Normal operation mode (default) / 1: No de-emphasis mode [1] Gen2 operation, 0: Normal operation mode (default) / 1: No de-emphasis mode	R/W
31:24	UI_LINK_WIDTH_CHANGE_ENABLE[7:0]	Link Width setting to change Assert UI_LINK_WIDTH_CHANGE_REQ and set Lane to 1 to operate when Link Width change request is issued. The lower bit (bit [24]) drives Lane 0 and the most significant bit (bit [31]) drives Lane 7. In this module, bit [25:24] is valid for Single Link structure, and bit [24] is valid for Multi Link structure or ch1.	R/W

The PCI_RC_PCCTRL2 register controls the link speed/width change in the PCI Express core.

56.3.1.41 PCI_RC_PCSTAT2 : PCIe Core Status 2 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0414

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	UI_LINK_WIDTH_CHANGE_DONE	UI_LINK_SPEED_CHANGE_DONE	—	—	—	STATE_UPCONFIGURE_CAPABLE	—	STATE_NEGOTIATED_LANE_END[2:0]			—	STATE_NEGOTIATED_LANE_START[2:0]		
Value after reset:	0	0	x	x	0	0	0	x	0	x	x	x	0	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	STATE_RECEIVER_DETECTED[7:0]								STATE_DATA_RATE_IDENTIFIER_RECEIVED[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	STATE_DATA_RATE_IDENTIFIER_RECEIVED[7:0]	Link Speed display supported by the opposite device Displays the TS-OS Data Rate Identifier feed received from the peer device. Bit 0: Reserved Bit 1: 2.5 GT/s Data Rate Supported. Must be set to 1. Bit 2: 5.0 GT/s Data Rate Supported. Must be set to 1 if Bit 3 is 1. Bit 3: 8.0 GT/s Data Rate Supported. Bits 4 to 7: Reserved	R
15:8	STATE_RECEIVER_DETECTED[7:0]	Connection status display with other device Receiver Detection results are displayed. [0] Detect opposite device on Lane 0 [1] Detect peer device on Lane 1 (only for x2) [2] to [7] reserved	R

Bit	Symbol	Function	R/W
18:16	STATE_NEGOTIATED_LANE_START[2:0]	Displays the position of Lane Number 0 after Link Negotiation with the opposite device during n-lane operation. Used to check the state of the current working lane before changing the Link Width. Available for Single Link structure. Prohibited to set for Multi Link structure or ch1. 0 0 0: Lane 0 is Lane Number 0 0 0 1: Lane 1 is Lane Number 0 Others: Reserved	R
19	—	This bit is read as 0.	R
22:20	STATE_NEGOTIATED_LANE_END[2:0]	Displays the position of Lane Number (n-1) (meaning Lane 1 when n=2) after Link Negotiation with the opposite device during n-lane operation. Used to check the state of the current working lane before changing the Link Width. Available for Single Link structure. Prohibited to set for Multi Link structure or ch1. 0 0 0: Lane 0 is Lane Number (n-1) 0 0 1: Lane 1 is Lane Number (n-1) Others: Reserved	R
23	—	This bit is read as 0.	R
24	STATE_UPCONFIGURE_CAPABLE	Upconfigure Capable bit display of opposite device Indicates whether the peer device supports changing the direction of widening the Link Width. If this bit is 0, changing the Link Width will not restore the original Link Width.	R
27:25	—	These bits are read as 0.	R
28	UI_LINK_SPEED_CHANGE_DONE	Link Speed Change operation complete display PCIe Core Status 2 Registers (Offset: 0x410) bit [0] Notifies completion of Speed Change (1) by setting UI_LINK_SPEED_CHANGE_REQ. It is set to 0 by setting UI_LINK_SPEED_CHANGE_REQ to 0.	R
29	UI_LINK_WIDTH_CHANGE_DONE	Link Width Change operation complete display Notifies completion of Width Change (1) by setting PCIe Core Status 2 Register (Offset: 0x410) bit [16] UI_LINK_WIDTH_CHANGE_REQ. It is set to 0 by setting UI_LINK_WIDTH_CHANGE_REQ to 0. Available for Single Link structure. Prohibited to set for Multi Link structure or ch1.	R
31:30	—	These bits are read as 0.	R

The PCI_RC_PCSTAT2 register indicates the status of the link speed/width change in the PCI Express core.

56.3.1.42 PCI_RC_PCSTAT5 : PCIe Core Status 5 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x042C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ORT_TRANSACTION_PENDING[1:0]	
Value after reset:	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ORT_TRANSACTION_PENDING[1:0]	Indicates whether or not there are Outstanding Requests (a state in which all Completions corresponding to Non-Posted Requests sent from the AXI side have not been received). Check with this bit that there is no Outstanding Request before requesting/permitting a transition to TxL0s/L1/L2. (bit 1: function1, bit 0: function0) 0: State without Outstanding Request 1: State with Outstanding Request	R
7:2	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
9:8	—	The read values are undefined.	R
31:10	—	These bits are read as 0.	R

The PCI_RC_PCSTAT5 register indicates the status in the PCI Express core.

56.3.1.43 PCI_RC_DMAINTVEC0 : DMA Interrupt Vector 0 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x04D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DMA_CH3_MSI_EN	DMA_CH3_V[4:0]					—	—	DMA_CH2_MSI_EN	DMA_CH2_V[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	DMA_CH1_MSI_EN	DMA_CH1_V[4:0]					—	—	DMA_CH0_MSI_EN	DMA_CH0_V[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DMA_CH0_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch0	R/W
5	DMA_CH0_MSI_EN	DMA Ch0 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DMA_CH1_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch1	R/W
13	DMA_CH1_MSI_EN	DMA Ch1 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
20:16	DMA_CH2_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch2	R/W
21	DMA_CH2_MSI_EN	DMA Ch2 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
28:24	DMA_CH3_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch3	R/W
29	DMA_CH3_MSI_EN	DMA Ch3 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: DMA_CHx_vec should be fixed to 0x0 for Root Complex Mode.

The PCI_RC_DMAINTVEC0 register specifies the interrupt vectors for interrupt notification from AXI to PCIe (MSI) during DMA transfer.

56.3.1.44 PCI_RC_DMAINTVEC1 : DMA Interrupt Vector 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x04D4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DMA_CH7_MSI_EN	DMA_CH7_V[4:0]					—	—	DMA_CH6_MSI_EN	DMA_CH6_V[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	DMA_CH5_MSI_EN	DMA_CH5_V[4:0]					—	—	DMA_CH4_MSI_EN	DMA_CH4_V[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DMA_CH4_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch4	R/W
5	DMA_CH4_MSI_EN	DMA Ch4 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DMA_CH5_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch5	R/W
13	DMA_CH5_MSI_EN	DMA Ch5 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
20:16	DMA_CH6_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch6	R/W
21	DMA_CH6_MSI_EN	DMA Ch6 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
28:24	DMA_CH7_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch7	R/W
29	DMA_CH7_MSI_EN	DMA Ch7 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Note 1. DMA_CHx_vec should be fixed to 0x0 for Root Complex Mode.

The PCI_RC_DMAINTVEC1 register specifies the interrupt vectors for interrupt notification from AXI to PCIe (MSI) during DMA transfer.

56.3.1.45 PCI_RC_MSIRCVEn : MSI Receive Enable n Register (n = 0 to 15)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0600 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	E	Enable Set enable/disable of the MSI Receive (Message Data discrimination) register group, which sets 1 register for + 16 bytes from Offset of this register. 0: disabled 1: Enabled	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_MSIRCVEn register enables control of MSI receive (Message Data discrimination) register group with the following as one set.

- MSI Receive Enable n Register (Offset 0x0600 + 0x10 × n)
- MSI Receive message Data n Register (Offset 0x0604 + 0x10 × n)
- MSI Receive Mask n Register (Offset 0x0608 + 0x10 × n)
- MSI Receive Status n Register (Offset 0x060C + 0x10 × n)

56.3.1.46 PCI_RC_MSIRCVMSGDATAn : MSI Receive Message Data n Register (n = 0 to 15)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0604 + 0x10 × n

Bit position:	31	5	4	3	2	1	0
Bit field:			0	0	0	0	0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Message Data Set Message Data for MSI reception judgment. The lower 5 bits are f ixed to 0.	R/W

The PCI_RC_MSIRCVMSGDATAn register is a Message data setting register for MSI reception judgment of registers enabled by MSI Receive Enable n Register.

56.3.1.47 PCI_RC_MSIRCVMSKn : MSI Receive Mask n Register (n = 0 to 15)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0608 + 0x10 × n

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	MSI Mask Set the interrupt Mask by each bit of MSI Status. 0: Interrupt enabled 1: Interrupt Mask	R/W

The PCI_RC_MSIRCVMSKn register masks control of interrupt by the MSI reception status of the register group enabled by MSI Receive Enable n Register.

56.3.1.48 PCI_RC_MSIRCVSTATn : MSI Receive Status n Register (n = 0 to 15)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x060C + 0x10 × n

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	MSI Status MSI receive Status. The bit corresponding to Data [4:0] is set when Message Data [31:5] and MSI reception Data [31:5] match. (When Data [4:0] = 0x00, MSI Status n [0] is set. When Data [4:0] = 0x01, MSI Status n [1] is set.)	R/W

56.3.1.49 PCI_RC_DMACTRL : DMAC Control Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0800

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	D_PMRS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	D_PMRS[2:0]	DMAC PCIe Max Read Request Size Set the upper limit of Read requests issued from the DMAC to PCIe. 0 0 0: 128 bytes (default) 0 0 1: 256 bytes 0 1 0: 512 bytes (not supported) 0 1 1: 1024 bytes (not supported) 1 0 0: 2048 bytes (not supported) 1 0 1: 4096 bytes (not supported) others: Reserved (prohibited)	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_DMACTRL register sets the maximum size of read requests which can be issued to the PCIe core as a DMAC function. Use the initial setting (128 bytes). This setting is common to all DMA channels.

56.3.1.50 PCI_RC_DMAINTE : DMAC Interrupt Enable Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0808

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CH7_ERR_EN	CH7_QUE_EMP_EN	CH7_STOP_EN	CH7_END_EN	CH6_ERR_EN	CH6_QUE_EMP_EN	CH6_STOP_EN	CH6_END_EN	CH5_ERR_EN	CH5_QUE_EMP_EN	CH5_STOP_EN	CH5_END_EN	CH4_ERR_EN	CH4_QUE_EMP_EN	CH4_STOP_EN	CH4_END_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CH3_ERR_EN	CH3_QUE_EMP_EN	CH3_STOP_EN	CH3_END_EN	CH2_ERR_EN	CH2_QUE_EMP_EN	CH2_STOP_EN	CH2_END_EN	CH1_ERR_EN	CH1_QUE_EMP_EN	CH1_STOP_EN	CH1_END_EN	CH0_ERR_EN	CH0_QUE_EMP_EN	CH0_STOP_EN	CH0_END_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CH0_END_EN	CH0 Completion Interrupt Enable 0: Disable 1: Enable	R/W
1	CH0_STOP_EN	CH0 Stop Interrupt Enable 0: Disable 1: Enable	R/W
2	CH0_QUE_EMP_EN	CH0 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
3	CH0_ERR_EN	CH0 Error Interrupt Enable 0: Disable 1: Enable	R/W
4	CH1_END_EN	CH1 Completion Interrupt Enable 0: Disable 1: Enable	R/W
5	CH1_STOP_EN	CH1 Stop Interrupt Enable 0: Disable 1: Enable	R/W
6	CH1_QUE_EMP_EN	CH1 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
7	CH1_ERR_EN	CH1 Error Interrupt Enable 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
8	CH2_END_EN	CH2 Completion Interrupt Enable 0: Disable 1: Enable	R/W
9	CH2_STOP_EN	CH2 Stop Interrupt Enable 0: Disable 1: Enable	R/W
10	CH2_QUE_EMP_EN	CH2 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
11	CH2_ERR_EN	CH2 Error Interrupt Enable 0: Disable 1: Enable	R/W
12	CH3_END_EN	CH3 Completion Interrupt Enable 0: Disable 1: Enable	R/W
13	CH3_STOP_EN	CH3 Stop Interrupt Enable 0: Disable 1: Enable	R/W
14	CH3_QUE_EMP_EN	CH3 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
15	CH3_ERR_EN	CH3 Error Interrupt Enable 0: Disable 1: Enable	R/W
16	CH4_END_EN	CH4 Completion Interrupt Enable 0: Disable 1: Enable	R/W
17	CH4_STOP_EN	CH4 Stop Interrupt Enable 0: Disable 1: Enable	R/W
18	CH4_QUE_EMP_EN	CH4 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
19	CH4_ERR_EN	CH4 Error Interrupt Enable 0: Disable 1: Enable	R/W
20	CH5_END_EN	CH5 Completion Interrupt Enable 0: Disable 1: Enable	R/W
21	CH5_STOP_EN	CH5 Stop Interrupt Enable 0: Disable 1: Enable	R/W
22	CH5_QUE_EMP_EN	CH5 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
23	CH5_ERR_EN	CH5 Error Interrupt Enable 0: Disable 1: Enable	R/W
24	CH6_END_EN	CH6 Completion Interrupt Enable 0: Disable 1: Enable	R/W
25	CH6_STOP_EN	CH6 Stop Interrupt Enable 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
26	CH6_QUE_EMP_EN	CH6 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
27	CH6_ERR_EN	CH6 Error Interrupt Enable 0: Disable 1: Enable	R/W
28	CH7_END_EN	CH7 Completion Interrupt Enable 0: Disable 1: Enable	R/W
29	CH7_STOP_EN	CH7 Stop Interrupt Enable 0: Disable 1: Enable	R/W
30	CH7_QUE_EMP_EN	CH7 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
31	CH7_ERR_EN	CH7 Error Interrupt Enable 0: Disable 1: Enable	R/W

The PCI_RC_DMAINTE register enables interrupts from the individual DMA channels.

56.3.1.51 PCI_RC_DMAINTS : DMAC Interrupt Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x080C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CH7_ERR	CH7_QUE_EMP	CH7_STOP	CH7_END	CH6_ERR	CH6_QUE_EMP	CH6_STOP	CH6_END	CH5_ERR	CH5_QUE_EMP	CH5_STOP	CH5_END	CH4_ERR	CH4_QUE_EMP	CH4_STOP	CH4_END
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CH3_ERR	CH3_QUE_EMP	CH3_STOP	CH3_END	CH2_ERR	CH2_QUE_EMP	CH2_STOP	CH2_END	CH1_ERR	CH1_QUE_EMP	CH1_STOP	CH1_END	CH0_ERR	CH0_QUE_EMP	CH0_STOP	CH0_END
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CH0_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
1	CH0_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
2	CH0_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
3	CH0_ERR	Set when an error occurs during DMA transfer.	R/W
4	CH1_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W

Bit	Symbol	Function	R/W
5	CH1_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
6	CH1_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
7	CH1_ERR	Set when an error occurs during DMA transfer.	R/W
8	CH2_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
9	CH2_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
10	CH2_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
11	CH2_ERR	Set when an error occurs during DMA transfer.	R/W
12	CH3_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
13	CH3_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
14	CH3_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
15	CH3_ERR	Set when an error occurs during DMA transfer.	R/W
16	CH4_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
17	CH4_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
18	CH4_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
19	CH4_ERR	Set when an error occurs during DMA transfer.	R/W
20	CH5_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
21	CH5_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
22	CH5_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
23	CH5_ERR	Set when an error occurs during DMA transfer.	R/W
24	CH6_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W

Bit	Symbol	Function	R/W
25	CH6_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
26	CH6_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
27	CH6_ERR	Set when an error occurs during DMA transfer.	R/W
28	CH7_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
29	CH7_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
30	CH7_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
31	CH7_ERR	Set when an error occurs during DMA transfer.	R/W

The PCI_RC_DMAINTS register indicates the state of interrupts from the individual DMA channels.

56.3.1.52 PCI_RC_DMACHCTLn : DMAC Channel Control Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0900 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	QUE_CLR	—	—	—	—	—	—	QUE_EN	RDMA_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RDMA_EN	Register method DMA transfer Enable Setting this bit to 1 starts DMA transfer (register method) and performs the transfer set by RDMA_SIZE. This bit is automatically cleared to 0 when DMA transfer is completed or an error is detected and DMA ends. It is also possible to stop DMA by writing 0. However, DMA will stop after completing the currently executing request (both PCIe and AXI). 0: Stop register-based DMA transfer 1: Register method DMA transfer start	R/W
1	QUE_EN	QUE Enable Setting this bit to 1 enables the descriptor list registered in the descriptor queue and starts DMA transfer (descriptor method). It is automatically cleared to 0 when DMA stops (both normal and abnormal). It is also possible to stop DMA by writing 0. However, DMA will stop after finishing the currently executing request (both PCIe and AXI).	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	QUE_CLR	QUE Clear Writing 1 clears QUE. All descriptor lists (waiting for execution and lists currently being executed) registered in QUE are cleared. Do not clear during DMA execution. The read value is always 0. 1Write is prohibited at the same time as QUE_EN Set	R/W

Bit	Symbol	Function	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_DMACHCTLn register sets the control method for each DMA channel. Set either register type or descriptor type.

- Setting QUE_EN = 1 and QUE_CLR = 1 is prohibited during register-type DMA transfer (TDMA_EN = 1).
- Setting RDMA_EN = 1 is prohibited during descriptor-type DMA transfer (QUE_EN = 1).

56.3.1.53 PCI_RC_DPSADRLn : Descriptor Start Address (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0908 + 0x80 × n

Bit position:	31																								6	5					0
Bit field:	QUE_ENTRY_DSA3106[25:0]																									QUE_ENTRY_DSA0500[5:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	QUE_ENTRY_DSA0500[5:0]	Descriptor list queue registration register. This area will be the DSA. Sets the lower 32 bits of the address where the first descriptor is stored. (16 byte align: Lower 6 bit is fixed to 0.)	R
31:6	QUE_ENTRY_DSA3106[25:0]	Descriptor list queue registration register. This area will be the DSA. Sets the lower 32 bits of the address where the first descriptor is stored.	R/W

Note: When these bits are read, the following contents will be the read value depending on the state of DMA.
DMA transfer in progress: The descriptor list in progress
DMA suspended (when QUE_EN is automatically cleared): Last executed list
DMA suspended (when QUE_EN S/W is cleared): Suspended (executing) list

The PCI_RC_DPSADRLn register sets the descriptor queue list.

The setting value is registered as the lower 32 bits of DSA (DMA Start Address) in the queue list.

56.3.1.54 PCI_RC_DPSADRUn : Descriptor Start Address (Upper) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x090C + 0x80 × n

Bit position:	31																														0
Bit field:	QUE_ENTRY_DSA6332[31:0]																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	QUE_ENTRY_DSA6332[31:0]	Descriptor list queue registration register. This area will be the DSA. Set the upper 32 bits of the address where the first descriptor is stored.	R/W

Note: When these bits are read, the following contents will be the read value depending on the state of DMA.
DMA transfer in progress: The descriptor list in progress
DMA suspended (when QUE_EN is automatically cleared): Last executed list
DMA suspended (when QUE_EN S/W is cleared): Suspended (executing) list

The PCI_RC_DPSADRUn register sets the descriptor queue list.

The setting value is registered as the upper 32 bits of DSA (DMA Start Address) in the queue list.

56.3.1.55 PCI_RC_QUEEn : QUE Entry Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0910 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	QUE_R[5:0]						QUE_ENTRY_EI	QUE_ENTRY_LS	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	QUE_ENTRY_LABEL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	QUE_ENTRY_LABEL[15:0]	There is no particular rule on how to set the labels in the list. You can set the value freely.	R/W
23:16	—	These bits are read as 0. The write value should be 0.	R/W
24	QUE_ENTRY_LS	Indicates whether to stop the DMA when processing of the descriptor list is complete. 0: do not stop 1: stop	R/W
25	QUE_ENTRY_EI	Indicates whether an interrupt (Interrupt Status CHx_END) is sent when the processing of the descriptor list is completed. 0: Do not signal interrupts 1: signal an interrupt	R/W
31:26	QUE_R[5:0]	QUE_Registration Enqueued by writing to [31:24]. The read value is always 0.	R/W

Note: When these bits are read, the following contents will be the read value depending on the state of DMA.
DMA transfer in progress: The descriptor list in progress
DMA suspended (when QUE_EN is automatically cleared): Last executed list
DMA suspended (when QUE_EN S/W is cleared): Suspended (executing) list

The PCI_RC_QUEEn register sets the descriptor queue list.

The setting value is registered as EI (End Interrupt), LS (List Stop), and LABEL of the queue list. Write to [31:24] to register to the queue.

56.3.1.56 PCI_RC_DMADPCTLn : DMA Descriptor Control (Descriptor 0x00) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0920 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DSCFM[3:0]				—	WBD	LE	LV	D	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	STS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	STS[15:0]	Shows the value of the STS field in the running descriptor table.	R

Bit	Symbol	Function	R/W
22:16	—	These bits are read as 0.	R
23	D	Shows the value of the D field in the running descriptor table.	R
24	LV	Shows the value of the LV field in the running descriptor table.	R
25	LE	Shows the value of the LE field in the running descriptor table.	R
26	WBD	Shows the value of the WBD field in the running descriptor table.	R
27	—	This bit is read as 0.	R
31:28	DSCFM[3:0]	Shows the value of the DSCFM field in the running descriptor table.	R

The PCI_RC_DMADPCTLn register indicates the field value at offset 0x00 in the descriptor table. Only effective when the descriptor-type DMA transfer is selected (the value read has no meaning in the case of register-type transfer).

56.3.1.57 PCI_RC_DMATCTLn : DMA Transaction Control (Descriptor 0x04) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0924 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	CCH_L[3:0]				CCH_D[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	DMA_TC[2:0]			—	—	DMA_ATB[1:0]		—	DMA_FUNC[2:0]			—	—	—	DMA_DIR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DMA_DIR	Sets the DMA transfer direction. 0: PCIe to AXI 1: AXI to PCIe	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
6:4	DMA_FUNC[2:0]	Specify the function number of the request issued to PCIe.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
9:8	DMA_ATB[1:0]	Attributes to publish to PCIe. bit [1]: Relaxed Ordering (Unsupported function: 0 fixed) bit [0]: No Snoop (0 recommended)	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
14:12	DMA_TC[2:0]	Traffic class to publish to PCIe. Specifies the traffic class value for requests issued to PCIe. Note: This IP does not support Virtual Channel, so please use it with 000b fixed.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
19:16	CCH_D[3:0]	Indicates the value of A*CACHE[3:0] to be issued to AXI. CCH_D is output when issuing an AXI request other than the output condition of CCH_L. The recommended value is 0x1 when DIR = 0 (PCIe→AXI) and 0x0 when DIR = 1 (AXI→PCIe). [3] 1: write allocatable / 0: not write allocatable [2] 1: Read assignable / 0: Read assignable [1] 1: cacheable / 0: non-cacheable [0] 1: Bufferable / 0: Not Bufferable	R/W

Bit	Symbol	Function	R/W
23:20	CCH_L[3:0]	Indicates the value of A*CACHE[3:0] to be issued to AXI. CCH_L is output when issuing an AXI request containing the last byte in a transfer indicated by SIZE. Recommended value is 0x0. [3] 1: write allocatable / 0: not write allocatable [2] 1: Read assignable / 0: Read assignable [1] 1: cacheable / 0: non-cacheable [0] 1: Bufferable / 0: Not Bufferable	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_DMATCTLn register controls DMA transfer to the AXI and PCIe.

56.3.1.58 PCI_RC_DMASIZEn : DMA Size (Descriptor 0x08) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0928 + 0x80 × n

Bit position:	31																										3	2		0
Bit field:	DMA_SIZE3104[27:0]																												DMA_SIZE 0300[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DMA_SIZE0300[3:0]	Sets the number of DMA transfer bytes. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R
31:4	DMA_SIZE3104[27:0]	Sets the number of DMA transfer bytes. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R/W

Note: When executing descriptor method DMA, writing is prohibited and the read value indicates the values of the CCH_L, CCH_D, TC, ATB, and DIR fields of the descriptor table being executed.

The PCI_RC_DMASIZEn register sets the number of bytes for DMA transfer. Reflects the field at offset 0x0C in the descriptor table.

56.3.1.59 PCI_RC_DMASLAn : DMA Source Lower Address (Descriptor 0x10) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0930 + 0x80 × n

Bit position:	31																										3	2		0
Bit field:	DMA_S_ADDR3104[27:0]																												DMA_S_ADDR0300[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DMA_S_ADDR0300[3:0]	Set the lower 32 bits of the transfer source start address for DMA transfer. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R
31:4	DMA_S_ADDR3104[27:0]	Set the lower 32 bits of the transfer source start address for DMA transfer. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R/W

Note: If the Source Address indicates the PCIe space (DIR = 0), the PCIe request address is set in combination with the DMA PCIe Upper Address (Descriptor 0x10). When descriptor-type DMA transfer is in process, writing to these bits is prohibited and the value read indicates the value of the SA field in the descriptor table being executed.

The PCI_RC_DMASLAn register sets the lower 32 bits of the source start address for DMA transfer. It is reflected to the offset 0x10 field of the descriptor table.

56.3.1.60 PCI_RC_DMASUAn : DMA Source Upper Address (Descriptor 0x14) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0934 + 0x80 × n

Bit position:	31																												0
Bit field:	DMA_S_ADDR6332[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DMA_S_ADDR6332[31:0]	Set the upper 32 bits of the transfer source start address for MA transfer.	R/W

Note: If the Source Address indicates the PCIe space (DIR = 0), the PCIe request address is set in combination with the DMA PCIe Upper Address (Descriptor 0x10). When descriptor-type DMA transfer is in process, writing to these bits is prohibited and the value read indicates the value of the SA field in the descriptor table being executed.

The PCI_RC_DMASUAn register sets the upper 32 bits of the source start address for DMA transfer. It is reflected to the offset 0x14 field of the descriptor table.

56.3.1.61 PCI_RC_DMADESTLAn : DMA Destination Lower Address (Descriptor 0x18) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0938 + 0x80 × n

Bit position:	31																										3	2	0
Bit field:	DMA_D_ADDR3104[27:0]																										DMA_D_ADDR0300[3:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DMA_D_ADDR0300[3:0]	Set the lower 32 bits of the transfer destination start address for DMA transfer. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R
31:4	DMA_D_ADDR3104[27:0]	Set the lower 32 bits of the transfer destination start address for DMA transfer. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R/W

Note: If the Destination Address indicates the PCIe space (DIR = 1), the PCIe request address is set in combination with the DMA PCIe Upper Address (Descriptor 0x10). When descriptor-type DMA transfer is in process, writing to these bits is prohibited and the value read indicates the value of the DA field in the descriptor table being executed.

The PCI_RC_DMADESTLAn register sets the lower 32 bits of the source start address for DMA transfer. It is reflected to the offset 0x18 field of the descriptor table.

56.3.1.62 PCI_RC_DMADESTUAn : DMA Destination Upper Address (Descriptor 0x1C) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x093C + 0x80 × n

Bit position:	31																												0
Bit field:	DMA_D_ADDR6332[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DMA_D_ADDR6332[31:0]	Set the upper 32 bits of the transfer destination start address for DMA transfer.	R/W

Note: If the Destination Address indicates the PCIe space (DIR = 1), the PCIe request address is set in combination with the DMA PCIe Upper Address (Descriptor 0x10). When descriptor-type DMA transfer is in process, writing to these bits is prohibited and the value read indicates the value of the DA field in the descriptor table being executed.

The PCI_RC_DMADESTUAn register sets the upper 32 bits of the source start address for DMA transfer. It is reflected to the offset 0x1C field of the descriptor table.

56.3.1.63 PCI_RC_DMADPLLPn : DMA Descriptor Lower Link Pointer (Descriptor 0x20) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0940 + 0x80 × n

Bit position:	31																													0
Bit field:	DMA_LP3100[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DMA_LP3100[31:0]	Indicates the value of the LP field of the running descriptor table.	R

The PCI_RC_DMADPLLPn register indicates the field value at offset 0x20 in the descriptor table. Only effective when the descriptor-type DMA transfer is selected (the value read has no meaning in the case of register-type transfer).

56.3.1.64 PCI_RC_DMADPULPn : DMA Descriptor Upper Link Pointer (Descriptor 0x24) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0944 + 0x80 × n

Bit position:	31																													0
Bit field:	DMA_LP6332[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DMA_LP6332[31:0]	Indicates the value of the LP field of the running descriptor table.	R

The PCI_RC_DMADPULPn register indicates the field value at offset 0x24 in the descriptor table. Only effective when the descriptor-type DMA transfer is selected (the value read has no meaning in the case of register-type transfer).

56.3.1.65 PCI_RC_DMARESTSIZEn : DMA Rest Size Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0950 + 0x80 × n

Bit position:	31																													0
Bit field:																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	DMA_REST_SIZE Displays the number of bytes for which DMA transfer has not been completed. (Register/descriptor method common)	R

The PCI_RC_DMARESTSIZEn register indicates the number of bytes for which DMA transfer has not yet been completed.

56.3.1.66 PCI_RC_AREQALn : AXI Request Address (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0960 + 0x80 × n

Bit position:	31																												0
Bit field:	AXI_REQ_ADDR3100[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	AXI_REQ_ADDR3100[31:0]	Displays the lower 32 bits of the address of the currently completed AXI transfer or the transfer that just completed. (Register/descriptor method common)	R

The PCI_RC_AREQALn register indicates the lower 32 bits of the address of the current or most recently completed AXI transfer.

56.3.1.67 PCI_RC_AREQAUn : AXI Request Address (Upper) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0964 + 0x80 × n

Bit position:	31																												0
Bit field:	AXI_REQ_ADDR6332[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	AXI_REQ_ADDR6332[31:0]	Displays the upper 32 bits of the address of the currently completed AXI transfer or the transfer that just completed. (Register/descriptor method common)	R

The PCI_RC_AREQAUn register indicates the upper 32 bits of the address of the current or most recently completed AXI transfer.

56.3.1.68 PCI_RC_PREQALn : PCIe Request Address (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0968 + 0x80 × n

Bit position:	31																												0
Bit field:	PCIE_REQ_ADDR3100[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PCIE_REQ_ADDR3100[31:0]	Displays the lower 32 bits of the address of the currently completed PCIe transfer or the transfer completed immediately before. (Register/descriptor method common)	R

The PCI_RC_PREQALn register indicates the lower 32 bits of the address during PCIe transfer that is in progress or for the most recently completed transfer.

56.3.1.69 PCI_RC_PREQAUn : PCIe Request Address (Upper) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x096C + 0x80 × n

Bit position:	31	0
Bit field:	PCIE_REQ_ADDR6332[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	PCIE_REQ_ADDR6332[31:0]	Displays the upper 32 bits of the address of the currently completed PCIe transfer or the transfer completed immediately before. (Register/descriptor method common)	R

The PCI_RC_PREQAUn register indicates the upper 32 bits of the address during PCIe transfer that is in progress or for the most recently completed transfer.

56.3.1.70 PCI_RC_QUESTAn : QUE Status Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0970 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	GO_LIST	LIST_NUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	LIST_NUM[3:0]	Displays the number of descriptor lists loaded on QUE (not including the list currently being executed). New registration to QUE (write access to QUE Entry) when this register indicates 0x8 is invalid (discarded).	R
4	GO_LIST	Shows whether there is a running descriptor list. 0: none 1: Yes	R
31:5	—	These bits are read as 0.	R

The PCI_RC_QUESTAn register indicates the state of the descriptor queue.

56.3.1.71 PCI_RC_DMACESTAn : DMAC Error Status Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0978 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CFG_BM_DIS_EP	BME_SUP	BME_DOWN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	MOR_CD_PERR	MOR_CH_PERR	MOR_EP_ERR	MOR_STATUS[2:0]			—	—	—	—	—	—	AXI_RESP[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

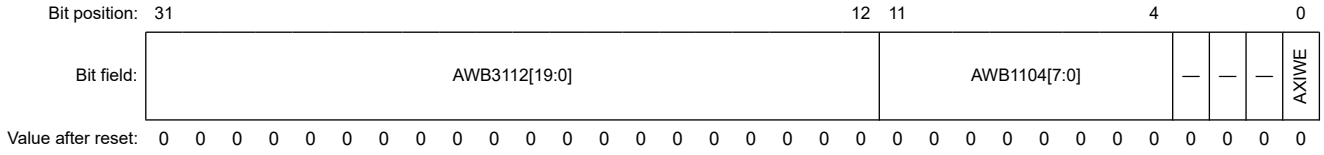
Bit	Symbol	Function	R/W
1:0	AXI_RESP[1:0]	Displays the slave response during AXI Master transactions. It updates when CHx_ERR is set and retains the value until the bit is cleared. 0 0: Initial value 0 1: Reserved	R
7:2	—	These bits are read as 0.	R
10:8	MOR_STATUS[2:0]	Indicates the value when MOR_STATUS is other than 000b (Success) as a set factor of CHx_ERR. Holds value until CHx_ERR is cleared. 0 0 0: initial value 0 0 1: Unsupported Request 0 1 0: CRS 0 1 1: Completion Timeout 1 0 0: Completer Abort 1 0 1: Unexpected Completion 1 1 0: Reserved 1 1 1: Mismatched Length (Length Overrun)	R
11	MOR_EP_ERR	It is set when it is Poisoned Completion as a set factor of CHx_ERR. Holds value until CHx_ERR is cleared.	R
12	MOR_CH_PERR	It is set when MOR_CH_PERR is detected as a set factor for CHx_ERR. Holds value until CHx_ERR is cleared.	R
13	MOR_CD_PERR	It is set when MOR_CD_PERR is detected as a set factor for CHx_ERR. Holds value until CHx_ERR is cleared.	R
15:14	—	These bits are read as 0.	R
16	BME_DOWN	It is set when a stop signal from the PCIe core is detected as a cause for setting CHx_ERR. Holds value until CHx_ERR is cleared.	R
17	BME_SUP	It is set when a sleep signal from the PCIe core is detected as a cause for setting CHx_ERR. Holds value until CHx_ERR is cleared. (Valid only in Endpoint mode)	R
18	CFG_BM_DIS_EP	CHx_ERR is set when the macro operates as an End Point and detects the state of Bus Master Enable Off (Configuration Space 0x004 bit [2] = 0). Holds value until CHx_ERR is cleared.	R
31:19	—	These bits are read as 0.	R

The PCI_RC_DMACESTAn register indicates the error status of the DMAC.

56.3.1.72 PCI_RC_AWBASELn : AXI Window Base (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1000 + 0x20 × n



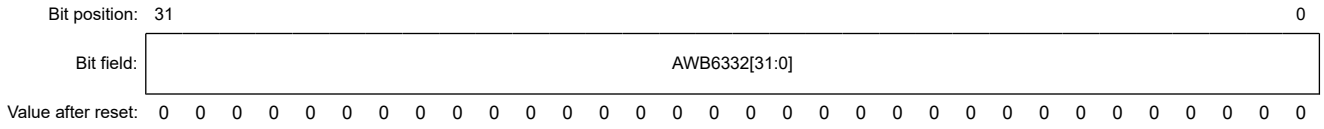
Bit	Symbol	Function	R/W
0	AXIWE	AXI Window Enable Enable setting of AXI Window. 0: Window disabled 1: Window enabled	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
11:4	AWB1104[7:0]	AWBase11:4 fixed 0x00	R
31:12	AWB3112[19:0]	AWBase31:12 Window setting register for address conversion when accessing AXI from PCIe. The configurable area is the 4K boundary.	R/W

The PCI_RC_AWBASELn register is for setting windows for lower address conversion in access from PCIe to AXI. It sets the base address on the PCI. The areas are set in 4-KB boundaries.

56.3.1.73 PCI_RC_AWBASEUn : AXI Window Base (Upper) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1004 + 0x20 × n



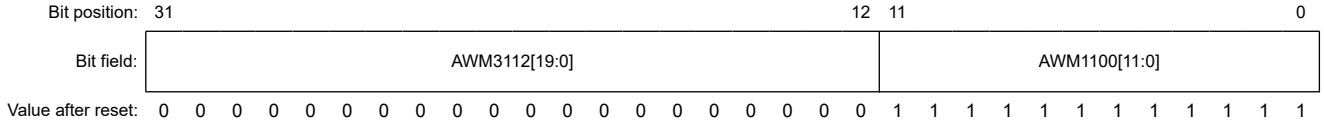
Bit	Symbol	Function	R/W
31:0	AWB6332[31:0]	AWBase63:32 Window setting register for address conversion when accessing AXI from PCIe.	R/W

The PCI_RC_AWBASEUn register is for setting windows for upper address conversion in access from PCIe to AXI.

56.3.1.74 PCI_RC_AWMASKLn : AXI Window Mask (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1008 + 0x20 × n



Bit	Symbol	Function	R/W
11:0	AWM1100[11:0]	AWMask11:0 Fixed 0xFFF	R

Bit	Symbol	Function	R/W
31:12	AWM3112[19:0]	AWMask31:12 Set the window in the area of the set number of bits from the address set in the AWBase register. Set 1 from the lower bit. Therefore, the area that can be set is 4K x 2^N bytes.	R/W

The PCI_RC_AWMASKLn register is for setting windows for lower address conversion in access from PCIe to AXI. The window is set as the area corresponding to the number of set bits from the address set in the AWBase register. The area which can be set is 4K × 2^N bytes.

56.3.1.75 PCI_RC_AWMASKUn : AXI Window Mask Upper Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x100C + 0x20 × n

Bit position:	31	30	0
Bit field:	—	AWM6232[30:0]	
Value after reset:	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
30:0	AWM6232[30:0]	AWMask62:32 Set the window in the area of the set number of bits from the address set in the AWBase register. Set 1 from the lower bit.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_RC_AWMASKUn register is for setting windows for upper address conversion in access from PCIe to AXI.

56.3.1.76 PCI_RC_ADESTLn : AXI Destination (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1010 + 0x20 × n

Bit position:	31	12	11	0
Bit field:	AD3112[19:0]		AD1100[11:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
11:0	AD1100[11:0]	ADest11:0 Fixed to 0x000	R
31:12	AD3112[19:0]	ADest31:12 Sets the window base point in the address space from the AXI side. The configurable area is the 4K boundary.	R/W

The PCI_RC_ADESTLn register is for setting windows for lower address conversion in access from PCIe to AXI. The base address of the window in the address space on the AXI is set. The areas are set in 4-KB boundaries.

56.3.1.77 PCI_RC_ADESTUn : AXI Destination (Upper) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1014 + 0x20 × n

Bit position:	31	0
Bit field:	AD6332[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	AD6332[31:0]	ADest63:32 Sets the window base point in the address space f rom the AXI side.	R/W

The PCI_RC_ADESTUn register is for setting windows for upper address conversion in access from PCIe to AXI.

56.3.1.78 PCI_RC_PWBASELn : PCIe Window Base (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1100 + 0x20 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PWB3112[19:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PWB3112[19:0]				PWB1104[7:0]								—	—	—	PCIWE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PCIWE	PCIe Window Enable Enable setting of PCIe Window. 0: Window disabled 1: Window enabled	R/W
3:1	—	This bit is read as 0. The write value should be 0.	R/W
11:4	PWB1104[7:0]	PWBase11:4 Fixed 0x00	R
31:12	PWB3112[19:0]	PWBase31:12 Sets the base point of the address on the AXI side. The configurable area is the 4K boundary.	R/W

The PCI_RC_PWBASELn register is for setting windows for lower address conversion in access from AXI to PCIe. It sets the base address on the AXI. The areas are set in 4-KB boundaries.

56.3.1.79 PCI_RC_PWBASEUn : PCIe Window Base (Upper) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1104 + 0x20 × n

Bit position:	31																													0
Bit field:	PWB6332[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PWB6332[31:0]	PWBase63:32 Sets the base point of the address on the AXI side. The configurable area is the 4K boundary.	R/W

The PCI_RC_PWBASEUn register is for setting windows for upper address conversion in access from AXI to PCIe.

56.3.1.80 PCI_RC_PWMASKLn : PCIe Window Mask (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1108 + 0x20 × n

Bit position:	31																		12	11										0
Bit field:	PWM3112[19:0]																			PWM1100[11:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
11:0	PWM1100[11:0]	PWMask Fixed 0xFFF	R
31:12	PWM3112[19:0]	PWMask31:12 Set the window in the area of the set number of bits from the address set in the PWBase register. Set 1 from the lower bit.	R/W

The PCI_RC_PWMASKLn register is for setting windows for address conversion in access from AXI to PCIe. The window is set as the area corresponding to the number of set bits from the address set in the PWBase register.

56.3.1.81 PCI_RC_PWMASKUn : PCIe Window Mask (Upper) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x110C + 0x20 × n

Bit position:	31	30																												0
Bit field:	—	PWM6232[30:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
30:0	PWM6232[30:0]	PWMask62:32 Set the window in the area of the set number of bits from the address set in the PWBase register. Set 1 from the lower bit.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_RC_PWMASKUn register is for setting windows for address conversion in access from AXI to PCIe. The window is set as the area corresponding to the number of set bits from the address set in the PWBase register.

56.3.1.82 PCI_RC_PDESTLn : PCIe Destination (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1110 + 0x20 × n

Bit position:	31																		12	11										0
Bit field:	PD3112[19:0]																			PD1100[11:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	PD1100[11:0]	PDest11:0 fixed 0x000	R
31:12	PD3112[19:0]	PDest31:12 Set the window base point in the address space on the PCIe side. The configurable area is the 4k boundary.	R/W

The PCI_RC_PDESTLn register is for setting windows for address conversion in access from AXI to PCIe. The base address of the window in the address space on the AXI is set. The areas are set in 4-KB boundaries.

56.3.1.83 PCI_RC_PDESTUn : PCIe Destination (Upper) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1114 + 0x20 × n

Bit position:	31																													0
Bit field:	PD6332[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PD6332[31:0]	PDest63:32 Set the window base point in the address space on the PCIe side.	R/W

The PCI_RC_PDESTUn register is for setting windows for upper address conversion in access from AXI to PCIe.

56.3.2 PCI Express Configuration Registers (Type1: Root Complex)

56.3.2.1 PCI_RC_VID : Vendor and Device ID Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6000

Bit position:	31																16	15												0
Bit field:	DID[15:0]																VID[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	VID[15:0]	Vendor ID The read value is undefined Used by the manufacturer specified by the Vendor ID to identify the manufactured device. Set a fixed value.	Hwinit
31:16	DID[15:0]	Device ID The read value is undefined Represents the manufacturer of the device. Set a fixed value.	Hwinit

The PCI_RC_VID register indicates the vendor and device ID. The PCI_RC_VID register can be written during initialization.

Table 56.16 Valid reset signal

Reset signal	Device ID	Vendor ID
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—

56.3.2.2 PCI_RC_COM_STA : Command and Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	SSE	—	—	—	—	—	—	—	—	—	CL	IS	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ID	—	SERRE	—	PER	—	—	—	BME	MSE	IOSE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IOSE	IO Space Enable 0 fixed. It does not support access to the I/O space.	R
1	MSE	Memory Space Enable Controls whether to respond to accesses to memory space.	R/W
2	BME	Bus Master Enable Controls whether to operate as a bus master.	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	PER	Parity Error Response	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	SERRE	SERR# Enable Controls the behavior when sending/receiving PoisonedTLP. Note: Error logging to the Detected Parity Error field of the Status register, the Device Status register of PCI Express Capability, and the Uncorrectable Error Status register of Advanced Error Reporting Capability is performed regardless of the setting of this bit.	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	ID	Interrupt Disable If set to 1, the Root Complex is notified of Non-Fatal Errors and Fatal Errors by Message Transaction. Note: Even if this bit is not set, if a bit related to Error Reporting in the Device Control register of PCI Express Capability is set to 1, an error is notified to the Root Complex by Message Transaction.	R/W
18:11	—	These bits are read as 0. The write value should be 0.	R/W
19	IS	Interrupt Status Suppress transmission of Assert_INTx Message. Not used for Root Complex devices.	R
20	CL	Capabilities List This bit is fixed to 1 because all PCI Express devices must implement PCI Express Capability.	R
29:21	—	These bits are read as 0. The write value should be 0.	R/W
30	SSE	Signaled System Error Set to 1 when the SERR Enable bit is 1 and this macro sends an ERR_FATAL or ERR_NONFATAL Message.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_RC_COM_STA register specifies the command and the status.

Table 56.17 Valid reset signal

Reset signal	Signaled System Error	Capabilities List	Interrupt Status	Interrupt Disable	SERR# Enable	Parity Error Response	Bus Master Enable	Memory Space Enable
RST_LOAD_B	—	—	—	—	—	—	—	—
RST_RSM_B	—	—	—	—	—	—	—	—
RST_CFG_B	✓	✓	✓	✓	✓	✓	✓	✓

56.3.2.3 PCI_RC_RID_CC : Revision ID and Class Code Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6008

Bit position:	31																						8	7							0
Bit field:	CC[23:0]																							RID[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RID[7:0]	Revision ID An 8-bit ID used to represent the revision of a specific device specified by Vendor ID and Device ID. Please set a fixed value.	Hwinit
31:8	CC[23:0]	Class Code This is information that indicates the device type and function, and the definition of the value is divided into the following two bytes by the PCI SIG. 31:24 base class: 23:16 sub-class: 15:8 programming interface: Please set a fixed value.	Hwinit

The PCI_RC_RID_CC register indicates the revision ID and the class code.

Table 56.18 Valid reset signal

Reset signal	Class Code	Capabilities List
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—

56.3.2.4 PCI_RC_CL_HT : Cache Line and Header Type Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x600C

Bit position:	31								23								16							7							0
Bit field:	—	—	—	—	—	—	—	—	HT[7:0]								—	—	—	—	—	—	—	—	CLS[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CLS[7:0]	Cache Line Size Implemented as a read/write field for legacy compatibility, but the value set has no effect on this device.	R
15:8	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
23:16	HT[7:0]	Header Type Root Complex: 0x01 fixed.	R
31:24	—	These bits are read as 0.	R

The PCI_RC_CL_HT register indicates the cache line and the header type.

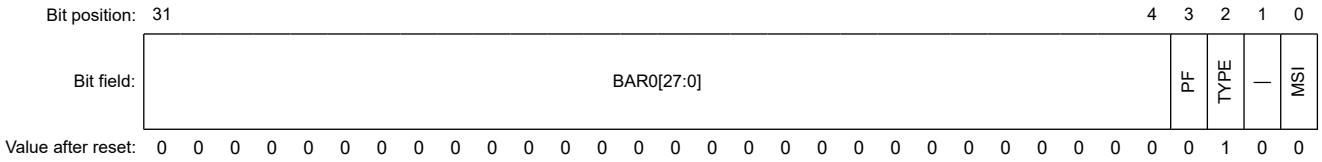
Table 56.19 Valid reset signal

Reset signal	Cache Line Size
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓

56.3.2.5 PCI_RC_BAR0 : Base Address Register 0

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6010



Bit	Symbol	Function	R/W
0	MSI	Memory Space Indicator Fixed to 0 to indicate the memory space.	R
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TYPE	Type 1 fixed to use 64-bit address 0: 32-bit address 1: 64-bit address	R
3	PF	Prefetch 0: Disable 1: Enable	Hwinit
31:4	BAR0[27:0]	Base Address Register 0 Indicates the base address. Depending on the size of the address block required, some lower bits of this field are implemented as Read Only bits fixed at 0. In this macro, the Read Only bits are located in the Base Address 0 Mask register (Offset: 0x0A0).	R/W

The PCI_RC_BAR0 register forms a 64-bit memory space in combination with Base Address Register 1 (BAR1).

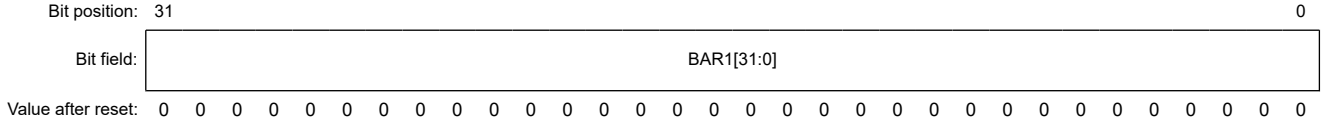
Table 56.20 Valid reset signal

Reset signal	Base Address Register 0	Prefetch
RST_LOAD_B	—	✓
RST_RSM_B	—	—
RST_CFG_B	✓	—

56.3.2.6 PCI_RC_BAR1 : Base Address Register 1

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6014



Bit	Symbol	Function	R/W
31:0	BAR1[31:0]	Base Address Register 1 (64-bit Upper Address) Shows the upper 32 bits of the base address.	R/W

The PCI_RC_BAR1 register forms a 64-bit memory space in combination with Base Address Register 0 (BAR0).

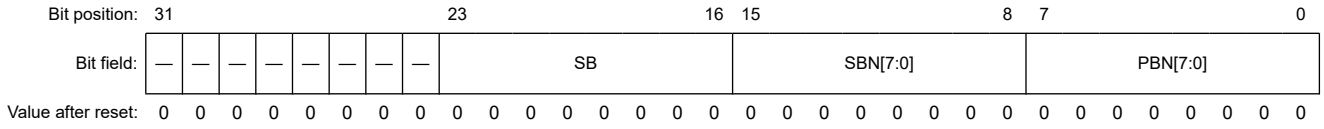
Table 56.21 Valid reset signal

Reset signal	Base Address Register 1
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓

56.3.2.7 PCI_RC_BNR : Bus Number Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6018



Bit	Symbol	Function	R/W
7:0	PBN[7:0]	Primary Bus Number Set the Primary Bus Number.	R/W
15:8	SBN[7:0]	Secondary Bus Number Set the Secondary Bus Number.	R/W
23:16	SB	Subordinate Bus Set the Subordinate Bus Number.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Table 56.22 Valid reset signal

Reset signal	Subordinate Bus Number	Secondary Bus Number	Primary Bus Number
RST_LOAD_B	—	—	—
RST_RSM_B	—	—	—
RST_CFG_B	✓	✓	✓

56.3.2.8 PCI_RC_IOBL_SS : I/O Base/ Limit and Secondary Status

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x601C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DPE	RSE	RMA	RTA	STA	—	—	MDPE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IOL[7:0]								IOB[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	IOB[7:0]	IO Base This is an unused field in this macros. (0x00h fixed)	R/W
15:8	IOL[7:0]	IO Limit This is an unused field in this macros. (0x00 fixed)	R/W
23:16	—	These bits are read as 0. The write value should be 0.	R/W
24	MDPE	Master Data Parity Error Set to 1 when the Parity Error Response bit is set to 1 and the following two conditions occur: 1) Requester (BME) received a Poisoned Completion TLP. 2) Requester (BME) sent Poisoned Write Request TLP. This bit is not set to 1 if the Parity Error Response bit is 0.	R/W
26:25	—	These bits are read as 0. The write value should be 0.	R/W
27	STA	Signaled Target Abort The Completion Status field is set to 1 when sending a Completion of Completer Abort (Posted or Non-Posted Request).	R/W
28	RTA	Received Target Abort Set to 1 when the Completion Status field receives a Completion for Completer Abort.	R/W
29	RMA	Received Master Abort Set to 1 when the Completion Status field receives a Completion of an Unsupported Request.	R/W
30	RSE	Received System Error Set to 1 when an ERR_FATAL or ERR_NONFATAL Message is received.	R/W
31	DPE	Detected Parity Error Set to 1 when a Poisoned TLP is received, regardless of the setting of the Parity Error Response Enable bit in the Bridge Control and Interrupt Register (Offset: 0x03C).	R/W

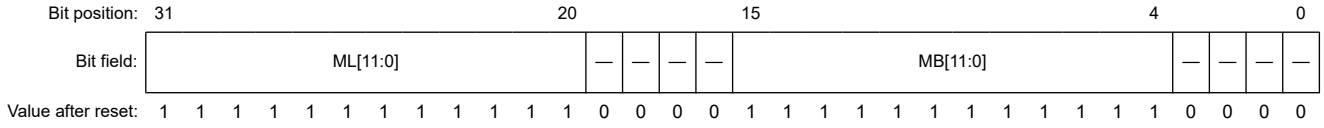
Table 56.23 Valid reset signal

Reset signal	Detected Parity Error	Received System Error	Received Master Abort	Received Target Abort	Signaled Target Abort	Master Data Parity Error	I/O Limit	I/O Base
RST_LOAD_B	—	—	—	—	—	—	—	—
RST_RSM_B	—	—	—	—	—	—	—	—
RST_CFG_B	✓	✓	✓	✓	✓	✓	✓	✓

56.3.2.9 PCI_RC_MEMBL : Memory Base/ Limit Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6020



Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
15:4	MB[11:0]	Memory Base Initial value 0xFFF0	R/W
19:16	—	These bits are read as 0. The write value should be 0.	R/W
31:20	ML[11:0]	Memory Limit Initial value 0xFFF0	R/W

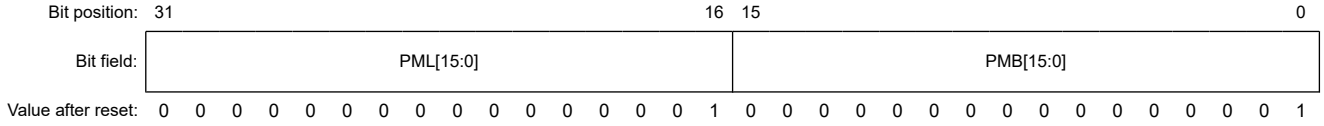
Table 56.24 Valid reset signal

Reset signal	Memory Limit	Memory Base
RST_LOAD_B	—	—
RST_RSM_B	—	—
RST_CFG_B	✓	✓

56.3.2.10 PCI_RC_PMBL : Prefetchable Memory Base/ Limit Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6024



Bit	Symbol	Function	R/W
15:0	PMB[15:0]	Prefetchable Memory Base Initial value 0xFFF0	R/W
31:16	PML[15:0]	Prefetchable Memory Limit Initial value 0xFFF0	R/W

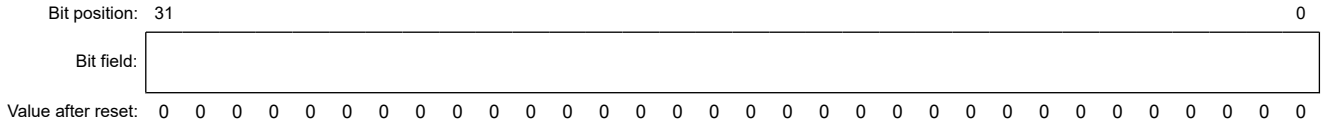
Table 56.25 Valid reset signal

Reset signal	Prefetchable Memory Limit	Prefetchable Memory Base
RST_LOAD_B	—	—
RST_RSM_B	—	—
RST_CFG_B	✓	✓

56.3.2.11 PCI_RC_PBUP32 : Prefetchable Base Upper 32 Bits Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6028



Bit	Symbol	Function	R/W
31:0	n/a	Prefetchable Base Upper 32 bits Initial value 0x0000_0000	R/W

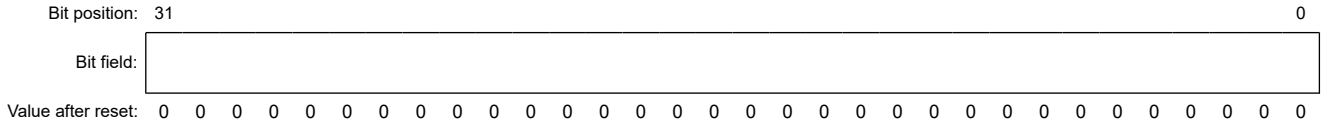
Table 56.26 Valid reset signal

Reset signal	Prefetchable Base Upper 32 bits
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓

56.3.2.12 PCI_RC_PLUP32 : Prefetchable Limit Upper 32 Bits Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x602C



Bit	Symbol	Function	R/W
31:0	n/a	Prefetchable Limit Upper 32 bits Initial value 0x0000_0000	R/W

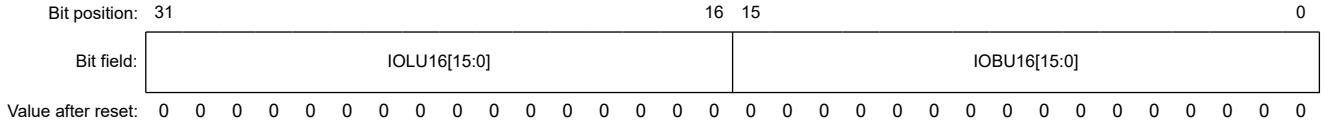
Table 56.27 Valid reset signal

Reset signal	Prefetchable Limit Upper 32 bits
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓

56.3.2.13 PCI_RC_IOBLUP16 : I/O Base/Limit Upper 16 bits Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6030



Bit	Symbol	Function	R/W
15:0	IOBU16[15:0]	IO Base Upper 16 bits This is an unused field in this macros.	R/W

Bit	Symbol	Function	R/W
31:16	IOLU16[15:0]	IO Limit Upper 16 bits This is an unused field in this macros.	R/W

The PCI_RC_IOBLUP16 register forms a 64-bit memory space in combination with Base Address Register 0 (BAR0).

Table 56.28 Valid reset signal

Reset signal	I/O Limit Upper 16 bits	I/O Base Upper 16 bits
RST_LOAD_B	—	—
RST_RSM_B	—	—
RST_CFG_B	✓	✓

56.3.2.14 PCI_RC_CP : Capabilities Pointer Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CP[7:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CP[7:0]	Capability Pointer Capability implementation start address 0x40 PCI Power Management Capability is implemented from 0x40. The lower 2 bits are fixed to 00b (reserved) and cannot be written from the UDL side.	Hwinit
31:8	—	These bits are read as 0.	R

The PCI_RC_CP register indicates the I/O base, limit, and secondary status.

Table 56.29 Valid reset signal

Reset signal	Capability Pointer[7:2]
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—

56.3.2.15 PCI_RC_BC_INT : Bridge Control and Interrupt

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x603C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	SBR	—	VGA16D	VGAE	ISAE	SERRE	PERE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IP[7:0]								IL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	IL[7:0]	Interrupt Line Fixed 0x00.	R/W
15:8	IP[7:0]	Interrupt Pin Fixed 0x00.	Hwinit
16	PERE	Parity Error Response Enable Writing 1 sets the Master Data Parity Error bit in the Secondary Status register when a Poisoned TLP is received.	R/W
17	SERRE	SERR# Enable 1 write enables INT_SERR* interrupt notification	R/W
18	ISAE	ISA Enable Initial value 0 Not used in this macro.	R/W
19	VGAE	VGA Enable Initial value 0 Not used in this macro.	R/W
20	VGA16D	VGA 16-bit Decode Initial value 0 Not used in this macro.	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	SBR	Secondary Bus Reset 1 Write to Hot Reset state	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

Table 56.30 Valid reset signal

Reset signal	Secondary Bus Reset	VGA 16-bit Decode	VGA Enable	ISA Enable	SERR# Enable	Parity Error Response Rnable	Interrupt Pin	Interrupt Line
RST_LOAD_B	—	—	—	—	—	—	✓	—
RST_RSM_B	—	—	—	—	—	—	—	—
RST_CFG_B	✓	✓	✓	✓	✓	✓	—	✓

56.3.2.16 PCI_RC_PMC : PM Capabilities

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PMES[4:0]					D2S	D1S	AUXC[2:0]			DSI	IRRD0	—	V[2:0]		
Value after reset:	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NCP[7:0]								CID[7:0]							
Value after reset:	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
7:0	CID[7:0]	Capability ID Indicates the PCI Power Management Capability. It is fixed at 0x01.	R
15:8	NCP[7:0]	Next Capability Pointer Indicates the PCI Express Capability start address.	R
18:16	V[2:0]	Version 011b is fixed. PCI Power Management Interface Specification Rev.1.2	Hwinit

Bit	Symbol	Function	R/W
19	—	This bit is read as 0.	R
20	IRRD0	Immediate_Readiness_on_Return_to_D0 0 fixed	R
21	DSI	Device Specific Initialization (DSI) Indicates whether or not DSI (Device Specific Initialization) is used. 0: not supported 1: Support	Hwinit
24:22	AUXC[2:0]	AUX_Current Indicates the 3.3Vaux auxiliary current (the maximum current value supplied from the auxiliary power supply). read value returns 000b. (AUX not supported) 0 0 0: 0 (self powered) 0 0 1: 55 mA 0 1 0: 100 mA 0 1 1: 160 mA 1 0 0: 220 mA 1 0 1: 250 mA 1 1 0: 320 mA 1 1 1: 375 mA	Hwinit
25	D1S	D1 Support Indicates whether the D1 Power Management State is supported. 0: not supported 1: Support	Hwinit
26	D2S	D2 Support Indicates whether D2 Power Management State is supported. 0: not supported 1: Support	Hwinit
31:27	PMES[4:0]	PME Support Indicates whether PME is supported in each Device State. xxxx1: supports D0 xxx1x: supports D1 xx1xx: supports D2 x1xxx: support D3hot 1xxxx: D3cold supported (not supported)	Hwinit

The PCI_RC_PMC register indicates various support information.

Table 56.31 Valid Reset Signal

Reset signal	PME Support	D2 Support	D1 Support	AUX_Current	DSI	Version	Next Capability Pointer[7:2]
RST_LOAD_B	✓	✓	✓	✓	✓	✓	✓
RST_RSM_B	—	—	—	—	—	—	—
RST_CFG_B	—	—	—	—	—	—	—

56.3.2.17 PCI_RC_PMSC : PM Status/Control Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6044

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PMES	—	—	—	—	—	—	PMEE	—	—	—	—	NSR	—	PS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
1:0	PS[1:0]	PowerState Set the PCI Device State. 0 0: D0 (Default) 0 1: D1 (Do not set) 1 0: D2 (Do not set) 1 1: D3hot	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	NSR	No_Soft_Reset Indicates that internal reset is not performed inside the device at the power state transition from D3hot to D0.	R
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PMEE	PME Enable Controls PME assertions. If 1, PME assertion is enabled. Assert PME if PME_Status is set at this time. PCI Express performs Link Wake-up processing, and then performs PME assert processing by sending PM_PME Message. Note: The specifications are as follows depending on the value of the PM Capabilities register PME Support[4] (PME Support in D3cold). PME Support[4] = 1 Reset: RST_RSM_B PME Support[4] = 0 Reset: RST_CFG_B	R/W
14:9	—	These bits are read as 0. The write value should be 0.	R/W
15	PMES	PME Status Indicates that a PME assert factor has occurred. 1 indicates that there is a PME assertion factor.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_PMSC register indicates and controls the PME status.

Table 56.32 Valid reset signal

Reset signal	PME Enable	PowerState
RST_LOAD_B	—	—
RST_RSM_B	✓	—
RST_CFG_B	—	✓

56.3.2.18 PCI_RC_PCIEC : PCI Express Capability Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	IMN[4:0]					SI	DPT[3:0]				CV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NCP[7:0]								CID[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CID[7:0]	Capability ID Indicates PCI Express Capability. 0x10 fixed.	R
15:8	NCP[7:0]	Next Capability Pointer Indicates that this Capability List is the final list (0x00 fixed). Lower 2 bits [9:8] is Reserved and fixed to 00b.	Hwinit
19:16	CV[3:0]	Capability Version Indicates the version of PCI Express Capability Structure. 0x2 fixed.	Hwinit
23:20	DPT[3:0]	DevicePort Type Indicates that it is a PCI Express Root Complex device. 0x0: PCI Express Endpoint device 0x1: Legacy PCI Express Endpoint device 0x4: Root Port of PCI Express Root Complex (Default) 0x5: Upstream Port of PCI Express Switch 0x6: Downstream Port of PCI Express Switch 0x7: PCI Express-to-PCI/PCI-X Bridge 0x8: PCI/PCI-X-to-PCI Express Bridge 0x9: Root Complex Integrated Endpoint Device 0xA: Root Complex Event Collector Others: Reserved	R
24	SI	Slot Implemented Setting it to 1 indicates that the PCI Express Link (Down Port) is connected to the Slot.	Hwinit
29:25	IMN[4:0]	Interrupt Message Number 0x00 fixed.	R
31:30	—	These bits are read as 0.	R

The PCI_RC_PCIEC register indicates the PCIe capability.

Table 56.33 Valid reset signal

Reset signal	Slot Implemented	Next Capability Pointer
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—

56.3.2.19 PCI_RC_DEVC : Device Capabilities Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RBER	—	—	—	—	—	—	—	—	—	—	—	—	MPSS[2:0]		
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	MPSS[2:0]	Max_Payload_Size Supported 0 0 0: 128 B max payload size 0 0 1: 256 B max payload size (Default) 0 1 0: 512 B max payload size 0 1 1: 1024 B max payload size 1 0 0: 2048 B max payload size 1 0 1: 4096 B max payload size 1 1 0: Reserved 1 1 1: Reserved	Hwinit
14:3	—	These bits are read as 0.	R
15	RBER	Role-Based Error Reporting 1 set if you have implemented the Error Reporting feature (Compliant with Rev1.1 or later). Fixed to 1 in this core.	R
31:16	—	These bits are read as 0.	R

The PCI_RC_DEVC register indicates the device capability.

Table 56.34 Valid reset signal

Reset signal	Max_Payload_Size Supported
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—

56.3.2.20 PCI_RC_DEVCS : Device Control/Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6068

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	URD	FED	NFED	CED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MRRS[2:0]			—	—	—	—	MPS[2:0]			ERO	URRE	FERE	NFERE	CERE
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	CERE	Correctable Error Reporting Enable Controls the generation of ERR_COR Messages. 1 enables Message generation.	R/W
1	NFERE	Non-Fatal Error Reporting Enable Controls generation of ERR_NONFATAL Message. 1 enables Message generation.	R/W
2	FERE	Fatal Error Reporting Enable Controls the generation of ERR_FATAL Messages. 1 enables Message generation.	R/W
3	URRE	Unsupported Request Reporting Enable ERR_NONFATAL due to Unsupported Request detection Or control the generation of ERR_FATAL Message. 1 enables Message generation.	R/W
4	ERO	Enable Relaxed Ordering Set whether or not to use Relaxed Ordering as a Requester. 0: Not supported 1: Support	R/W
7:5	MPS[2:0]	Max_Payload_Size Set Max_Payload_Size. 0 0 0: 128 B max payload size (Default) 0 0 1: 256 B max payload size 0 1 0: 512 B max payload size 0 1 1: 1024 B max payload size 1 0 0: 2048 B max payload size 1 0 1: 4096 B max payload size 1 1 0: Reserved 1 1 1: Reserved	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
14:12	MRRS[2:0]	Max_Read_Request_Size Set Max_Read_request_Size. 0 0 0: 128 B max read request size 0 0 1: 256 B max read request size 0 1 0: 512 B max read request size (Default) 0 1 1: 1024 B max read request size 1 0 0: 2048 B max read request size 1 0 1: 4096 B max read request size 1 1 0: Reserved 1 1 1: Reserved	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	CED	Correctable Error Detected Indicates that a correctable error was detected. Indicates that an Error is detected in 1.	R/W
17	NFED	Non-Fatal Error Detected Indicates that a Non-Fatal Error was detected. Indicates that an Error is detected in 1.	R/W
18	FED	Fatal Error Detected Indicates that a Fatal Error was detected. Indicates that an Error is detected in 1.	R/W
19	URD	Unsupported Request Detected Indicates that an Unsupported Request Error was detected. Indicates that an Error is detected in 1.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_DEVCS register controls the device and indicates the device status.

Table 56.35 Valid reset signal

Reset signal	Unsupported Request Detected	Fatal Error Detected	Non-Fatal Error Detected	Correctable Error Detected	Max_Read_Request_Size	Max_Payload_Size	Enable Relaxed Ordering	Unsupported Request Reporting Enable	Fatal Error Reporting Enable	Non-Fatal Error Reporting Enable	Correctable Error Reporting Enable
RST_LOAD_B	—	—	—	—	—	—	—	—	—	—	—
RST_RSM_B	—	—	—	—	—	—	—	—	—	—	—
RST_CFG_B	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

56.3.2.21 PCI_RC_LINKC : Link Capabilities Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x606C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PN[7:0]								—	ASPMOC	LBNC	DLLLARC	—	—	L1EL[2:1]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	L1EL	L0EL[2:0]			ASPMS[1:0]		MLW[5:0]						SLS[3:0]			
Value after reset:	1	1	0	0	1	1	0	0	0	0	*1	*1	0	0	1	1

Bit	Symbol	Function	R/W
3:0	SLS[3:0]	Supported Link Speed 0x1: 2.5 GT/s link speed supported 0x2: 5.0 GT/s and 2.5 GT/s Link speeds supported 0x3: 8.0 GT/s link speed supported (default) 0x4: 16.0 GT/s link speed supported (prohibited) Others: Reserved	Hwinit
9:4	MLW[5:0]	Maximum Link Width 0x00: Reserved 0x01: x1 (Default when using ch1.) 0x02: x2 (Default when using ch0. Do not set for ch1.) 0x04: x4 (prohibited) 0x08: x8 (prohibited) 0x0C: x12 (prohibited) 0x10: x16 (prohibited) 0x20: x32 (prohibited)	Hwinit
11:10	ASPMS[1:0]	Active State Power Management (ASPM) Support 0 0: Reserved 0 1: L0s entry supported 1 0: L1 entry supported 1 1: L0s and L1 entry supported (default)	Hwinit

Bit	Symbol	Function	R/W
14:12	L0EL[2:0]	L0s Exit Latency 0 0 0: Less than 64 ns 0 0 1: 64 ns to less than 128 ns 0 1 0: 128 ns to less than 256 ns 0 1 1: 256 ns to less than 512 ns 1 0 0: 512 ns to less than 1 µs (default) 1 0 1: 1 µs to less than 2 µs 1 1 0: 2 µs to 4 µs 1 1 1: More than 4 µs	Hwinit
17:15	L1EL[2:0]	L1 Exit Latency 0 0 0: Less than 1 µs 0 0 1: 1 µs to less than 2 µs 0 1 0: 2 µs to less than 4 µs 0 1 1: 4 µs to less than 8 µs 1 0 0: 8 µs to less than 16 µs 1 0 1: 16 µs to less than 32 µs (default) 1 1 0: 32 µs to 64 µs 1 1 1: More than 64 µs	Hwinit
19:18	—	These bits are read as 0.	R
20	DLLLARC	Data Link Layer Link Active Reporting Capable in the Data Link Control and Management State Machine Indicates support for the DL_Active state reporting feature.	Hwinit
21	LBNC	Link Bandwidth Notification Capability Indicates support for Link Bandwidth Notification status and Interrupt functions.	Hwinit
22	ASPMOC	ASPM Optionality Compliance 1 fixed	Hwinit
23	—	This bit is read as 0.	R
31:24	PN[7:0]	Port Number Indicates the port number of PCI Express Link.	Hwinit

Note 1. Depends on the channel setting.

The PCI_RC_LINKC register indicates the link capabilities. The PCI_RC_LINKC register can be written during initialization.

Table 56.36 Valid reset signal

Reset signal	ASPM Optionality Compliance	Link Bandwidth Notification Capability	Data Link Layer Link Active Reporting Capable	L1 Exit Latency	L0s Exit Latency	Active State Power Management Support	Maximum Link Width	Supported Link Speed
RST_LOAD_B	✓	✓	✓	✓	✓	✓	✓	✓
RST_RSM_B	—	—	—	—	—	—	—	—
RST_CFG_B	—	—	—	—	—	—	—	—

56.3.2.22 PCI_RC_LINKCS : Link Control/Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	LABS	LBMS	DLLLA	SCC	LT	—	NLW[5:0]						CLS[3:0]			
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	LABIE	LBMIE	HAWD	—	ES	CCC	RL	LD	RCB[3:0]	—	ASPMC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ASPMC[1:0]	Active State Power Management (ASPM) Control Sets the permission level for Active State Power Management. 0 0: Disabled (Default) 0 1: L0s Entry Supported 1 0: L1 Entry Supported 1 1: L0s and L1 Entry Supported	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	RCB[3:0]	Read Completion Boundary (RCB) 0: 64 bytes 1: 128 bytes (Default)	R
4	LD	Link Disable Setting it to 1 causes the LTSSM to transition to the Disabled state.	R/W
5	RL	Retrain Link Setting it to 1 causes the LTSSM to transition to the Recovery state and start Link Retraining. Note that reading is always 0.	R/W
6	CCC	Common Clock Configuration Sets whether the Common Reference Clock is used. Only Common Reference Clock is supported for Single Link. 0: Provided by non-Common Reference Clock (Default) 1: Supplied by Common Reference Clock	R/W
7	ES	Extended Synch If set to 1, 4096 FTS Ordered-sets will be sent when transitioning from L0s to L0. Also, 1024 TS1 Ordered-sets are transmitted at the beginning of the Recovery state when transitioning from L1 to L0. The default value will be 0.	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	HAWD	Hardware Autonomous Width Disable Set the Link Width Change function to Disable. 0: Link Width Change Enable (Default) 1: Link Width Change Disable	R/W
10	LBMIE	Link Bandwidth Management Interrupt Enable Interrupt generation is controlled by the Link Bandwidth Status bit (bit 30). 0: Interrupt Disable (Default) 1: Interrupt Enable	R/W
11	LABIE	Link Autonomous Bandwidth Interrupt Enable Interrupt generation is controlled by the Link Autonomous Bandwidth Management Status bit (bit 31). 0: Interrupt Disable (Default) 1: Interrupt Enable	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
19:16	CLS[3:0]	Current Link Speed 0x0: During reset period 0x1: 2.5 GT/s PCI Express Link 0x2: 5.0 GT/s PCI Express Link 0x3: 8.0 GT/s PCI Express Link 0x4: 16.0 GT/s PCI Express Link	R
25:20	NLW[5:0]	Negotiated Link Width Indicates the Link width established as a result of negotiation. 0x01: x1 0x02: x2 0x04: x4 0x08: x8 0x0C: x12 0x10: x16 0x20: x32 Others: Reserved	R
26	—	This bit is read as 0. The write value should be 0.	R/W
27	LT	Link Training 1 indicates that the Physical layer LTSSM is in Configuration state or Recovery state. This bit is cleared when exiting the Configuration/Recovery state.	R
28	SCC	Slot Clock Configuration Indicates that the reference clock shared with the EP is used. 0: Do not use Connector Reference Clock 1: Use Connector Reference Clock (Default)	Hwinit
29	DLLLA	Data Link Layer Link Active 1 indicates that the Data Link Layer is in the Link Active state.	R
30	LBMS	Link Bandwidth Management Status This bit is output when Link Bandwidth Management Interrupt Enable is asserted. Cleared by writing 1. Writing 0 is ignored.	R/W
31	LABS	Link Autonomous Bandwidth Status This bit is output when Link Autonomous Bandwidth Interrupt Enable is asserted. Cleared by writing 1. Writing 0 is ignored.	R/W

The PCI_RC_LINKCS register controls the link and indicates the link status.

Table 56.37 Valid reset signal

Reset signal	Link Autonomous Bandwidth Status	Link Bandwidth Management Status	Slot Clock Configuration	Link Autonomous Bandwidth Interrupt Enable	Link Bandwidth Management Interrupt Enable	Hardware Autonomous Width Disable	Extended Synch	Common Clock Configuration	Retrain Link	Link Disable	Active State Power Management Control
RST_LOAD_B	—	—	✓	—	—	—	—	—	—	—	—
RST_RSM_B	—	—	—	—	—	—	—	—	—	—	—
RST_CFG_B	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓

56.3.2.23 PCI_RC_SLOTC : Slot Capabilities

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6074

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSN[12:0]													NCCS	EIP	SPLS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPLS	SPLV[7:0]								HPC	HPS	PIP	AIP	MRLSP	PCP	ABP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ABP	Attention Button Present this bit indicates that an Attention Button for this slot is electrically controlled by the chassis.	Hwinit
1	PCP	Power Control Present this bit indicates that a software programmable Power Controller is implemented for this slot/adapter (depending on form factor).	Hwinit
2	MRLSP	MRL Sensor Present this bit indicates that an MRL Sensor is implemented on the chassis for this slot.	Hwinit
3	AIP	Attention Indicator Present this bit indicates that an Attention Indicator is electrically controlled by the chassis.	Hwinit
4	PIP	Power Indicator Present this bit indicates that a Power Indicator is electrically controlled by the chassis for this slot.	Hwinit
5	HPS	Hot-Plug Surprise this bit indicates that an adapter present in this slot might be removed from the system without any prior notification. This is a form factor specific capability. This bit is an indication to the operating system to allow for such removal without impacting continued software operation.	Hwinit
6	HPC	Hot-Plug Capable this bit indicates that this slot is capable of supporting hot-plug operations.	Hwinit
14:7	SPLV[7:0]	Slot Power Limit Value n combination with the Slot Power Limit Scale Register value, sets the upper limit of Power for the Slot. This register is enabled when the Slot Implemented bit (Express Capability Register (offset: 0x060), bit 24) is set. 0xF0: 250 W Slot Power Limit 0xF1: 275 W Slot Power Limit 0xF2: 300 W Slot Power Limit Others: reserved	Hwinit
16:15	SPLS[1:0]	Slot Power Limit Scale Sets the Scale for the Slot Power Limit Value. This register is enabled when the Slot Implemented bit (Express Capability Register (offset: 0x060), bit 24) is set. 0 0: 1.0x 0 1: 0.1x 1 0: 0.01x 1 1: 0.001x	Hwinit
17	EIP	Electromechanical Interlock Present Indicates that Electromechanical Interlock is implemented.	Hwinit
18	NCCS	No Command Completed Support Indicates that the Slot will not issue a software notification if the issued command is completed by the Hot-Plug Controller.	Hwinit
31:19	PSN[12:0]	Physical Slot Number Indicates the slot number connected to the port.	Hwinit

When PCI Express Capability Structure: Configuration Space Bit [24]: Slot Implemented is 0, there is no implementation of the slot. Therefore, this bit will be all 0.

Table 56.38 Valid reset signal

Reset signal	Physical Slot Number	No Command Completed Support	Electromechanical Interlock Present	Slot Power Limit Scale	Slot Power Limit Value	Hot-Plug Capable	Hot-Plug Surprise	Power Indicator Present	Attention Indicator Present	MRL Sensor Present	Power Control Present	Attention Button
Present	—	—	—	—	—	—	—	—	—	—	—	—
RST_LOAD_B	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RST_RSM_B	—	—	—	—	—	—	—	—	—	—	—	—
RST_CFG_B	—	—	—	—	—	—	—	—	—	—	—	—

56.3.2.24 PCI_RC_SLOTCS : Slot Control/Status

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6078

Bit position:	31																													0
Bit field:																														
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Slot ControlStatus Since PCI Express Capability Structure: Configuration Space Bit [24]: Slot Implemented is 0, the slot is not implemented, so all registers other than bit [22]: Presence Detect State are 0.	R

When PCI Express Capability Structure: Configuration Space Bit [24]: Slot Implemented is 0, there is no implementation of the slot. Therefore, bits other than bit [22]: Presence Detect State will be all 0.

56.3.2.25 PCI_RC_ROOTCC : Root Control/Capabilities

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x607C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PMEIE	SEFEE	SENFEE	SECEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SECEE	System Error on Correctable Error Enable This bit makes it possible to generate a system error when NON_ERR_FATAL occurs anywhere in the hierarchy below the root port.	R/W

Bit	Symbol	Function	R/W
1	SENFEE	System Error on Non-Fatal Error Enable This bit makes it possible to generate a system error when NON_ERR_FATAL occurs anywhere in the hierarchy below the root port.	R/W
2	SEFEE	System Error on Fatal Error Enable This bit makes it possible to generate a system error when ERR_FATAL occurs anywhere in the hierarchy below the root port.	R/W
3	PMEIE	PME Interrupt Enable When a PME Message is received, it is possible to generate a PME interrupt along with setting the PME status bit. An interrupt will also occur if this bit is enabled after PME status is already set.	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Table 56.39 Valid reset signal

Reset signal	PME Interrupt Enable	System Error on Fatal Error Enable	System Error on Non-Fatal Error Enable	System Error on Correctable Error Enable
RST_LOAD_B	—	—	—	—
RST_RSM_B	—	—	—	—
RST_CFG_B	✓	✓	✓	✓

56.3.2.26 PCI_RC_ROOTS : Root Status

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PMEP	PMES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PMERID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PMERID[15:0]	PME Requester ID This field indicates the PCI Requester ID of the most recent PME requestor. This field is only valid when the PME status bit is 1.	R
16	PMES	PME Status This bit indicates that a PME was received by the requestor indicated by the PME Requester ID field. Subsequent PMEs are held off until this bit is cleared by software.	R/W
17	PMEP	PME Pending This bit indicates that the PME Status bit is 1 and PME is pending. When the PME status bit is cleared by software, pending PMEs are resent in hardware by resetting the PME status bit and updating the Requester ID field appropriately. The PME pending bit is cleared by hardware when there are no more pending PMEs.	R
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Table 56.40 Valid reset signal

Reset signal	PME Pending	PME Status	PME Requester ID
RST_LOAD_B	—	—	—
RST_RSM_B	—	—	—
RST_CFG_B	✓	✓	✓

56.3.2.27 PCI_RC_DEVC2 : Device Capabilities 2 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CTDS	CTRS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Bit	Symbol	Function	R/W
3:0	CTRS[3:0]	Completion Timeout Ranges Supported Set Completion Timeout Range. Range A: 50 µs to 10 ms Range B: 10 ms to 250 ms Range C: 250 ms to 4 s Range D: 4 s to 64 s The above four patterns are determined, and the following combinations can be set. The initial value described in UM is 0x2, but please change the initial value according to the installed system/device performance, or specify the initial value as a product request. 0x0: Program setting Not supported 0x1: Range A 0x2: Range B 0x3: Ranges A and B 0x6: Ranges B and C 0x7: Ranges A, B, and C 0xE: Ranges B, C and D 0xF: Ranges A, B, C, and D Others: Reserved	Hwinit
4	CTDS	Completion Timeout Disable Supported Set whether to support the Completion Timeout Disable function. 0: Not supported 1: Supported (default)	Hwinit
31:5	—	These bits are read as 0.	R

The PCI_RC_DEVC2 register indicates the device capability.

Table 56.41 Valid reset signal

Reset signal	Completion Timeout Disable Supported	Completion Timeout Ranges Supported
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—

56.3.2.28 PCI_RC_DEVCS2 : Device Control 2/Status 2 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6088

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CTD	CTV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CTV[3:0]	Completion Timeout Value Set Completion Timeout Range. The default 0x0 sets a longer time than the Base Spec's default 50 µsec lower limit, but this takes into account the Base Spec's recommended 10 msec lower limit. 0x0: 10 ms to 50 ms (default) 0x1: 50 µs to 100 µs 0x2: 1 ms to 10 ms 0x5: 16 ms to 55 ms 0x6: 65 ms to 210 ms 0x9: 260 ms to 900 ms 0xA: 1 s to 3.5 s 0xD: 4 s to 13 s 0xE: 17 s to 64 s Others: reserved (setting prohibited)	R/W
4	CTD	Completion Timeout Disable Setting this bit enables the Completion Timeout Disable function.	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_DEVCS2 register controls the device and indicates the device status.

Table 56.42 Valid reset signal

Reset signal	Completion Timeout Disable	Completion Timeout Value
RST_LOAD_B	—	—
RST_RSM_B	—	—
RST_CFG_B	✓	✓

56.3.2.29 PCI_RC_LINKC2 : Link Capabilities 2 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x608C

Bit position:	31	7	1	0
Bit field:	—	SLSV[6:0]		—
Value after reset:	0	0 0 0 0 0 0 0	1 1 1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
7:1	SLSV[6:0]	Supported Link Speeds Vector The read value is undefined Indicates Support Link Speed Bit 0: 2.5 GT/s Bit 1: 5.0 GT/s Bit 2: 8.0 GT/s Bit 3: 16.0 GT/s (Setting prohibited) Bits 6 to 4: Reserved	Hwinit
31:8	—	These bits are read as 0.	R

The PCI_RC_LINKC2 register indicates the link capabilities.

Table 56.43 Valid reset signal

Reset signal	Supported Link Speeds Vector
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—

56.3.2.30 PCI_RC_LINCS2 : Link Control 2/Status 2 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	LER	EP3S	EP2S	EP1S	EQC	CDL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPD[3:0]				CSOS	EMC	TM[2:0]			SD	HASD	ETC	TLS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
3:0	TLS[3:0]	Target Link Speed Set the Link Speed value for notification to the opposite device during training. 0x01: 2.5 GT/s Target Link Speed 0x02: 5.0 GT/s Target Link Speed 0x03: 8.0 GT/s Target Link Speed (Default) 0x04: 16.0 GT/s Target Link Speed (prohibited) Others: Reserved	R/W
4	ETC	Enter Compliance Setting to 1 allows transition to Compliance mode. At this time, the Link Speed will be the value set in the Target Link Speed field.	R/W
5	HASD	Hardware Autonomous Speed Disable Controls the Link Speed Change function. 0: Support of Link Speed Change (Enable) 1: not Support of Link Speed Change (Disable)	R/W
6	SD	Selectable Deemphasis The read value is undefined De-emphasis setting register for RC Gen2 operation. 0: -6 dB 1: -3.5 dB	R/W
9:7	TM[2:0]	Transmit Margin Adjust the voltage level of the Transmitter. 000b Normal operating range See 001b to 111b Base Spec Section 8.3.4.	R/W

Bit	Symbol	Function	R/W
10	EMC	Enter Modified Compliance Setting bit for transmission of Modified Compliance pattern. 0: Compliance Pattern (Default) 1: Modified Compliance Pattern	R/W
11	CSOS	Compliance SOS When this bit is set to 1, insert the SKP Ordered-set periodically during compliance pattern transmission.	R/W
15:12	CPD[3:0]	Compliance Preset De-emphasis Set the De-emphasis Level in Polling.Compliance State. 0x00: -6 dB (Default) 0x01: -3.5 dB	R/W
16	CDL	Current De-emphasis Level This is a Status register that indicates the De-emphasis Level during Gen2 operation. Note: The initial value of this bit is 0 at reset and after Gen2-Linkup. However, although it indicates 1 at Gen1-Linkup, ignore this bit as it has no meaning in Gen1. 0: -6 dB (Default) 1: -3.5 dB	R
17	EQC	Equalization Complete When set to 1, this bit indicates that the Transmitter Equalization procedure has completed. The default value of this bit is 0. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Components that only support speeds below 8.0 GT/s are permitted to hardwire this bit to 0.	R
18	EP1S	Equalization Phase 1 Successful When set to 1, this bit indicates that Phase 1 of the Transmitter Equalization procedure has successfully completed. The default value of this bit is 0. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Components that only support speeds below 8.0 GT/s are permitted to hardwire this bit to 0.	R
19	EP2S	Equalization Phase 2 Successful When set to 1, this bit indicates that Phase 2 of the Transmitter Equalization procedure has successfully completed. The default value of this bit is 0. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Components that only support speeds below 8.0 GT/s are permitted to hardwire this bit to 0.	R
20	EP3S	Equalization Phase 3 Successful When set to 1, this bit indicates that Phase 3 of the Transmitter Equalization procedure has successfully completed. The default value of this bit is 0. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Components that only support speeds below 8.0 GT/s are permitted to hardwire this bit to 0.	R
21	LER	Link Equalization Request This bit is Set by hardware to request the Link equalization process to be performed on the Link. The default value of this bit is 0. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Components that only support speeds below 8.0 GT/s are permitted to hardwire this bit to 0.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_LINCS2 register controls the link and indicates the link status.

Table 56.44 Valid reset signal

Reset signal	Link Equalization Request	Compliance Preset/De-emphasis	Compliance SOS	Enter Modified Compliance	Transmit Margin	Hardware Autonomous Speed Disable	Enter Compliance	Target Link Speed
RST_LOAD_B	—	—	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓	✓	✓
RST_CFG_B	—	—	—	—	—	—	—	—

56.3.2.31 PCI_RC_BARMSK00L : Base Address Register Mask00 (Lower)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60A0

Bit position:	31																												0
Bit field:	BARM00L[31:0]																												
Value after reset:	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	BARM00L[31:0]	Base Address Register Mask00 (Lower) Mask register for Base Address Register 0 (BAR0).	Hwinit

The PCI_RC_BARMSK00L register indicates mask information for Base Address Register 0 (BAR0).

Table 56.45 Valid reset signal

Reset signal	Base Address Register Mask00 (Lower)
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—

56.3.2.32 PCI_RC_BARMSK00U : Base Address Register Mask00 (Upper)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60A4

Bit position:	31																												0
Bit field:	BARM00U[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	BARM00U[31:0]	Base Address Register Mask00 (Upper) Mask register for Base Address Register 1 (BAR1).	Hwinit

The PCI_RC_BARMSK00U register indicates mask information for Base Address Register 1 (BAR1).

Table 56.46 Valid reset signal (1 of 2)

Reset signal	Base Address Register Mask00 (Upper)
RST_LOAD_B	✓

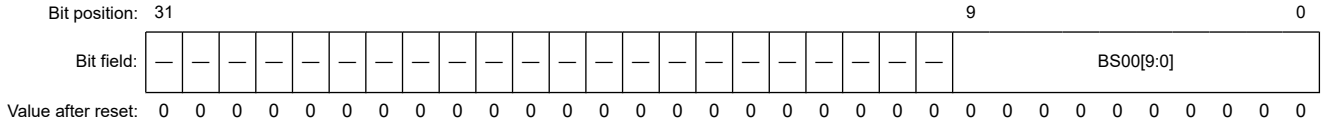
Table 56.46 Valid reset signal (2 of 2)

Reset signal	Base Address Register Mask00 (Upper)
RST_RSM_B	—
RST_CFG_B	—

56.3.2.33 PCI_RC_BSIZE00_01 : Base Size 00/01 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60C8



Bit	Symbol	Function	R/W
9:0	BS00[9:0]	Base Size 00 Sets the size of the TLP (DW Size) that will be accepted in Address Space 00 set in the Base Address Register and Base Address Mask Register. If a TLP with a packet length larger than the size set here is received (even if it is smaller than the Max Payload Size), a Completer Abort Error (CA) will be detected. Note that the default value is 0x000, which disables this function.	Hwinit
31:10	—	These bits are read as 0.	R

The PCI_RC_BSIZE00_01 register sets the acceptable TLP size.

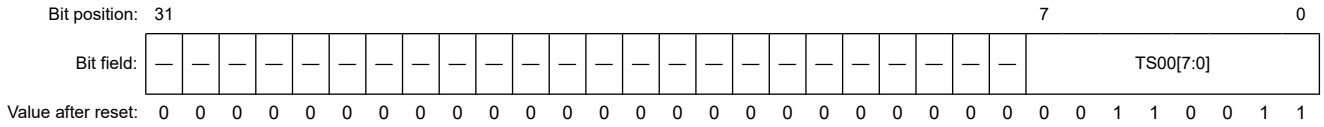
Table 56.47 Valid reset signal

Reset signal	Base Size 00
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—

56.3.2.34 PCI_RC_TSUPPORT00_01_02 : Type Supported 00/01/02 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60D8



Bit	Symbol	Function	R/W
7:0	TS00[7:0]	Type Supported 00 Set the Transaction Types that can be supported by Space00 (CFG_SPACE00_BASE). Bit 0: Memory read 32 bit Bit 1: Memory read 64 bit Bit 2: Memory read lock 32 bit Bit 3: Memory read lock 64 bit Bit 4: Memory write 32 bit Bit 5: Memory write 64 bit Bit 6: IO Read Bit 7: IO Write	Hwinit
31:8	—	These bits are read as 0.	R

The PCI_RC_TSUPPORT00_01_02 register indicates the transaction type which can be supported by the memory space.

Table 56.48 Valid reset signal

Reset signal	Type Supported 00
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—

56.3.2.35 PCI_RC_ADVERC : Advanced Error Reporting Capability Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6100

Bit position:	31											20	19			16	15													0
Bit field:	NCO[11:0]												CV[3:0]				PCIEECID[15:0]													
Value after reset:	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
15:0	PCIEECID[15:0]	PCI Express Extended Capability ID Indicates the Advanced Error Reporting Capability. Default: 0x0001	R
19:16	CV[3:0]	Capability Version Indicates the version of the Capability Structure. Default: 0x01	Hwinit
31:20	NCO[11:0]	Next Capability Offset Indicates the starting address for DeviceSerialNumberCapability.	Hwinit

The PCI_RC_ADVERC register indicates the advanced error reporting capability.

Table 56.49 Valid reset signal

Reset signal	Next Capability Offset	Capability Version
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—

56.3.2.36 PCI_RC_UNCESTS : Uncorrectable Error Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	URES	ECRCESO	MTLPS	ROSO	UCS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CASO	CTS	—	PTLPRS	—	—	—	—	—	—	—	DLPES	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	DLPES	Data Link Protocol Error Status Indicates that a Sequence Number error was detected in the Data Link Layer. 0: No error detected 1: Error detected	R/W
11:5	—	These bits are read as 0. The write value should be 0.	R/W
12	PTLPRS	Poisoned TLP Received Status Indicates that a Poisoned TLP (with payload and the EP field in the header is 1) has been received. 0: No error detected 1: Error detected	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	CTS	Completion Timeout Status Indicates that the corresponding Completion was not received within the specified time after sending a Non-Posted Request. 0: No error detected 1: Error detected	R/W
15	CASO	Completer Abort Status (Optional) Indicates that a Completion whose Completion Status is Completer Abort (CA) was returned after receiving a Non-Posted Request. 0: No error detected 1: Error detected	R/W
16	UCS	Unexpected Completion Status Indicates that a Completion was received, but there is no record of a corresponding Non-Posted Request sent (the Transaction Descriptor did not match). 0: No error detected 1: Error detected	R/W
17	ROSO	Receiver Overflow Status (Optional) Indicates that a TLP with a size larger than the free credits in the receive buffer was received. 0: No error detected 1: Error detected	R/W
18	MTLPS	Malformed TLP Status Indicates that a Malformed TLP was received. 0: No error detected 1: Error detected	R/W
19	ECRCESO	ECRC Error Status (Optional) Indicates that an ECRC Error was received. 0: No error detected 1: Error detected	R/W
20	URES	Unsupported Request Error Status Indicates that an unsupported TLP was received. 0: No error detected 1: Error detected	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_UNCESTS register indicates the uncorrectable error status.

Table 56.50 Valid reset signal (1 of 2)

Reset signal	Unsupported Request Error Status	ECRC Error Status	Malformed TLP Status	Receiver Overflow Status	Unexpected Completion Status	Completer Abort Status	Completion Timeout Status	Poisoned TLP Received Status	Data Link Protocol Error Status
RST_LOAD_B	—	—	—	—	—	—	—	—	—

Table 56.50 Valid reset signal (2 of 2)

Reset signal	Unsupported Request Error Status	ECRC Error Status	Malformed TLP Status	Receiver Overflow Status	Unexpected Completion Status	Completer Abort Status	Completion Timeout Status	Poisoned TLP Received Status	Data Link Protocol Error Status
RST_RSM_B	✓	✓	✓	✓	✓	✓	✓	✓	✓
RST_CFG_B	—	—	—	—	—	—	—	—	—

56.3.2.37 PCI_RC_UNCEMASK : Uncorrectable Error Mask Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	UREM	ECRC EMO	MTLP M	ROMO	UCM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CAMO	CTM	—	PTLP RM	—	—	—	—	—	—	—	DLPE M	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	DLPEM	Data Link Protocol Error Mask Masks error notifications when a Data Link Protocol Error is detected. 0: No mask 1: Send Error Message, Mask Update of First Error Pointer	R/W
11:5	—	These bits are read as 0. The write value should be 0.	R/W
12	PTLPRM	Poisoned TLP Received Mask Masks error notification when Poisoned TLP Error is detected. 0: No mask 1: Send Error Message, record header to Header Log register, mask Update of First Error Pointer	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	CTM	Completion Timeout Mask Masks the error notification when Completion Timeout Error is detected. 0: No mask 1: Send Error Message, Mask Update of First Error Pointer	R/W
15	CAMO	Completer Abort Mask (Optional) Masks the error notification when a Completer Abort Error is detected. 0: No mask 1: Send Error Message, record header to Header Log register, mask Update of First Error Pointer	R/W
16	UCM	Unexpected Completion Mask Masks error notification when Unexpected Completion Error is detected. 0: No mask 1: Send Error Message, record header to Header Log register, mask Update of First Error Pointer	R/W

Bit	Symbol	Function	R/W
17	ROMO	Receiver Overflow Mask (Optional) Masks error notification when Receiver Overflow Error is detected. 0: No mask 1: Send Error Message, Mask Update of First Error Pointer	R/W
18	MTLPM	Malformed TLP Mask Masks error notification when Malformed TLP Error is detected. 0: No mask 1: Send Error Message, record header to Header Log register, mask Update of First Error Pointer	R/W
19	ECRCEMO	ECRC Error Mask (Optional) Masks error notification when ECRC Error is detected. 0: No mask 1: Mask	R/W
20	UREM	Unsupported Request Error Mask Masks the error notification when an Unsupported Request Error is detected. 0: No mask 1: Send Error Message, record header to Header Log register, mask Update of First Error Pointer	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_UNCEMASK register masks the uncorrectable error status.

Table 56.51 Valid reset signal

Reset signal	Unsupported Request Error Mask	ECRC Error Mask	Malformed TLP Mask	Receiver Overflow Mask	Unexpected Completion Mask	Completer Abort Mask	Completion Timeout Mask	Poisoned TLP Received Mask	Data Link Protocol Error Mask
RST_LOAD_B	—	—	—	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓	✓	✓	✓
RST_CFG_B	—	—	—	—	—	—	—	—	—

56.3.2.38 PCI_RC_UNCESVY : Uncorrectable Error Severity Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x610C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	URES	ECRCESO	MTLPS	ROSO	UCS
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CASO	CTS	—	PTLPRS	—	—	—	—	—	—	—	DLPES	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	DLPES	Data Link Protocol Error Severity Sets the Error Severity when a Data Link Protocol Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
11:5	—	These bits are read as 0. The write value should be 0.	R/W
12	PTLPRS	Poisoned TLP Received Severity Sets the Error Severity when Poisoned TLP Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	CTS	Completion Timeout Severity Sets the Error Severity when Completion Timeout Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
15	CASO	Completer Abort Severity (Optional) Sets the Error Severity when a Completer Abort Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
16	UCS	Unexpected Completion Severity Sets the Error Severity when an Unexpected Completion Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
17	ROSO	Receiver Overflow Severity (Optional) Set the Error Severity when Receiver Overflow Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
18	MTLPS	Malformed TLP Severity Sets the Error Severity when Malformed TLP Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
19	ECRCESO	ECRC Error Severity (Optional) Sets the Error Severity when receiving an ECRC Error. 0: Non-Fatal Error 1: Fatal Error	R/W
20	URES	Unsupported Request Error Severity Set the Error Severity when Unsupported Request Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	—	This bit is read as 1. The write value should be 1.	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_UNCESVY register sets the uncorrectable error severity.

Table 56.52 Valid reset signal (1 of 2)

Reset signal	Unsupported Request Error Severity	ECRC Error Severity	Malformed TLP Severity	Receiver Overflow Severity	Unexpected Completion Severity	Completer Abort Severity	Completion Timeout Severity	Poisoned TLP Received Severity	Data Link Protocol Error Severity
RST_LOAD_B	—	—	—	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 56.52 Valid reset signal (2 of 2)

Reset signal	Unsupported Request Error Severity	ECRC Error Severity	Malformed TLP Severity	Receiver Overflow Severity	Unexpected Completion Severity	Completer Abort Severity	Completion Timeout Severity	Poisoned TLP Received Severity	Data Link Protocol Error Severity
RST_CFG_B	—	—	—	—	—	—	—	—	—

56.3.2.39 PCI_RC_CESTS : Correctable Error Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6110

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	ANFES	RTTS	—	—	—	REPLAYNUMRS	BDLLPS	BTLPS	—	—	—	—	—	RES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RES	Receiver Error Status (optional) 0: No error detected 1: Error detected	R/W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	BTLPS	Bad TLP Status Indicates that a TLP CRC error or Sequence Number error was detected. 0: No error detected 1: Error detected	R/W
7	BDLLPS	Bad DLLP Status Indicates that a DLLP CRC error was detected. 0: No error detected 1: Error detected	R/W
8	REPLAYNUMRS	REPLAY_NUM Rollover Status Replay occurred four times in a row, indicating REPLAY_NUM rolled over from 11b to 00b. 0: No error detected 1: Error detected	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	RTTS	Replay Timer Timeout Status Indicates that a Timeout error occurred when a TLP was sent and an Ack or Nak DLLP could not be received within the specified time. 0: No error detected 1: Error detected	R/W
13	ANFES	Advisory Non-Fatal Error Status Indicates that an Advisory Non-Fatal Error was detected. 0: No error detected 1: Error detected	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_CESTS register indicates the correctable error status.

Table 56.53 Valid reset signal

Reset signal	Advisory Non-Fatal Error Status	Replay Timer Timeout Status	REPLAY_NUM Rollover Status	Bad DLLP Status	Bad TLP Status	Receiver Error Status
RST_LOAD_B	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓
RST_CFG_B	—	—	—	—	—	—

56.3.2.40 PCI_RC_CEMASK : Correctable Error Mask Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6114

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	ANFEM	RTTM	—	—	—	REPLAYNUMRM	BDLLPM	BTLPM	—	—	—	—	—	REM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	REM	Receiver Error Mask (optional) 0: No mask 1: Mask error message transmission	R/W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	BTLPM	Bad TLP Mask Masks error notification when Bad TLP Error is detected. 0: No mask 1: Mask error message transmission	R/W
7	BDLLPM	Bad DLLP Mask Masks error notification when Bad DLLP Error is detected. 0: No mask 1: Mask error message transmission	R/W
8	REPLAYNUMRM	REPLAY_NUM Rollover Mask REPLAY_NUM Mask error notification when Roll Over Error is detected. 0: No mask 1: Mask error message transmission	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	RTTM	Replay Timer Timeout Mask Masks error notification when Replay Timer Timeout Error is detected. 0: No mask 1: Mask error message transmission	R/W
13	ANFEM	Advisory Non-Fatal Error Mask Advisory Masks error notifications when non-fatal errors are detected. 0: No mask 1: Mask Advisory Non-Fatal Error handling (Mask updating First Error Pointer and Header Logging and sending Error Message)	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_CEMASK register masks the correctable error status.

Table 56.54 Valid reset signal

Reset signal	Advisory Non-Fatal Error Mask	Replay Timer Timeout Mask	REPLAY_NUM Rollover Mask	Bad DLLP Mask	Bad TLP Mask	Receiver Error Mask
RST_LOAD_B	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓
RST_CFG_B	—	—	—	—	—	—

56.3.2.41 PCI_RC_ADVECC : Advanced Error Capabilities and Control Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6118

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ECRCCE	ECRCCC	ECRCGE	ECRCGC	FEP[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	FEP[4:0]	First Error Pointer Indicates the field value of the Uncorrectable Error Status register for the first detected Uncorrectable Error Bit [4:0] First Error Pointer is reset by RST_GP_B.	R
5	ECRCGC	ECRC Generation Capable Presence or absence of ECRC Generation function 0: None 1: Yes (Default)	R
6	ECRCGE	ECRC Generation Enable ECRC Generation Enable setting 0: Disable (Default) 1: Enable	R/W
7	ECRCCC	ECRC Check Capable Presence or absence of ECRC Check function 0: None 1: Yes (Default)	R
8	ECRCCE	ECRC Check Enable ECRC Check Enable setting 0: Disable (Default) 1: Enable	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_RC_ADVECC register indicates and controls the advanced error capabilities.

Table 56.55 Valid reset signal

Reset signal	ECRC Check Enable	ECRC Generation Enable
RST_LOAD_B	—	—
RST_RSM_B	✓	✓
RST_CFG_B	—	—

56.3.2.42 PCI_RC_HLOG0 : Header Log Register 0

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x611C

Bit position:	31																												0
Bit field:	HTLPE0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	HTLPE0[31:0]	Header of TLP associated with error 0 For the first detected Uncorrectable Error, indicates the 1st DW of Header.	R

The PCI_RC_HLOG0 register indicates the header log.

56.3.2.43 PCI_RC_HLOG1 : Header Log Register 1

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6120

Bit position:	31																												0
Bit field:	HTLPE1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	HTLPE1[31:0]	Header of TLP associated with error 1 For the first detected Uncorrectable Error, indicates the 2nd DW of Header.	R

The PCI_RC_HLOG1 register indicates the header log.

56.3.2.44 PCI_RC_HLOG2 : Header Log Register 2

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6124

Bit position:	31																												0
Bit field:	HTLPE2[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	HTLPE2[31:0]	Header of TLP associated with error 2 For the first detected Uncorrectable Error, indicates the 3rd DW of Header.	R

The PCI_RC_HLOG2 register indicates the header log.

56.3.2.45 PCI_RC_HLOG3 : Header Log Register 3

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6128

Bit position:	31																													0
Bit field:	HTLPE3[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	HTLPE3[31:0]	Header of TLP associated with error 3 For the first detected Uncorrectable Error, indicates the 4th DW of Header.	R

The PCI_RC_HLOG3 register indicates the header log.

56.3.2.46 PCI_RC_ROOTEC : Root Error Command

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x612C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	FERE	NFERE	CERE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CERE	Correctable Error Reporting Enable This bit enables an interrupt to be generated when ERR_COR occurs in any of the layers below the root port.	R/W
1	NFERE	Non-Fatal Error Reporting Enable This bit enables an interrupt to be generated when NON_ERR_FATAL occurs in any of the layers below the root port.	R/W
2	FERE	Fatal Error Reporting Enable This bit enables an interrupt to be generated when ERR_FATAL occurs in any of the layers below the root port.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Table 56.56 Valid reset signal

Reset signal	Fatal Error Reporting Enable	Non-Fatal Error Reporting Enable	Correctable Error Reporting Enable
RST_LOAD_B	—	—	—
RST_RSM_B	—	—	—
RST_CFG_B	✓	✓	✓

56.3.2.47 PCI_RC_ROOTES : Root Error Status

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6130

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	FEMR	NFEMR	FUF	MERRFATALNONFATALR	ERRFATALNONFATALR	MERRCORR	ERRCORR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ERRCORR	ERR_COR Received Indicates that ERR_COR was received with the ERR_COR Received bit cleared.	R/W
1	MERRCORR	Multiple ERR_COR Received It means that another ERR_COR was received while the ERR_COR Received bit was already set.	R/W
2	ERRFATALNONFATALR	ERR_FATALNONFATAL Received Indicates that ERR_FATAL or ERR_NON_FATAL was received with the ERR_FATAL/NONFATAL Received bit cleared.	R/W
3	MERRFATALNONFATALR	Multiple ERR_FATALNONFATAL Received This means that ERR_FATAL or ERR_NON_FATAL was received while the ERR_FATAL/NONFATAL Received bit was already set.	R/W
4	FUF	First Uncorrectable Fatal Set when the first uncorrectable error is received and the error was ERR_FATAL.	R/W
5	NFEMR	Non-Fatal Error Message Received Set when one or more ERR_NON_FATAL is received.	R/W
6	FEMR	Fatal Error Message Received Set when one or more ERR_FATAL is received.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

Table 56.57 Valid reset signal

Reset signal	Fatal Error Message Received	Non-Fatal Error Message Received	First Uncorrectable Fatal	Multiple ERR_FATAL/NONFATAL Received	ERR_FATAL/NONFATAL Received	Multiple ERR_COR Received	ERR_COR Received
RST_LOAD_B	—	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓	✓
RST_CFG_B	—	—	—	—	—	—	—

56.3.2.48 PCI_RC_ERRSI : Error Source Identification Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6134

Bit position:	31	16	15	0
Bit field:	ERRFATALNONFATALSI[15:0]		ERRCORSI[15:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
15:0	ERRCORSI[15:0]	ERR_COR Source Identification When the ERR_COR Received bit in the Root Error Status register is not set and ERR_COR is received, the error requester ID is captured.	R
31:16	ERRFATALNONFATALSI[15:0]	ERR_FATALNONFATAL Source Identification When ERR_FATAL or ERR_NONFATAL is received when the ERR_FATAL/NONFATAL Received bit of the Root Error Status register is not set, the requester ID of the error is captured.	R

Table 56.58 Valid reset signal

Reset signal	ERR_FATAL/NONFATAL Source Identification	ERR_COR Source Identification
RST_LOAD_B	—	—
RST_RSM_B	✓	✓
RST_CFG_B	—	—

56.3.2.49 PCI_RC_DEVSNEXTC : Device Serial Number Extended Capability Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6150

Bit position:	31											20	19			16	15													0
Bit field:	NCO[11:0]												CV[3:0]				PCIEECID[15:0]													
Value after reset:	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
15:0	PCIEECID[15:0]	PCI Express Extended Capability ID Indicates the Device Serial Number Extended Capability. Default: 0x0003	R
19:16	CV[3:0]	Capability Version Indicates the version of the Capability Structure. Default: 0x1	Hwinit
31:20	NCO[11:0]	Next Capability Offset Indicates the starting address of the Secondary PCI Express Extended Capability Header.	R

The PCI_RC_DEVSNEXTC register specifies the device serial number extended capability.

Table 56.59 Valid reset signal

Reset signal	Next Capability Offset	Capability Version
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—

56.3.2.50 PCI_RC_SNL : Serial Number Register (Lower DW)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6154

Bit position:	31																													0
Bit field:	PCIEDSN_LDW[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PCIEDSN_LDW[31:0]	PCI Express Device Serial Number (Lower DW) The read value is undefined The lower 32 bits of the IEEE standard 64-bit unique ID (EUI-64). EUI-64 consists of a 24-bit company ID and a 40-bit vendor-defined extension.	Hwinit

The PCI_RC_SNL register specifies the serial number of the device.

Table 56.60 Valid reset signal

Reset signal	PCI Express Device Serial Number (Lower DW)
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—

56.3.2.51 PCI_RC_SNU : Serial Number Register (Upper DW)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6158

Bit position:	31																												0
Bit field:	PCIEDSN_UDW[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PCIEDSN_UDW[31:0]	PCI Express Device Serial Number (Upper DW) The read value is undefined The Upper 32 bits of the IEEE standard 64-bit unique ID (EUI-64). EUI-64 consists of a 24-bit company ID and a 40-bit vendor-defined extension.	Hwinit

The PCI_RC_SNU register specifies the serial number of the device.

Table 56.61 Valid reset signal

Reset signal	PCI Express Device Serial Number (Upper DW)
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—

56.3.2.52 PCI_RC_SPEECH : Secondary PCI Express Extended Capability Header

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x61B0

Bit position:	31												20	19			16	15											0
Bit field:	NCO[11:0]													CV[3:0]				PCIEECID[15:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1

Bit	Symbol	Function	R/W
15:0	PCIEECID[15:0]	PCI Express Extended Capability ID Indicates the Secondary PCI Express Extended Capability Header. Default: 0x0019	R

Bit	Symbol	Function	R/W
19:16	CV[3:0]	Capability Version The read value is undefined Indicates the version of the Capability Structure. Default: 0x1	Hwinit
31:20	NCO[11:0]	Next Capability Offset Indicates that this Capability List is the final List. 0x000 fixed	R

The PCI_RC_SPEECH register specifies the Secondary PCI Express Extended Capability Header.

Table 56.62 Valid reset signal

Reset signal	Capability Version
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—

56.3.2.53 PCI_RC_LINC3 : Link Control 3 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x61B4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ELSKPOSGV[6:0]							—	—	—	—	—	—	—	LERIE	PE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PE	Perform Equalization When this bit is 1 and a 1 is written to the Retrain Link bit with the Target Link Speed field set to 8.0 GT/s or higher, the Downstream Port must perform Link Equalization. This bit is RW for Downstream Ports and for Upstream Ports when Crosslink Supported is 1. This bit is not applicable and is RsvdP for Upstream Ports when the Crosslink Supported bit is 0. The default value is 0. RC: Default 0 EP: Fixed to 0 because Crosslink is not supported	R/W
1	LERIE	Link Equalization Request Interrupt Enable When Set, this bit enables the generation of an interrupt to indicate that the Link Equalization 8.0 GT/s Request bit or the Link Equalization Request 16.0 GT/s bit has been set. This bit is RW for Downstream Ports and for Upstream Ports when Crosslink Supported is 1. This bit is not applicable and is RsvdP for Upstream Ports when the Crosslink Supported bit is 0. The default value for this bit is 0. RC: Default 0 EP: Fixed to 0 because Crosslink is not supported	R/W
8:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:9	ELSKPOSGV[6:0]	Enable Lower SKP OS Generation Vector When the Link is in L0 and the bit in this field corresponding to the current Link speed is Set, SKP Ordered Sets are scheduled at the rate defined for SRNS, overriding the rate required based on the clock tolerance architecture. Bit definitions within this field are: Bit 0: 2.5 GT/s Bit 1: 5.0 GT/s Bit 2: 8.0 GT/s Bit 3: 16.0 GT/s (prohibited) Bits 6 to 4: RsvdP Bits in this field are RW if the corresponding bit in the Lower SKP OS Generation Supported Speeds Vector is Set, otherwise they are permitted to be hardwired to 0. Behavior is undefined if a bit is Set in this field and the corresponding bit in the Lower SKP OS Generation Supported Speeds Vector is not Set. The default value of this field is 0x00.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Table 56.63 Valid reset signal

Reset signal	Enable Lower SKP OS Generation Vector	Link Equalization Request Interrupt Enable	Perform Equalization
RST_LOAD_B	—	—	—
RST_RSM_B	—	—	—
RST_CFG_B	✓	✓	✓

56.3.2.54 PCI_RC_LESTA : Lane Error Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x61B8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	LESB[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	LESB[3:0]	Lane Error Status Bits Each bit indicates if the corresponding Lane detected a Lane-based error. A value of 1 indicates that a Lane based-error was detected on the corresponding Lane Number. The default value of each bit is 0. For Links that are narrower than 32 bits, the unused upper bits [31: Bits Maximum Link Width] are RsvdZ. Default 0	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Table 56.64 Valid reset signal

Reset signal	Lane Error Status Bits
RST_LOAD_B	—
RST_RSM_B	✓
RST_CFG_B	—

56.3.2.55 PCI_RC_LEQCTL : Lane Equalization Control Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x61BC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	UPRPH1[2:0]			UPTP1[3:0]				—	DPRPH1[2:0]			DPTP1[3:0]			
Value after reset:	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	UPRPH0[2:0]			UPTP0[3:0]				—	DPRPH0[2:0]			DPTP0[3:0]			
Value after reset:	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	DPTP0[3:0]	Downstream Port 8.0 GTs Transmitter Preset Lane0 The read value is undefined. Transmitter Presetused for equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0, this field is RsvdP. Otherwise, this field is Hwinit. The default value is 0xF. RC: Default 0xF EP: Fixed to 0 because Crosslink is not supported	Hwinit
6:4	DPRPH0[2:0]	Downstream Port 8.0 GTs Receiver Preset Hint Lane0 The read value is undefined. Receiver Preset Hint may be used as a hint for receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0, this field is RsvdP. Otherwise, this field is Hwinit. The default value is 0x7. RC: Default 0x7 EP: Fixed to 0 because Crosslink is not supported	Hwinit
7	—	This bit is read as 0.	R
11:8	UPTP0[3:0]	Upstream Port 8.0 GTs Transmitter Preset Lane0 The read value is undefined. Field contains the Transmit Preset value sent or received during Link Equalization. Field usage varies as follows: A: Downstream Port Field contains the value sent on the associated Lane during Link Equalization. Field is Hwinit. B: Upstream Port, Crosslink Supported = 0 Field is intended for debug and diagnostics. It contains the value captured from the associated Lane during Link Equalization. Field is RO. When crosslinks are supported, case C (below) applies and this captured information is not visible to software. Vendors are encouraged to provide an alternate mechanism to obtain this information. C: Upstream Port, Crosslink Supported = 1 Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A (above) applies. Field is Hwinit. The default value is 0xF. RC: Default 0xF EP: Input signal	Hwinit

Bit	Symbol	Function	R/W
14:12	UPRPH0[2:0]	Upstream Port 8.0 GTs Receiver Preset Hint Lane0 The read value is undefined. Field contains the Receiver Preset Hint value sent or received during Link Equalization. Field usage varies as follows: A: Downstream Port Field contains the value sent on the associated Lane during Link Equalization. Field is Hwinit. B: Upstream Port, Crosslink Supported = 0 Field is intended for debug and diagnostics. It contains the value captured from the associated Lane during Link Equalization. Field is RO. When crosslinks are supported, case C (below) applies and this captured information is not visible to software. Vendors are encouraged to provide an alternate mechanism to obtain this information. C: Upstream Port, Crosslink Supported = 1 Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A (above) applies. Field is Hwinit. The default value is 0x7. RC: Default 0x7 EP: Input signal	Hwinit
15	—	This bit is read as 0.	R
19:16	DPTP1[3:0]*1	Downstream Port 8.0 GTs Transmitter Preset Lane1 The read value is undefined. Transmitter Presetused for equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0, this field is RsvdP. Otherwise, this field is Hwinit. The default value is 0xF. RC: Default 0xF EP: Fixed to 0 because Crosslink is not supported	Hwinit
22:20	DPRPH1[2:0]*1	Downstream Port 8.0 GTs Receiver Preset Hint Lane1 The read value is undefined. Receiver Preset Hint may be used as a hint for receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0, this field is RsvdP. Otherwise, this field is Hwinit. The default value is 0x7. RC: Default 0x7 EP: Fixed to 0 because Crosslink is not supported	Hwinit
23	—	This bit is read as 0.	R
27:24	UPTP1[3:0]*1	Upstream Port 8.0 GTs Transmitter Preset Lane1 The read value is undefined. Field contains the Transmit Preset value sent or received during Link Equalization. Field usage varies as follows: A: Downstream Port Field contains the value sent on the associated Lane during Link Equalization. Field is Hwinit. B: Upstream Port, Crosslink Supported = 0 Field is intended for debug and diagnostics. It contains the value captured from the associated Lane during Link Equalization. Field is RO. When crosslinks are supported, case C (below) applies and this captured information is not visible to software. Vendors are encouraged to provide an alternate mechanism to obtain this information. C: Upstream Port, Crosslink Supported = 1 Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A (above) applies. Field is Hwinit. The default value is 0xF. RC: Default 0xF EP: Input signal	Hwinit

Bit	Symbol	Function	R/W
30:28	UPRPH1[2:0]*1	Upstream Port 8.0 GTs Receiver Preset Hint Lane1 The read value is undefined. Field contains the Receiver Preset Hint value sent or received during Link Equalization. Field usage varies as follows: A: Downstream Port Field contains the value sent on the associated Lane during Link Equalization. Field is Hwinit. B: Upstream Port, Crosslink Supported = 0 Field is intended for debug and diagnostics. It contains the value captured from the associated Lane during Link Equalization. Field is RO. When crosslinks are supported, case C (below) applies and this captured information is not visible to software. Vendors are encouraged to provide an alternate mechanism to obtain this information. C: Upstream Port, Crosslink Supported = 1 Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A (above) applies. Field is Hwinit. The default value is 0x7. RC: Default 0x7 EP: Input signal	Hwinit
31	—	This bit is read as 0.	R

Note 1. Not available for m = 1.

Table 56.65 Valid reset signal

Reset signal	Upstream Port 8.0 GT/s Receiver Preset Hint	Upstream Port 8.0 GT/s Transmitter Preset	Downstream Port 8.0 GT/s Receiver Preset Hint	Downstream Port 8.0 GT/s Transmitter Preset	Upstream Port 8.0 GT/s Receiver Preset Hint	Upstream Port 8.0 GT/s Transmitter Preset	Downstream Port 8.0 GT/s Receiver Preset Hint	Downstream Port 8.0 GT/s Transmitter Preset
RST_LOAD_B	✓	✓	✓	✓	✓	✓	✓	✓
RST_RSM_B	—	—	—	—	—	—	—	—
RST_CFG_B	—	—	—	—	—	—	—	—

56.4 Register Descriptions (Endpoint Mode)

The registers in this section are availabe when the MODE_PORT bit in PCIE_MODE register is set to 0.

The registers are implemented in both ch0 and ch1, but the registers related to lane 1 are reserved for ch1.

Registers can be accessed from the AXI and PCIe. The attributes change depending on the access source. R/W column shows the attribute from AXI. When attribute is different between AXI and PCIe, note is added. When attribute is same, no note is added.

56.4.1 AXI Bridge Register

56.4.1.1 PCI_EP_REQDATAn : Request Data Register n (n = 0 to 2)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0080 + 0x4 × n

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Request Data Issue various requests to PCIe See Table 56.66.	R/W*1

Note 1. The access type from PCIe is read-only.

The PCI_EP_REQDATAn register issues various requests.

Table 56.66 Valid reset signal

Transaction type	Request Data Register 0	Request Data Register 1	Request Data Register 2
Zero-Length Read Request	Invalid	Invalid	Invalid
Message Request	3rd Header	4th Header	Invalid
Message Request with data payload	3rd Header	4th Header	Message data

Note: The bits should be set to 0 for the requests indicated as "Invalid".

56.4.1.2 PCI_EP_REQRCVDAT : Request Receive Data Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x008C

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Request Receive Data After issuing a read request, the data read on reception of the completion response are set in these bits. However, these bits are invalid for Zero-Lentgh Read and any kind of various write requests.	R

The PCI_EP_REQRCVDAT register indicates the data read on reception of the completion response after issuing a read request.

56.4.1.3 PCI_EP_REQADR1 : Request Address Register 1

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0090

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Request Address Set the Address, etc. when issuing a Request. See Table 56.67.	R/W*1

Note 1. The access type from PCIe is read-only.

The PCI_EP_REQADR1 register issues requests.

Table 56.67 Valid reset signal (1 of 2)

Request Address Register 1	[31:27]	[26:24]	[23:19]	[18:16]	[15:12]	[11:8]	[7:2]	[1:0]
Zero-Length Read Request	Address							Reserved

Table 56.67 Valid reset signal (2 of 2)

Request Address Register 1	[31:27]	[26:24]	[23:19]	[18:16]	[15:12]	[11:8]	[7:2]	[1:0]
Message Request	Reserved	Routing type	Reserved	Reserved	Reserved	Reserved	Message code	
Message Request with data payload	Reserved	Routing type	Reserved	Reserved	Reserved	Reserved	Message code	

Note: The bits should be set to 0 for the requests indicated as "Reserved".

56.4.1.4 PCI_EP_REQADR2 : Request Address Register 2

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0094

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Request Address63:32 See Table 56.68.	R/W*1

Note 1. The access type from PCIe is read-only.

The PCI_EP_REQADR2 register issues requests.

Table 56.68 Valid reset signal

Transaction type	[31:0]
Zero-Length Read Request	Address
Message Request	Invalid
Message Request with data payload	Invalid

Note: The bits should be set to 0 for the requests indicated as "Invalid".

56.4.1.5 PCI_EP_REQBE : Request Byte Enable Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0098

Bit position:	31																											3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RBE[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
3:0	RBE[3:0]	Request Byte Enable Specify Byte Enable when issuing Cfg Request as required. Normally use 0xF. 0: Byte Enable enabled 1: Byte Enable disabled	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_REQBE register specifies the first byte enable bit within the TLP header (1st DW byte) when issuing a request to PCIe.

Table 56.69 Valid reset signal

Transaction type	RBE[3:0]
Zero-Length Read Request	0x0
Message Request	Invalid (0xF)
Message Request with data payload	Invalid (0xF)

56.4.1.6 PCI_EP_REQISS : Request Issue Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x009C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	RR	MOR_CD_PERR	MOR_CH_PERR	MOR_EP_ERR	MOR_STATUS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	FUNC[2:0]			TR_TYPE[3:0]				—	—	—	—	—	—	—	RI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RI	Request Issue 0: Write: No operation. Read: Request can be accepted. (Indicates that the processing of the issued Request has ended.) 1: Write: Request issuance. Read: Processing Request. (Indicates that the issued Request is being processed.)	R/W*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W
11:8	TR_TYPE[3:0]	Set the type of Request. See Table 56.70.	R/W*1
14:12	FUNC[2:0]	Set the function of Request.	R/W*1
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	MOR_STATUS[2:0]	These bits retain the MOR Status of Completion TLP for Non-Posted requests issued. It is not updated at the time of Posted request. 0 0 0: Successful Completion (SC) 0 0 1: Unsupported Request (UR) 0 1 0: Configuration Request Retry Status (CRS) (not supported) 0 1 1: Completion Timeout 1 0 0: Completer Abort (CA) 1 0 1: Unexpected Completion and mismatched type (Lock Completion responds to non-Lock Request) 1 1 0: Reserved 1 1 1: Overrun Completion length	R
19	MOR_EP_ERR	It is set to 1 when a Poisoned Completion TLP for a Non-Posted request issued by this register is received. Normally not used. It is not updated at the time of Posted request.	R
20	MOR_CH_PERR	It is set to 1 when a header error occurs in the Completion TLP for the Non-Posted request issued by this register. Normally not used. It is not updated at the time of Posted request.	R

Bit	Symbol	Function	R/W
21	MOR_CD_PERR	It is set to 1 when a data error occurs in the Completion TLP for the Non-Posted request issued by this register. Normally not used. It is not updated at the time of Posted request.	R
22	RR	Request Rejection Indicates that the stopped or hibernated state of the PCI Express transmit (TX) side was detected and the process was forcibly terminated. 0: Normal state (Request issued) 1: Rejection	R
31:23	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_REQISS register issues a request to PCIe.

Table 56.70 Valid reset signal

Transaction type	TR_TYPE[11:8]	Posted/Non-posted	Device type	
			Root complex	Endpoint
Zero-Length Read Request	0x0	Non-posted	Issuable	Issuable
Configuration Read Type0	0x4	Non-posted	Issuable	Issuing prohibited
Configuration Write Type0	0x5	Non-posted	Issuable	Issuing prohibited
Configuration Read Type1	0x6	Non-posted	Issuable	Issuing prohibited
Configuration Write Type1	0x7	Non-posted	Issuable	Issuing prohibited
Message Request	0x8	Posted	Issuable	Issuable
Message Request with data payload	0x9	Posted	Issuable	Issuable
	Others	—	Issuing prohibited	Issuing prohibited

56.4.1.7 PCI_EP_INTXOUTS : PCI INTx Out Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0118

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	INTDS	INTCS	INTBS	INTAS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INTAS	INTA Status It is set by sending an Assert INTA Message and cleared by sending a Deassert INTA Message. 0: Deassert 1: Assert	R
1	INTBS	INTB Status It is set by sending an Assert INTB Message and cleared by sending a Deassert INTB Message. 0: Deassert 1: Assert	R

Bit	Symbol	Function	R/W
2	INTCS	INTC Status It is set by sending an Assert INTC Message and cleared by sending a Deassert INTC Message. 0: Deassert 1: Assert	R
3	INTDS	INTD Status It is set by sending an Assert INTD Message and cleared by sending a Deassert INTD Message. 0: Deassert 1: Assert	R
31:4	—	These bits are read as 0.	R

The PCI_EP_INTXOUTS register can confirm the PCI INTx status issued by the unit with the interrupt input signal (INTX_EP_F*: active high). In addition, the value of this register is not reflected if the INTx message is asserted or deasserted by issuing a special request (prohibited operation). The PCI_EP_INTXOUTS register is only valid in endpoint mode.

56.4.1.8 PCI_EP_MSGRCVIE : Message Receive Interrupt Enable Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0120

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	MRIE	—	—	—	—	PMASNRI	PMPMERIE	PMETORIE	PMETOARIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	PMETOARIE	PME_TO_Ack Receive Interrupt Enable Enable control of MSG_INT assertion by PME_TO_Ack reception 0: Disable 1: Enable	R/W
17	PMETORIE	PME_Turn_Off Receive Interrupt Enable Enable control of MSG_INT assertion by PME_Turn_Off reception 0: Disable 1: Enable	R/W
18	PMPMERIE	PM_PME Receive Interrupt Enable Enable control of MSG_INT assert by PM_PME reception 0: Disable 1: Enable	R/W
19	PMASNRI	PM_Active_State_Nak Receive Interrupt Enable control of MSG_INT assertion by PM_Active_State_Nak reception 0: Disable 1: Enable	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	MRIE	Message Receive Interrupt Enable Enable control of MSG_INT assert by message reception 0: enable 1: disable	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_MSGRCVIE register controls enabling of MSG_INT in response to the reception of message requests other than INTx and error-related messages.

56.4.1.9 PCI_EP_MSGRCVIS : Message Receive Interrupt Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0124

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	MRI	—	—	—	—	PMASNR	PMPMERI	PMETORI	PMETOARI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	PMETOARI	PME_TO_Ack Receive Interrupt Set when receiving a PM_TO_Ack message	R/W
17	PMETORI	PME_Turn_Off Receive Interrupt Disabled due to RC	R/W
18	PMPMERI	PM_PME Receive Interrupt Set when receiving a PM_PME message	R/W
19	PMASNR	PM_Active_State_Nak Receive Disabled due to RC	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	MRI	Message Receive Interrupt Set when receiving a message (does not depend on the type of message)	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_MSGRCVIS register is a status register that indicates the reception of message requests other than INTx and error-related messages. The value of the PCI_EP_MSGRCVIS register is reflected in MSG_INT. Only the message code is used to decide the message type, and the corresponding message is assumed to have been received while the validity of the routing and the validity of the Msg/MsgD selection are not verified.

56.4.1.10 PCI_EP_MSGCODE : Message Code Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0130

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MC[7:0]								R[2:0]			—	—	—	—	MP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

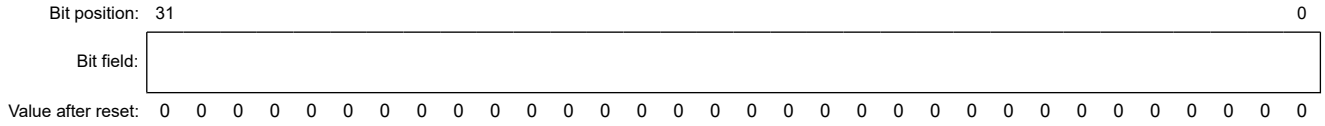
Bit	Symbol	Function	R/W
0	MP	Message Payload Stores the presence or absence of the data payload of the last received Message. When Power Management Message (PME_TO_Ack Message, PME_Turn_Off Message, PM_PME Message, PM_Active_State_Nak Message) is received, this register is not written. 0: Msg (Without Payload) 1: MsgD (with Payload)	R
4:1	—	These bits are read as 0.	R
7:5	R[2:0]	Routing Stores the Routing of the last received Message.	R
15:8	MC[7:0]	Message Code Stores the Code of the last received Message.	R
31:16	—	These bits are read as 0.	R

The PCI_EP_MSGCODE register stores the code and routing of the last received message.

56.4.1.11 PCI_EP_MSGDATA : Message Data Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0134



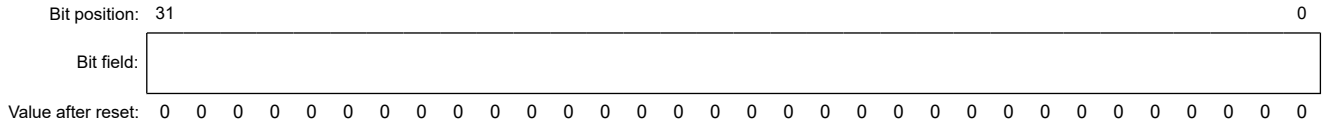
Bit	Symbol	Function	R/W
31:0	n/a	Message Data Stores the 1DW data of the last received Message. It is updated only when MsgD (with Data) is received, and the previous value is retained when Msg (without Data) is received. When Power Management Message (PME_TO_Ack Message, PME_Turn_Off Message, PM_PME Message, PM_Active_State_Nak Message) is received, this register is not written.	R

The PCI_EP_MSGDATA register stores the data of the last received message.

56.4.1.12 PCI_EP_MSGH3DW : Message Header 3rdDW Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0138



Bit	Symbol	Function	R/W
31:0	n/a	Message Header 3rdDW Stores the Header (3rdDW) of the last received Message.	R

The PCI_EP_MSGH3DW register stores the header (the 3rd DW) of the last received message.

56.4.1.13 PCI_EP_MSGH4DW : Message Header 4thDW Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x013C

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Message Header 4thDW Registers Stores the Header (4thDW) of the last received Message.	R

The PCI_EP_MSGH4DW register stores the header (the 4th DW) of the last received message.

56.4.1.14 PCI_EP_INTTABLE : Interrupt Table Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0140

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMA_INT[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	AXI_ERR_IN	PCIE_EVT_INT	MSG_INT	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0.	R
8	MSG_INT	Message interrupt signal monitor	R
9	PCIE_EVT_INT	Event interrupt signal monitor	R
10	AXI_ERR_IN	Error interrupt signal monitor	R
23:11	—	These bits are read as 0.	R
31:24	DMA_INT[7:0]	DMA_INT interrupt signal monitor	R

The PCI_EP_INTTABLE register is an index of interrupt factors. The interrupt signal (active high) status of each category can be monitored in a list.

56.4.1.15 PCI_EP_PEIE0 : PCIe Event Interrupt Enable 0 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	UI_LINK_WIDTH_CHANGE_DONE_EN	UI_LINK_SPEED_CHANGE_DONE_EN	RDEN	—	—	AXI_PERR_EN	CA_EN	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BME_PERR_EN	—	—	RX_DLLP_PM_ENTER_L23_EN	—	ASPML1REN	DLUDEN	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	—	These bits are read as 0. The write value should be 0.	R/W
9	DLUDEN	DL_UpDown EN Interrupt enable on DL state change 0: Disable 1: Enable	R/W
10	ASPML1REN	ASPM L1 Rejected interrupt enable 0: Disable 1: Enable	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	RX_DLLP_PM_ENTER_L23_EN	RX_DLLP_PM_ENTER_L23 interrupt enable 0: Disable 1: Enable	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	BME_PERR_EN	Enable BME Parity Error interrupt. (Parity Error in AXI bridge internal RAM) 0: Disable 1: Enable	R/W
23:16	—	These bits are read as 0. The write value should be 0.	R/W
24	CA_EN	CA (Completer Abort) interrupt enable 0: Disable 1: Enable	R/W
25	AXI_PERR_EN	Enable AXIM RAM parity error interrupt 0: Disable 1: Enable	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	RDEN	Request complete interrupt enable 0: Disable 1: Enable	R/W
29	UI_LINK_SPEED_CHANGE_DONE_EN	Speed change operation completion interrupt enable 0: Disable 1: Enable	R/W
30	UI_LINK_WIDTH_CHANGE_DONE_EN	Up/Down Configure operation complete interrupt enable 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_EP_PEIE0 register enables interrupts of each PCI Express event factors. The PCI_EP_PEIE0 register enables writing to the PCIe Event Interrupt Status 0 Register (address: <PCI_S0_REG_base> + 0x0204).

56.4.1.16 PCI_EP_PEIS0 : PCIe Event Interrupt Status 0 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0204

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	UI_LINK_WIDTH_CHANGE_DONE	UI_LINK_SPEED_CHANGE_DONE	RD	—	—	AXIM_PERR	CA	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BME_PERR	—	—	RX_DLLP_PM_ENTER_L23	—	ASPML1	DLUDEN	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	—	These bits are read as 0. The write value should be 0.	R/W
9	DLUDEN	DL_UpDown EN Set to 1 on transition from DL_Down state to DL_Up state, or DL_Up state to DL_Down state. Check the DL_Down/DL_Up status with PCIe Core Status 1 Registers (Offset: 0x408).	R/W
10	ASPML1	ASPM L1 It is recommended not to use (permit) in Endpoint mode.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	RX_DLLP_PM_ENTER_L23	Indicates transition to L2/L3 State in Power Management control.	R/W
14:13	—	These bits are read as 0. The write value should be 0.	
15	BME_PERR	Indicates that a parity error has occurred in the BME RAM.	
23:16	—	These bits are read as 0. The write value should be 0.	R/W
24	CA	Indicates that the device has responded with CA (Completer Abort).	R/W
25	AXIM_PERR	Indicates that a parity error has occurred in the AXIM RAM. (Parity Error in AXI bridge internal RAM)	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	RD	Request Done For requests submitted in the Request Issue Registers (Offset: 0x9C): Non-Posted: Indicates that Completion has been received. Posted: Indicates that the request has finished being submitted.	R/W
29	UI_LINK_SPEED_CHANGE_DONE	Indicates completion of Speed Change operation.	R/W
30	UI_LINK_WIDTH_CHANGE_DONE	Indicates completion of Up/DownConfigure operation.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_EP_PEIS0 register is a status register to indicate interrupts of each PCI Express event factors. Set to 1 by the factor in the table. After checking the factor, write 1 to the corresponding bit to clear it.

56.4.1.17 PCI_EP_PEIE1 : PCIe Event Interrupt Enable 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0208

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXB_PARITY_ERR_EN	ERR_RPC_REPLAYFIFO_PERR_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ERR_REPLAY_UPPER_CORRECTABLE_ERROR_EN	ERR_REPLAY_LOWER_CORRECTABLE_ERROR_EN	—	—	—	—	—	—	ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR_EN	ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR_EN	Enable ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR interrupts. Interrupt notification enable setting when an error of ECC 2-bit or more (Uncorrectable Error) occurs in the lower 64-bit data bus of the Replay FIFO installed in the Data Link Layer 0: Disable 1: Enable	R/W
1	ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR_EN	Enable ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR interrupts. Interrupt notification enable setting when an error of ECC 2-bit or more (Uncorrectable Error) occurs in the upper 64-bit data bus of the Replay FIFO installed in the Data Link Layer 0: Disable 1: Enable	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	ERR_REPLAY_LOWER_CORRECTABLE_ERROR_EN	Enable ERR_REPLAY_LOWER_CORRECTABLE_ERROR interrupts. Interrupt notification enable setting when an ECC 1-bit error (Correctable Error) occurs in the lower 64-bit data bus of the Replay FIFO installed in the Data Link Layer 0: Disable 1: Enable	R/W
9	ERR_REPLAY_UPPER_CORRECTABLE_ERROR_EN	Enable ERR_REPLAY_UPPER_CORRECTABLE_ERROR interrupts. Interrupt notification enable setting when an ECC 1-bit error (Correctable Error) occurs in the upper 64-bit data bus of the Replay FIFO installed in the Data Link Layer 0: Disable 1: Enable	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	ERR_RPC_REPLAYFIFO_PERR_EN	Enable ERR_RPC_REPLAYFIFO_PERR interrupts. Parity error interrupt notification enable setting for Replay FIFO installed in Data Link Layer 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
17	TXB_PARITY_ERR_EN	Enable the TXB_PARITY_ERR interrupt. Parity error interrupt notification enable setting for TX Buffer installed in Transaction Layer 0: Do not allow 1: Allow	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_PEIE1 register enables parity error and ECC error interrupts. When each bit is set to the valid setting, the value of each corresponding status bit of the PCIe Event Interrupt Status 1 register (address: <PCI_S0_REG_base> + 0x020C) becomes valid.

56.4.1.18 PCI_EP_PEIS1 : PCIe Event Interrupt Status 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x020C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXB_PARITY_ERR	ERR_RPC_REPLAYFIFO_PERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ERR_REPLAY_UPPER_CORRECTABLE_ERROR	ERR_REPLAY_LOWER_CORRECTABLE_ERROR	—	—	—	—	—	—	ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR	ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR	ERR_REPLAY_LOWER_UNCORRECTABLE_ERROR interrupt Interrupt notification when an ECC 2-bit or more error (Uncorrectable Error) occurs in the lower 64-bit data bus of the Replay FIFO installed in the Data Link Layer	R/W
1	ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR	ERR_REPLAY_UPPER_UNCORRECTABLE_ERROR interrupt Interrupt notification when an ECC 2-bit or more error (Uncorrectable Error) occurs in the upper 64-bit data bus of the Replay FIFO installed in the Data Link Layer	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	ERR_REPLAY_LOWER_CORRECTABLE_ERROR	ERR_REPLAY_LOWER_CORRECTABLE_ERROR interrupt Interrupt notification when an ECC 1-bit error (Correctable Error) occurs in the lower 64-bit data bus of the Replay FIFO installed in the Data Link Layer	R/W
9	ERR_REPLAY_UPPER_CORRECTABLE_ERROR	ERR_REPLAY_UPPER_CORRECTABLE_ERROR interrupt Interrupt notification when an ECC 1-bit error (Correctable Error) occurs in the upper 64-bit data bus of the Replay FIFO installed in the Data Link Layer	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	ERR_RPC_REPLAYFIFO_PERR	ERR_RPC_REPLAYFIFO_PERR interrupt Parity error interrupt notification of Replay FIFO installed in Data Link Layer	R/W
17	TXB_PARITY_ERR	TXB_PARITY_ERR interrupt Parity error interrupt notification of TX Buffer installed in Transaction Layer	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_PEIS1 register is a status register that indicates parity error and ECC error interrupts. After checking the factor, write 1 to the corresponding bit to clear it.

56.4.1.19 PCI_EP_AMEIE : AXI Master Error Interrupt Enable Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0210

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WMSTERRINTEN[3:0]				—	—	—	—	RMSTERRINTEN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	RMSTERRINTEN[3:0]	Read MSTERR INT Enable Each corresponding bit can be turned on/off individually. 0: Disable 1: Enable	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	WMSTERRINTEN[3:0]	Write MSTERR INT Enable Each corresponding bit can be turned on/off individually. 0: Disable 1: Enable	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_AMEIE register enables the AXI master error interrupt.

56.4.1.20 PCI_EP_AMEIS : AXI Master Error Interrupt Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0214

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	WERRID[3:0]				—	—	—	—	RERRID[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WMSTERRINT[3:0]				—	—	—	—	RMSTERRINT[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	RMSTERRINT[3:0]	Read MSTERR INT Indicates that an error was detected in the AXI Master Port. Only the first detected error is saved, and when bit [3:0] is cleared, a new error can be saved. Cleared by writing 1. Bit 3: Length Error When the length of the data received on the TER and the data channel do not match. Bit 2: ID Mismatch When the MARID and MRID values received on the data channel are different. Bit 1: When DECERR is received Bit 0: When SLVERR is received	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
11:8	WMSTERRINT[3:0]	Write MSTERR INT Indicates that an error was detected in the AXI Master Port. Only the first detected error is saved, and when bit [11:8] is cleared, a new error can be saved. Cleared by writing 1. Bit 11: Length Error When the length of the data sent by the TEF and the data channel do not match. Bit 10: ID Mismatch When the MBID value received on the MAWID response channel is different. Bit 9: When DECERR is received Bit 8: When SLVERR is received	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RERRID[3:0]	Read ERR ID Save the ID of the first DECERR/SLVERR received. A new error ID can be saved when bit [3:0] is cleared. 0x0: normal access Others: Error ID	R
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	WERRID[3:0]	Write ERR ID Save the ID of the first DECERR/SLVERR received. When bit [11:8] is cleared, a new error ID can be saved. 0x0: normal access Others: Error ID	R
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_AMEIS register indicates the AXI master error interrupt status.

56.4.1.21 PCI_EP_ASEIE1 : AXI Slave Error Interrupt Enable 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0220

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WSLVERRINTEN[3:0]				—	—	—	—	—	—	RSLVERRINTEN[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RSLVERRINTEN[1:0]	Read SLVERR INT EN Enable Read SLVERR INT. Each corresponding bit can be turned on/off individually. 0: Disable 1: Enable	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
11:8	WSLVERRINTEN[3:0]	Write SLVERR INT EN Enable Write SLVERR INT. Each corresponding bit can be turned on/off individually. 0: Disable 1: Enable	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_ASEIE1 register enables the AXI slave error interrupt.

56.4.1.22 PCI_EP_ASEIS1 : AXI Slave Error Interrupt Status 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0224

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WSLVERRINT[3:0]				—	—	—	—	—	—	RSLVERRINT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RSLVERRINT[1:0]	Read SLVERR INT Indicates that an unrecoverable error was detected in the AXI Slave Port. (Transaction response will be SLVERR.) Cleared by writing 1. Bit 1: Burst Type Disabled When SARBURST is 11b (undefined). When SARBURST is 10b (wrapping) and burst length is other than 2, 4, 8, or 16. Bit 0: Data Size Invalid When SARSIZE is (exceeding AXI Bus width/not supported). Each bit means: 0: No error detected 1: error detection	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
11:8	WSLVERRINT[3:0]	Write SLVERR INT Indicates that an unrecoverable error was detected in the AXI Slave Port. (Transaction response will be SLVERR.) Cleared by writing 1. Bit 11: Burst Length Error When the SAWLEN and the burst length of the data received on the data channel do not match. Bit 10: ID Mismatch When the SAWID and SWID values received on the data channel are different. Bit 9: Burst Type Disabled When SAWBURST is 11b (undefined). When SAWBURST is 10b (wrapping) and burst length is other than 2, 4, 8, or 16. Bit 8: Data Size Invalid When SAWSIZE is between 100b and 111b (Exceeding AXI Bus width/not supported). Each bit means: 0: No error detected 1: error detection	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_ASEIS1 register indicates the AXI slave error interrupt status.

56.4.1.23 PCI_EP_ASEIS3 : AXI Slave Error Interrupt Status 3 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0230

Bit position:	31	0
Bit field:		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	n/a	ERR ID Save the ID when the first error occurred in the AXI Slave Error Interrupt Status1 register (Offset 0x224). Only the first detected error is saved, and when bit[11:8] and bit[1:0] of the AXI Slave Error Interrupt Status1 register are all cleared, a new error ID can be saved.	R

The PCI_EP_ASEIS3 register indicates the AXI ID at the time of the first AXI slave error.

56.4.1.24 PCI_EP_PEIE2 : PCIe Event Interrupt Enable 2 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0240

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LTR_MECHANISM_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	D3_HOT_ERR_EN[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
15:8	D3_HOT_ERR_EN[7:0]	D3_hot_err_EN When Completion is in Pending state (ORT_TRANSACTION_PENDING = 1), transition to D3hot state (CFG_POWERSTATE! = 0) or receive PME_TURN_OFF Message, detect the state when D3_EVENT_ACK/TURN_OFF_EVENT_ACK is asserted Interrupt enable for D3_hot_err status detection Bit[15:10] are not used. Lower 2 bits corresponds to each function.	R/W
16	LTR_MECHANISM_EN	Interrupt enable for the ability to detect changes in CFG_LTR_MECHANISM	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_PEIE2 register enables interrupts of the various PCI Express event factors. The PCI_EP_PEIE2 register enables writing to the PCIe Event Interrupt Status 2 Register (PCI_EP_PEIS2).

56.4.1.25 PCI_EP_PEIS2 : PCIe Event Interrupt Status 2 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0244

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LTR_MECHANISM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	D3_HOT_ERR[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	D3_HOT_ERR[7:0]	D3_hot_err Status that detects and notif ies the state when D3_EVENT_ACK/ TURN_OFF_EVENT_ACK is asserted when transition to D3hot state (CFG_POWERSTATE! = 0) or PME_TURN_OFF Message is received in Completion Pending state (ORT_TRANSACTION_PENDING = 1) Bit[15:10] are not used. Lower 2 bit corresponds to each function. Cleared by writing 1.	R/W
16	LTR_MECHANISM	Status to detect and notify changes in CFG_LTR_MECHANISM Cleared by writing 1.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_PEIS2 register indicates the state of various PCI Express events.

56.4.1.26 PCI_EP_PERM : Permission Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PHY_REG_CLK_EN	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	PHY_REG_CLK_EN	Write enable for Physical Layer Control/Status Registers. 0: Disable 1: Enable	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

56.4.1.27 PCI_EP_RESET : Reset Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0310

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FTD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	RST_PREG_B	RST_OUT_B	RST_PS_B	RST_LOAD_B	RST_CFG_B	RST_RSM_B	RST_GP_B	RST_B
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
0	RST_B	Reset to PCI Express core part inside macro 0: reset 1: normal operation	R/W
1	RST_GP_B	Reset to the PCI Express core part (ACLK domain) inside the macro 0: reset 1: normal operation	R/W
2	RST_RSM_B	POWERGOOD reset of AUX Power (AUX not supported) Reset to Sticky register. Reset to PHY. 0: reset 1: normal operation	R/W
3	RST_CFG_B	Reset to Configuration Register 0: Reset 1: normal operation	R/W
4	RST_LOAD_B	Reset to Configuration Register Reset to bits not initialized by RST_CFG_B. 0: reset 1: normal operation	R/W
5	RST_PS_B	Reset to the PCI Express core part (PCLK domain) inside the macro 0: reset 1: normal operation	R/W
6	RST_OUT_B	RST_OUT_B output 0: reset 1: normal operation	R/W
7	RST_PREG_B	Not used in this macro. The setting value does not affect macro behavior. 0: reset 1: normal operation	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	FTD0	force to D0 After sending PME_TO_Ack, force PM Control to transition to D0 State. Auto-cleared when PM Control transitions to D0. This bit is normally prohibited as it can create inconsistencies with the power state of the entire system. 0: No operation 1: Transition to D0	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_RESET register resets the PCIe core. Supplied to the internal core. The write value is saved when accessed from the AXI, but a low-level pulse is generated when the PCIe writes 0. However, when the AXI has already written 0, the signal remains at the low level.

Writing 1 from the PCIe is ignored.

56.4.1.28 PCI_EP_MSET0 : Mode Set 0 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0314

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	AWPROT[2:0]			AWCACHE_L[3:0]				—	—	AWLOCK[1:0]		AWCACHE_D[3:0]			
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ARPROT[2:0]			—	—	—	—	—	—	ARLOCK[1:0]		ARCACHE[3:0]			
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ARCACHE[3:0]	Lock type for PCIe-to-AXI transactions. This bit provides information about the atomic nature of the transfer. 0 0: Normal access 0 1: Exclusive access 1 0: Locked access 1 1: Reserved	R/W
5:4	ARLOCK[1:0]	Lock type for PCIe-to-AXI transactions. This bit provides information about the atomic nature of the transfer. 0 0: Normal access 0 1: Exclusive access 1 0: Locked access 1 1: Reserved	R/W
11:6	—	These bits are read as 0. The write value should be 0.	R/W
14:12	ARPROT[2:0]	Sets the protection type for PCIe→AXI transactions. This bit indicates whether the transaction's protection level is Normal, Privileged, or Secure, and whether the transaction is a data or instruction access. [2] 1: instruction access / 0: data access [1] 1: non-secure access / 0: secure access [0] 1: privileged access / 0: normal access	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
19:16	AWCACHE_D[3:0]	Indicates the value of MAWCACHE[3:0] to be issued to AXI. This setting is output when issuing an AXI request other than the output condition of AWCACHE_L. *Recommended value is 0x1. [3] 1: write allocatable / 0: not write allocatable [2] 1: Read assignable / 0: Read assignable [1] 1: cacheable / 0: non-cacheable [0] 1: Bufferable / 0: Not Bufferable	R/W
21:20	AWLOCK[1:0]	Lock type for PCIe-to-AXI transactions. This signal provides information about the atomic nature of the transfer. 0 0: Normal access 0 1: Exclusive access 1 0: Locked access 1 1: Reserved	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
27:24	AWCACHE_L[3:0]	Indicates the value of MAWCACHE[3:0] to be issued to AXI. This setting is printed when issuing an AXI request containing the last byte. *Recommended value is 0x0. [3] 1: write allocatable / 0: not write allocatable [2] 1: Read assignable / 0: Read assignable [1] 1: cacheable / 0: non-cacheable [0] 1: Bufferable / 0: Not Bufferable	R/W
30:28	AWPROT[2:0]	Sets the protection type for PCIe→AXI transactions. This bit indicates whether the transaction's protection level is Normal, Privileged, or Secure, and whether the transaction is a data or instruction access. [2] 1: instruction access / 0: data access [1] 1: non-secure access / 0: secure access [0] 1: privileged access / 0: normal access	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_EP_MSET0 register sets AXI mode.

56.4.1.29 PCI_EP_MSET1 : Mode Set 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0318

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AXIMIW[3:0]				AXIMIR[3:0]				AXIMMB[3:0]				—	—	RAMPE	PCIERO
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	0

Bit	Symbol	Function	R/W
0	PCIERO	PCIe Request Order Issue a Read Request to PCIe from the same AXI master without waiting for Completion. Set to 1 if you want to strictly follow the order of Requests to the Completer. 0: Do not wait for Completion. 1: Wait for Completion.	R/W
1	RAMPE	RAM Parity Enable Sets whether or not to check the parity of the internal SRAM. The default value is Enable, but it is ignored by non-Parity macros. 0: RAM parity check disabled 1: RAM parity check enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	AXIMMB[3:0]	AXI Master Max Burst Sets the maximum burst length as an AXI Master operation.	R/W
11:8	AXIMIR[3:0]	AXI Max Issue Read Set the number of read issues that can be issued by AXI Master. Set within the range allowed by Interconnect. 0x0: possible number 1 0x1: possible number 2 ⋮ 0xF: Possible number 16	R/W
15:12	AXIMIW[3:0]	AXI Max Issue Write Set the number of writes that can be issued by AXI Master. Set within the range allowed by Interconnect. 0x0: possible number 1 0x1: possible number 2 ⋮ 0xF: Possible number 16	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_MSET1 register sets AXI mode.

56.4.1.30 PCI_EP_MSET3 : Mode Set 3 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0380

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ASPML1IT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ASPML1IT[7:0]	ASPM L1 Idle Time Sets the idle period for AXI transactions that the macro checks on ASPM L1 transitions. One of the conditions for ASPM L1 transition is that the idle period is confirmed for the number of cycles of the 8 bits set by this bit plus 0xFF to the lower 8 bits. 0x00: 256 [ACLK] 0x01: 512 [ACLK] ⋮ 0xFF: 65536 [ACLK]	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_MSET3 register outputs the setting value as the ASPM L1 Idle Time bit.

56.4.1.31 PCI_EP_MSET4 : Mode Set 4 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0384

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MODE_EQ_RX_EVAL_WAIT[19:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MODE_EQ_RX_EVAL_WAIT[19:0]				—	—	—	—	MODE_8GT_EQTS2	MODE_EQ_PHASE23_ENABLE1	MODE_EQ_AUTONOMOUS1	—	—	UI_EQ_REQTYPE[1:0]		UI_EQ_REQUEST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UI_EQ_REQUEST	Set when sending an Equalization Request with Quiesce Guarantee = 1 in the Recovery.RcvrCfg state. 1 set prohibited except when MODE_QUIESCE_GUARANTEE = 1. After setting 1, hold until confirmation of UI_EQ_DONE = 1. 0: Do not send Equalization Request 1: Send Equalization Request	R/W

Bit	Symbol	Function	R/W
2:1	UI_EQ_REQTYPE[1:0]	Operation specification when sending Equalization Request with Quiesce Guarantee = 1 in Recovery.RcvrCfg state 0 0: Equalization Request transmission with Equalization Request Data Rate = 0 (8G) 0 1: Send Equalization Request with Equalization Request Data Rate = 1 (16G) (Setting prohibited) 1 0: Operation of 0 + set Link Equalization Request 8.0 GT/s register 1 1: Set 1 operation + Link Equalization Request 16.0 GT/s register (setting prohibited)	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	MODE_EQ_AUTONOMOUS1	Fixed 0	R/W
6	MODE_EQ_PHASE23_ENABLE1	Fixed 0	R/W
7	MODE_8GT_EQTS2	8GT EQ TS2OS transmission enable/disable setting for Upstream Port Note: Not used with Downsream Port. (fixed to 0) 0: Do not send 1: Send	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
31:12	MODE_EQ_RX_EVAL_WAIT[19:0]	During RxEval (Downstream Port Phase3, Upstream Port Phase2), wait time setting until RxEval is executed when Block Alignment cannot be obtained. Set by the count number of PCLK. Note: It is necessary to set the register value (recommended: equivalent to 1 msec) before starting. Please set from the local CPU.	R/W

56.4.1.32 PCI_EP_MSET5 : Mode Set 5 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0388

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MODE_8GT_EQTS2_PRESET[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MODE_8GT_EQTS2_PRESET[15:0]	Transmitter Preset value of 8GT EQ TS2OS to transmit at Upstream Port (USP). No setting is required if 8GT_EQTS2 is not sent from USP (DSP Preset is not specified). Settings must be made in advance if necessary. 4-bit configuration for each lane. Gen3 supports up to x2 lane configurations. The correspondence between each bit and lane number is as follows. [3:0] Lane #0 [7:4] Lane #1 (Not used in case of ch1) [11:8] Lane #2 (Not used. Fixed to 0x0) [15:12] Lane #3 (Not used. Fixed to 0x0) The above lane number is the default lane number for circuit implementation. Note that this is not a negotiated lane number. Note: Not used with Downsream Port. (fixed to 0x0000_0000)	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

56.4.1.33 PCI_EP_MSTAT0 : Mode Status 0 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0390

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	RX_EQ_REQTYPE[1:0]		RX_EQ_REQUEST	UI_EQ_DONE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UI_EQ_DONE	Set Equalization Request with Quiesce Guarantee=1 in Recovery .RcvrCfg state when sending Related flow: "Set UI_EQ_REQUEST = 1 → Confirm UI_EQ_DONE = 1 → Clear UI_EQ_REQUEST = 0" 0: No Equalization Request sent 1: Sent Equalization Request	R
1	RX_EQ_REQUEST	Status display when receiving 8 consecutive TS2OS with Request Equalization bit (Symbol6, bit7) = 1 in Recovery.RcvrCfg state 0: No Equalization Request received 1: Equalization Request received	R
3:2	RX_EQ_REQTYPE[1:0]	Quiesce Guarantee and Equalization Request Data Rate bit display when receiv ing 8 consecutive TS2OS with Request Equalization bit (Symbol6, bit7) = 1 in Recovery.RcvrCfg state EP (USP) (1) When Request Equalization is received bit 0: Quiesce Guarantee (TS2OS, Symbol6, bit6) (= 1 fixed) bit 1: Equalization Request Data Rate (TS2OS, Symbol6, bit5) (2) When receiving EQTS2OS bit 0: 0 (fixed) bit 1: Equalization Request Data Rate (0: EQTS2, 1: 8GT EQTS2)	R
31:4	—	These bits are read as 0.	R

56.4.1.34 PCI_EP_PCMSET1 : PCIe Core Mode Set 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0400

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	ASPM_L1_INTERVAL_TIME[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MODE_LINK_UPCONFIGURE_CAPABILITY_DISABLED	—	MODE_TXSWING	—	MODE_SELECTABLE_DEEMPHASIS	—	—	—	—	—	—	MODE_PORT	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	MODE_PORT	Device type setting register. When the bit value of this register is fixed to the initial value 1, the setting of the MODE_PORT bit in PCIE_MODE register becomes valid. When the MODE_PORT bit in PCIE_MODE register is fixed to 1, the setting by this register bit becomes valid. 0: Endpoint 1: Root Complex	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	MODE_SELECTABLE_DEEMPHASIS	De-emphasis setting in Endpoint mode For Endpoint, set the De-emphasis value for 5.0 GT/s operation. Initial value of select_deemphasis variable described in PCIe Base Spec. 0: −6 dB (default) 1: −3.5dB	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	MODE_TXSWING	SerDes serial output amplitude control 0: Full swing mode (default) 1: Half swing mode	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	MODE_LINK_UPCONFIGURE_CAPABILITY_DISABLED	Setting of Link Upconfigure Capability bit of Training Sequence Ordered-set (TS-OS) When connecting with a Gen1 PCIe device, LinkUp may not occur unless this bit is 0. In that case, change it to 0 in F/W. 0: Link Upconfigure Capability bit = 1 setting 1: Link Upconfigure Capability bit = 0 setting (Gen1 x1)	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
27:16	ASPM_L1_INTERVAL_TIME[11:0]	Interval settings for ASPM L1 requests The PCIe Base Spec stipulates that ASPM L1 transition requests must not be accepted continuously within 10 µs, and this field sets the timer value to guard against this. Set so that ACLK cycle x set value is 10 µs or more. At ACLK: 400 [MHz] or higher, this bit setting value should be the value (1/16) of the lower 4 bits of the setting value of the above specifications, and set the clock number as the timer value setting. Settings of this product (Default): When ACLK = 400 MHz(2.5 ns): 4000(d)/16 = FA(h)	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_PCMSET1 register sets the operating mode of the PCI Express core.

56.4.1.35 PCI_EP_PCCTRL1 : PCIe Core Control 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0404

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	BLB_RELAX_ORDERING_EN	—	—	—	—	—	UI_ENTER_L1S	UI_ENTER_L1_STATUS	—	—	—	—	UI_ENTER_TXL0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MODE_QUIESCE_GUARANTEE	UI_ENTER_TXMODE_SRIS	MODE_EQ_AUTONOMOUS	MODE_EQ_PHASE23_ENABLE	MODE_RESET_EIEOS_INTERVALL0S	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	MODE_RESET_EIEOS_INTERVALL0S	Gen3 features: Reset EIEOS Interval information of bit 2, symbol 6 of TS1OS transmitted in Recovery.Equalization state	R/W
9	MODE_EQ_PHASE23_ENABLE	Gen3 features: Setting whether to execute EQ PHASE2 and EQ PHASE3 in RC mode (MODE_PORT=1). 0: Do not execute EQ PHASE2/3 1: Execute EQ PHASE2/3	R/W
10	MODE_EQ_AUTONOMOUS	Gen3 feature: Autonomous Equalization Basically only changeable during the reset period 0: Do not use Autonomous Mechanism 1: Use Autonomous Mechanisms	R/W
11	UI_ENTER_TXMODE_SRIS	Setting Clock Tolerance Compensation 0: SRNS (default) 1: SRIS (not supported)	R/W
12	MODE_QUIESCE_GUARANTEE	Symbol6 bit6 Quiesce Guarantee control bit of TS2OS 0: Set 0 to TS2OS (default) 1: Set 1 to TS2OS	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	UI_ENTER_TXL0S	TxL0s transition control 0: Do not perf orm ASPM L0s transition (default) 1: Execute ASPM L0s transition when internal conditions are satisfied	R/W
20:17	—	These bits are read as 0. The write value should be 0.	R/W
21	UI_ENTER_L1_STATUS	L1 transition status Read: 0: Normal operation (default) 1: L1 transition processing Write: Writing 1 clears the bit. When clearing the ALLOW_ENTER_L1 bit in the PCIE_MISC register during L1 transition process, clear this bit if this bit indicates 1. See section 56.7.9.2. Active State Power Management (ASPM).	R/W
22	UI_ENTER_L1S	L1SubState transition permission setting 0: L1SubState transition disabled (default) 1: L1SubState transition permission (setting prohibited)	R/W
27:23	—	These bits are read as 0. The write value should be 0.	R/W
28	BLB_RELAX_ORDERING_EN	Control of RO bit of Request to be sent 0: RO bit of Request TLP to be sent is always 0 (default) 1: A TLP can be sent with the RO bit of the Request TLP to be sent set to 1.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_PCCTRL1 register controls power management and LTSSM (Link Training Sequence State Machine) state transitions.

56.4.1.36 PCI_EP_PCSTAT1 : PCIe Core Status 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0408

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	TURN_OFF_EVENT_ACK	TURN_OFF_EVENT	—	—	—	—	—	—	—	—	BD	—
Value after reset:	0	0	0	0	x	0	0	0	0	0	0	0	0	0	x	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	LTSSM_STATE[6:0]							PMU_LINKSTATE[3:0]				—	—	STATE_VC0_NEGOTIATION_PENDING	DLDS
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x

Bit	Symbol	Function	R/W
0	DLDS	DL_Down status Indicates whether PCIe Core is in DL_Down or DL_Up state 0: DL_Up Status 1: DL_Down Status	R
1	STATE_VC0_NEGOTIATION_PENDING	Flow Control initialization operation monitor If this bit is 1, do not initiate a transaction from the AXI side. Check that this bit is 0 and DL_Down Status (bit[0]) is 0. 0: Indicates that Flow Control initialization is complete 1: Indicates that Flow Control initialization has not completed	R
3:2	—	These bits are read as 0.	R
7:4	PMU_LINKSTATE[3:0]	L-state monitor of power management control unit 0x4: L1 state 0x8: L2 state	R
14:8	LTSSM_STATE[6:0]	Indicates the state of the Link Training & Status State Machine in the PCIe Core Link. The following states are indicated by the upper 5 bits [14:10]. 000xxb: Detect 001xxb: Polling 010xxb: Config 01100b: L0 01101b: L1 0111xb: L2 100xxb: Recovery 101xxb: Disable 110xxb: Loopback	R
16:15	—	These bits are read as 0.	R
17	BD	bme_down Indicates that the PCIe core transmitter is in an unusable state.	R
25:18	—	These bits are read as 0.	R
26	TURN_OFF_EVENT	TURN_OFF_EVENT output signal monitor	R
27	TURN_OFF_EVENT_ACK	TURN_OFF_EVENT_ACK input signal monitor	R
31:28	—	These bits are read as 0.	R

The PCI_EP_PCSTAT1 register indicates the status of the power management in the PCI Express core.

56.4.1.37 PCI_EP_PCCTRL2 : PCIe Core Control 2 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0410

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	UI_LINK_WIDTH_CHANGE_ENABLE[7:0]								MODE_NODEEMPHASIS[1:0]		MODE_PRESET_ENABLE[4:0]					UI_LINK_WIDTH_CHANGE_REQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	UI_LINK_SPEED_CHANGE[1:0]		—	—	—	UI_LINK_CHANGE_AUTONOMOUS	—	—	—	UI_LINK_SPEED_CHANGE_REQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UI_LINK_SPEED_CHANGE_REQ	Link Speed change request control Setting this bit to 1 requests to change the Link Speed to the speed set in bit [8] UI_LINK_SPEED_CHANGE field. By asserting it in the L0 state, it transitions to the Recovery state and performs negotiation with the peer device. PCIe Core Status 2 Register (Offset: 0x414) bit [28] Set to 0 after confirming that UI_LINK_SPEED_CHANGE_DONE has been asserted.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	UI_LINK_CHANGE_AUTONOMOUS	Link Width/Speed change reason setting 0: reliability reason (change for reliability, direction of bandwidth reduction) 1: autonomous reason (intentional change)	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
9:8	UI_LINK_SPEED_CHANGE[1:0]	Link speed setting Set the Link Speed you want to change. 0 0: 2.5 GT/s 0 1: 5.0 GT/s 1 0: 8.0 GT/s 1 1: 16.0 GT/s (Setting prohibited)	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	UI_LINK_WIDTH_CHANGE_REQ	Link Width change request control Setting this bit to 1 issues a request to change the Link Width to the configuration set in bit [31:24] UI_LINK_WIDTHCHANGE_ENABLE field. By asserting it in the L0 state, it transitions from the Recovery state to the Configuration state, and performs negotiation with the other device. Set to 0 after confirming that PCIe Core Status 2 Register (Offset: 0x414) bit [29] UI_LINK_WIDTH_CHANGE_DONE is asserted. Available for Single Link structure. Prohibited to set for Multi Link structure or ch1.	R/W
21:17	MODE_PRESET_ENABLE[4:0]	Reduced Swing mode setting for Gen3 operation [0] 8GT/s Preset P0, 0: disable / 1: enable (default) [1] 8GT/s Preset P2, 0: disable / 1: enable (default) [2] 8GT/s Preset P7, 0: disable / 1: enable (default) [3] 8GT/s Preset P8, 0: disable / 1: enable (default) [4] 8GT/s Preset P10, 0: disable / 1: enable (default)	R/W
23:22	MODE_NODEEMPHASIS[1:0]	No de-emphasis mode setting for Gen1/Gen3 operation [0] Gen1 operation, 0: Normal operation mode (default) / 1: No de-emphasis mode [1] Gen3 operation, 0: Normal operation mode (default) / 1: No de-emphasis mode	R/W

Bit	Symbol	Function	R/W
31:24	UI_LINK_WIDTH_CHANGE_ENABLE[7:0]	Link Width setting to change Assert UI_LINK_WIDTH_CHANGE_REQ and set Lane to 1 to operate when Link Width change request is issued. The lower bit (bit [24]) drives Lane 0 and the most significant bit (bit [31]) drives Lane 7. In this module, bit [25:24] is valid for Single Link structure, and bit [24] is valid for Multi Link structure or ch1.	R/W

The PCI_EP_PCCTRL2 register controls the link speed/width change in the PCI Express core.

56.4.1.38 PCI_EP_PCSTAT2 : PCIe Core Status 2 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0414

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	UI_LINK_WIDTH_CHANGE_DONE	UI_LINK_SPEED_CHANGE_DONE	—	—	—	STATE_UPCONFIGURE_CAPABLE	—	STATE_NEGOTIATED_LANE_END[2:0]			—	STATE_NEGOTIATED_LANE_START[2:0]		
Value after reset:	0	0	x	x	0	0	0	x	0	x	x	x	0	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	STATE_RECEIVER_DETECTED[7:0]								STATE_DATA_RATE_IDENTIFIER_RECEIVED[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	STATE_DATA_RATE_IDENTIFIER_RECEIVED[7:0]	Link Speed display supported by the opposite device Displays the TS-OS Data Rate Identifier feed received from the peer device. Bit 0: Reserved Bit 1: 2.5 GT/s Data Rate Supported. Must be set to 1. Bit 2: 5.0 GT/s Data Rate Supported. Must be set to 1 if Bit 3 is 1. Bit 3: 8.0 GT/s Data Rate Supported. Bits 4 to 7: Reserved	R
15:8	STATE_RECEIVER_DETECTED[7:0]	Connection status display with other device Receiver Detection results are displayed. [0] Detect opposite device on Lane 0 [1] Detect peer device on Lane 1 (only for x2) [2] to [7] reserved	R
18:16	STATE_NEGOTIATED_LANE_START[2:0]	Displays the position of Lane Number 0 after Link Negotiation with the opposite device during n-lane operation. Used to check the state of the current working lane before changing the Link Width. Available for Single Link structure. Prohibited to set for Multi Link structure or ch1. 0 0 0: Lane 0 is Lane Number 0 0 0 1: Lane 1 is Lane Number 0 Others: Reserved	R
19	—	This bit is read as 0.	R
22:20	STATE_NEGOTIATED_LANE_END[2:0]	Displays the position of Lane Number (n-1) (meaning Lane 1 when n=2) after Link Negotiation with the opposite device during n-lane operation. Used to check the state of the current working lane before changing the Link Width. Available for Single Link structure. Prohibited to set for Multi Link structure or ch1. 0 0 0: Lane 0 is Lane Number (n-1) 0 0 1: Lane 1 is Lane Number (n-1) Others: Reserved	R
23	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
24	STATE_UPCONFIGURE_CAPABLE	Upconfigure Capable bit display of opposite device Indicates whether the peer device supports changing the direction of widening the Link Width. If this bit is 0, changing the Link Width will not restore the original Link Width.	R
27:25	—	These bits are read as 0.	R
28	UI_LINK_SPEED_CHANGE_DONE	Link Speed Change operation complete display PCIe Core Status 2 Registers (Offset: 0x410) bit [0] Notifies completion of Speed Change (1) by setting UI_LINK_SPEED_CHANGE_REQ. It is set to 0 by setting UI_LINK_SPEED_CHANGE_REQ to 0.	R
29	UI_LINK_WIDTH_CHANGE_DONE	Link Width Change operation complete display Notifies completion of Width Change (1) by setting PCIe Core Status 2 Register (Offset: 0x410) bit [16] UI_LINK_WIDTH_CHANGE_REQ. It is set to 0 by setting UI_LINK_WIDTH_CHANGE_REQ to 0. Available for Single Link structure. Prohibited to set for Multi Link structure or ch1.	R
31:30	—	These bits are read as 0.	R

The PCI_EP_PCSTAT2 register indicates the status of the link speed/width change in the PCI Express core.

56.4.1.39 PCI_EP_PCSTAT5 : PCIe Core Status 5 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x042C

Bit position:	31							24	23							16	15							8	7							0
Bit field:	SBME[7:0]								D3_EVENT[7:0]								D3_EVENT_ACK[7:0]								ORT_TRANSACTION_PENDING[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ORT_TRANSACTION_PENDING[7:0]	Outstanding Request Monitor by Function Other than the lower 2 bits Reserved (fixed to 0) Indicates whether or not there are Outstanding Requests (a state in which all Completions corresponding to Non-Posted Requests sent from the AXI side have not been received). Check with this bit that there is no Outstanding Request before requesting/permitting a transition to TxL0s/L1/L2. 0: State without Outstanding Request 1: State with Outstanding Request	R
15:8	D3_EVENT_ACK[7:0]	D3_EVENT_ACK signal monitor for each function Other than the lower 2 bits Reserved (fixed to 0)	R
23:16	D3_EVENT[7:0]	D3_EVENT signal monitor for each function Other than the lower 2 bits Reserved (fixed to 0)	R
31:24	SBME[7:0]	Suspend_bme Indicates the state in which the use of the PCIe core transmitter should be suppressed for each function. Other than the lower 2 bits Reserved (fixed to 0)	R

The PCI_EP_PCSTAT5 register indicates the status in the PCI Express core.

56.4.1.40 PCI_EP_DMAINTVEC0 : DMA Interrupt Vector 0 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x04D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DMA_CH3_MSI_EN	DMA_CH3_V[4:0]					—	—	DMA_CH2_MSI_EN	DMA_CH2_V[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	DMA_CH1_MSI_EN	DMA_CH1_V[4:0]					—	—	DMA_CH0_MSI_EN	DMA_CH0_V[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DMA_CH0_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch0	R/W
5	DMA_CH0_MSI_EN	DMA Ch0 MSI Enable Reset to PHY. For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DMA_CH1_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch1	R/W
13	DMA_CH1_MSI_EN	DMA Ch1 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
20:16	DMA_CH2_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch2	R/W
21	DMA_CH2_MSI_EN	DMA Ch2 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
28:24	DMA_CH3_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch3	R/W
29	DMA_CH3_MSI_EN	DMA Ch3 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_DMAINTVEC0 register specifies the interrupt vectors for interrupt notification from AXI to PCIe (MSI) during DMA transfer.

56.4.1.41 PCI_EP_DMAINTVEC1 : DMA Interrupt Vector 1 Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x04D4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DMA_CH7_MSI_EN	DMA_CH7_V[4:0]					—	—	DMA_CH6_MSI_EN	DMA_CH6_V[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	DMA_CH5_MSI_EN	DMA_CH5_V[4:0]					—	—	DMA_CH4_MSI_EN	DMA_CH4_V[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DMA_CH4_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch4	R/W
5	DMA_CH4_MSI_EN	DMA Ch4 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DMA_CH5_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch5	R/W
13	DMA_CH5_MSI_EN	DMA Ch5 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
20:16	DMA_CH6_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch6	R/W
21	DMA_CH6_MSI_EN	DMA Ch6 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
28:24	DMA_CH7_V[4:0]	Vector value of MSI interrupt transmitted by DMAC Ch7	R/W
29	DMA_CH7_MSI_EN	DMA Ch7 MSI Enable For the RC function, MSI transmission is prohibited, so please fix it to 0. 0: Do not use MSI (DMA interrupt notification by DMA_INT interrupt) 1: Use MSI	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_DMAINTVEC1 register specifies the interrupt vectors for interrupt notification from AXI to PCIe (MSI) during DMA transfer.

56.4.1.42 PCI_EP_DMACTRL : DMAC Control Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0800

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	D_PMRS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	D_PMRS[2:0]	DMAC PCIe Max Read Request Size Set the upper limit of Read requests issued from the DMAC to PCIe. 0 0 0: 128 bytes (default) 0 0 1: 256 bytes 0 1 0: 512 bytes (not supported) 0 1 1: 1024 bytes (not supported) 1 0 0: 2048 bytes (not supported) 1 0 1: 4096 bytes (not supported) others: Reserved (prohibited)	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_DMACTRL register sets the maximum size of read requests which can be issued to the PCIe core as a DMAC function. Use the initial setting (128 bytes). This setting is common to all DMA channels.

56.4.1.43 PCI_EP_DMAINTE : DMAC Interrupt Enable Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0808

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CH7_ERR_EN	CH7_QUE_EMP_EN	CH7_STOP_EN	CH7_END_EN	CH6_ERR_EN	CH6_QUE_EMP_EN	CH6_STOP_EN	CH6_END_EN	CH5_ERR_EN	CH5_QUE_EMP_EN	CH5_STOP_EN	CH5_END_EN	CH4_ERR_EN	CH4_QUE_EMP_EN	CH4_STOP_EN	CH4_END_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CH3_ERR_EN	CH3_QUE_EMP_EN	CH3_STOP_EN	CH3_END_EN	CH2_ERR_EN	CH2_QUE_EMP_EN	CH2_STOP_EN	CH2_END_EN	CH1_ERR_EN	CH1_QUE_EMP_EN	CH1_STOP_EN	CH1_END_EN	CH0_ERR_EN	CH0_QUE_EMP_EN	CH0_STOP_EN	CH0_END_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CH0_END_EN	CH0 Completion Interrupt Enable 0: Disable 1: Enable	R/W
1	CH0_STOP_EN	CH0 Stop Interrupt Enable 0: Disable 1: Enable	R/W
2	CH0_QUE_EMP_EN	CH0 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
3	CH0_ERR_EN	CH0 Error Interrupt Enable 0: Disable 1: Enable	R/W
4	CH1_END_EN	CH1 Completion Interrupt Enable 0: Disable 1: Enable	R/W
5	CH1_STOP_EN	CH1 Stop Interrupt Enable 0: Disable 1: Enable	R/W
6	CH1_QUE_EMP_EN	CH1 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
7	CH1_ERR_EN	CH1 Error Interrupt Enable 0: Disable 1: Enable	R/W
8	CH2_END_EN	CH2 Completion Interrupt Enable 0: Disable 1: Enable	R/W
9	CH2_STOP_EN	CH2 Stop Interrupt Enable 0: Disable 1: Enable	R/W
10	CH2_QUE_EMP_EN	CH2 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
11	CH2_ERR_EN	CH2 Error Interrupt Enable 0: Disable 1: Enable	R/W
12	CH3_END_EN	CH3 Completion Interrupt Enable 0: Disable 1: Enable	R/W
13	CH3_STOP_EN	CH3 Stop Interrupt Enable 0: Disable 1: Enable	R/W
14	CH3_QUE_EMP_EN	CH3 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
15	CH3_ERR_EN	CH3 Error Interrupt Enable 0: Disable 1: Enable	R/W
16	CH4_END_EN	CH4 Completion Interrupt Enable 0: Disable 1: Enable	R/W
17	CH4_STOP_EN	CH4 Stop Interrupt Enable 0: Disable 1: Enable	R/W
18	CH4_QUE_EMP_EN	CH4 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
19	CH4_ERR_EN	CH4 Error Interrupt Enable 0: Disable 1: Enable	R/W
20	CH5_END_EN	CH5 Completion Interrupt Enable 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
21	CH5_STOP_EN	CH5 Stop Interrupt Enable 0: Disable 1: Enable	R/W
22	CH5_QUE_EMP_EN	CH5 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
23	CH5_ERR_EN	CH5 Error Interrupt Enable 0: Disable 1: Enable	R/W
24	CH6_END_EN	CH6 Completion Interrupt Enable 0: Disable 1: Enable	R/W
25	CH6_STOP_EN	CH6 Stop Interrupt Enable 0: Disable 1: Enable	R/W
26	CH6_QUE_EMP_EN	CH6 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
27	CH6_ERR_EN	CH6 Error Interrupt Enable 0: Disable 1: Enable	R/W
28	CH7_END_EN	CH7 Completion Interrupt Enable 0: Disable 1: Enable	R/W
29	CH7_STOP_EN	CH7 Stop Interrupt Enable 0: Disable 1: Enable	R/W
30	CH7_QUE_EMP_EN	CH7 Que Empty Interrupt Enable 0: Disable 1: Enable	R/W
31	CH7_ERR_EN	CH7 Error Interrupt Enable 0: Disable 1: Enable	R/W

The PCI_EP_DMAINTE register enables interrupts from the individual DMA channels.

56.4.1.44 PCI_EP_DMAINTS : DMAC Interrupt Status Register

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x080C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CH7_ERR	CH7_QUE_EMP	CH7_STOP	CH7_END	CH6_ERR	CH6_QUE_EMP	CH6_STOP	CH6_END	CH5_ERR	CH5_QUE_EMP	CH5_STOP	CH5_END	CH4_ERR	CH4_QUE_EMP	CH4_STOP	CH4_END
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CH3_ERR	CH3_QUE_EMP	CH3_STOP	CH3_END	CH2_ERR	CH2_QUE_EMP	CH2_STOP	CH2_END	CH1_ERR	CH1_QUE_EMP	CH1_STOP	CH1_END	CH0_ERR	CH0_QUE_EMP	CH0_STOP	CH0_END
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CH0_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
1	CH0_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
2	CH0_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
3	CH0_ERR	Set when an error occurs during DMA transfer.	R/W
4	CH1_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
5	CH1_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
6	CH1_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
7	CH1_ERR	Set when an error occurs during DMA transfer.	R/W
8	CH2_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
9	CH2_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
10	CH2_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
11	CH2_ERR	Set when an error occurs during DMA transfer.	R/W
12	CH3_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
13	CH3_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
14	CH3_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
15	CH3_ERR	Set when an error occurs during DMA transfer.	R/W
16	CH4_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
17	CH4_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
18	CH4_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
19	CH4_ERR	Set when an error occurs during DMA transfer.	R/W

Bit	Symbol	Function	R/W
20	CH5_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
21	CH5_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
22	CH5_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
23	CH5_ERR	Set when an error occurs during DMA transfer.	R/W
24	CH6_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
25	CH6_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
26	CH6_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
27	CH6_ERR	Set when an error occurs during DMA transfer.	R/W
28	CH7_END	Set when the DMAC ends normally. Normal termination refers to the following conditions. ● Transfer for the amount indicated by DMA_SIZE is completed. ● At the end of the descriptor list when the EI field is 1.	R/W
29	CH7_STOP	Set when the DMAC is prematurely stopped. DMA stops under the following conditions: ● When RDMA_EN or QUE_EN is cleared to 0 by the S/W during DMA transfer execution and the processing of the request being issued ends.	R/W
30	CH7_QUE_EMP	Set when the list is removed from the descriptor queue (transferred to the execution descriptor list) and the QUE is empty.	R/W
31	CH7_ERR	Set when an error occurs during DMA transfer.	R/W

The PCI_EP_DMAINTS register indicates the state of interrupts from the individual DMA channels. Writing 1 clears the bit.

56.4.1.45 PCI_EP_DMACHCTLn : DMAC Channel Control Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0900 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	QUE_CLR	—	—	—	—	—	—	QUE_EN	RDMA_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RDMA_EN	Register method DMA transfer Enable Setting this bit to 1 starts DMA transfer (register method) and performs the transfer set by RDMA_SIZE. This bit is automatically cleared to 0 when DMA transfer is completed or an error is detected and DMA ends. It is also possible to stop DMA by writing 0. However, DMA will stop after completing the currently executing request (both PCIe and AXI). 0: Stop register-based DMA transfer 1: Register method DMA transfer start	R/W
1	QUE_EN	QUE Enable Setting this bit to 1 enables the descriptor list registered in the descriptor queue and starts DMA transfer (descriptor method). It is automatically cleared to 0 when DMA stops (both normal and abnormal). It is also possible to stop DMA by writing 0. However, DMA will stop after finishing the currently executing request (both PCIe and AXI).	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	QUE_CLR	QUE Clear Writing 1 clears QUE. All descriptor lists (waiting for execution and lists currently being executed) registered in QUE are cleared. Do not clear during DMA execution. The read value is always 0. 1Write is prohibited at the same time as QUE_EN Set	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_DMACHCTLn register sets the control method for each DMA channel. Set either register type or descriptor type.

- Setting QUE_EN = 1 and QUE_CLR = 1 is prohibited during register-type DMA transfer (TDMA_EN = 1).
- Setting RDMA_EN = 1 is prohibited during descriptor-type DMA transfer (QUE_EN = 1).

56.4.1.46 PCI_EP_DPSADRLn : Descriptor Start Address (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0908 + 0x80 × n

Bit position:	31																								6	5					0
Bit field:	QUE_ENTRY_DSA3106[25:0]																									QUE_ENTRY_DSA0500[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	QUE_ENTRY_DSA0500[5:0]	Descriptor list queue registration register. This area will be the DSA. Sets the lower 32 bits of the address where the first descriptor is stored. (16 byte align: Lower 6 bit is fixed to 0.)	R
31:6	QUE_ENTRY_DSA3106[25:0]	Descriptor list queue registration register. This area will be the DSA. Sets the lower 32 bits of the address where the first descriptor is stored.	R/W

Note: When these bits are read, the following contents will be the read value depending on the state of DMA.
DMA transfer in progress: The descriptor list in progress
DMA suspended (when QUE_EN is automatically cleared): Last executed list
DMA suspended (when QUE_EN S/W is cleared): Suspended (executing) list

The PCI_EP_DPSADRLn register sets the descriptor queue list.

The setting value is registered as the lower 32 bits of DSA (DMA Start Address) in the queue list.

56.4.1.47 PCI_EP_DPSADRUn : Descriptor Start Address (Upper) Registers n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x090C + 0x80 × n

Bit position:	31																												0
Bit field:	QUE_ENTRY_DSA6332[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	QUE_ENTRY_DSA6332[31:0]	Descriptor list queue registration register. This area will be the DSA. Set the upper 32 bits of the address where the first descriptor is stored.	R/W

Note: When these bits are read, the following contents will be the read value depending on the state of DMA.
DMA transfer in progress: The descriptor list in progress
DMA suspended (when QUE_EN is automatically cleared): Last executed list
DMA suspended (when QUE_EN S/W is cleared): Suspended (executing) list

The PCI_EP_DPSADRUn register sets the descriptor queue list.

The setting value is registered as the upper 32 bits of DSA (DMA Start Address) in the queue list.

56.4.1.48 PCI_EP_QUEEn : QUE Entry Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0910 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	QUE_R[5:0]						QUE_ENTRY_EI	QUE_ENTRY_LS	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	QUE_ENTRY_LABEL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	QUE_ENTRY_LABEL[15:0]	There is no particular rule on how to set the labels in the list. You can set the value freely.	R/W
23:16	—	These bits are read as 0. The write value should be 0.	R/W
24	QUE_ENTRY_LS	Indicates whether to stop the DMA when processing of the descriptor list is complete. 0: do not stop 1: stop	R/W
25	QUE_ENTRY_EI	Indicates whether an interrupt (Interrupt Status CHx_END) is sent when the processing of the descriptor list is completed. 0: Do not signal interrupts 1: signal an interrupt	R/W
31:26	QUE_R[5:0]	QUE_Registration Enqueued by writing to [31:24]. The read value is always 0.	R/W

Note: When these bits are read, the following contents will be the read value depending on the state of DMA.
DMA transfer in progress: The descriptor list in progress
DMA suspended (when QUE_EN is automatically cleared): Last executed list
DMA suspended (when QUE_EN S/W is cleared): Suspended (executing) list

The PCI_EP_QUEEn register sets the descriptor queue list.

The setting value is registered as EI (End Interrupt), LS (List Stop), and LABEL of the queue list. Write to [31:24] to register to the queue.

56.4.1.49 PCI_EP_DMADPCTLn : DMA Descriptor Control (Descriptor 0x00) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0920 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DSCFM[3:0]				—	WBD	LE	LV	D	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	STS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	STS[15:0]	Shows the value of the STS field in the running descriptor table.	R
22:16	—	These bits are read as 0.	R
23	D	Shows the value of the D field in the running descriptor table.	R
24	LV	Shows the value of the LV field in the running descriptor table.	R
25	LE	Shows the value of the LE field in the running descriptor table.	R
26	WBD	Shows the value of the WBD field in the running descriptor table.	R
27	—	This bit is read as 0.	R
31:28	DSCFM[3:0]	Shows the value of the DSCFM field in the running descriptor table.	R

The PCI_EP_DMADPCTLn register indicates the field value at offset 0x00 in the descriptor table. Only effective when the descriptor-type DMA transfer is selected (the value read has no meaning in the case of register-type transfer).

56.4.1.50 PCI_EP_DMATCTLn : DMA Transaction Control (Descriptor 0x04) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0924 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	CCH_L[3:0]				CCH_D[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	DMA_TC[2:0]			—	—	DMA_ATB[1:0]		—	DMA_FUNC[2:0]			—	—	—	DMA_DIR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DMA_DIR	Sets the DMA transfer direction. 0: PCIe to AXI 1: AXI to PCIe	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
6:4	DMA_FUNC[2:0]	Specify the function number of the request issued to PCIe.	R/W

Bit	Symbol	Function	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
9:8	DMA_ATB[1:0]	Attributes to publish to PCIe. bit [1]: Relaxed Ordering (Unsupported function: 0 fixed) bit [0]: No Snoop (0 recommended)	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
14:12	DMA_TC[2:0]	Traffic class to publish to PCIe. Specifies the traffic class value for requests issued to PCIe. Note: This IP does not support Virtual Channel, so please use it with 000b fixed.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
19:16	CCH_D[3:0]	Indicates the value of A*CACHE[3:0] to be issued to AXI. CCH_D is output when issuing an AXI request other than the output condition of CCH_L. The recommended value is 0x1 when DIR = 0 (PCIe→AXI) and 0x0 when DIR = 1 (AXI→PCIe). [3] 1: write allocatable / 0: not write allocatable [2] 1: Read assignable / 0: Read assignable [1] 1: cacheable / 0: non-cacheable [0] 1: Bufferable / 0: Not Bufferable	R/W
23:20	CCH_L[3:0]	Indicates the value of A*CACHE[3:0] to be issued to AXI. CCH_L is output when issuing an AXI request containing the last byte in a transfer indicated by SIZE. Recommended value is 0x0. [3] 1: write allocatable / 0: not write allocatable [2] 1: Read assignable / 0: Read assignable [1] 1: cacheable / 0: non-cacheable [0] 1: Bufferable / 0: Not Bufferable	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: When executing descriptor method DMA, writing is prohibited and the read value indicates the values of the CCH_L, CCH_D, TC, ATB, and DIR fields of the descriptor table being executed.

The PCI_EP_DMATCTLn register controls DMA transfer to the AXI and PCIe.

56.4.1.51 PCI_EP_DMASIZEn : DMA Size (Descriptor 0x08) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0928 + 0x80 × n

Bit position:	31																										3	2		0
Bit field:	DMA_SIZE3104[27:0]																										DMA_SIZE 0300[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DMA_SIZE0300[3:0]	Sets the number of DMA transfer bytes. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R
31:4	DMA_SIZE3104[27:0]	Sets the number of DMA transfer bytes. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R/W

Note: When executing descriptor method DMA, writing is prohibited and the read value indicates the value of the SIZE field of the descriptor table being executed. The number of transfer bytes when 0x0000_0000 is set is 0x1_0000_0000.

The PCI_EP_DMASIZEn register sets the number of bytes for DMA transfer. Reflects the field at offset 0x0C in the descriptor table.

56.4.1.52 PCI_EP_DMASLAn : DMA Source Lower Address (Descriptor 0x10) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0930 + 0x80 × n

Bit position:	31																											3	2	0
Bit field:	DMA_S_ADDR3104[27:0]																											DMA_S_ADDR0300[3:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DMA_S_ADDR0300[3:0]	Set the lower 32 bits of the transfer source start address for DMA transfer. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R
31:4	DMA_S_ADDR3104[27:0]	Set the lower 32 bits of the transfer source start address for DMA transfer. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R/W

Note: If the Source Address indicates the PCIe space (DIR = 0), the PCIe request address is set in combination with the DMA PCIe Upper Address (Descriptor 0x10). When descriptor-type DMA transfer is in process, writing to these bits is prohibited and the value read indicates the value of the SA field in the descriptor table being executed.

The PCI_EP_DMASLAn register sets the lower 32 bits of the source start address for DMA transfer. It is reflected to the offset 0x10 field of the descriptor table.

56.4.1.53 PCI_EP_DMASUAn : DMA Source Upper Address (Descriptor 0x14) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0934 + 0x80 × n

Bit position:	31																													0
Bit field:	DMA_S_ADDR6332[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DMA_S_ADDR6332[31:0]	Set the upper 32 bits of the transfer source start address for MA transfer.	R/W

Note: If the Source Address indicates the PCIe space (DIR = 0), the PCIe request address is set in combination with the DMA PCIe Upper Address (Descriptor 0x10). When descriptor-type DMA transfer is in process, writing to these bits is prohibited and the value read indicates the value of the SA field in the descriptor table being executed.

The PCI_EP_DMASUAn register sets the upper 32 bits of the source start address for DMA transfer. It is reflected to the offset 0x14 field of the descriptor table.

56.4.1.54 PCI_EP_DMADESTLAn : DMA Destination Lower Address (Descriptor 0x18) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0938 + 0x80 × n

Bit position:	31																											3	2	0
Bit field:	DMA_D_ADDR3104[27:0]																											DMA_D_ADDR0300[3:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DMA_D_ADDR0300[3:0]	Set the lower 32 bits of the transfer destination start address for DMA transfer. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R
31:4	DMA_D_ADDR3104[27:0]	Set the lower 32 bits of the transfer destination start address for DMA transfer. The lower 4 bits are fixed to 0 because it is an 16-byte aligned setting.	R/W

Note: If the Destination Address indicates the PCIe space (DIR = 1), the PCIe request address is set in combination with the DMA PCIe Upper Address (Descriptor 0x10). When descriptor-type DMA transfer is in process, writing to these bits is prohibited and the value read indicates the value of the DA field in the descriptor table being executed.

The PCI_EP_DMADESTLAn register sets the lower 32 bits of the source start address for DMA transfer. It is reflected to the offset 0x18 field of the descriptor table.

56.4.1.55 PCI_EP_DMADESTUAn : DMA Destination Upper Address (Descriptor 0x1C) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x093C + 0x80 × n

Bit position:	31																													0
Bit field:	DMA_D_ADDR6332[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DMA_D_ADDR6332[31:0]	Set the upper 32 bits of the transfer destination start address for DMA transfer.	R/W

Note: If the Destination Address indicates the PCIe space (DIR = 1), the PCIe request address is set in combination with the DMA PCIe Upper Address (Descriptor 0x10). When descriptor-type DMA transfer is in process, writing to these bits is prohibited and the value read indicates the value of the DA field in the descriptor table being executed.

The PCI_EP_DMADESTUAn register sets the upper 32 bits of the source start address for DMA transfer. It is reflected to the offset 0x1C field of the descriptor table.

56.4.1.56 PCI_EP_DMADPLLPn : DMA Descriptor Lower Link Pointer (Descriptor 0x20) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0940 + 0x80 × n

Bit position:	31																													0
Bit field:	DMA_LP3100[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DMA_LP3100[31:0]	Indicates the value of the LP field of the running descriptor table.	R

The PCI_EP_DMADPLLPn register indicates the field value at offset 0x1C in the descriptor table. Only effective when the descriptor-type DMA transfer is selected (the value read has no meaning in the case of register-type transfer).

56.4.1.57 PCI_EP_DMADPULPn : DMA Descriptor Upper Link Pointer (Descriptor 0x24) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0944 + 0x80 × n

Bit position:	31																												0
Bit field:	DMA_LP6332[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DMA_LP6332[31:0]	Indicates the value of the LP field of the running descriptor table.	R

The PCI_EP_DMADPULPn register indicates the field value at offset 0x1C in the descriptor table. Only effective when the descriptor-type DMA transfer is selected (the value read has no meaning in the case of register-type transfer).

56.4.1.58 PCI_EP_DMARESTSIZEn : DMA Rest Size Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0950 + 0x80 × n

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	DMA_REST_SIZE Displays the number of bytes for which DMA transfer has not been completed. (Register/ descriptor method common)	R

The PCI_EP_DMARESTSIZEn register indicates the number of bytes for which DMA transfer has not yet been completed.

56.4.1.59 PCI_EP_AREQALn : AXI Request Address (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0960 + 0x80 × n

Bit position:	31																												0
Bit field:	AXI_REQ_ADDR3100[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	AXI_REQ_ADDR3100[31:0]	Displays the lower 32 bits of the address of the currently completed AXI transfer or the transfer that just completed. (Register/descriptor method common)	R

The PCI_EP_AREQALn register indicates the lower 32 bits of the address of the current or most recently completed AXI transfer.

56.4.1.60 PCI_EP_AREQAUn : AXI Request Address (Upper) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0964 + 0x80 × n

Bit position:	31																												0
Bit field:	AXI_REQ_ADDR6332[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	AXI_REQ_ADDR6332[31:0]	Displays the upper 32 bits of the address of the currently completed AXI transfer or the transfer that just completed. (Register/descriptor method common)	R

The PCI_EP_AREQAUn register indicates the upper 32 bits of the address of the current or most recently completed AXI transfer.

56.4.1.61 PCI_EP_PREQALn : PCIe Request Address (Lower) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0968 + 0x80 × n

Bit position:	31																												0
Bit field:	PCIE_REQ_ADDR3100[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PCIE_REQ_ADDR3100[31:0]	Displays the lower 32 bits of the address of the currently completed PCIe transfer or the transfer completed immediately before. (Register/descriptor method common)	R

The PCI_EP_PREQALn register indicates the lower 32 bits of the address during PCIe transfer that is in progress or for the most recently completed transfer.

56.4.1.62 PCI_EP_PREQAUn : PCIe Request Address (Upper) Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x096C + 0x80 × n

Bit position:	31																												0
Bit field:	PCIE_REQ_ADDR6332[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PCIE_REQ_ADDR6332[31:0]	Displays the upper 32 bits of the address of the currently completed PCIe transfer or the transfer completed immediately before. (Register/descriptor method common)	R

The PCI_EP_PREQAUn register indicates the upper 32 bits of the address during PCIe transfer that is in progress or for the most recently completed transfer.

56.4.1.63 PCI_EP_QUESTAn : QUE Status Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0970 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	GO_LIST	LIST_NUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	LIST_NUM[3:0]	Displays the number of descriptor lists loaded on QUE (not including the list currently being executed). New registration to QUE (write access to QUE Entry) when this register indicates 0x8 is invalid (discarded).	R
4	GO_LIST	Shows whether there is a running descriptor list. 0: none 1: Yes	R
31:5	—	These bits are read as 0.	R

The PCI_EP_QUESTAn register indicates the state of the descriptor queue.

56.4.1.64 PCI_EP_DMACESTAn : DMAC Error Status Register n (n = 0 to 7)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0978 + 0x80 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CFG_BM_DIS_EP	BME_SUP	BME_DOWN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	MOR_CD_PERR	MOR_CH_PERR	MOR_EP_ERR	MOR_STATUS[2:0]			—	—	—	—	—	—	AXI_RESP[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	AXI_RESP[1:0]	Displays the slave response during AXI Master transactions. It updates when CHx_ERR is set and retains the value until the bit is cleared. 0 0: Initial value 0 1: Reserved	R
7:2	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
10:8	MOR_STATUS[2:0]	Indicates the value when MOR_STATUS is other than 000b (Success) as a set factor of CHx_ERR. Holds value until CHx_ERR is cleared. 0 0 0: initial value 0 0 1: Unsupported Request 0 1 0: CRS 0 1 1: Completion Timeout 1 0 0: Completer Abort 1 0 1: Unexpected Completion 1 1 0: Reserved 1 1 1: Mismatched Length (Length Overrun)	R
11	MOR_EP_ERR	It is set when it is Poisoned Completion as a set factor of CHx_ERR. Holds value until CHx_ERR is cleared.	R
12	MOR_CH_PERR	It is set when MOR_CH_PERR is detected as a set factor for CHx_ERR. Holds value until CHx_ERR is cleared.	R
13	MOR_CD_PERR	It is set when MOR_CD_PERR is detected as a set factor for CHx_ERR. Holds value until CHx_ERR is cleared.	R
15:14	—	These bits are read as 0.	R
16	BME_DOWN	It is set when a stop signal from the PCIe core is detected as a cause for setting CHx_ERR. Holds value until CHx_ERR is cleared.	R
17	BME_SUP	It is set when a sleep signal from the PCIe core is detected as a cause for setting CHx_ERR. Holds value until CHx_ERR is cleared. (Valid only in Endpoint mode)	R
18	CFG_BM_DIS_EP	CHx_ERR is set when the macro operates as an End Point and detects the state of Bus Master Enable Off (Configuration Space 0x004 bit [2] = 0). Holds value until CHx_ERR is cleared.	R
31:19	—	These bits are read as 0.	R

The PCI_EP_DMACESTAn register indicates the error status of the DMAC.

56.4.1.65 PCI_EP_AWBASELn_Fi : AXI Window Base (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1000 + 0x0200 × i + 0x20 × n

Bit position:	31																			12	11								4				0
Bit field:	AWB3112[19:0]																				AWB1104[7:0]								—	—	—	AXIWE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AXIWE	AXI Window Enable Enable setting of AXI Window. 0: Window disabled 1: Window enabled	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
11:4	AWB1104[7:0]	AWBase[11:4] fixed 0x00	R
31:12	AWB3112[19:0]	AWBase[31:12] Window setting register for address conversion when accessing AXI from PCIe. The configurable area is the 4K boundary.	R/W

The PCI_EP_AWBASELn_Fi register is for setting windows for lower address conversion in access from PCIe to AXI. It sets the base address on the PCI. The areas are set in 4-KB boundaries.

56.4.1.66 PCI_EP_AWBASEUn_Fi : AXI Window Base (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1004 + 0x0200 × i + 0x20 × n

Bit position:	31																												0
Bit field:	AWB6332[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	AWB6332[31:0]	AWBase[63:32] Window setting register for address conversion when accessing AXI from PCIe.	R/W

The PCI_EP_AWBASEUn_Fi register is for setting windows for upper address conversion in access from PCIe to AXI.

56.4.1.67 PCI_EP_AWMASKLn_Fi : AXI Window Mask (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1008 + 0x0200 × i + 0x20 × n

Bit position:	31																			12	11									0
Bit field:	AWM3112[19:0]																				AWM1100[11:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	AWM1100[11:0]	AWMask[11:0] Fixed 0xFFF	R
31:12	AWM3112[19:0]	AWMask[31:12] Set the window in the area of the set number of bits from the address set in the AWBase register. Set 1 from the lower bit. Therefore, the area that can be set is 4K x 2^N bytes.	R/W

The PCI_EP_AWMASKLn_Fi register is for setting windows for lower address conversion in access from PCIe to AXI. The window is set as the area corresponding to the number of set bits from the address set in the AWBase register. The area which can be set is 4K × 2^N bytes.

56.4.1.68 PCI_EP_AWMASKUn_Fi : AXI Window Mask (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x100C + 0x0200 × i + 0x20 × n

Bit position:	31	30																											0
Bit field:	—	AWM6232[30:0]																											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
30:0	AWM6232[30:0]	AWMask[62:32] Set the window in the area of the set number of bits from the address set in the AWBase register. Set 1 from the lower bit.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_EP_AWMASKUn_Fi register is for setting windows for upper address conversion in access from PCIe to AXI.

56.4.1.69 PCI_EP_ADESTLn_Fi : AXI Destination (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1010 + 0x0200 × i + 0x20 × n

Bit position:	31																		12	11											0
Bit field:	AD3112[19:0]																			AD1100[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	AD1100[11:0]	ADest[11:0] Fixed to 0x000	R
31:12	AD3112[19:0]	ADest[31:12] Sets the window base point in the address space from the AXI side. The configurable area is the 4K boundary.	R/W

The PCI_EP_ADESTLn_Fi register is for setting windows for lower address conversion in access from PCIe to AXI. It sets the base address on the AXI. The areas are set in 4-KB boundaries.

56.4.1.70 PCI_EP_ADESTUn_Fi : AXI Destination (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1014 + 0x0200 × i + 0x20 × n

Bit position:	31																														0
Bit field:	AD6332[31:0]																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	AD6332[31:0]	ADest[63:32] Sets the window base point in the address space f rom the AXI side.	R/W

The PCI_EP_ADESTUn_Fi register is for setting windows for upper address conversion in access from PCIe to AXI.

56.4.1.71 PCI_EP_PWBASELn_Fi : PCIe Window Base (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1100 + 0x0200 × i + 0x20 × n

Bit position:	31																		12	11							4				0
Bit field:	PWB3112[19:0]																			PWB114[7:0]							—	—	—	PCIWE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

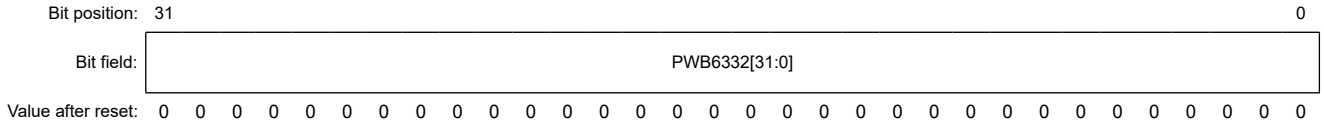
Bit	Symbol	Function	R/W
0	PCIWE	PCIe Window Enable Enable setting of PCIe Window. 0: Window disabled 1: Window enabled	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
11:4	PWB114[7:0]	PWBase[11:4] Fixed 0x00	R
31:12	PWB3112[19:0]	PWBase[31:12] Sets the base point of the address on the AXI side. The configurable area is the 4K boundary.	R/W

The PCI_EP_PWBASELn_Fi register is for setting windows for lower address conversion in access from AXI to PCIe. It sets the base address on the AXI. The areas are set in 4-KB boundaries.

56.4.1.72 PCI_EP_PWBASEUn_Fi : PCIe Window Base (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1104 + 0x0200 × i + 0x20 × n



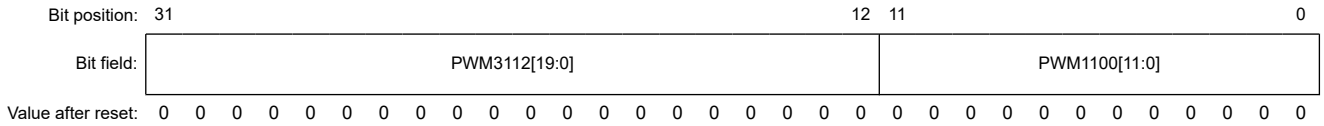
Bit	Symbol	Function	R/W
31:0	PWB6332[31:0]	PWBase[63:32] Sets the base point of the address on the AXI side. The configurable area is the 4K boundary.	R/W

The PCI_EP_PWBASEUn_Fi register is for setting windows for upper address conversion in access from AXI to PCIe.

56.4.1.73 PCI_EP_PWMASKLn_Fi : PCIe Window Mask (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1108 + 0x0200 × i + 0x20 × n



Bit	Symbol	Function	R/W
11:0	PWM1100[11:0]	PWMask[11:0] Fixed 0xFFF	R
31:12	PWM3112[19:0]	PWMask[31:12] Set the window in the area of the set number of bits from the address set in the PWBase register. Set 1 from the lower bit.	R/W

The PCI_EP_PWMASKLn_Fi register is for setting windows for address conversion in access from AXI to PCIe. The window is set as the area corresponding to the number of set bits from the address set in the PWBase register.

56.4.1.74 PCI_EP_PWMASKUn_Fi : PCIe Window Mask (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x110C + 0x0200 × i + 0x20 × n

Bit position:	31	30	0
Bit field:	—	PWM6232[30:0]	
Value after reset:	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
30:0	PWM6232[30:0]	PWMask[62:32] Set the window in the area of the set number of bits from the address set in the PWBase register. Set 1 from the lower bit.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The PCI_EP_PWMASKUn_Fi register is for setting windows for address conversion in access from AXI to PCIe. The window is set as the area corresponding to the number of set bits from the address set in the PWBase register.

56.4.1.75 PCI_EP_PDESTLn_Fi : PCIe Destination (Lower) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1110 + 0x0200 × i + 0x20 × n

Bit position:	31	12	11	0
Bit field:	PD3112[19:0]		PD1100[11:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
11:0	PD1100[11:0]	PDest[11:0] fixed 0x000	R
31:12	PD3112[19:0]	PDest[31:12] Set the window base point in the address space on the PCIe side. The configurable area is the 4k boundary.	R/W

The PCI_EP_PDESTLn_Fi register is for setting windows for address conversion in access from AXI to PCIe. The base address of the window in the address space on the AXI is set. The areas are set in 4-KB boundaries.

56.4.1.76 PCI_EP_PDESTUn_Fi : PCIe Destination (Upper) Register n (Function #i) (n = 0 to 7, i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1114 + 0x0200 × i + 0x20 × n

Bit position:	31	0
Bit field:	PD6332[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	PD6332[31:0]	PDest[63:32] Set the window base point in the address space on the PCIe side.	R/W

The PCI_EP_PDESTUn_Fi register is for setting windows for upper address conversion in access from AXI to PCIe.

56.4.2 PCI Express Configuration Registers (Type0: Endpoint)

56.4.2.1 PCI_EP_VID_Fi : Vendor and Device ID (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6000 + 0x1000 × i

Bit position:	31																16	15														0
Bit field:	DID[15:0]																	VID[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	VID[15:0]	Vendor ID Used by the manufacturer specified by the Vendor ID to identify the manufactured device. Set a fixed value.	R/W*1
31:16	DID[15:0]	Device ID Represents the manufacturer of the device. Set a fixed value.	R/W*1

Note 1. The access type from PCIe is read-only.

The PCI_EP_VID_Fi register indicates the vendor and device ID.

Table 56.71 Valid reset signal

Reset signal	Device ID	Vendor ID
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—
FLR	—	—

56.4.2.2 PCI_EP_COM_STA_Fi : Command and Status (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6004 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DPE	SSE	RMA	RTA	STA	—	—	MDPE	—	—	—	CL	IS	—	—	IR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ID	—	SERRE	—	PER	—	—	—	BME	MSE	IOSE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IOSE	IO Space Enable 0 fixed. It does not support access to the I/O space.	R
1	MSE	Memory Space Enable Controls whether to respond to accesses to memory space.	R/W
2	BME	Bus Master Enable Controls whether to operate as a bus master.	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	PER	Parity Error Response Controls the behavior when sending/receiving PoisonedTLP. Error logging to the Detected Parity Error field of the Status register, the Device Status register of PCI Express Capability, and the Uncorrectable Error Status register of Advanced Error Reporting Capability is performed regardless of the setting of this bit.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	SERRE	SERR# Enable When set to 1, the Root Complex is notified of Non-Fatal Errors and Fatal Errors by Message Transaction. Even if this bit is not set, if a bit related to Error Reporting in the Device Control register of PCI Express Capability is set to 1, an error is notified to the Root Complex by Message Transaction.	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	ID	Interrupt Disable Suppress transmission of Assert_INTx Message. When set to 1, INTx Message cannot be sent. If this bit is set to 1 while an Assert_INTx Message has been sent, a Deassert_INTx Message must be sent.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	IR	Immediate Readiness Fixed 0	R
18:17	—	These bits are read as 0. The write value should be 0.	R/W
19	IS	Interrupt Status Indicates the interrupt status of the device.	R
20	CL	Capabilities List This bit is fixed to 1 because all PCI Express devices must implement PCI Express Capability.	R
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	MDPE	Master Data Parity Error Set to 1 when the Parity Error Response bit is set to 1 and the following two conditions occur: 1. Requester (BME) received a Poisoned Completion TLP. 2. Requester (BME) sent Poisoned Write Request TLP. This bit is not set to 1 if the Parity Error Response bit is 0.	R/W
26:25	—	These bits are read as 0. The write value should be 0.	R/W
27	STA	Signaled Target Abort Set to 1 when the Completion Status field has sent a Completion of Completer Abort.	R/W*1
28	RTA	Received Target Abort Set to 1 when the Completion Status field receives a Completion for Completer Abort.	R/W*1
29	RMA	Received Master Abort Set to 1 when the Completion Status field receives a Completion of an Unsupported Request.	R/W*1
30	SSE	Signaled System Error Set to 1 when the SERR Enable bit is 1 and this macro sends an ERR_FATAL or ERR_NONFATAL Message.	R/W*1
31	DPE	Detected Parity Error Set to 1 when a Poisoned TLP is received, regardless of the setting of the Parity Error Response bit.	R/W*1

Note 1. Writing 1 from PCIe clears the bit. Writing 0 from PCIe is ignored.

The PCI_EP_COM_STA_Fi register specifies the command and the status.

Table 56.72 Valid reset signal

Reset signal	Detected Parity Error	Signaled System Error	Received Master Abort	Received Target Abort	Signaled Target Abort	Master Data Parity Error	Interrupt Disable	SERR# Enable	Parity Error Response	Bus Master Enable	Memory Space Enable
RST_LOAD_B	—	—	—	—	—	—	—	—	—	—	—
RST_RSM_B	—	—	—	—	—	—	—	—	—	—	—
RST_CFG_B	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FLR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

56.4.2.3 PCI_EP_RID_CC_Fi : Revision ID and Class Code (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6008 + 0x1000 × i

Bit position:	31																							8	7							0
Bit field:	CC[23:0]																								RID[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RID[7:0]	Revision ID An 8-bit ID used to represent the revision of a specific device specified by Vendor ID and Device ID. Please set a fixed value.	R/W*1
31:8	CC[23:0]	Class Code This is information that indicates the device type and function, and the definition of the value is divided into the following two bytes by the PCI SIG. 31:24 base class: 23:16 sub-class: 15:8 programming interface: Please set a fixed value.	R/W*1

Note 1. The access type from PCIe is read-only.

The PCI_EP_RID_CC_Fi register indicates the revision ID and the class code.

Table 56.73 Valid reset signal

Reset signal	Class Code	Capabilities List
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—
FLR	—	—

56.4.2.4 PCI_EP_CL_HT : Cache Line and Header Type (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x600C + 0x1000 × i

Bit position:	31								23								16							7							0
Bit field:	—	—	—	—	—	—	—	—	HT[7:0]								—	—	—	—	—	—	—	—	CLS[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CLS[7:0]	Cache Line Size Implemented as a read/write field for legacy compatibility, but the value set has no effect on this device.	R*1
15:8	—	These bits are read as 0.	R
23:16	HT[7:0]	Header Type Multi-Function Device bit = 1 (0x80 fixed)	R
31:24	—	These bits are read as 0.	R

Note 1. The access type from PCIe is read-write.

The PCI_EP_CL_HT register indicates the cache line size and the header type.

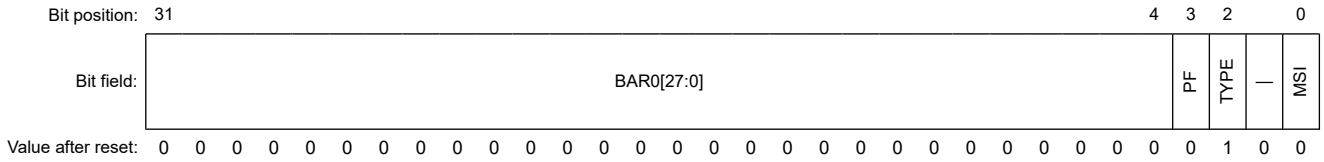
Table 56.74 Valid reset signal

Reset signal	Cache Line Size
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓
FLR	✓

56.4.2.5 PCI_EP_BAR0_Fi : Base Address Register 0 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6010 + 0x1000 × i



Bit	Symbol	Function	R/W
0	MSI	Memory Space Indicator Fixed to 0 to indicate the memory space.	R/W*1
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TYPE	Type 1 fixed to use 64-bit address 0: 32-bit address 1: 64-bit address	R/W*1
3	PF	Prefetch 0: Disable 1: Enable	R/W*1
31:4	BAR0[27:0]	Base Address Register 0 Indicates the base address. Depending on the size of the address block required, some lower bits of this field are implemented as Read Only bits fixed at 0. In this macro, the Read Only bits are located in the Base Address 0 Mask register (Offset: 0x0A0).	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_BAR0_Fi register forms a 64-bit memory space in combination with Base Address Register 1 (BAR1).

Table 56.75 Valid reset signal (1 of 2)

Reset signal	Base Address Register 0	Prefetch	Type	Memory Space Indicator
RST_LOAD_B	—	✓	✓	✓
RST_RSM_B	—	—	—	—
RST_CFG_B	✓	—	—	—

Table 56.75 Valid reset signal (2 of 2)

Reset signal	Base Address Register 0	Prefetch	Type	Memory Space Indicator
FLR	✓	—	—	—

56.4.2.6 PCI_EP_BAR1_Fi : Base Address Register 1 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6014 + 0x1000 × i

Bit position:	31																													0
Bit field:	BAR1[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	BAR1[31:0]	Base Address Register 1 (64-bit Upper Address) Shows the upper 32 bits of the base address.	R/W

The PCI_EP_BAR1_Fi register forms a 64-bit memory space in combination with Base Address Register 0 (BAR0).

Table 56.76 Valid reset signal

Reset signal	Base Address Register 1
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓
FLR	✓

56.4.2.7 PCI_EP_BAR2_Fi : Base Address Register 2 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6018 + 0x1000 × i

Bit position:	31																										4	3	2		0
Bit field:	BAR2[27:0]																										PF	TYPE	—	MSI	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	

Bit	Symbol	Function	R/W
0	MSI	Memory Space Indicator Fixed to 0 to indicate the memory space.	R
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TYPE	Type 1 fixed to use 64-bit address 0: 32-bit address 1: 64-bit address	R/W*1
3	PF	Prefetch 0: Disable 1: Enable	R/W*1
31:4	BAR2[27:0]	Base Address Register 2 Indicates the base address. Depending on the size of the address block required, some lower bits of this field are implemented as Read Only bits fixed at 0. In this macro, the Read Only bits are located in the Base Address 0 Mask register (Offset: 0x0A8).	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_BAR2_Fi register forms a 64-bit memory space in combination with Base Address Register 3 (BAR3).

Table 56.77 Valid reset signal

Reset signal	Base Address Register 2	Prefetch	Type	Memory Space Indicator
RST_LOAD_B	—	✓	✓	✓
RST_RSM_B	—	—	—	—
RST_CFG_B	✓	—	—	—
FLR	✓	—	—	—

56.4.2.8 PCI_EP_BAR3_Fi : Base Address Register 3 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x601C + 0x1000 × i

Bit position:	31																														0
Bit field:	BAR3[31:0]																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	BAR3[31:0]	Base Address Register 3 (64-bit Upper Address) Shows the upper 32 bits of the base address.	R/W

The PCI_EP_BAR3_Fi register forms a 64-bit memory space in combination with Base Address Register 2 (BAR2).

Table 56.78 Valid reset signal

Reset signal	Base Address Register 3
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓
FLR	✓

56.4.2.9 PCI_EP_BAR4_Fi : Base Address Register 4 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6020 + 0x1000 × i

Bit position:	31																											4	3	2		0
Bit field:	BAR4[27:0]																											PF	TYPE	—	MSI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Bit	Symbol	Function	R/W
0	MSI	Memory Space Indicator Fixed to 0 to indicate the memory space.	R
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TYPE	Type 1 fixed to use 64-bit address 0: 32-bit address 1: 64-bit address	R/W*1

Bit	Symbol	Function	R/W
3	PF	Prefetch 0: Disable 1: Enable	R/W*1
31:4	BAR4[27:0]	Base Address Register 4 Indicates the base address. Depending on the size of the address block required, some lower bits of this field are implemented as Read Only bits fixed at 0. In this macro, the Read Only bits are located in the Base Address 0 Mask register (Offset: 0x0B0).	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_BAR4_Fi register forms a 64-bit memory space in combination with Base Address Register 5 (BAR5).

Table 56.79 Valid reset signal

Reset signal	Base Address Register 4	Prefetch	Type	Memory Space Indicator
RST_LOAD_B	—	✓	✓	✓
RST_RSM_B	—	—	—	—
RST_CFG_B	✓	—	—	—
FLR	✓	—	—	—

56.4.2.10 PCI_EP_BAR5_Fi : Base Address Register 5 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6024 + 0x1000 × i

Bit position:	31																													0
Bit field:	BAR5[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	BAR5[31:0]	Base Address Register 5 (64-bit Upper Address) Shows the upper 32 bits of the base address.	R/W

The PCI_EP_BAR5_Fi register forms a 64-bit memory space in combination with Base Address Register 4 (BAR4).

Table 56.80 Valid reset signal

Reset signal	Base Address Register 5
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓
FLR	✓

56.4.2.11 PCI_EP_SUBSID_Fi : Subsystem ID (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x602C + 0x1000 × i

Bit position:	31															16	15													0
Bit field:	SID[15:0]																SVID[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SVID[15:0]	Subsystem Vendor ID A 16-bit ID representing the manufacturer of the add-in card or subsystem that contains this device. Set fixed value	R/W*1
31:16	SID[15:0]	Subsystem ID Sub System Vendor A 16-bit ID used to identify subsystems manufactured by the manufacturer specified by the ID. Set fixed value	R/W*1

Note 1. The access type from PCIe is read-only.

The PCI_EP_SUBSID_Fi register indicates the subsystem ID.

Table 56.81 Valid reset signal

Reset signal	Subsystem ID	Subsystem Vendor ID
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—
FLR	—	—

56.4.2.12 PCI_EP_CP_Fi : Capabilities Pointer (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6034 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CP[7:0]	Capabilities Pointer Capability implementation start address 0x40 PCI Power Management Capability is implemented from 0x40.	R
31:8	—	These bits are read as 0.	R

The PCI_EP_CP_Fi register indicates start address for implementing the capability.

56.4.2.13 PCI_EP_INT_Fi : Interrupt Register (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x603C + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IP[7:0]								IL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	IL[7:0]	Interrupt Line Indicates which line of the system's interrupt lines the device's interrupt output is connected to as a result of system configuration. Values are set by system initialization programs and read by device drivers and the OS as needed. This is a software register and there is no relation between device operation and register value.	R/W

Bit	Symbol	Function	R/W
15:8	IP[7:0]	Interrupt Pin Specifies the message to be issued on the INTX_EP_Fi bit in PCIE_INTX register. 0x01: Assert_INTA# (default) 0x02: Assert_INTB# 0x03: Assert_INTC# 0x04: Assert_INTD# Others: Reserved	R/W*1
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_INT_Fi register assigns an interrupt to the terminal.

Table 56.82 Valid reset signal

Reset signal	Interrupt Pin	Interrupt Line
RST_LOAD_B	✓	—
RST_RSM_B	—	—
RST_CFG_B	—	✓
FLR	—	✓

56.4.2.14 PCI_EP_PMC_Fi : PM Capabilities (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6040 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PMES[4:0]					D2S	D1S	AUXC[2:0]			DSI	IRORTD0	—	V[2:0]		
Value after reset:	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NCP[7:0]								CID[7:0]							
Value after reset:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
7:0	CID[7:0]	Capability ID Indicates the PCI Power Management Capability. It is fixed at 0x01.	R
15:8	NCP[7:0]	Next Capability Pointer Indicates the MSI Capability start address.	R
18:16	V[2:0]	Version 011b is fixed. PCI Power Management Interface Specification Rev .1.2	R/W*1
19	—	This bit is read as 0. The write value should be 0.	R/W
20	IRORTD0	Immediate_Readiness_on_Return_to_D0 0 fixed	R
21	DSI	Device Specific Initialization Indicates whether or not DSI (Device Specific Initialization) is used. 0: Not supported 1: Support	R/W*1

Bit	Symbol	Function	R/W
24:22	AUXC[2:0]	AUX_Current Indicates the 3.3Vaux auxiliary current (the maximum current value supplied from the auxiliary power supply). Read value returns 000b. (AUX not supported) 0 0 0: 0 (self powered) 0 0 1: 55 mA 0 1 0: 100 mA 0 1 1: 160 mA 1 0 0: 220 mA 1 0 1: 250 mA 1 1 0: 320 mA 1 1 1: 375 mA	R/W*1
25	D1S	D1 Support Indicates whether the D1 Power Management State is supported. 0: Not supported 1: Support	R/W*1
26	D2S	D2 Support Indicates whether the D2 Power Management State is supported. 0: Not supported 1: Support	R/W*1
31:27	PMES[4:0]	PME Support xxxx1: Supports D0 xxx1x: Supports D1 xx1xx: Supports D2 x1xxx: Support D3hot 1xxxx: D3cold supported (not supported)	R/W*1

Note 1. The access type from PCIe is read-only.

The PCI_EP_PMC_Fi register indicates various support information.

Table 56.83 Valid reset signal

Reset signal	PME Support	D2 Support	D1 Support	AUX_Current	DSI	Version
RST_LOAD_B	✓	✓	✓	✓	✓	✓
RST_RSM_B	—	—	—	—	—	—
RST_CFG_B	—	—	—	—	—	—
FLR	—	—	—	—	—	—

56.4.2.15 PCI_EP_PMSC_Fi : PM Status/Control (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6044 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PMES	—	—	—	—	—	—	PMEE	—	—	—	—	NSR	—	PS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
1:0	PS[1:0]	PowerState Set the PCI Device State. 0 0: D0 (Default) 0 1: D1 (Do not set) 1 0: D2 (Do not set) 1 1: D3hot	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	NSR	No_Soft_Reset Indicates that internal reset is not performed inside the device at the power state transition from D3hot to D0.	R
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PMEE	PME Enable Controls PME assertions. If 1, PME assertion is enabled. Assert PME if PME_Status is set at this time. PCI Express perf orms Link Wake-up processing, and then performs PME assert processing by sending PM_PME Message. The specifications are as f ollows depending on the value of the PM Capabilities register PME Status[4] (PME Support in D3cold). PME Status[4] = 1: ● Reset: RST_RSM_B ● EP UDL: RW ● EP PCIe: RWS PME Status[4] = 0: ● Reset: RST_RSM_B ● EP UDL: RW ● EP PCIe: RW	R/W
14:9	—	These bits are read as 0. The write value should be 0.	R/W
15	PMES	PME Status Indicates that a PME assert factor has occurred. 1 indicates that there is a PME assertion factor. Cleared by writing 1.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_PMSC_Fi register indicates and controls the PME status.

Table 56.84 Valid reset signal

Reset signal	PME Enable	PowerState
RST_LOAD_B	—	—
RST_RSM_B	✓	—
RST_CFG_B	—	✓
FLR	✓	✓

56.4.2.16 PCI_EP_PCIEC_Fi : PCI Express Capability (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6060 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	IMN[4:0]					—	DPT[3:0]				CV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NCP[7:0]								CID[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CID[7:0]	Capability ID Indicates PCI Express Capability. 0x10 fixed.	R
15:8	NCP[7:0]	Next Capability Pointer Indicates that this Capability List is the final list. 0x00 fixed.	R
19:16	CV[3:0]	Capability Version Indicates the version of PCI Express Capability Structure. 0x2 fixed.	R/W*1
23:20	DPT[3:0]	DevicePort Type Indicates that it is a PCI Express Endpoint device. 0x0: PCI Express Endpoint device 0x1: Legacy PCI Express Endpoint device 0x4: Root Port of PCI Express Root Complex (Default) 0x5: Upstream Port of PCI Express Switch 0x6: Downstream Port of PCI Express Switch 0x7: PCI Express-to-PCI/PCI-X Bridge 0x8: PCI/PCI-X-to-PCI Express Bridge 0x9: Root Complex Integrated Endpoint Device 0xA: Root Complex Event Collector Others: Reserved	R/W*1
24	—	This bit is read as 0. The write value should be 0.	R/W
29:25	IMN[4:0]	Interrupt Message Number Indicates the MSI vector used in interrupt messages associated with any status bit in this Capability Structure	R/W*1
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_PCIEC_Fi register indicates the PCIe capability.

Table 56.85 Valid reset signal

Reset signal	Interrupt Message Number	Device/Port Type	Capability Version
RST_LOAD_B	✓	✓	✓
RST_RSM_B	—	—	—
RST_CFG_B	—	—	—
FLR	—	—	—

56.4.2.17 PCI_EP_DEVC_Fi : Device Capabilities (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6064 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	FLRC	CSPLS[1:0]		CSPLV[7:0]								—	—
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RBER	—	—	—	EL1AL[2:0]			EL0AL[2:0]			ETFS	—	—	MPSS[2:0]		
Value after reset:	1	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	MPSS[2:0]	Max_Payload_Size Supported 0 0 0: 128 B max payload size 0 0 1: 256 B max payload size (Default) 0 1 0: 512 B max payload size 0 1 1: 1024 B max payload size 1 0 0: 2048 B max payload size 1 0 1: 4096 B max payload size 1 1 0: Reserved 1 1 1: Reserved	R/W*1
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	ETFS	Extended Tag Field Supported Extended Tag support 0: 5-bit Tag field supported (Default) 1: 8-bit Tag field supported	R/W*1
8:6	EL0AL[2:0]	Endpoint L0 Acceptable Latency 0 0 0: max 64 ns 0 0 1: max 128 ns 0 1 0: max 256 ns 0 1 1: max 512 ns 1 0 0: max 1 µs 1 0 1: max 2 µs 1 1 0: max 4 µs 1 1 1: No limit (Default)	R/W*1
11:9	EL1AL[2:0]	Endpoint L1 Acceptable Latency 0 0 0: max 1 µs 0 0 1: max 2 µs 0 1 0: max 4 µs 0 1 1: max 8 µs 1 0 0: max 16 µs 1 0 1: max 32 µs 1 1 0: max 64 µs 1 1 1: No limit (Default)	R/W*1
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	RBER	Role-Based Error Reporting Set to 1 if you are implementing the Error Reporting ECN feature. Fixed to 1 for PCI Express Base Spec 1.1 and later.	R
17:16	—	These bits are read as 0. The write value should be 0.	R/W
25:18	CSPLV[7:0]	Captured Slot Power Limit Value Indicates the Slot Power Limit (Watt) value. Set by the received Set_Slot_Power_Limit Message.	R/W*1
27:26	CSPLS[1:0]	Captured Slot Power Limit Scale Indicates the Scale of the Captured Slot Power Limit Value. Set by the received Set_Slot_Power_Limit Message.	R/W*1
28	FLRC	Function Level Reset Capability Set when the Function Level Reset function is implemented.	R/W*1
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_DEVC_Fi register indicates the device capability.

Table 56.86 Valid reset signal

Reset signal	Function Level Reset Capability	Captured Slot Power Limit Scale	Captured Slot Power Limit Value	Endpoint L1 Acceptable Latency	Endpoint L0 Acceptable Latency	Extended Tag Field Supported	Max_Payload_ Size Supported
RST_LOAD_B	✓	—	—	✓	✓	✓	✓
RST_RSM_B	—	—	—	—	—	—	—
RST_CFG_B	—	✓	✓	—	—	—	—

56.4.2.18 PCI_EP_DEVCS_Fi : Device Control/Status (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6068 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	TP	—	URD	FED	NFED	CED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IFLR	MRRS[2:0]			—	—	—	—	MPS[2:0]			ERO	URRE	FERE	NFERE	CERE
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	CERE	Correctable Error Reporting Enable Controls the generation of ERR_COR Messages. 1 enables Message generation.	R/W
1	NFERE	Non-Fatal Error Reporting Enable Controls generation of ERR_NONFATAL Message. 1 enables Message generation.	R/W
2	FERE	Fatal Error Reporting Enable Controls the generation of ERR_FATAL Messages. 1 enables Message generation.	R/W
3	URRE	Unsupported Request Reporting Enable ERR_NONFATAL due to Unsupported Request detection Or control the generation of ERR_FATAL Message. 1 enables Message generation.	R/W
4	ERO	Enable Relaxed Ordering Set whether or not to use Relaxed Ordering as a Requester. 0: Not supported 1: Support (Default)	R/W
7:5	MPS[2:0]	Max_Payload_Size Set Max_Payload_Size. 0 0 0: 128 B max payload size (Default) 0 0 1: 256 B max payload size 0 1 0: 512 B max payload size 0 1 1: 1024 B max payload size 1 0 0: 2048 B max payload size 1 0 1: 4096 B max payload size 1 1 0: Reserved 1 1 1: Reserved	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
14:12	MRRS[2:0]	Max_Read_Request_Size Set Max_Read_request_Size. 0 0 0: 128 B max read request size 0 0 1: 256 B max read request size 0 1 0: 512 B max read request size (Default) 0 1 1: 1024 B max read request size 1 0 0: 2048 B max read request size 1 0 1: 4096 B max read request size 1 1 0: Reserved 1 1 1: Reserved	R/W
15	IFLR	Initiate Function Level Reset A write of 1 initiates Function Level Reset to the Function. The value read by software from this bit is always 0.	R/W
16	CED	Correctable Error Detected Indicates that a correctable error was detected. Indicates that an Error is detected in 1.	R/W*1

Bit	Symbol	Function	R/W
17	NFED	Non-Fatal Error Detected Indicates that a Non-Fatal Error was detected. Indicates that an Error is detected in 1.	R/W*1
18	FED	Fatal Error Detected Indicates that a Fatal Error was detected. Indicates that an Error is detected in 1.	R/W*1
19	URD	Unsupported Request Detected Indicates that an Unsupported Request Error was detected. Indicates that an Error is detected in 1.	R/W*1
20	—	This bit is read as 0. The write value should be 0.	R/W
21	TP	Transaction Pending Indicates that the transaction is Pending because Completion has not been received for the sent Non-posted Request. 1 indicates Pending.	R
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 from PCIe clears the bit. Writing 0 from PCIe is ignored.

The PCI_EP_DEVCS_Fi register controls the device and indicates the device status.

Table 56.87 Valid reset signal

Reset signal	Unsupported Request Detected	Fatal Error Detected	Non-Fatal Error Detected	Correctable Error Detected	Initiate Function Level Reset	Max_Read_Request_Size	Max_Payload_Size	Enable Relaxed Ordering	Unsupported Request Reporting Enable	Fatal Error Reporting Enable	Non-Fatal Error Reporting Enable	Correctable Error Reporting Enable
RST_LOAD_B	—	—	—	—	—	—	—	—	—	—	—	—
RST_RSM_B	—	—	—	—	—	—	—	—	—	—	—	—
RST_CFG_B	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FLR	✓	✓	✓	✓	—	✓	—	✓	✓	✓	✓	✓

56.4.2.19 PCI_EP_LINKC_Fi : Link Capabilities (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x606C + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PN[7:0]								—	ASPM OC	—	—	—	CPM	L1EL[2:1]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	L1EL	L0EL[2:0]			ASPMS[1:0]		MLW[5:0]						MLS[3:0]			
Value after reset:	1	1	0	0	1	1	0	0	0	0	*1	*1	0	0	1	1

Bit	Symbol	Function	R/W
3:0	MLS[3:0]	Max Link Speed 0x1: 2.5 GT/s Link speed supported 0x2: 5.0 GT/s and 2.5 GT/s Link speeds supported 0x3: 8.0 GT/s Link speed supported (Default) 0x4: 16.0 GT/s Link speed supported (prohibited) Others: Reserved	R/W*2

Bit	Symbol	Function	R/W
9:4	MLW[5:0]	Maximum Link Width 0x00: Reserved 0x01: x1 (Default when using ch1.) 0x02: x2 (Default when using ch0. Don't set for ch1.) 0x04: x4 (prohibited) 0x08: x8 (prohibited) 0x0C: x12 (prohibited) 0x10: x16 (prohibited) 0x20: x32 (prohibited)	R/W*2
11:10	ASPMS[1:0]	Active State Power Management (ASPM) Support 0 0: Reserved 0 1: L0s Entry Supported 1 0: L1 Entry Supported 1 1: L0s and L1 Entry Supported (Default)	R/W*2
14:12	L0EL[2:0]	L0s Exit Latency 0 0 0: Less than 64 ns 0 0 1: 64 ns to less than 128 ns 0 1 0: 128 ns to less than 256 ns 0 1 1: 256 ns to less than 512 ns 1 0 0: 512 ns to less than 1 µs (Default) 1 0 1: 1 µs to less than 2 µs 1 1 0: 2 µs to 4 µs 1 1 1: More than 4 µs	R/W*2
17:15	L1EL[2:0]	L1 Exit Latency 0 0 0: Less than 1 µs 0 0 1: 1 µs to less than 2 µs 0 1 0: 2 µs to less than 4 µs 0 1 1: 4 µs to less than 8 µs 1 0 0: 8 µs to less than 16 µs 1 0 1: 16 µs to less than 32 µs (Default) 1 1 0: 32 µs to 64 µs 1 1 1: More than 64 µs	R/W*2
18	CPM	Clock Power Management Indicates whether the CLKREQ# mechanism is supported in L1 and L2/L3 Ready. 0: CLKREQ# mechanism not supported. 1: Supports CLKREQ# mechanism. (Setting prohibited)	R/W*2
21:19	—	These bits are read as 0. The write value should be 0.	R/W
22	ASPMOC	ASPM Optionality Compliance 1 fixed	R/W*2
23	—	This bit is read as 0. The write value should be 0.	R/W
31:24	PN[7:0]	Port Number Indicates the port number of PCI Express Link.	R/W*2

Note 1. Depends on the channel setting.
Note 2. The access type from PCIe is read-only.

The PCI_EP_LINKC_Fi register indicates the link capabilities.

Table 56.88 Valid reset signal

Reset signal	ASPM Optionality Compliance	Clock Power Management	L1 Exit Latency	L0s Exit Latency	Active State Power Management Support	Maximum Link Width	Max Link Speed
RST_LOAD_B	✓	✓	✓	✓	✓	✓	✓
RST_RSM_B	—	—	—	—	—	—	—
RST_CFG_B	—	—	—	—	—	—	—
FLR	—	—	—	—	—	—	—

56.4.2.20 PCI_EP_LINKCS_Fi : Link Control/Status (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6070 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DLLLA	SCC	—	—	NLW[5:0]						CLS[3:0]			
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	HAWD	ECPM	ES	CCC	—	—	RCB	—	ASPMC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
1:0	ASPMC[1:0]	Active State Power Management (ASPM) Control Sets the permission level for Active State Power Management. 0 0: Disabled (Default) 0 1: L0s Entry Supported 1 0: L1 Entry Supported 1 1: L0s and L1 Entry Supported	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	RCB	Read Completion Boundary (RCB) 0: 64 bytes 1: 128 bytes (Default)	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	CCC	Common Clock Configuration Sets whether the Common Reference Clock is used. Only Common Reference Clock is supported for Single Link. 0: Provided by non-Common Reference Clock (Default) 1: Supplied by Common Reference Clock	R/W
7	ES	Extended Synch If set to 1, 4096 FTS Ordered-sets will be sent when transitioning from L0s to L0. Also, 1024 TS1 Ordered-sets are transmitted at the beginning of the Recovery state when transitioning from L1 to L0. The default value will be 0.	R/W
8	ECPM	Enable Clock Power Management Controls Enable/Disable of CLKREQ# mechanism in L1 and L2/L3 Ready.	R/W
9	HAWD	Hardware Autonomous Width Disable Controls Link width change. Reserved in Function #1 0: Enable of Link Width Change 1: Disable	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
19:16	CLS[3:0]	Current Link Speed 0x0: During reset period 0x1: 2.5 GT/s PCI Express Link 0x2: 5.0 GT/s PCI Express Link 0x3: 8.0 GT/s PCI Express Link 0x4: 16.0 GT/s PCI Express Link	R

Bit	Symbol	Function	R/W
25:20	NLW[5:0]	Negotiated Link Width Indicates the Link width established as a result of negotiation. 0x01: x1 0x02: x2 0x04: x4 0x08: x8 0x0C: x12 0x10: x16 0x20: x32 Others: Reserved	R
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	SCC	Slot Clock Configuration Indicates whether or not the reference clock supplied to the Connector is used when the Add-in Card is used. 0: Not using Connector Reference Clock 1: Use Connector Reference Clock (Default)	R/W*1
29	DLLLA	Data Link Layer Link Active 1 indicates that the Data Link Layer is in the Link Active state.	R
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_LINKCS_Fi register controls the link and indicates the link status.

Table 56.89 Valid reset signal

Reset signal	Slot Clock Configuration	Hardware Autonomous Width Disable	Enable Clock Power Management	Extended Synch	Common Clock Configuration	Read Completion Boundary	Active State Power Management Control
RST_LOAD_B	✓	—	—	—	—	—	—
RST_RSM_B	—	—	—	—	—	—	—
RST_CFG_B	—	✓	✓	✓	✓	✓	✓
FLR	—	—	—	—	—	—	—

56.4.2.21 PCI_EP_DEVC2_Fi : Device Capabilities 2 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6084 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CTDS	CTRS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Bit	Symbol	Function	R/W
3:0	CTRS[3:0]	Completion Timeout Ranges Supported Set Completion Timeout Range. Range A: 50 µs to 10 ms Range B: 10 ms to 250 ms Range C: 250 ms to 4 s Range D: 4 s to 64 s The above four patterns are determined, and the following combinations can be set. The initial value described in UM is 0x2, but please change the initial value according to the installed system/device performance, or specify the initial value as a product request. 0x0: Program setting Not supported 0x1: Range A 0x2: Range B 0x3: Ranges A and B 0x6: Ranges B and C 0x7: Ranges A, B, and C 0xE: Ranges B, C and D 0xF: Ranges A, B, C, and D Others: Reserved	R/W*1
4	CTDS	Completion Timeout Disable Supported Set whether to support the Completion Timeout Disable function. 0: Not supported 1: Supported (default)	R/W*1
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_DEVC2_Fi register indicates the device capability.

Table 56.90 Valid reset signal

Reset signal	Completion Timeout Disable Supported	Completion Timeout Ranges Supported
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—
FLR	—	—

56.4.2.22 PCI_EP_DEVCS2_Fi : Device Control 2/Status 2 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6088 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CTD	CTV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CTV[3:0]	Completion Timeout Value Set Completion Timeout Range. The default 0x0 sets a longer time than the Base Spec's default 50 µsec lower limit, but this takes into account the Base Spec's recommended 10 msec lower limit. 0x0: 10 ms to 50 ms (default) 0x1: 50 µs to 100 µs 0x2: 1 ms to 10 ms 0x5: 16 ms to 55 ms 0x6: 65 ms to 210 ms 0x9: 260 ms to 900 ms 0xA: 1 s to 3.5 s 0xD: 4 s to 13 s 0xE: 17 s to 64 s Others: reserved (setting prohibited)	R/W
4	CTD	Completion Timeout Disable Setting this bit enables the Completion Timeout Disable function.	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_DEVCS2_Fi register controls the device and indicates the device status.

Table 56.91 Valid reset signal

Reset signal	Completion Timeout Disable	Completion Timeout Value
RST_LOAD_B	—	—
RST_RSM_B	—	—
RST_CFG_B	✓	✓
FLR	✓	✓

56.4.2.23 PCI_EP_LINKC2_Fi : Link Capabilities 2 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x608C + 0x1000 × i

Bit position:	31																							7						1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SLSV[6:0]							—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
7:1	SLSV[6:0]	Supported Link Speeds Vector The read value is undefined Indicates Support Link Speed Bit 0: 2.5 GT/s Bit 1: 5.0 GT/s Bit 2: 8.0 GT/s Bit 3: 16.0 GT/s (Setting prohibited) Bits 6 to 4: Reserved	R/W*1
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_LINKC2_Fi register indicates the link capabilities.

Table 56.92 Valid reset signal (1 of 2)

Reset signal	Supported Link Speeds Vector
RST_LOAD_B	✓

Table 56.92 Valid reset signal (2 of 2)

Reset signal	Supported Link Speeds Vector
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.24 PCI_EP_LINCS2_Fi : Link Control 2/Status 2 (Function #n) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6090 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	LER	EP3S	EP2S	EP1S	EQC	CDL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPD[3:0]				CSOS	EMC	TM[2:0]			—	HASD	ETC	TLS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
3:0	TLS[3:0]	Target Link Speed Set the Link Speed value for notification to the Root Complex during training. Reserved in Function #1 0x1: 2.5 GT/s Target Link Speed 0x2: 5.0 GT/s Target Link Speed 0x3: 8.0 GT/s Target Link Speed (Default) 0x4: 16.0 GT/s Target Link Speed (prohibited) Others: Reserved	R/W
4	ETC	Enter Compliance Setting to 1 allows transition to Compliance mode. At this time, the Link Speed will be the value set in the Target Link Speed field. Reserved in Function #1	R/W
5	HASD	Hardware Autonomous Speed Disable Controls the Link Speed Change function. Reserved in Function #1 0: Support of Link Speed Change (Enable) 1: Not Support of Link Speed Change (Disable)	R/W
6	—	These bits are read as 0. The write value should be 0.	R/W
9:7	TM[2:0]	Transmit Margin Adjust the voltage level of the Transmitter. 000b Normal operating range See 001b to 111b Base Spec Section 8.3.4. Reserved in Function #1	R/W
10	EMC	Enter Modified Compliance Setting bit for transmission of Modified Compliance pattern. Reserved in Function #1 0: Compliance Pattern (Default) 1: Modified Compliance Pattern	R/W
11	CSOS	Compliance SOS When this bit is set to 1, insert the SKP Ordered-set periodically during compliance pattern transmission. Reserved in Function #1	R/W

Bit	Symbol	Function	R/W
15:12	CPD[3:0]	Compliance Preset De-emphasis Set the De-emphasis Level in Polling.Compliance State. Reserved in Function #1 0x0: −6 dB (Default) 0x1: −3.5 dB	R/W
16	CDL	Current De-emphasis Level This is a Status register that indicates the De-emphasis Level during Gen3 operation. The initial value of this bit is 0 at reset and after Gen3-Linkup. However, although it indicates 1 at Gen1-Linkup, ignore this bit as it has no meaning in Gen1. 0: −6 dB (Default) 1: −3.5 dB	R
17	EQC	Equalization Complete When set to 1, this bit indicates that the Transmitter Equalization procedure has completed. The default value of this bit is 0. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Components that only support speeds below 8.0 GT/s are permitted to hardwire this bit to 0.	R
18	EP1S	Equalization Phase 1 Successful When set to 1, this bit indicates that Phase 1 of the Transmitter Equalization procedure has successfully completed. The default value of this bit is 0. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Components that only support speeds below 8.0 GT/s are permitted to hardwire this bit to 0.	R
19	EP2S	Equalization Phase 2 Successful When set to 1, this bit indicates that Phase 2 of the Transmitter Equalization procedure has successfully completed. The default value of this bit is 0. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Components that only support speeds below 8.0 GT/s are permitted to hardwire this bit to 0.	R
20	EP3S	Equalization Phase 3 Successful When set to 1, this bit indicates that Phase 3 of the Transmitter Equalization procedure has successfully completed. The default value of this bit is 0. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Components that only support speeds below 8.0 GT/s are permitted to hardwire this bit to 0.	R
21	LER	Link Equalization Request This bit is Set by hardware to request the Link equalization process to be performed on the Link. The default value of this bit is 0. For Multi-Function Upstream Port, this bit must be implemented in Function 0 and RsvdZ in other Functions. Components that only support speeds below 8.0 GT/s are permitted to hardwire this bit to 0.	R/W*1
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 from PCIe clears the bit. Writing 0 from PCIe is ignored.

The PCI_EP_LINCS2_Fi register controls the link and indicates the link status.

Table 56.93 Valid reset signal (1 of 2)

Reset signal	Link Equalization Request	Compliance Preset/De-emphasis	Compliance SOS	Enter Modified Compliance	Transmit Margin	Hardware Autonomous Speed Disable	Enter Compliance	Target Link Speed
RST_LOAD_B	—	—	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓	✓	✓

Table 56.93 Valid reset signal (2 of 2)

Reset signal	Link Equalization Request	Compliance Preset/De-emphasis	Compliance SOS	Enter Modified Compliance	Transmit Margin	Hardware Autonomous Speed Disable	Enter Compliance	Target Link Speed
RST_CFG_B	—	—	—	—	—	—	—	—
FLR	—	—	—	—	—	—	—	—

56.4.2.25 PCI_EP_BARMSK00L_Fi : Base Address Register Mask00 (Lower) (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60A0 + 0x1000 × i

Bit position:	31																												0
Bit field:	BARM00L[31:0]																												
Value after reset:	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	BARM00L[31:0]	Base Address Register Mask00 (Lower) Mask register for Base Address Register 0 (BAR0).	R/W*1

Note 1. The access type from PCIe is read-only.

The PCI_EP_BARMSK00L_Fi register indicates mask information for Base Address Register 0 (BAR0).

Table 56.94 Valid reset signal

Reset signal	Base Address Register Mask00 (Lower)
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.26 PCI_EP_BARMSK00U_Fi : Base Address Register Mask00 (Upper) (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60A4 + 0x1000 × i

Bit position:	31																												0
Bit field:	BARM00U[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	BARM00U[31:0]	Base Address Register Mask00 (Upper) Mask register for Base Address Register 1 (BAR1).	R/W*1

Note 1. The access type from PCIe is read-only.

The PCI_EP_BARMSK00U_Fi register indicates mask information for Base Address Register 1 (BAR1).

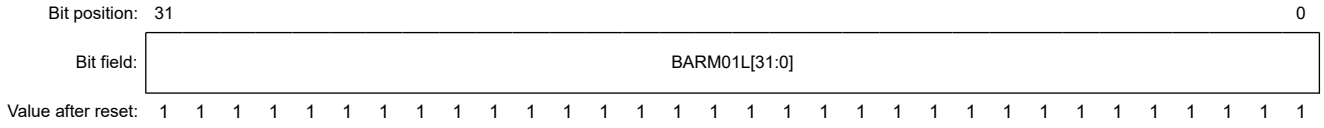
Table 56.95 Valid reset signal

Reset signal	Base Address Register Mask00 (Upper)
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.27 PCI_EP_BARMSK01L_Fi : Base Address Register Mask01 (Lower) (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60A8 + 0x1000 × i



Bit	Symbol	Function	R/W
31:0	BARM01L[31:0]	Base Address Register Mask01 (Lower) Mask register for Base Address Register 2 (BAR2).	R/W*1

Note 1. The access type from PCIe is read-only.

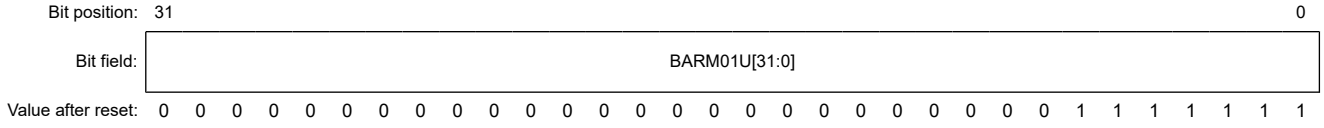
Table 56.96 Valid reset signal

Reset signal	Base Address Register Mask01 (Lower)
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.28 PCI_EP_BARMSK01U_Fi : Base Address Register Mask01 (Upper) (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60AC + 0x1000 × i



Bit	Symbol	Function	R/W
31:0	BARM01U[31:0]	Base Address Register Mask01 (Upper) Mask register for Base Address Register 3 (BAR3). 0x0000_007F (Function0) 0x0000_00FF (Function1)	R/W*1

Note 1. The access type from PCIe is read-only.

Table 56.97 Valid reset signal (1 of 2)

Reset signal	Base Address Register Mask01 (Upper)
RST_LOAD_B	✓

Table 56.97 Valid reset signal (2 of 2)

Reset signal	Base Address Register Mask01 (Upper)
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.29 PCI_EP_BARMSK02L_Fi : Base Address Register Mask02 (Lower) (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60B0 + 0x1000 × i

Bit position:	31																												0
Bit field:	BARM02L[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	BARM02L[31:0]	Base Address Register Mask02 (Lower) Mask register for Base Address Register 4 (BAR4).	R/W*1

Note 1. The access type from PCIe is read-only.

Table 56.98 Valid reset signal

Reset signal	Base Address Register Mask02 (Lower)
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.30 PCI_EP_BARMSK02U_Fi : Base Address Register Mask02 (Upper) (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60B4 + 0x1000 × i

Bit position:	31																												0
Bit field:	BARM02U[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	BARM02U[31:0]	Base Address Register Mask02 (Upper) Mask register for Base Address Register 5 (BAR5).	R/W*1

Note 1. The access type from PCIe is read-only.

Table 56.99 Valid reset signal

Reset signal	Base Address Register Mask02 (Upper)
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.31 PCI_EP_BSIZE00_01_Fi : Base Size 00/01 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60C8 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	BS01[9:0]										—	—	—	—	—	—	BS00[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	BS00[9:0]	Base Size 00 Sets the size of the TLP (DW Size) that will be accepted in Address Space 01 set in the Base Address Register and Base Address Mask Register. If a TLP with a packet length larger than the size set here is received (even if it is smaller than the Max Payload Size), a Completer Abort Error (CA) will be detected. Note that the default value is 0x000, which disables this function.	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	BS01[9:0]	Base Size 01 Sets the size of the TLP (DW Size) that will be accepted in Address Space 00 set in the Base Address Register and Base Address Mask Register. If a TLP with a packet length larger than the size set here is received (even if it is smaller than the Max Payload Size), a Completer Abort Error (CA) will be detected. Note that the default value is 0x000, which disables this function.	R/W*1
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_BSIZE00_01_Fi register sets the acceptable TLP size.

Table 56.100 Valid reset signal

Reset signal	Base Size 01	Base Size 00
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—
FLR	—	—

56.4.2.32 PCI_EP_BSIZE02_03_Fi : Base Size 02/03 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60CC + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	BS03[9:0]										—	—	—	—	—	—	BS02[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	BS02[9:0]	Base Size 02 Sets the size of the TLP (DW Size) that will be accepted in Address Space 01 set in the Base Address Register and Base Address Mask Register. If a TLP with a packet length larger than the size set here is received (even if it is smaller than the Max Payload Size), a Completer Abort Error (CA) will be detected. Note that the default value is 0x000, which disables this function.	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
25:16	BS03[9:0]	Base Size 03 Sets the size of the TLP (DW Size) that will be accepted in Address Space 01 set in the Base Address Register and Base Address Mask Register. If a TLP with a packet length larger than the size set here is received (even if it is smaller than the Max Payload Size), a Completer Abort Error (CA) will be detected. Note that the default value is 0x000, which disables this function.	R/W*1
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_BSIZE02_03_Fi register sets the acceptable TLP size.

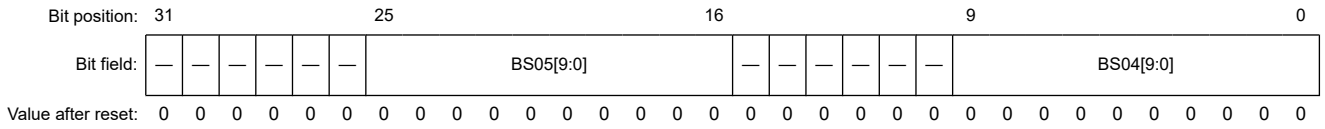
Table 56.101 Valid reset signal

Reset signal	Base Size 03	Base Size 02
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—
FLR	—	—

56.4.2.33 PCI_EP_BSIZE04_05_Fi : Base Size 04/05 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60D0 + 0x1000 × i



Bit	Symbol	Function	R/W
9:0	BS04[9:0]	Base Size 04 Sets the size of the TLP (DW Size) that will be accepted in Address Space 01 set in the Base Address Register and Base Address Mask Register. If a TLP with a packet length larger than the size set here is received (even if it is smaller than the Max Payload Size), a Completer Abort Error (CA) will be detected. Note that the default value is 0x000, which disables this function.	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	BS05[9:0]	Base Size 05 Sets the size of the TLP (DW Size) that will be accepted in Address Space 01 set in the Base Address Register and Base Address Mask Register. If a TLP with a packet length larger than the size set here is received (even if it is smaller than the Max Payload Size), a Completer Abort Error (CA) will be detected. Note that the default value is 0x000, which disables this function.	R/W*1
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_BSIZE04_05_Fi register sets the acceptable TLP size.

Table 56.102 Valid reset signal

Reset signal	Base Size 05	Base Size 04
RST_LOAD_B	✓	✓
RST_RSM_B	—	—
RST_CFG_B	—	—
FLR	—	—

56.4.2.34 PCI_EP_BSIZE06_Fi : Base Size 06 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60D4 + 0x1000 × i

Bit position:	31																9													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BS06[9:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	BS06[9:0]	Base Size 06 Sets the size of the TLP (DW Size) that will be accepted in Address Space 01 set in the Base Address Register and Base Address Mask Register. If a TLP with a packet length larger than the size set here is received (even if it is smaller than the Max Payload Size), a Completer Abort Error (CA) will be detected. Note that the default value is 0x000, which disables this function.	R/W*1
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_BSIZE06_Fi register sets the acceptable TLP size.

Table 56.103 Valid reset signal

Reset signal	Base Size 06
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.35 PCI_EP_TSUPPORT00_01_02_Fi : Type Supported 00/01/02 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60D8 + 0x1000 × i

Bit position:	31								23							16	15							8	7						0
Bit field:	—	—	—	—	—	—	—	—	TS02[7:0]								TS01[7:0]								TS00[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

Bit	Symbol	Function	R/W
7:0	TS00[7:0]	Type Supported 00 Set the Transaction Types that can be supported by Space00 (CFG_SPACE00_BASE). Bit 0: Memory read 32 bit Bit 1: Memory read 64 bit Bit 2: Memory read lock 32 bit Bit 3: Memory read lock 64 bit Bit 4: Memory write 32 bit Bit 5: Memory write 64 bit Bit 6: IO Read Bit 7: IO Write	R/W*1

Bit	Symbol	Function	R/W
15:8	TS01[7:0]	Type Supported 01 Set the Transaction Types that can be supported by Space00 (CFG_SPACE01_BASE). Bit 0: Memory read 32 bit Bit 1: Memory read 64 bit Bit 2: Memory read lock 32 bit Bit 3: Memory read lock 64 bit Bit 4: Memory write 32 bit Bit 5: Memory write 64 bit Bit 6: IO Read Bit 7: IO Write	R/W*1
23:16	TS02[7:0]	Type Supported 02 Set the Transaction Types that can be supported by Space00 (CFG_SPACE02_BASE). Bit 0: Memory read 32 bit Bit 1: Memory read 64 bit Bit 2: Memory read lock 32 bit Bit 3: Memory read lock 64 bit Bit 4: Memory write 32 bit Bit 5: Memory write 64 bit Bit 6: IO Read Bit 7: IO Write	R/W*1
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The access type from PCIe is read-only.

The PCI_EP_TSUPPORT00_01_02_Fi register indicates the transaction type which can be supported by the memory space.

Table 56.104 Valid reset signal

Reset signal	Type Supported 02	Type Supported 01	Type Supported 00
RST_LOAD_B	✓	✓	✓
RST_RSM_B	—	—	—
RST_CFG_B	—	—	—
FLR	—	—	—

56.4.2.36 PCI_EP_MSICAP_Fi : MSI Capability (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60E0 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	PVMC	AC64	MME[2:0]			MMC[2:0]			MSIE
Value after reset:	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NCP[7:0]								CID[7:0]							
Value after reset:	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
7:0	CID[7:0]	Capability ID Indicates MSI Capability. 0x05 fixed.	R
15:8	NCP[7:0]	Next Capability Pointer Indicates the starting address of the PCI Express Capability. Default: 0x60	R
16	MSIE	MSI Enable Controls whether or not to generate MSI Messages. 1 enables MSI generation.	R/W

Bit	Symbol	Function	R/W
19:17	MMC[2:0]	Multiple Message Capable 0 0 0: 1 0 0 1: 2 0 1 0: 4 0 1 1: 8 1 0 0: 16 1 0 1: 32 (default) 1 1 0: Reserved 1 1 1: Reserved	R
22:20	MME[2:0]	Multiple Message Enable Set the generation permission and number of Multiple Messages. Since 101b is specified in the Multiple Message Capable field, up to 101b (32) can be set as software settings. 0 0 0: 1 (default) 0 0 1: 2 0 1 0: 4 0 1 1: 8 1 0 0: 16 1 0 1: 32 1 1 0: Reserved 1 1 1: Reserved	R/W
23	AC64	64-bit Address Capable Indicates that a 64-bit Address MSI Message can be generated. 1 fixed.	R
24	PVMC	Per-vector masking capable Indicates support for MSI per-vector masking.	R
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_MSICAP_Fi register specifies the MSI Capability.

Table 56.105 Valid reset signal

Reset signal	Multiple Message Enable	MSI Enable
RST_LOAD_B	—	—
RST_RSM_B	—	—
RST_CFG_B	✓	✓
FLR	✓	✓

56.4.2.37 PCI_EP_MSGADR_Fi : Message Address (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60E4 + 0x1000 × i

Bit position:	31																										2	0
Bit field:	MA[29:0]																										—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
31:2	MA[29:0]	Message Address Set the MSI Message Destination Address [31:2].	R/W

The PCI_EP_MSGADR_Fi register sets the destination address of MSI messages.

Table 56.106 Valid reset signal

Reset signal	Message Address
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓
FLR	✓

56.4.2.38 PCI_EP_MSGUADR_Fi : Message Upper Address (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60E8 + 0x1000 × i

Bit position:	31																												0
Bit field:	MUA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MUA[31:0]	Message Upper Address Set the MSI Message Destination Address [63:32].	R/W

The PCI_EP_MSGUADR_Fi register sets the upper destination address of MSI messages.

Table 56.107 Valid reset signal

Reset signal	Mask Bits
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓
FLR	✓

56.4.2.39 PCI_EP_MSGDAT_Fi : Message Data (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60EC + 0x1000 × i

Bit position:	31																15												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MD[15:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MD[15:0]	Message Data Set Data[15:0] to be set in the MSI Message. If the Multiple Message Enable field is 011b (8), the Function (UDL) side can change the lower 3 bits of the Message data, and 8 types of vectors can be notified to the system as Message_Data. Similarly, in the case of 010b (4 pieces), the lower 2 bits can be changed by the function (UDL) side, and 4 types of vector can be changed. In the case of 001b (2 pieces), the lower 1 bit can be changed by the function (UDL) side 2 types of vector f or 000b (1 piece), 1 type of vector for the set data only. Note: Specify 0 for invalid high-order bits (in the above example of Multiple Message Enable setting 4, high-order bits other than the low-order 2 bits to be changed on the Function side) for the vector specification value on each Function side.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

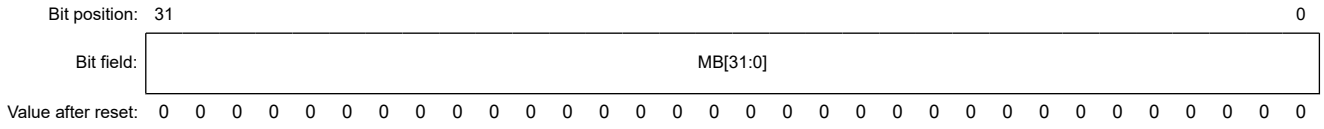
Table 56.108 Valid reset signal

Reset signal	Message Data
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓
FLR	✓

56.4.2.40 PCI_EP_MSKBIT_Fi : Mask Bits (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60F0 + 0x1000 × i



Bit	Symbol	Function	R/W
31:0	MB[31:0]	Mask Bits Bit set to 1 to mask message transmission of vector	R/W

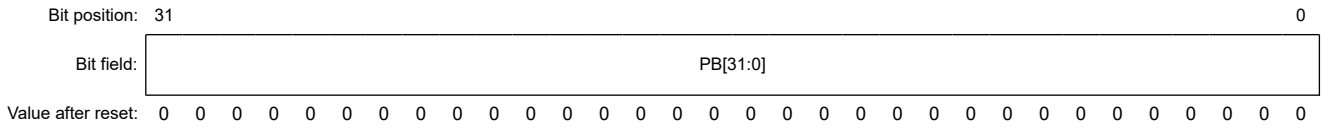
Table 56.109 Valid reset signal

Reset signal	Mask Bits
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓
FLR	✓

56.4.2.41 PCI_EP_PENDBIT_Fi : Pending Bits (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60F4 + 0x1000 × i



Bit	Symbol	Function	R/W
31:0	PB[31:0]	Pending Bits Indicates MSI Message pending status for each Function.	R/W

The PCI_EP_PENDBIT_Fi register indicates the pending state of an MSI message for the given function.

56.4.2.42 PCI_EP_ADVERC_Fi : Advanced Error Reporting Capability (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6100 + 0x1000 × i

Bit position:	31												20	19			16	15													0
Bit field:	NCO[11:0]													CV[3:0]				PCIEECID[15:0]													
Value after reset:	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
15:0	PCIEECID[15:0]	PCI Express Extended Capability ID Indicates the Advanced Error Reporting Capability. Default: 0x0001	R
19:16	CV[3:0]	Capability Version Indicates the version of the Capability Structure. Default: 0x01	R/W*1
31:20	NCO[11:0]	Next Capability Offset Indicates the starting address for DeviceSerialNumberCapability.	R

Note 1. The access type from PCIe is read-only.

The PCI_EP_ADVERC_Fi register indicates the advanced error reporting capability.

Table 56.110 Valid reset signal

Reset signal	Capability Version
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.43 PCI_EP_UNCESTS_Fi : Uncorrectable Error Status Register (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6104 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	URES	ECRCESO	MTLPS	ROSO	UCS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CASO	CTS	—	PTLPRS	—	—	—	—	—	—	—	DLPES	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	DLPES	Data Link Protocol Error Status*1 Indicates that a Sequence Number error was detected in the Data Link Layer. 0: No error detected 1: Error detected	R/W
11:5	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	PTLPRS	Poisoned TLP Received Status*1 Indicates that a Poisoned TLP (with payload and the EP field in the header is 1) has been received. 0: No error detected 1: Error detected	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	CTS	Completion Timeout Status*1 Indicates that the corresponding Completion was not received within the specified time after sending a Non-Posted Request. 0: No error detected 1: Error detected	R/W
15	CASO	Completer Abort Status (Optional)*1 Indicates that a Completion whose Completion Status is Completer Abort (CA) was returned after receiving a Non-Posted Request. 0: No error detected 1: Error detected	R/W
16	UCS	Unexpected Completion Status*1 Indicates that a Completion was received, but there is no record of a corresponding Non-Posted Request sent (the Transaction Descriptor did not match). 0: No error detected 1: Error detected	R/W
17	ROSO	Receiver Overflow Status (Optional)*1 Indicates that a TLP with a size larger than the free credits in the receive buffer was received. 0: No error detected 1: Error detected	R/W
18	MTLPS	Malformed TLP Status*1 Indicates that a Malformed TLP was received. 0: No error detected 1: Error detected	R/W
19	ECRCESO	ECRC Error Status (Optional)*1 Indicates that an ECRC Error was received. 0: No error detected 1: Error detected	R/W
20	URES	Unsupported Request Error Status*1 Indicates that an unsupported TLP was received. 0: No error detected 1: Error detected	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 clears the bit. Writing 0 is ignored.

The PCI_EP_UNCESTS_Fi register indicates the uncorrectable error status.

Table 56.111 Valid reset signal

Reset signal	Unsupported Request Error Status	ECRC Error Status	Malformed TLP Status	Receiver Overflow Status	Unexpected Completion Status	Completer Abort Status	Completion Timeout Status	Poisoned TLP Received Status	Data Link Protocol Error Status
RST_LOAD_B	—	—	—	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓	✓	✓	✓
RST_CFG_B	—	—	—	—	—	—	—	—	—
FLR	—	—	—	—	—	—	—	—	—

56.4.2.44 PCI_EP_UNCEMASK_Fi : Uncorrectable Error Mask Register (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6108 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	UREM	ECRCEMO	MTLPM	ROMO	UCM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CAMO	CTM	—	PTLPRM	—	—	—	—	—	—	—	DLPEM	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	DLPEM	Data Link Protocol Error Mask Masks error notifications to the Root Complex when a Data Link Protocol Error is detected. 0: No mask 1: Send Error Message, Mask Update of First Error Pointer	R/W
11:5	—	These bits are read as 0. The write value should be 0.	R/W
12	PTLPRM	Poisoned TLP Received Mask Masks error notification to Root Complex when Poisoned TLP Error is detected. 0: No mask 1: Send Error Message, record header to Header Log register, mask Update of First Error Pointer	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	CTM	Completion Timeout Mask Masks the error notification to Root Complex when Completion Timeout Error is detected. 0: No mask 1: Send Error Message, Mask Update of First Error Pointer	R/W
15	CAMO	Completer Abort Mask (Optional) Masks the error notification to the Root Complex when a Completer Abort Error is detected. 0: No mask 1: Send Error Message, record header to Header Log register, mask Update of First Error Pointer	R/W
16	UCM	Unexpected Completion Mask Masks error notification to the Root Complex when Unexpected Completion Error is detected. 0: No mask 1: Send Error Message, record header to Header Log register, mask Update of First Error Pointer	R/W
17	ROMO	Receiver Overflow Mask (Optional) Masks error notification to the Root Complex when Receiver Overflow Error is detected. 0: No mask 1: Send Error Message, Mask Update of First Error Pointer	R/W
18	MTLPM	Malformed TLP Mask Masks error notification to the Root Complex when Malformed TLP Error is detected. 0: No mask 1: Send Error Message, record header to Header Log register, mask Update of First Error Pointer	R/W
19	ECRCEMO	ECRC Error Mask (Optional) Masks error notification to the Root Complex when ECRC Error is detected. 0: No mask 1: Mask	R/W

Bit	Symbol	Function	R/W
20	UREM	Unsupported Request Error Mask Masks the error notification to the Root Complex when an Unsupported Request Error is detected. 0: No mask 1: Send Error Message, record header to Header Log register, mask Update of First Error Pointer	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_UNCEMASK_Fi register masks the uncorrectable error status.

Table 56.112 Valid reset signal

Reset signal	Unsupported Request Error Mask	ECRC Error Mask	Malformed TLP Mask	Receiver Overflow Mask	Unexpected Completion Mask	Completer Abort Mask	Completion Timeout Mask	Poisoned TLP Received Mask	Data Link Protocol Error Mask
RST_LOAD_B	—	—	—	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓	✓	✓	✓
RST_CFG_B	—	—	—	—	—	—	—	—	—
FLR	—	—	—	—	—	—	—	—	—

56.4.2.45 PCI_EP_UNCESVY_Fi : Uncorrectable Error Severity Register (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x610C + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	URES	ECRCESO	MTLPS	ROSO	UCS
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CASO	CTS	—	PTLPRS	—	—	—	—	—	—	—	DLPES	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	DLPES	Data Link Protocol Error Severity Sets the Error Severity when a Data Link Protocol Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
11:5	—	These bits are read as 0. The write value should be 0.	R/W
12	PTLPRS	Poisoned TLP Received Severity Sets the Error Severity when Poisoned TLP Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
14	CTS	Completion Timeout Severity Sets the Error Severity when Completion Timeout Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
15	CASO	Completer Abort Severity (Optional) Sets the Error Severity when a Completer Abort Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
16	UCS	Unexpected Completion Severity Sets the Error Severity when an Unexpected Completion Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
17	ROSO	Receiver Overflow Severity (Optional) Set the Error Severity when Receiver Overflow Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
18	MTLPS	Malformed TLP Severity Sets the Error Severity when Malformed TLP Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
19	ECRCESO	ECRC Error Severity (Optional) Sets the Error Severity when receiving an ECRC Error. 0: Non-Fatal Error 1: Fatal Error	R/W
20	URES	Unsupported Request Error Severity Set the Error Severity when Unsupported Request Error is detected. 0: Non-Fatal Error 1: Fatal Error	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	—	This bit is read as 1. The write value should be 1.	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_UNCESVY_Fi register sets the uncorrectable error severity.

Table 56.113 Valid reset signal

Reset signal	Unsupported Request Error Severity	ECRC Error Severity	Malformed TLP Severity	Receiver Overflow Severity	Unexpected Completion Severity	Completer Abort Severity	Completion Timeout Severity	Poisoned TLP Received Severity	Data Link Protocol Error Severity
RST_LOAD_B	—	—	—	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓	✓	✓	✓
RST_CFG_B	—	—	—	—	—	—	—	—	—
FLR	—	—	—	—	—	—	—	—	—

56.4.2.46 PCI_EP_CESTS_Fi : Correctable Error Status Register (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6110 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	ANFES	RTTS	—	—	—	REPLAYNUMRS	BDLLPS	BTLPS	—	—	—	—	—	RES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RES	Receiver Error Status (optional) 0: No error detected 1: Error detected	R*1
5:1	—	These bits are read as 0.	R
6	BTLPS	Bad TLP Status Indicates that a TLP CRC error or Sequence Number error was detected. 0: No error detected 1: Error detected	R*1
7	BDLLPS	Bad DLLP Status Indicates that a DLLP CRC error was detected. 0: No error detected 1: Error detected	R*1
8	REPLAYNUMRS	REPLAY_NUM Rollover Status Replay occurred four times in a row, indicating REPLAY_NUM rolled over from 11b to 00b. 0: No error detected 1: Error detected	R*1
11:9	—	These bits are read as 0.	R
12	RTTS	Replay Timer Timeout Status Indicates that a Timeout error occurred when a TLP was sent and an Ack or Nak DLLP could not be received within the specified time. 0: No error detected 1: Error detected	R*1
13	ANFES	Advisory Non-Fatal Error Status Indicates that an Advisory Non-Fatal Error was detected. 0: No error detected 1: Error detected	R*1
31:14	—	These bits are read as 0.	R

Note 1. The access type from PCIe is read-write. Writing 1 clears the bit. Writing 0 is ignored.

The PCI_EP_CESTS_Fi register indicates the correctable error status.

Table 56.114 Valid reset signal

Reset signal	Advisory Non-Fatal Error Status	Replay Timer Timeout Status	REPLAY_NUM Rollover Status	Bad DLLP Status	Bad TLP Status	Receiver Error Status
RST_LOAD_B	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓
RST_CFG_B	—	—	—	—	—	—
FLR	—	—	—	—	—	—

56.4.2.47 PCI_EP_CEMASK_Fi : Correctable Error Mask Register (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6114 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	ANFEM	RTTM	—	—	—	REPLAYNUMRM	BDLLPM	BTLPM	—	—	—	—	—	REM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	REM	Receiver Error Mask (optional) 0: No mask 1: Mask error message transmission	R/W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	BTLPM	Bad TLP Mask Masks error notification to Root Complex when Bad TLP Error is detected. 0: No mask 1: Mask error message transmission	R/W
7	BDLLPM	Bad DLLP Mask Masks error notification to Root Complex when Bad DLLP Error is detected. 0: No mask 1: Mask error message transmission	R/W
8	REPLAYNUMRM	REPLAY_NUM Rollover Mask REPLAY_NUM Mask error notification to Root Complex when Roll Over Error is detected. 0: No mask 1: Mask error message transmission	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	RTTM	Replay Timer Timeout Mask Masks error notification to Root Complex when Replay Timer Timeout Error is detected. 0: No mask 1: Mask error message transmission	R/W
13	ANFEM	Advisory Non-Fatal Error Mask Advisory Masks error notifications to Root Complex when non-fatal errors are detected. 0: No mask 1: Mask Advisory Non-Fatal Error handling (Mask updating First Error Pointer and Header Logging and sending Error Message)	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_CEMASK_Fi register masks the correctable error status.

Table 56.115 Valid reset signal

Reset signal	Advisory Non-Fatal Error Mask	Replay Timer Timeout Mask	REPLAY_NUM Rollover Mask	Bad DLLP Mask	Bad TLP Mask	Receiver Error Mask
RST_LOAD_B	—	—	—	—	—	—
RST_RSM_B	✓	✓	✓	✓	✓	✓
RST_CFG_B	—	—	—	—	—	—
FLR	—	—	—	—	—	—

56.4.2.48 PCI_EP_ADVECC_Fi : Advanced Error Capabilities and Control Register (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6118 + 0x1000 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ECRCCE	ECRCCC	ECRCGE	ECRCGC	FEP[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	FEP[4:0]	First Error Pointer Indicates the field value of the Uncorrectable Error Status register for the first detected Uncorrectable Error	R
5	ECRCGC	ECRC Generation Capable Presence or absence of ECRC Generation function 0: None 1: Yes (Default)	R
6	ECRCGE	ECRC Generation Enable ECRC Generation Enable setting 0: Disable (Default) 1: Enable	R/W
7	ECRCCC	ECRC Check Capable Presence or absence of ECRC Check function 0: None 1: Yes (Default)	R
8	ECRCCE	ECRC Check Enable ECRC Check Enable setting 0: Disable (Default) 1: Enable	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The PCI_EP_ADVECC_Fi register indicates and controls the advanced error capabilities.

Table 56.116 Valid reset signal

Reset signal	ECRC Check Enable	ECRC Generation Enable
RST_LOAD_B	—	—
RST_RSM_B	✓	✓
RST_CFG_B	—	—
FLR	—	—

56.4.2.49 PCI_EP_HLOG0_Fi : Header Log Register 0 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x611C + 0x1000 × i

Bit position:	31	0
Bit field:	HTLPE0[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	HTLPE0[31:0]	Header of TLP associated with error 0 For the first detected Uncorrectable Error, indicates the 1st DW of Header.	R

The PCI_EP_HLOG0_Fi register indicates the header log.

56.4.2.50 PCI_EP_HLOG1_Fi : Header Log Register 1 (Function #i)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6120 + 0x1000 × i

Bit position:	31																													0
Bit field:	HTLPE1[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	HTLPE1[31:0]	Header of TLP associated with error 1 For the first detected Uncorrectable Error, indicates the 2nd DW of Header.	R

The PCI_EP_HLOG1_Fi register indicates the header log.

56.4.2.51 PCI_EP_HLOG2_Fi : Header Log Register 2 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6124 + 0x1000 × i

Bit position:	31																													0
Bit field:	HTLPE2[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	HTLPE2[31:0]	Header of TLP associated with error 2 For the first detected Uncorrectable Error, indicates the 3rd DW of Header.	R

The PCI_EP_HLOG2_Fi register indicates the header log.

56.4.2.52 PCI_EP_HLOG3_Fi : Header Log Register 3 (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6128 + 0x1000 × i

Bit position:	31																													0
Bit field:	HTLPE3[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	HTLPE3[31:0]	Header of TLP associated with error 3 For the first detected Uncorrectable Error, indicates the 4th DW of Header.	R

The PCI_EP_HLOG3_Fi register indicates the header log.

56.4.2.53 PCI_EP_DEVSNEXTC_Fi : Device Serial Number Extended Capability Register (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6150 + 0x1000 × i

Bit position:	31											20	19			16	15													0
Bit field:	NCO[11:0]												CV[3:0]				PCIEECID[15:0]													
Value after reset:	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
15:0	PCIEECID[15:0]	PCI Express Extended Capability ID Indicates the Device Serial Number Extended Capability. Default: 0x0003	R
19:16	CV[3:0]	Capability Version Indicates the version of the Capability Structure. Default: 0x1	R/W*1
31:20	NCO[11:0]	Next Capability Offset Indicates the starting address of the Secondary PCI Express Extended Capability Header. 0x1B0 fixed. Function#1 is fixed at 0x000.	R

Note 1. The access type from PCIe is read-only.

The PCI_EP_DEVSNEXTC_Fi register specifies the device serial number extended capability.

Table 56.117 Valid reset signal

Reset signal	Capability Version
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.54 PCI_EP_SNL_Fi : Serial Number Register (Lower DW) (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6154 + 0x1000 × i

Bit position:	31																													0
Bit field:	PCIEDSN_LDW[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PCIEDSN_LDW[31:0]	PCI Express Device Serial Number (Lower DW) The lower 32 bits of the IEEE standard 64-bit unique ID (EUI-64). EUI-64 consists of a 24-bit company ID and a 40-bit vendor-defined extension. When using a unique ID, write from the UDL side.	R/W*1

Note: "PCI Express Device Serial Number" value must be the same value for all functions including Lower/Upper.
Note 1. The access type from PCIe is read-only.

The PCI_EP_SNL_Fi register specifies the serial number of the device.

Table 56.118 Valid reset signal (1 of 2)

Reset signal	PCI Express Device Serial Number (Lower DW)
RST_LOAD_B	✓

Table 56.118 Valid reset signal (2 of 2)

Reset signal	PCI Express Device Serial Number (Lower DW)
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.55 PCI_EP_SNU_Fi : Serial Number Register (Upper DW) (Function #i) (i = 0, 1)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x6158 + 0x1000 × i

Bit position:	31																													0
Bit field:	PCIEDSN_UDW[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PCIEDSN_UDW[31:0]	PCI Express Device Serial Number (Upper DW) The Upper 32 bits of the IEEE standard 64-bit unique ID (EUI-64). EUI-64 consists of a 24-bit company ID and a 40-bit vendor-defined extension. When using a unique ID, write from the UDL side.	R/W*1

Note 1. The access type from PCIe is read-only.

The PCI_EP_SNU_Fi register specifies the serial number of the device.

Table 56.119 Valid reset signal

Reset signal	PCI Express Device Serial Number (Upper DW)
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.56 PCI_EP_SPEECH_F0 : Secondary PCI Express Extended Capability Header (Function #0)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x61B0

Bit position:	31											20	19			16	15													0
Bit field:	NCO[11:0]												CV[3:0]				PCIEECID[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1

Bit	Symbol	Function	R/W
15:0	PCIEECID[15:0]	PCI Express Extended Capability ID Indicates the Secondary PCI Express Extended Capability Header. Default: 0x0019	R
19:16	CV[3:0]	Capability Version Indicates the version of the Capability Structure. Default: 0x1	R/W*1
31:20	NCO[11:0]	Next Capability Offset Indicates that this Capability List is the final List. 0x000 fixed	R

Note 1. The access type from PCIe is read-only.

The PCI_EP_SPEECH_F0 register specifies the Secondary PCI Express Extended Capability Header.

Table 56.120 Valid reset signal

Reset signal	Capability Version
RST_LOAD_B	✓
RST_RSM_B	—
RST_CFG_B	—
FLR	—

56.4.2.57 PCI_EP_LINC3_F0 : Link Control 3 Register (Function #0)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x61B4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ELSKPOSGV[6:0]							—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	—	These bits are read as 0. The write value should be 0.	R/W
15:9	ELSKPOSGV[6:0]	Enable Lower SKP OS Generation Vector When the Link is in L0 and the bit in this field corresponding to the current Link speed is Set, SKP Ordered Sets are scheduled at the rate defined for SRNS, overriding the rate required based on the clock tolerance architecture. Bit definitions within this field are: Bit 0: 2.5 GT/s Bit 1: 5.0 GT/s Bit 2: 8.0 GT/s Bit 3: 16.0 GT/s (prohibited) Bits 6 to 4: RsvdP Bits in this field are RW if the corresponding bit in the Lower SKP OS Generation Supported Speeds Vector is Set, otherwise they are permitted to be hardwired to 0. Behavior is undefined if a bit is Set in this field and the corresponding bit in the Lower SKP OS Generation Supported Speeds Vector is not Set. The default value of this field is 0x000_0000.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Table 56.121 Valid reset signal

Reset signal	Enable Lower SKP OS Generation Vector
RST_LOAD_B	—
RST_RSM_B	—
RST_CFG_B	✓
FLR	✓

56.4.2.58 PCI_EP_LESTA_F0 : Lane Error Status Register (Function #0)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x61B8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LESB[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	LESB[1:0]	Lane Error Status Bits Each bit indicates if the corresponding Lane detected a Lane-based error. A value of 1 indicates that a Lane based-error was detected on the corresponding Lane Number. The default value of each bit is 0. Cleared by writing 1 from AXI.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Table 56.122 Valid reset signal

Reset signal	Lane Error Status Bits
RST_LOAD_B	—
RST_RSM_B	✓
RST_CFG_B	—
FLR	—

56.4.2.59 PCI_EP_LEQCTL_F0 : Lane Equalization Control Register (Function #0)

Base address: PCIEm = 0x9210_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x61BC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	UPRPH1[2:0]			UPTP1[3:0]				—	DPRPH1[2:0]			DPTP1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	UPRPH0[2:0]			UPTP0[3:0]				—	DPRPH0[2:0]			DPTP0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DPTP0[3:0]	Downstream Port 8.0 GTs Transmitter Preset Lane0 The read value is undefined. Transmitter Presetused for equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0, this field is RsvdP. Otherwise, this field is Hwinit. The default value is 0xF. EP: Fixed to 0 because Crosslink is not supported	R

Bit	Symbol	Function	R/W
6:4	DPRPH0[2:0]	Downstream Port 8.0 GTs Receiver Preset Hint Lane0 The read value is undefined. Receiver Preset Hint may be used as a hint for receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0, this field is RsvdP. Otherwise, this field is Hwinit. The default value is 0x7. EP: Fixed to 0 because Crosslink is not supported	R
7	—	This bit is read as 0.	R
11:8	UPTP0[3:0]	Upstream Port 8.0 GTs Transmitter Preset Lane0 The read value is undefined. Field contains the Transmit Preset value sent or received during Link Equalization. Field usage varies as follows: A: Downstream Port Field contains the value sent on the associated Lane during Link Equalization. Field is Hwinit. B: Upstream Port, Crosslink Supported = 0 Field is intended for debug and diagnostics. It contains the value captured from the associated Lane during Link Equalization. Field is RO. When crosslinks are supported, case C (below) applies and this captured information is not visible to software. Vendors are encouraged to provide an alternate mechanism to obtain this information. C: Upstream Port, Crosslink Supported = 1 Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A (above) applies. Field is Hwinit. The default value is 0xF. EP: Input signal	R
14:12	UPRPH0[2:0]	Upstream Port 8.0 GTs Receiver Preset Hint Lane0 The read value is undefined. Field contains the Receiver Preset Hint value sent or received during Link Equalization. Field usage varies as follows: A: Downstream Port Field contains the value sent on the associated Lane during Link Equalization. Field is Hwinit. B: Upstream Port, Crosslink Supported = 0 Field is intended for debug and diagnostics. It contains the value captured from the associated Lane during Link Equalization. Field is RO. When crosslinks are supported, case C (below) applies and this captured information is not visible to software. Vendors are encouraged to provide an alternate mechanism to obtain this information. C: Upstream Port, Crosslink Supported = 1 Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A (above) applies. Field is Hwinit. The default value is 0x7. EP: Input signal	R
15	—	This bit is read as 0.	R
19:16	DPTP1[3:0]*1	Downstream Port 8.0 GTs Transmitter Preset Lane1 The read value is undefined. Transmitter Presetused for equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0, this field is RsvdP. Otherwise, this field is Hwinit. The default value is 0xF. EP: Fixed to 0 because Crosslink is not supported	R

Bit	Symbol	Function	R/W
22:20	DPRPH1[2:0]*1	Downstream Port 8.0 GTs Receiver Preset Hint Lane1 The read value is undefined. Receiver Preset Hint may be used as a hint for receiver equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. For an Upstream Port if Crosslink Supported is 0, this field is RsvdP. Otherwise, this field is Hwinit. The default value is 0x7. EP: Fixed to 0 because Crosslink is not supported	R
23	—	This bit is read as 0.	R
27:24	UPTP1[3:0]*1	Upstream Port 8.0 GTs Transmitter Preset Lane1 The read value is undefined. Field contains the Transmit Preset value sent or received during Link Equalization. Field usage varies as follows: A: Downstream Port Field contains the value sent on the associated Lane during Link Equalization. Field is Hwinit. B: Upstream Port, Crosslink Supported = 0 Field is intended for debug and diagnostics. It contains the value captured from the associated Lane during Link Equalization. Field is RO. When crosslinks are supported, case C (below) applies and this captured information is not visible to software. Vendors are encouraged to provide an alternate mechanism to obtain this information. C: Upstream Port, Crosslink Supported = 1 Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A (above) applies. Field is Hwinit. The default value is 0xF. EP: Input signal	R
30:28	UPRPH1[2:0]*1	Upstream Port 8.0 GTs Receiver Preset Hint Lane1 The read value is undefined. Field contains the Receiver Preset Hint value sent or received during Link Equalization. Field usage varies as follows: A: Downstream Port Field contains the value sent on the associated Lane during Link Equalization. Field is Hwinit. B: Upstream Port, Crosslink Supported = 0 Field is intended for debug and diagnostics. It contains the value captured from the associated Lane during Link Equalization. Field is RO. When crosslinks are supported, case C (below) applies and this captured information is not visible to software. Vendors are encouraged to provide an alternate mechanism to obtain this information. C: Upstream Port, Crosslink Supported = 1 Field is not used or affected by the current Link Equalization. Field value will be used if a future crosslink negotiation switches the Operating Port Direction so that case A (above) applies. Field is Hwinit. The default value is 0x7. EP: Input signal	R
31	—	This bit is read as 0.	R

Note 1. Not available for m = 1.

56.5 Register Descriptions (Physical Layer Control/Status)

These registers control the pins for setting the PHY built into the module.

In the multi-link configuration, the area for ch.1 is reserved, because the control is performed by ch.0.

When these registers are accessed for writing, PHY_REG_CLK_EN bit in permission register must be set to 1 before a write access is made. In addition, rewrite the registers while the PCIe logic and PHY resets (other than ARESETn, RST_CFG_B, and RST_LOAD_B) are asserted.

56.5.1 PCI_PHY_XCFGDn : XCFGD Setting Register n (n = 0 to 26)

Base address: PCIE_PHY = 0x9210_2000

Offset address: 0x000 + 0x10 × n

Bit position:	31																												0
Bit field:	XCFGD[31:0]																												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	XCFGD[31:0]	Sets each bit of the PHY setting pin XCFGD.	R/W

The PCI_PHY_XCFGDn registers are used to set PHY of the PCI Express core.

Set the value according to Table 56.123 shown below by software during initialization phase.

Table 56.123 Setting value of XCFGD setting registers

n	Offset address	Value	n	Offset address	Value	n	Offset address	Value
0	0x2000	0x0000_2000	9	0x2090	0x0000_0000	18	0x2120	0x0000_0000
1	0x2010	0x00C0_0090	10	0x20A0	0x5DBB_8000	19	0x2130	0x0000_0000
2	0x2020	0x0000_01E0	11	0x20B0	0x1E00_0000	20	0x2140	0x0000_0000
3	0x2030	0x0200_0000	12	0x20C0	0x0000_0000	21	0x2150	0x0000_0000
4	0x2040	0x0000_0000	13	0x20D0	0x0000_0000	22	0x2160	0x0000_0000
5	0x2050	0x0052_0154	14	0x20E0	0x0000_0000	23	0x2170	0x0000_0000
6	0x2060	0x0000_0000	15	0x20F0	0x0000_0000	24	0x2180	0x0000_0000
7	0x2070	0x0000_0000	16	0x2100	0x0000_0000	25	0x2190	0x0000_0000
8	0x2080	0x4444_0000	17	0x2110	0x0000_0000	26	0x21A0	0x0000_0000

56.5.2 PCI_PHY_XCFGA_CMNn : XCFGA_CMN Setting Register n (n = 0 to 15)

Base address: PCIE_PHY = 0x9210_2000

Offset address: 0x400 + 0x10 × n

Bit position:	31																												0
Bit field:	XCFGA_CMN[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	XCFGA_CMN[31:0]	Sets each bit of the PHY setting pin XCFGA_CMN.	R/W

The PCI_PHY_XCFGA_CMNn registers are used to set PHY of the PCI Express core.

Set the value according to Table 56.124 shown below by software during initialization phase.

Table 56.124 Setting value of XCFGA_CMN setting registers (1 of 2)

n	Offset address	Value	n	Offset address	Value	n	Offset address	Value
0	0x2400	0x0000_0080	6	0x2460	0x3204_0400	12	0x24C0	0x010A_9291
1	0x2410	0x6094_0060	7	0x2470	0x2191_4064	13	0x24D0	0xE044_039C
2	0x2420	0x00C1_2000	8	0x2480	0x5294_8A03	14	0x24E0	0x0880_0807
3	0x2430	0x6000_0BF8	9	0x2490	0x219C_E008	15	0x24F0	0x0004_1002

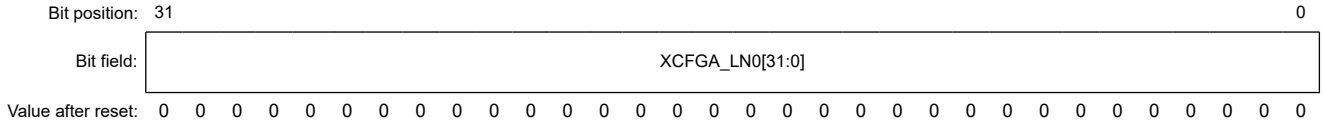
Table 56.124 Setting value of XCFGA_CMN setting registers (2 of 2)

n	Offset address	Value	n	Offset address	Value	n	Offset address	Value
4	0x2440	0x8083_4238	10	0x24A0	0x0C86_7F02	—	—	—
5	0x2450	0x0000_1118	11	0x24B0	0x4064_3228	—	—	—

56.5.3 PCI_PHY_XCFGA_L0n : XCFGA_LN0 Setting Register n (n = 0 to 5)

Base address: PCIE_PHY = 0x9210_2000

Offset address: 0x500 + 0x10 × n



Bit	Symbol	Function	R/W
31:0	XCFGA_LN0[31:0]	Sets each bit of the PHY setting pin XCFGA_LN0.	R/W

The PCI_PHY_XCFGA_L0n registers are used to set PHY of the PCI Express core.

Set the value according to Table 56.125 shown below by software during initialization phase.

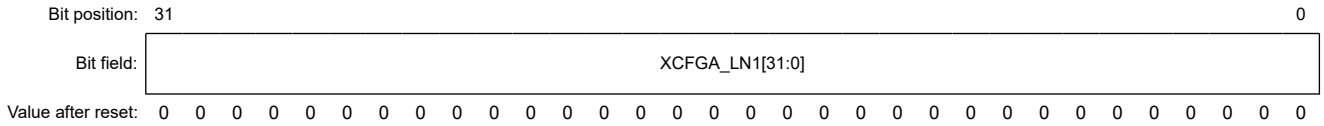
Table 56.125 Setting value of XCFGA_LN0 setting registers

n	Offset address	Value	n	Offset address	Value	n	Offset address	Value
0	0x2500	0x0800_0000	2	0x2520	0xE000_3300	4	0x2540	0x9609_02A0
1	0x2510	0x0005_0400	3	0x2530	0xC040_0FBF	5	0x2550	0x0002_0860

56.5.4 PCI_PHY_XCFGA_L1n : XCFGA_LN1 Setting Register n (n = 0 to 5)

Base address: PCIE_PHY = 0x9210_2000

Offset address: 0x560 + 0x10 × n



Bit	Symbol	Function	R/W
31:0	XCFGA_LN1[31:0]	Sets each bit of the PHY setting pin XCFGA_LN1.	R/W

The PCI_PHY_XCFGA_L1n registers are used to set PHY of the PCI Express core.

Set the value according to Table 56.126 shown below by software during initialization phase.

Table 56.126 Setting value of XCFGA_LN1 setting registers

n	Offset address	Value	n	Offset address	Value	n	Offset address	Value
0	0x2560	0x0800_0000	2	0x2580	0xE000_3300	4	0x25A0	0x9609_02A0
1	0x2570	0x0005_0400	3	0x2590	0xC040_0FBF	5	0x25B0	0x0002_0860

56.6 Register Descriptions (Supplemental Setting)

56.6.1 PCIE_INTX : PCIe INTX Register

Base address: PCIE_SPLm = 0x8029_2000 + 0x0030 × m (m = 0, 1)

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INTX_EP_F1	INTX_EP_F0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INTX_EP_F0	Legacy Interrupt (Assert_INTx/Deassert_INTx message) Trigger for Endpoint Funtion #0 0: Issue Deassert_INTx message 1: Issue Assert_INTx message	R/W
1	INTX_EP_F1	Legacy Interrupt (Assert_INTx/Deassert_INTx message) Trigger for Endpoint Funtion #1 0: Issue Deassert_INTx message 1: Issue Assert_INTx message	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The PCIE_INTX is the register for Endpoint to generate legacy interrupt (Assert_INTx/Deassert_INTx) to Root Complex.

56.6.2 PCIE_MSI1 : PCIe MSI Register 1

Base address: PCIE_SPLm = 0x8029_2000 + 0x0030 × m (m = 0, 1)

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	UI_EXTMSI_VAL4	UI_EXTMSI_VAL3	UI_EXTMSI_VAL2	UI_EXTMSI_VAL1	UI_EXTMSI_VAL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UI_EXTMSI_VAL0	MSI 0 Interrupt Trigger for Endpoint When writing 1, triggers MSI 0 interrupt. When reading, 0 is read.	W
1	UI_EXTMSI_VAL1	MSI 1 Interrupt Trigger for Endpoint When writing 1, triggers MSI 1 interrupt. When reading, 0 is read.	W
2	UI_EXTMSI_VAL2	MSI 2 Interrupt Trigger for Endpoint When writing 1, triggers MSI 2 interrupt. When reading, 0 is read.	W
3	UI_EXTMSI_VAL3	MSI 3 Interrupt Trigger for Endpoint When writing 1, triggers MSI 3 interrupt. When reading, 0 is read.	W

Bit	Symbol	Function	R/W
4	UI_EXTMSI_VAL4	MSI 4 Interrupt Trigger for Endpoint When writing 1, triggers MSI 4 interrupt. When reading, 0 is read.	W
31:5	—	The write value should be 0.	W

The PCIE_MSI1 is the register for Endpoint to generate MSI interrupt to Root Complex.

56.6.3 PCIE_MSI2 : PCIe MSI Register 2

Base address: PCIE_SPLm = 0x8029_2000 + 0x0030 × m (m = 0, 1)

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	UI_EXTMSI_VEC3[4:0]					—	—	—	UI_EXTMSI_VEC2[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	UI_EXTMSI_VEC1[4:0]					—	—	—	UI_EXTMSI_VEC0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	UI_EXTMSI_VEC0[4:0]	Specify the interrupt vector of MSI 0 interrupt for Endpoint.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
12:8	UI_EXTMSI_VEC1[4:0]	Specify the interrupt vector of MSI 1 interrupt for Endpoint.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	UI_EXTMSI_VEC2[4:0]	Specify the interrupt vector of MSI 2 interrupt for Endpoint.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	UI_EXTMSI_VEC3[4:0]	Specify the interrupt vector of MSI 3 interrupt for Endpoint.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

The PCIE_MSI2 is the register for Endpoint to specify the interrupt vector of MSI interrupt to Root Complex.

56.6.4 PCIE_MSI3 : PCIe MSI Register 3

Base address: PCIE_SPLm = 0x8029_2000 + 0x0030 × m (m = 0, 1)

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	UI_EXTMSI_VEC4[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	UI_EXTMSI_VEC4[4:0]	Specify the interrupt vector of MSI 4 interrupt for Endpoint.	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

The PCIE_MSI3 is the register for Endpoint to specify the interrupt vector of MSI interrupt to Root Complex.

56.6.5 PCIE_MSI4 : PCIe MSI Register 4

Base address: PCIE_SPLm = 0x8029_2000 + 0x0030 × m (m = 0, 1)

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	UI_EXTMSI_FUNC3[2:0]			—	—	—	—	—	UI_EXTMSI_FUNC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	UI_EXTMSI_FUNC1[2:0]			—	—	—	—	—	UI_EXTMSI_FUNC0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	UI_EXTMSI_FUNC0[2:0]	Specify the interrupt function number of MSI 0 interrupt for Endpoint. Set 000b or 001b.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	UI_EXTMSI_FUNC1[2:0]	Specify the interrupt function number of MSI 1 interrupt for Endpoint. Set 000b or 001b.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	UI_EXTMSI_FUNC2[2:0]	Specify the interrupt function number of MSI 2 interrupt for Endpoint. Set 000b or 001b.	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
26:24	UI_EXTMSI_FUNC3[2:0]	Specify the interrupt function number of MSI 3 interrupt for Endpoint. Set 000b or 001b.	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

The PCIE_MSI4 is the register for Endpoint to specify the interrupt function number of MSI interrupt to Root Complex.

56.6.6 PCIE_MSI5 : PCIe MSI Register 5

Base address: PCIE_SPLm = 0x8029_2000 + 0x0030 × m (m = 0, 1)

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	UI_EXTMSI_FUNC4[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	UI_EXTMSI_FUNC4[2:0]	Specify the interrupt function number of MSI 4 interrupt for Endpoint. Set 000b or 001b.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The PCIE_MSI5 is the register for Endpoint to specify the interrupt function number of MSI interrupt to Root Complex.

56.6.7 PCIE_PME : PCIe PME Register

Base address: PCIE_SPLm = 0x8029_2000 + 0x0030 × m (m = 0, 1)

Offset address: 0x018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFG_PMCSR_PME_STATUS_F1	CFG_PMCSR_PME_STATUS_F0	—	—	—	—	—	—	—	PME_TIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PME_TIM	PM_PME Message Clock for Endpoint Toggle the bit every 100 to 150 ms.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	CFG_PMCSR_PME_STATUS_F0	Power Management Event Setting for Endpoint Function #0 This value is set to the PME_Status bit (bit 15) in PM Status/Control Register.	R/W
9	CFG_PMCSR_PME_STATUS_F1	Power Management Event Setting for Endpoint Function #1 This value is set to the PME_Status bit (bit 15) in PM Status/Control Register.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The PCIE_PME is the register for Endpoint to set PME.

56.6.8 PCIE_ACK : PCIe ACK Register

Base address: PCIE_SPLm = 0x8029_2000 + 0x0030 × m (m = 0, 1)

Offset address: 0x01C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	D3_EVENT_ACK_F1	D3_EVENT_ACK_F0	—	—	—	—	—	—	—	TURN_OFF_EVENT_ACK
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TURN_OFF_EVENT_ACK	Acknowledge to PME_Turn_Off Message for Endpoint When writing 1, sends PME_TO_Ack Message. When reading, 0 is read.	W
7:1	—	The write value should be 0.	W
8	D3_EVENT_ACK_F0	Acknowledge to Power State Transition for Endpoint Function #0 When writing 1, acknowledges the state transition. When reading, 0 is read.	W
9	D3_EVENT_ACK_F1	Acknowledge to Power State Transition for Endpoint Function #1 When writing 1, acknowledges the state transition. When reading, 0 is read.	W
31:10	—	The write value should be 0.	W

The PCIE_ACK is the register for Endpoint to acknowledge to Power Management.

56.6.9 PCIE_MISC : PCIe MISC Register

Base address: PCIE_SPLm = 0x8029_2000 + 0x0030 × m (m = 0, 1)

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FLR_REQ1	FLR_REQ0	FLR_RESET1	FLR_RESET0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ALLOW_ENTER_L1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ALLOW_ENTER_L1	ASPM L1 State Transition Permission 0: Not permitted to transit from ASPM L0 to ASPM L1 state. 1: Permitted to transit from ASPM L0 to ASPM L1 state.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
16	FLR_RESET0	Function Level Reset (FLR) for Function #0 When writing 1, performs FLR. When reading, 0 is read.	W
17	FLR_RESET1	Function Level Reset (FLR) for Function #1 When writing 1, performs FLR. When reading, 0 is read.	W
18	FLR_REQ0	Indicates FLR Request for Funtion #0	R
19	FLR_REQ1	Indicates FLR Request for Funtion #1	R
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The PCIE_MISC is the register to set ASPM and FLR.

56.6.10 PCIE_MODE : PCIe Mode Register

Base address: PCIE_SPLm = 0x8029_2000 + 0x0030 × m (m = 0, 1)

Offset address: 0x024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MODE_PORT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MODE_PORT	Device Type Select 0: Endpoint 1: Root Complex	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The PCIE_MODE is the register to set device type.

56.6.11 PCIE_LINKMODE : PCIe Link Mode Register

Base address: PCIE_LNK = 0x8029_2060

Offset address: 0x0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LINK_MASTER[1:0]		—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
9:8	LINK_MASTER[1:0]	Channel Configuration Dynamic change is prohitebed. Change the bits during reset. 0 1: 2 lanes x 1 channel 1 1: 1 lane x 2 channels Others: Setting prohibited	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The PCIE_LINKMODE is the register to set channel configuration.

56.7 Functional Description

56.7.1 PCIe Core Functions

The functional description of the unit is given below.

This module is configured based on the Base Spec 4.0. For the detailed specification, refer to the Base Spec 4.0. This section mainly describes the parts specific to Renesas Electronics.

56.7.2 Memory Map

56.7.2.1 PCIe to AXI (AXI Window)

Figure 56.4 shows the relation between AXI Window and registers. They are used when MWr/MRd data from PCI Express (RX) is transferred to AXI bus.

Set AWBase and AWMask in the BAR region. (BAR: PCI Express Configuration Register, Base Address Register)

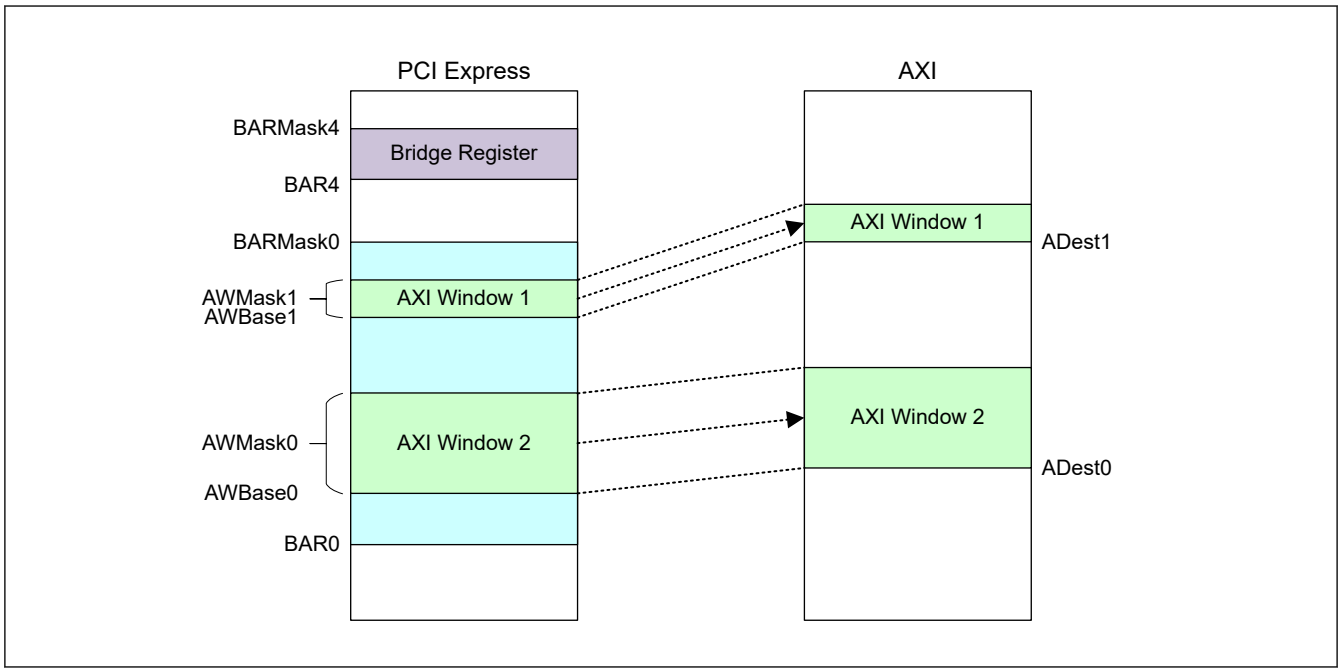


Figure 56.4 Relationship between AXI Window and registers

Set BAR to base address and BARMask to region size. It is possible to have up to 4 windows in that region. Access to AXI is performed within the range of AWMask x from Adest x (x: 0 to 3 corresponding to window number). AXI access address is shown as below. 64-bit address is converted to 32-bit address.

AXI Access Address = PCIe Access Address – BAR0 -AWBase x + ADest x

Note: Address is converted based on BAR.

BAR/BARmask, AWBase x/AWMask x, AWMask x/ADest x should be set not to generate carry bit when two registers are added. Minimum size of the region by BAR/BARMask is 4 KB. Each window and bridge register region should be set not to be overlapped each other.

Table 56.127 and Figure 56.5 are the example of the register setting for PCIe to AXI access.

Table 56.127 Example of the register setting for PCIe to AXI access

BAR0	0x0000_0000_1000_0000	PCIe Configuration Register
BARMask0	0x0000_0000_0FFF_FFFC	PCIe Configuration Register
Assigned region	0x0000_0000_1000_0000 to 0x0000_0000_1FFF_FFFF (256 MB)	
AWBase0	0x0000_0000_0010_0000	AXI Bridge Register
AWMask0	0x0000_0000_0003_FFFF	AXI Bridge Register
ADest0	0x0000_0004_0100_0000	AXI Bridge Register
AXI Window	0x0100_0000 to 0x0103_FFFF (256 KB)	
BAR4	0x0000_0000_2000_0000	PCIe Configuration Register
BARMask4	0x0000_0000_0000_7FFC	PCIe Configuration Register
Assigned region	0x0000_0000_2000_0000 to 0x0000_0000_2000_7FFF (32 KB)	

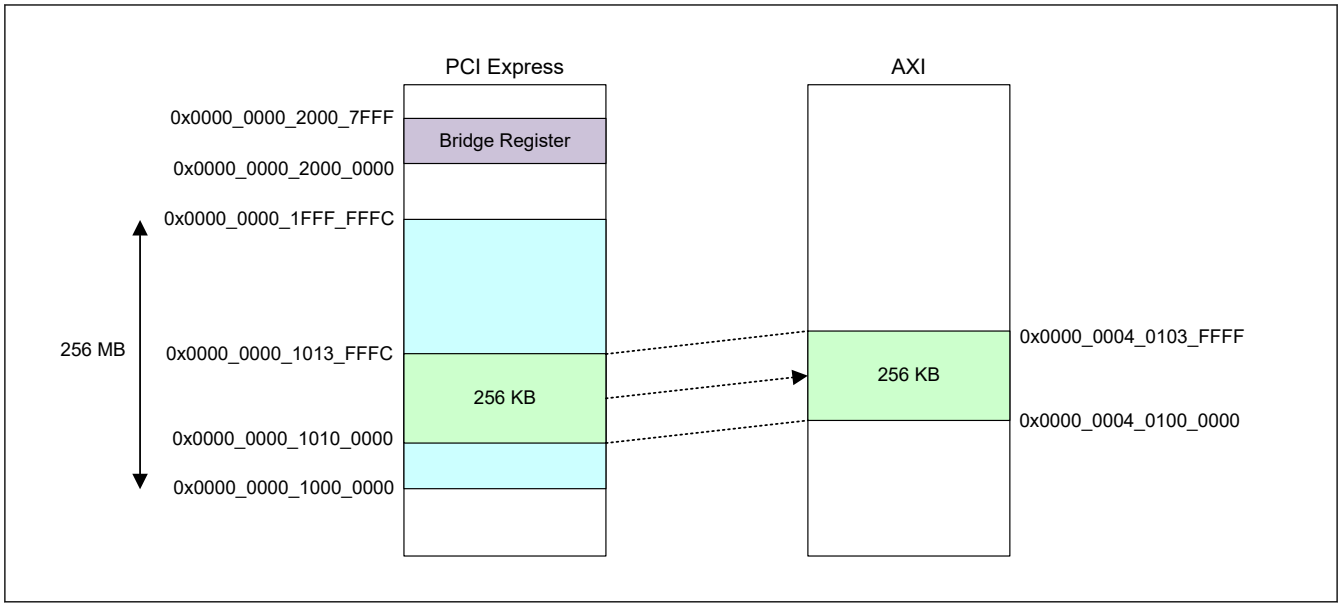


Figure 56.5 Example of the PCI-AXI access for PCIe to AXI access

56.7.2.2 AXI to PCIe (PCIe Window)

Figure 56.6 shows the relation between PCIe Window and registers. They are used when data from AXI bus is transferred to PCI Express (TX) as MWr/MRd.

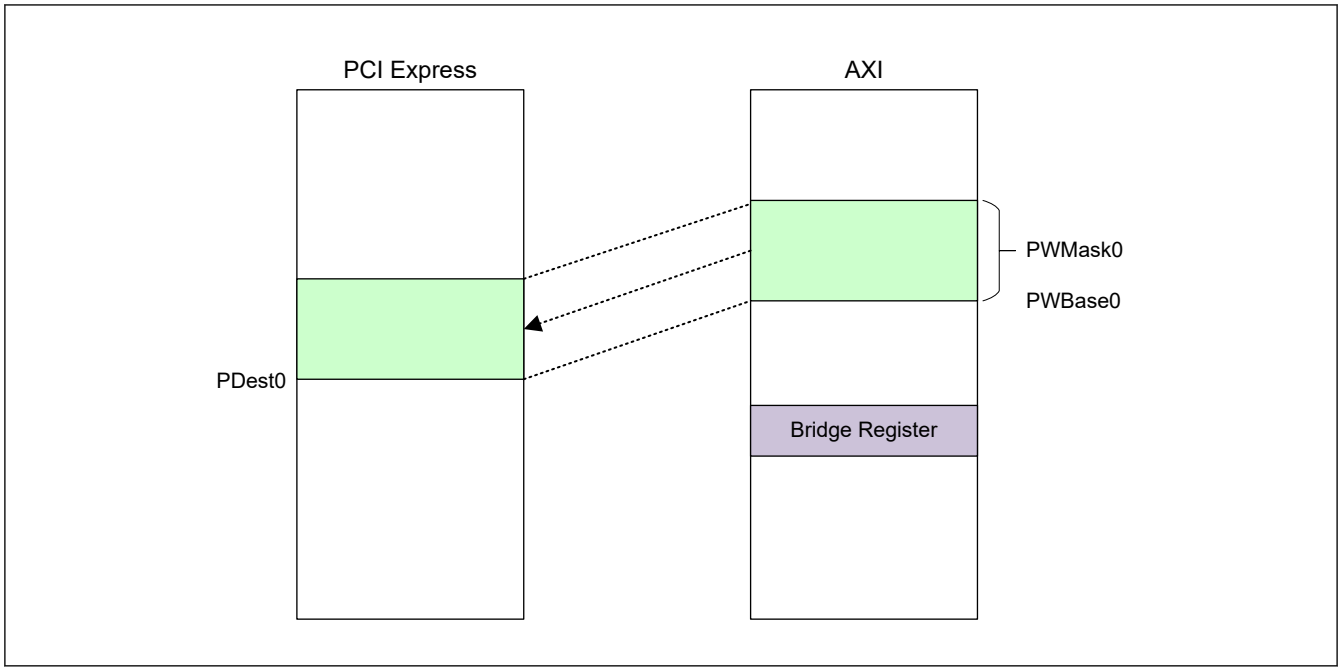


Figure 56.6 Relationship between PCIe Window and registers

It is possible to have up to 4 windows. Access to PCIe is performed within the range of PWMask x from PDest x (x: 0 to 3 corresponding to window number). PCIe access address is shown as below. 32-bit address is converted to 64-bit address.

PCIe Access Address = AXI Access Address – PWBase x + PDest x

Note: No relation to BAR.

Each window and bridge register region should be set not to be overlapped each other.

Table 56.128 and Figure 56.7 are the example of the register setting for AXI to PCIe access.

Table 56.128 Example of the register setting for AXI to PCIe access

BAR0	0x0000_0000_1000_0000	PCIe Configuration Register
BARMask0	0x0000_0000_07FF_FFFC	PCIe Configuration Register
Assigned region	0x0000_0000_1000_0000 to 0x0000_0000_17FF_FFFF (128 MB)	
PWBase0	0x0000_0000_0070_0000	AXI Bridge Register
PWMask0	0x0000_0000_000F_FFFF	AXI Bridge Register
PDest0	0x0000_0004_0070_0000	AXI Bridge Register
PCIe Window	0x0070_0000 to 0x007F_FFFF (1 MB)	

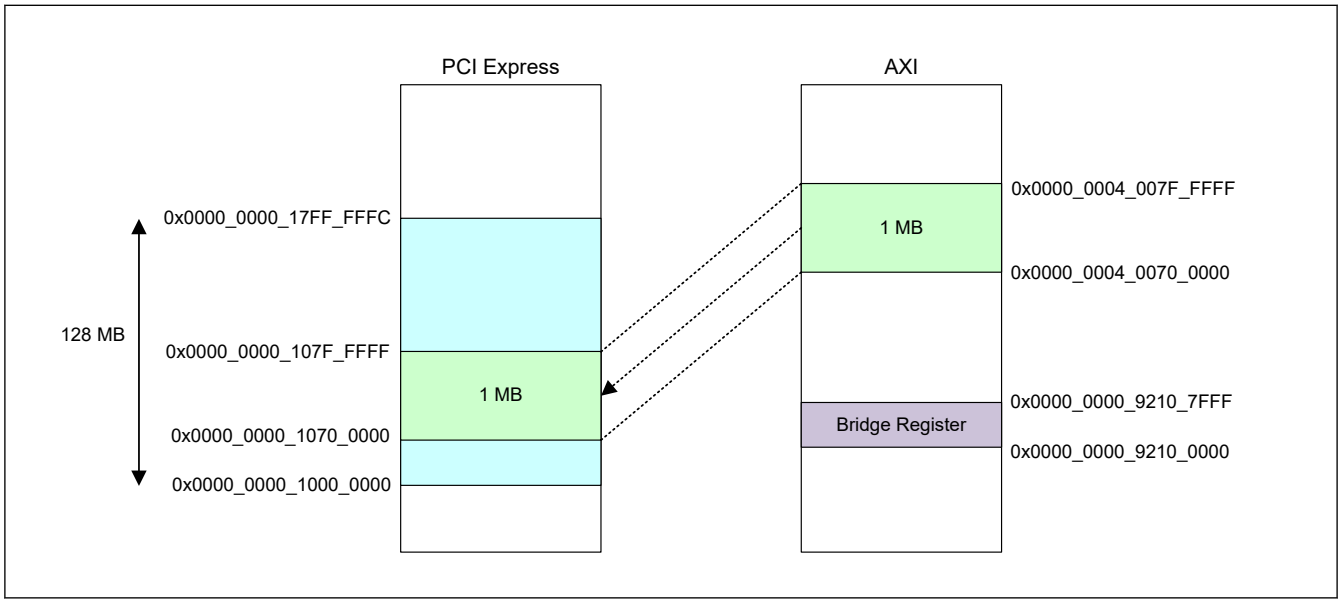


Figure 56.7 Example of the PCI-AXI access for AXI to PCIe access

56.7.3 Issuing of PCIe Requests and Register Access (by AXI)

The following gives functional description on access to the PCI Express core by the AXI bus.

56.7.3.1 PCIe Requests which can be Issued (Supported Commands for TX)

Requests to be issued via pcie windows

MRd: Memory Read Request

MWr: Memory Write Request

Requests to be issued by the registers

MRd: Zero-Length Memory Read Request

CfgRd0: Configuration Read Type 0

CfgWr0: Configuration Write Type 0

CfgRd1: Configuration Read Type 1

CfgWr1: Configuration Write Type 1

Msg: Message Request

MsgD: Message Request with data payload

Unsupported requests (issuance prohibited)

IORd: I/O Read Request

IOWr: I/O Write Request

MRdLk: Memory Read Request-Locked

CplDLk: Completion for Locked Memory Read

Note: If non-posted request with data (CfgWr0/CfgWr1/IOWr) which should not be transmitted is issued inadvertently in EPS mode, an MRd (non-posted request without data) which can be normally transmitted in EP mode may not also be transmitted after that.

56.7.3.2 Register Access

1. Internal register read/write
2. Configuration register read/write

The following restrictions apply to access to the configuration registers.

- Byte-lane transfer is only acceptable if the transfer consists of consecutive valid bytes.
- If the valid bytes are not consecutive, only values which are no greater than 32 bits (1 dword) and do not span the dword alignment are acceptable.
  Example) A value which can represent a dword TLP for PCIe

56.7.3.3 Issuing Memory Requests

Access from the AXI side is converted into a PCIe request and then issued. Up to eight windows can be allocated. Note that only incremental bursts and fixed bursts (of 1 beat) are acceptable. Wrapping bursts and fixed bursts (of 2 or more beats) are prohibited. If these prohibited bursts are received, the unit operates as follows.

- Although the response will in general be "OKAY" (except in cases that involved a protocol error of the AXI), this is beyond the scope of guaranteed operation.
- Unexpected requests may be issued to the PCIe bus.
- Unexpected registers may be modified.

To maintain the order between memory writing and memory reading, only issue next transactions after the reception of responses to writing. Buffering is not available in the issuing of memory write requests. (Refer to the description of AWCACHE* of Mode Set 0 Register (offset: 0x314).)

When accessing to the region out of the window set. the response will be "OKAY" but request is discarded in the module.

(1) Memory Write Transaction from the AXI

A write transaction from the AXI via a window is converted into a MWr command and then issued.

- Number of write transactions which can be accepted at a time: 1
- Write data are held in an internal buffer.
- The order between memory write transactions is preserved (with the exception of transactions for messages and configuration).
- The order of memory write and other transactions is not preserved (a preceding memory read may be overtaken).
- To preserve the order between memory writing and memory reading or of Msg and MsgD, make sure that a next transaction is only issued after a response is returned. Buffering is not available in the issuing of memory write requests.
- Writing does not proceed if the PCI power state is not D0.
- Transactions where all bits of WSTRB are 0 during a burst return an "OKAY" response, but the written data are not reflected.
- Memory requests are not issued after an acknowledgement for PME_Turn_Off Message reception/Non-D0 State transition request reception has been asserted (Precautions: Only for Endpoint mode).

(2) Memory Read Transaction from the AXI

A read transaction from the AXI via a window is converted into an MRd command and then issued.

- Number of read transactions which can be accepted at a time: 1 to 8

- Read data are held in an internal buffer to preserve the order of transactions with the same ID.
- To preserve the order of memory read transactions from a given master ID, the system can also wait until the indicator of completion of a preceding read transaction is returned. The method of waiting is selectable as either of the above by the setting of an internal register as listed in the table below.

PCIe request order*1	Method of waiting	Performance	Severity of order
0 (initial value)	Data read are held in an internal buffer.	✓	—
1	Have the system wait for a read request to be issued.	—	✓

Note 1. The order can be set by using the PCIe Request Order bit (bit [0]) of Mode Set 1 Register (offset: 0x318).

- The order of memory read transactions from a different master ID is not preserved.
- The order of memory read and other transactions is not preserved.
- To preserve the order between memory reading and memory writing, make sure that a next transaction is only issued after a response is returned. Buffering is not available in the issuing of memory write requests.
- MRdLk requests are not supported and therefore cannot be issued.
- Reading does not proceed if the PCI power state is not D0.
- Memory requests are not issued after an acknowledgement for PME_Turn_Off Message reception/Non-D0 State transition request reception has been asserted (Precautions: Only for Endpoint mode).

56.7.3.4 Issuing Special Requests

The following requests can be issued by controlling the internal registers.

- Configuration Read Type0/1
- Configuration Write Type0/1
- Zero-Length Memory Read Request
- Message Request
- Message Request with data payload

The internal registers are only writable from the AXI side. Attempted writing from the PCIe side is ignored and the write operation returns an "OKAY" response, but the written data are not reflected.

The figure below shows the flow of issuing requests by the registers.

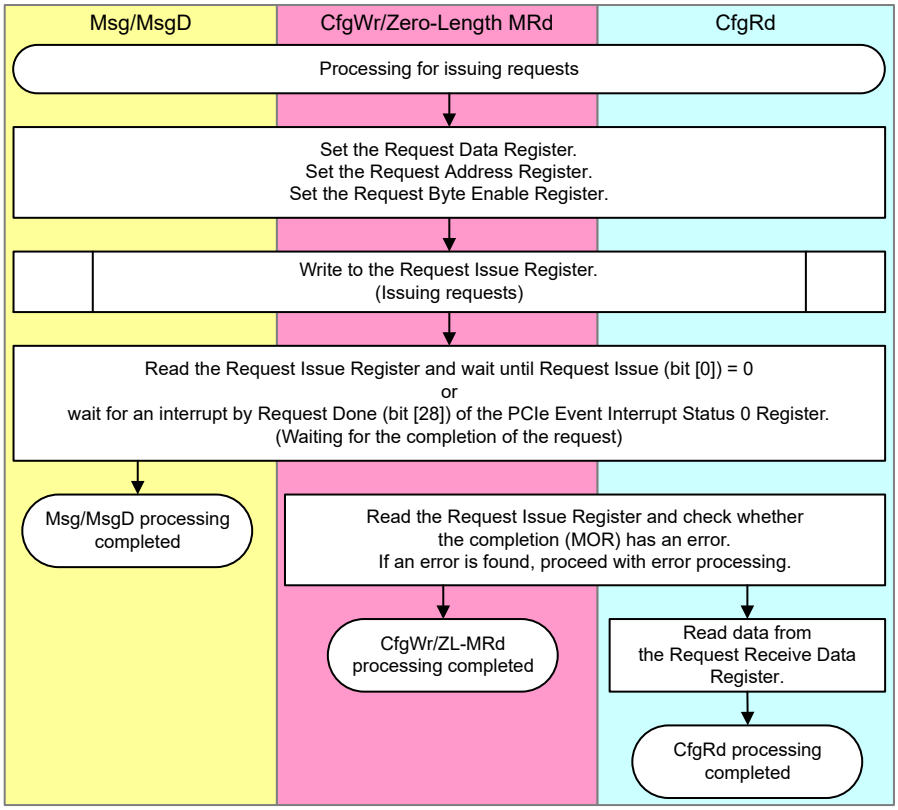


Figure 56.8 Flow of issuing requests

(1) Issuing Configuration Requests

A configuration request is automatically issued by setting the Request Issue bit (bit [0]) of the Request Issue Register after access to an internal register through the AXI slave interface and setting a destination in Request Address Register 1.

(a) Flow of Issuing Configuration Requests

An example of the flow of issuing configuration requests is described below.

1. Set a destination in Request Address Register 1.

Bits [31:24]: Bus Number[7:0]	Specify the number from 0 to 255.
Bits [23:19]: Device Number[4:0]	Speciy the number from 0 to 31.
Bits [18:16]: Function Number[2:0]	Specify the number from 0 to 7.
Bits [15:12]: Reserved	Fixed to 0000b.
Bits [11:8]: Extend Register Number[3:0]	Set the number of address.
Bits [7:2]: Register Number[5:0]	Set the number of address.
Bits [1:0]: Reserved	Fixed to 00b.

2. Set the Byte Enable in Request Byte Enable Register. When reading/writing each byte, set Enable/Disable each corresponding byte lane (Default: 1111b).
3. For CfgWr, set write data in Request Data Registers 2 (not required in case of CfgRd).
4. Configuration requests are automatically issued by setting the Request Issue bit (bit [0]) of the Request Issue Register to 1 (the TR type bits (bits [11:8]) of the Request Issue Register are used to set CfgWr/CfgRd).
5. Read the Request Issue Register and wait (polling) or wait for an interrupt of Request Done (bit [28]) of PCIe Event Interrupt Status 0 Register until the Request Issue bit (bit [0]) is set to 0, wait for the request to end.
6. For CfgRd, CpID is stored to Request Receive Data Register from facing Endpoint device. Read this register.

(2) Issuing Message Requests

A message request is automatically issued by setting the Request Issue bit (bit [0]) of the Request Issue Register after access to an internal register through the AXI slave interface and setting a destination in Request Address Register 1.

(a) Flow of Issuing Message Requests

An example of the flow of issuing message requests is described below.

1. Set a destination in Request Address Register 1.

   Bits [31:27]: Reserved Fixed to 0_0000b.

   Bits [26:24]: Routing Type Specify the message routing.

   Bits [23:8]: Reserved Fixed to 0x0000.

   Bits [7:0]: Message CodeSpecify the message code.

2. Set the 3rd header in Request Data Register 0 as required.
3. Set the 4th header in Request Data Register 1 as required.
4. For MsgD, set write data in Request Data Register 2 (not required in the case of Msg).
5. Message requests are automatically issued by setting the Request Issue bit (bit [0]) of the Request Issue Register to 1 (the TR type bits (bits [11:8]) of the Request Issue Register are used to set Msg/MsgD).
6. Read the Request Issue Register and wait (polling) or wait for an interrupt of Request Done (bit [28]) of PCIe Event Interrupt Status 0 Register until the Request Issue bit (bit [0]) is set to 0, wait for the request to end.

(b) Notes when Issuing Message Requests

Points to note when issuing message requests are described below.

- INTx and error type messages are automatically issued. They do not require issuing as message requests by the user (prohibited).
- Set message headers appropriately.
- A value to be set in a message header must be 0 except when sending a vendor defined message.

(3) Issuing Zero-Length Read Requests

A zero-length read request is automatically issued by setting a destination in Request Address Register 1 after access to an internal register through the AXI slave interface and setting the Request Issue bit (bit [0]) of the Request Issue Register.

(a) Flow of Issuing Zero-Length Read Requests

An example of the flow of issuing zero-length read requests is described below.

1. Set a destination in Request Address Register 1.

   Bits [31:2]: Address Set the address.

   Bits [1:0]: Reserved Fixed to 00b.

2. Set a destination in Request Address Register 1.

   Bits [64:32]: Address Set the address.

3. Set Byte Enable (0000b) in Request Byte Enable Register.
4. Zero-length read requests are automatically issued by setting the Request Issue bit (bit[0]) of the Request Issue Register to 1 (the TR type bits (bits [11:8]) of the Request Issue Register are used to set zero-length read).
5. Read the Request Issue Register and wait (polling) or wait for an interrupt of Request Done (bit [28]) of PCIe Event Interrupt Status 0 Register until the Request Issue bit (bit [0]) is set to 0, wait for the request to end.

56.7.4 Initiation of AXI Transactions and Register Access (by PCIe)

The following gives functional description on access to the AXI bus by the PCI Express core.

56.7.4.1 PCIe Requests which can be Received (Supported Commands for RX)

MRd:	Memory Read Request
MWr:	Memory Write Request
CfgRd0:	Configuration Read Type 0
CfgWr0:	Configuration Write Type 0
Msg:	Message Request
Cpl:	Completion
CplD:	Completion with Data

The following requests are not supported.

MRdLk:	Memory Read Request-Locked
IORd:	I/O Read Request
IOWr:	I/O Write Request
CplLK:	Completion for Locked Memory Read without Data
CplDLk:	Completion for Locked Memory Read
CfgRd1:	Configuration Read Type 1
CfgWr1:	Configuration Write Type 1

AtomicOP is not supported.

56.7.4.2 Initiation of AXI Transactions

Access from the PCIe side is converted into an AXI transaction and then issued. Up to eight windows can be allocated.

To avoid a read-after-write (RAW) hazard, only issue a next memory write request following reception of the completion TLP for the previous one.

Table 56.129 AXI transaction from PCIe request

PCIe request	AXI burst type	AXI bus size	Division
MWr/MRd	Incremental	64	Yes
		32	No
IOWr/IORd	Unsupported request		
MRdLk	Unsupported request		

(1) Memory Write Transaction from the PCIe

- Allowable number of write requests to be issued: Variable (the value is set by the AXI Max Issue Write bits (bits [15:12]) of Mode Set 1 Register).
- Write data are not held in an internal buffer (buffering within the PCIe core).
- The order between memory write transactions is preserved.
- The order of memory writing and memory reading is not preserved.
- To preserve the order between memory writing and memory reading, make sure that a next request is only issued after a completion TLP is returned.
- AWCACHE and AWPROT can be set by using Mode Set 0 Register.
- In cases of contention between DMA transfer (from PCIe to AXI) and memory write transaction, requests are accepted in turn in round-robin fashion.

(2) Memory Read Transaction from the PCIe

- Allowable number of read requests to be issued: Variable (the value is set by the AXI Max Issue Read bits (bits [11:8]) of Mode Set 1 Register).
- Since the PCI specification requires dword-aligned memory addresses in transfer, invalid byte lanes may be read. If an AXI slave as a read destination has a register or FIFO buffer which has been cleared by being read, the transaction may be non-compliant.
- Data read are held in an internal buffer.
- The order between memory read operations is preserved.
- A preceding memory write is not overtaken.
  Data read are held in an internal buffer until a preceding memory write is completed.
  When a zero-length read request is received, an "OKAY" response is returned, but the written data are not reflected and a completion TLP is transmitted after waiting for the completion of the preceding memory write.
  Data read are held in an internal buffer until a preceding memory write is completed.
- ARCACHE and ARPORT can be set by using Mode Set 0 Register.
- In cases of contention between DMA transfer (from AXI to PCIe) and memory read transaction, requests are accepted in turn in round-robin fashion.

56.7.4.3 Narrow Transfer

Narrow transfer to be initiated by the AXI transfer only supports the following AXI transactions.

- MAxBURST = INCR
- MAxSIZE = 0x0 to 0x2 (8 bits / 16 bits / 32 bits)
- MAxLEN = 0x0 (1 beat)

The following restrictions apply to TLPs from the PCIe module in narrow transfer.

- The length is 1 dword.
- The First Byte Enable bit only supports the following.
  - MAxSIZE = 0x0 (8 bits): 1000b / 0100b / 0010b / 0001b
  - MAxSIZE = 0x1 (16 bits): 1100b / 0011b
  - MAxSIZE = 0x2 (32 bits): 1111b

56.7.5 DMAC Functions

This section explains functions of the DMAC within the unit. Control by registers and by descriptors are both supported as methods of DMA control. The method of control is independently selected per channel.

56.7.5.1 Register-Type Transfer

Table 56.130 Registers related to register-type DMA transfer (1 of 2)

Common control	
0x800	DMAC Control Register
0x808	DMAC Interrupt Enable Register
0x080C	DMAC Interrupt Status Register
Channel control	
0x900 + channel offset	DMA Channel Control Register m
DMA setting	
0x924 + channel offset	DMA Transaction Control Register m

Table 56.130 Registers related to register-type DMA transfer (2 of 2)

Common control	
0x928 + channel offset	DMA Size Register m
0x930 + channel offset	DMA Source Lower Address Register m
0x934 + channel offset	DMA Source Upper Address Register m
0x938 + channel offset	DMA Destination Lower Address Register m
0x93C + channel offset	DMA Destination Upper Address Register m
DMA status	
0x950 + channel offset	DMA Rest Size Register m
0x960 + channel offset	AXI Request Address (Lower) Register m
0x964 + channel offset	AXI Request Address (Upper) Register m
0x968 + channel offset	PCIe Request Address(Lower) Register m
0x96C + channel offset	PCIe Request Address(Upper) Register m
0x978 + channel offset	DMAC Error Status Register m

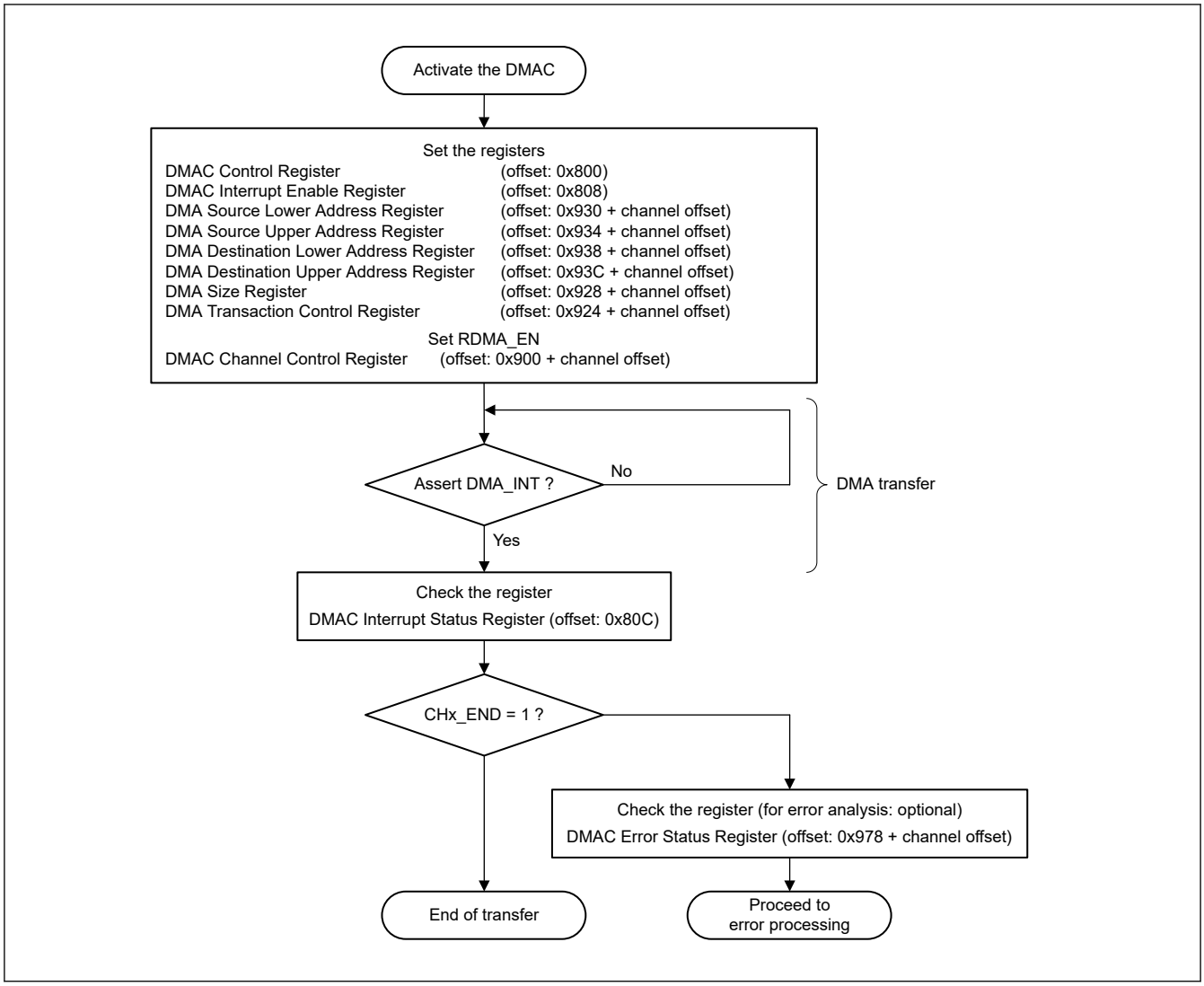
Note: One transfer byte size is up to 4 GB for register-type DMA transfer.

(1) Flow of Operations

The following describes the procedures for settings to activate and stop the DMAC in the case of register-type transfer.

(a) Activation and Normal Operation

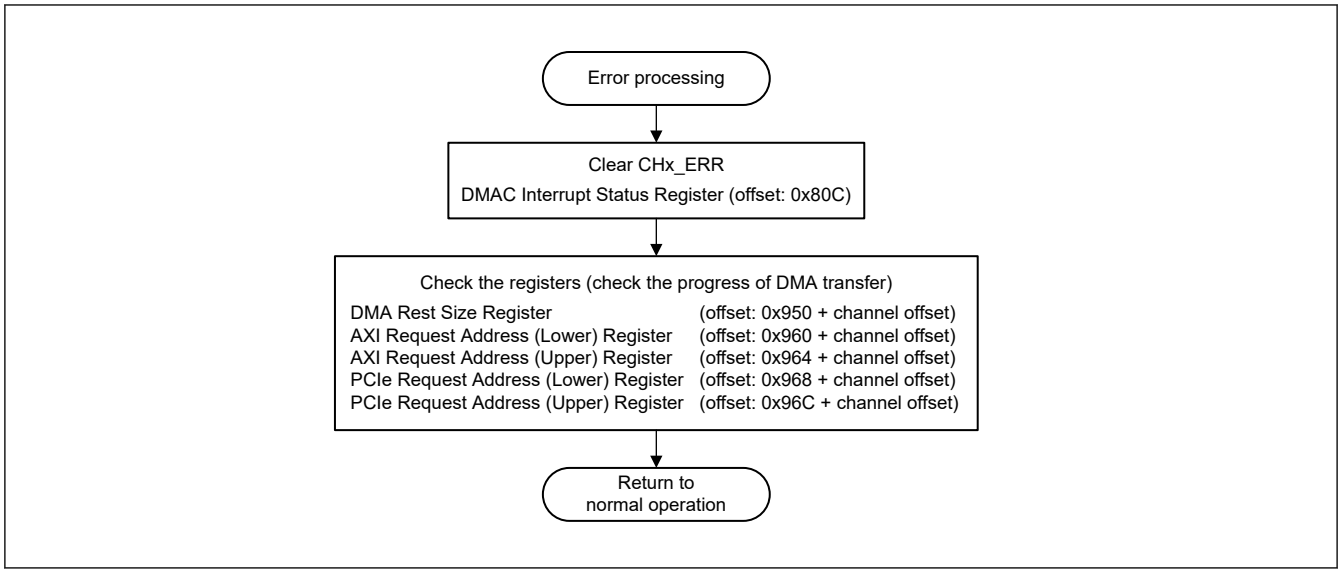
Before activating the DMAC, set the PCIe and AXI windows. After that, set the registers of the DMAC to start the DMAC.



(b) Error Processing

If an error is detected in the AXI bus or a PCIe link while DMA transfer is in progress, the DMA_INT interrupt signal is asserted. Check the channel where the error was found (CHx_ERR = 1) by using the DMAC interrupt status register and also the type of error by using the DMAC error status register (offset: 0x978 + channel offset) as required.

Clear the CHx_ERR bit that is currently set in the DMAC interrupt status register (offset: 0x80C) to 0 by writing to it. The address to which transfer was most recently completed before it was stopped and the remaining number of bytes for transfer can be checked by reading the DMA Status registers (offset: 0x950 + channel offset to 0x978 + channel offset) for the given channel.



56.7.5.2 Descriptor-Type Transfer

Consecutive DMA transfer is achieved by sequentially reading descriptors which indicate parameters of DMA transfer. Descriptors are allocated to the AXI memory space and the descriptor lists indicate the addresses where the descriptors start. This DMAC has a queue for storing multiple descriptor lists and these descriptor lists are written by software. The first list loaded in the queue is executed after being moved to become the descriptor list for execution following the detection of the condition for starting DMA transfer (the DMAC retains up to nine lists, including the one currently being executed).

Descriptors can have a chained configuration; execution of a descriptor list ends on detection of the last descriptor, and if the queue has a next list, execution of the next descriptor follows.

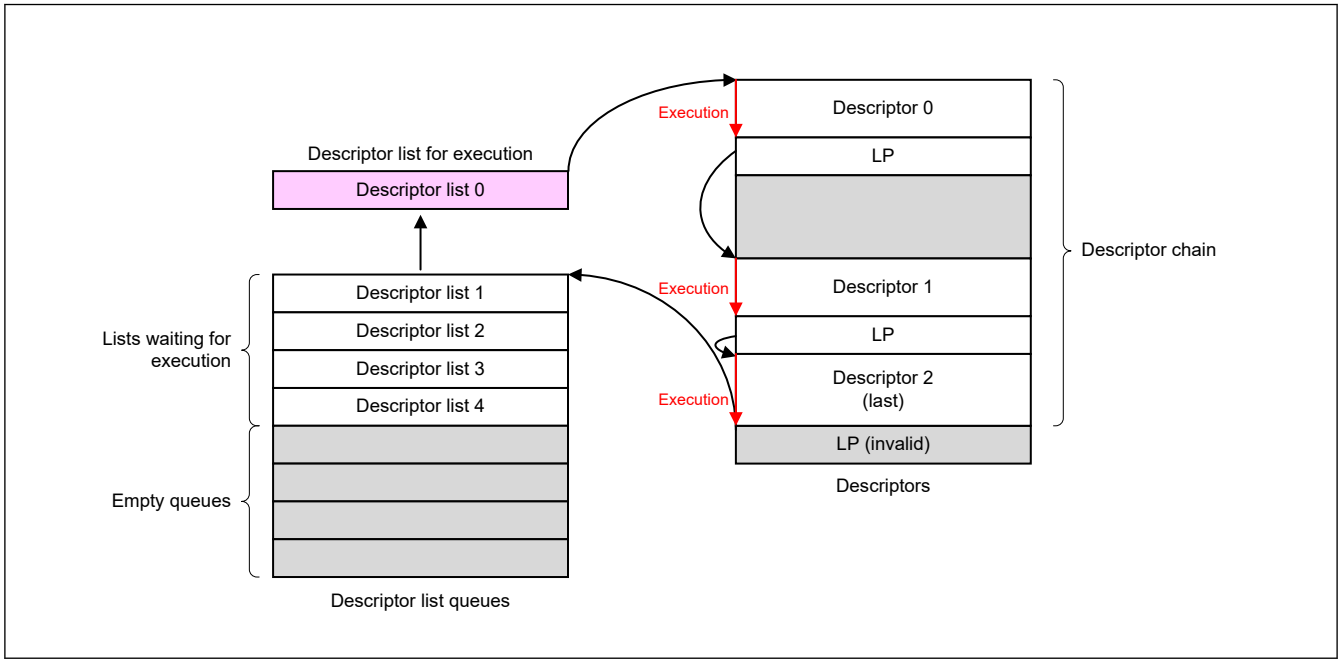


Figure 56.9 Descriptor chain

The table below lists the DMAC registers for execution of descriptor-type DMA transfer.

Table 56.131 Registers related to descriptor-type DMA transfer (1 of 2)

Common control	
0x800	DMAC Control Register

Table 56.131 Registers related to descriptor-type DMA transfer (2 of 2)

Common control	
0x808	DMAC Interrupt Enable Register
0x80C	DMAC Interrupt Status Register
Channel control	
0x900 + channel offset	DMA Channel Control Register m
0x908 + channel offset	Descriptor Start Address (Lower) Register m
0x90C + channel offset	Descriptor Start Address (Upper) Register m
0x910 + channel offset	QUE Entry Register
DMA setting	
0x920 + channel offset	DMA Descriptor Control (Descriptor 0x00) Register m
0x924 + channel offset	DMA Transaction Control (Descriptor 0x04) Register m
0x928 + channel offset	DMA Size (Descriptor 0x08) Register m
0x930 + channel offset	DMA Source Lower Address (Descriptor 0x10) Register m
0x934 + channel offset	DMA Source Upper Address (Descriptor 0x14) Register m
0x938 + channel offset	DMA Destination Lower Address (Descriptor 0x18) Register m
0x93C + channel offset	DMA Destination Upper Address (Descriptor 0x1C) Register m
0x940 + channel offset	DMA Descriptor Link Lower Pointer (Descriptor 0x20) Register m
0x944 + channel offset	DMA Descriptor Link Upper Pointer (Descriptor 0x24) Register m
DMA status	
0x950 + channel offset	DMA Rest Size Register m
0x960 + channel offset	AXI Request Address (Lower) Register m
0x964 + channel offset	AXI Request Address (Upper) Register m
0x968 + channel offset	PCIe Request Address(Lower) Register m
0x96C + channel offset	PCIe Request Address(Upper) Register m
0x970 + channel offset	QUE Status Register m
0x978 + channel offset	DMAC Error Status Register m

(1) Descriptor List Queues

The DMAC has queues for storing descriptor lists for each of the channels. The number of FIFO queue stages is eight, so it can hold up to eight descriptor list entries (not including the one currently being executed).

Software places descriptor list entries in the queue. The descriptor lists to be entered are set by the Descriptor Start Address (Upper), Descriptor Start Address (Lower) and QUE_ENTRY registers.

Since the target registers are also accessible in 8-bit units, making the settings in byte units through multiple accesses is also possible. A descriptor list entry is placed in the queue at the time when the QUE_ENTRY (upper) bits [31:24] are written. When making the settings and entering lists in the queue, take care with the order of register access (values written to the QUE_ENTRY (upper) bits [31:26] will not affect the register values and settings because they are readonly bits).

When the DMAC detects the presence of a list which has been entered in the queue, the first list entry is moved to become the pointer to the next descriptor for execution and DMA transfer starts.

(2) Descriptor List Format

The table below is the format of a descriptor list.

Table 56.132 Descriptor list format (1 of 2)

Offset	Byte 3			Byte 2	Byte 1	Byte 0
0x00	DSA[31:0]					
0x04	DSA[63:32]					
0x08	Reserved	EI	LS	Reserved	LABEL[15:0]	

Table 56.132 Descriptor list format (2 of 2)

Field name	Description
DSA[63:0]	Descriptor start address for DMA. This field indicates the address where the first descriptor to be executed is stored. Since the setting is for 16-byte alignment and allocation of a single descriptor (0x00 to 0x24) to straddle a 4-K boundary is prohibited, the 6 lower-order bits [5:0] are fixed to 000000b.
EI	End Interrupt bit. This bit indicates whether an interrupt (Interrupt Status CHx_END of the DMAC Interrupt Status Register (offset: 0x80C)) is or is not conveyed when processing of this descriptor list is completed. 0: The interrupt is not conveyed. 1: The interrupt is conveyed.
LS	List Stop bit. This bit indicates whether DMA transfer is or is not to be stopped on completion of processing for this descriptor list entry. 0: Not stopped 1: Stopped On completion of the list in which the setting of this bit is 1, the QUE_EN bit of the DMAC control register is cleared.
LABEL[15:0]	This field represents the label of the list. There are no special rules for the setting procedures. The value can be set as desired.

Operations by the settings of the EI and LS bits following the completion of the descriptor list are as follows.

Table 56.133 Details of the EI/LS bits

EI	LS	Description
0	0	No interrupt. DMA transfer is not stopped. If the queue has a next list, execution of the next descriptor follows.
0	1	No interrupt. DMA transfer is stopped (QUE_EN is cleared).
1	0	Interrupt (DMAC Interrupt Status Register (Offset: 0x80C), Interrupt Status CHx_END) is present. DMA transfer is not stopped. If the queue has a next list, execution of the next descriptor follows.
1	1	Interrupt (DMAC Interrupt Status Register (Offset: 0x80C), Interrupt Status CHx_END) is present. DMA transfer is stopped (QUE_EN is cleared).

(3) Descriptor Format

The table below is the format of descriptors.

Table 56.134 Descriptor list format (1 of 4)

Offset	Byte 3								Byte 2								Byte 1								Byte 0							
0x00	DSCFM[3:0]				—	WBD	LE	LV	D	—	—	—	—	—	—	—	STS[15:0] (not used)															
0x04	—	—	—	—	—	—	—	—	CCH_L[3:0]				CCH_D[3:0]				—	TC[2:0]			—	—	ATB[1:0]		—	DMA_FUNC[2:0]				—	—	—	DIR
0x08	SIZE[31:0]																															
0x0C	Reserved																															
0x10	SA[31:0]																															
0x14	SA[63:32]																															
0x18	DA[31:0]																															

Table 56.134 Descriptor list format (2 of 4)

Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x1C	DA[63:32]			
0x20	LP[31:0]			
0x24	LP[63:32]			

Table 56.134 Descriptor list format (3 of 4)

Field name	Description
DSCFM[3:0]	The Descriptor Format field specifies the format of descriptors. This DMAC only supports 0001b. Do not set any other value.
WBD	Write Back Disable bit Indicates whether the DMAC writes 0 back to the LV bit when DMA transfer specified by a single descriptor is completed. 0: The LV bit is written back. 1: The LV bit is not written back.
LE	List End bit Indicates the end of the current descriptor chain. 0: The current descriptor is not the last of the chain. 1: The current descriptor is the last of the chain.
LV	Link Valid bit Indicates that the descriptor is valid. When DMA transfer is completed, the DMAC writes 0 back to this bit. When DMA transfer ends due to an error, write-back does not proceed. 0: The descriptor is not valid (DMA transfer specified by the descriptor is completed). 1: The descriptor is valid (DMA transfer specified by the descriptor is not completed).
D	Descriptor error bit Indicates a descriptor access error. If LV = 0 when the descriptor is read, the DMAC writes 1 back to the LV bit. When a descriptor error occurs, if the setting of the LE bit is 0 (indicating that the current descriptor is not the last of the chain), the DMAC continues transfer in accord with the descriptor the LP bits indicate. 0: No error 1: A descriptor access error occurred.
STS[15:0]	This field has no effect when DSCFM = 0001b. The DMAC does not use this field.
CCH_L[3:0]	This field indicates the value of A*CACHE[3:0] to be issued through the AXI. The value of CCH_L[3:0] is output when an AXI request including the last byte is issued in transfer specified in the SIZE bits. Bit [0]: Buffer enabled Bit [1]: Cache enabled Bit [2]: Read allocation Bit [3]: Write allocation The recommend value is 0000b regardless of whether DIR is 0 or 1.
CCH_D[3:0]	This field indicates the value of A*CACHE[3:0] to be issued through the AXI. The value of CCH_D[3:0] is output when an AXI request other than the condition for the output of CCH_L[3:0] is issued. Bit [0]: Buffer enabled Bit [1]: Cache enabled Bit [2]: Read allocation Bit [3]: Write allocation The recommended value is 0001b when DIR = 0 (PCIe → AXI) and 0000b when DIR = 1 (AXI → PCIe).
TC[2:0]	This field specifies the value of the traffic class of the request to be issued through the PCIe interface. The value must be fixed to 000b.
ATB[1:0]	This field indicates the value of the attribute to be issued through the PCIe interface. Bit [0]: No-snoop Bit [1]: Relaxed ordering If neither relaxed ordering nor no-snoop is used, this field should be set to 00b (recommended).
DMA_FUNC[2:0]	This field specifies the function number of the request to be issued through the PCIe interface.

Table 56.134 Descriptor list format (4 of 4)

Field name	Description
DIR	This bit indicates the direction of data transfer. 0: PCIe → AXI 1: AXI → PCIe
SIZE[63:0]	This field indicates the number of bytes for transfer. Since the setting is for 16-byte alignment, do not set a value other than 0000b to the 4 lower-order bits ([3:0]).
SA[63:0]	The Source Address field indicates the address of the source data for transfer. Since the setting is for 16-byte alignment, do not set a value other than 0000b to the 4 lower-order bits ([3:0]).
DA[63:0]	The Destination Address field indicates the destination address. Since the setting is for 16-byte alignment, do not set a value other than 0000b to the 4 lower-order bits ([3:0]).
LP[63:0]	This field indicates the address where a next descriptor is stored. Since the setting is for 16-byte alignment and allocation of a single descriptor (0x00 to 0x24) to straddle a 4-K boundary is prohibited, the 6 lower-order bits [5:0] are fixed to 000000b.

The conditions where the descriptor is written back after the descriptor is read, the timings, and the corresponding bits are listed below.

Condition	Timing	Corresponding bit and value
WBD = 0 && LV = 1	After DMA transfer specified by the descriptor is completed	LV = 0
WBD = (1 or 0) && LV = 0	After the descriptor is read (DMA transfer does not proceed)	D = 1
WBD = 1 && LV = 1	Write-back does not proceed.	—

(4) Flow of Operations

The following describes the procedures for settings to activate and stop the DMAC in the case of descriptor-type transfer.

(a) Activation and Normal Operation

Before activating the DMAC, set the PCIe and AXI windows. After that, set the registers of the DMAC to start the DMAC.

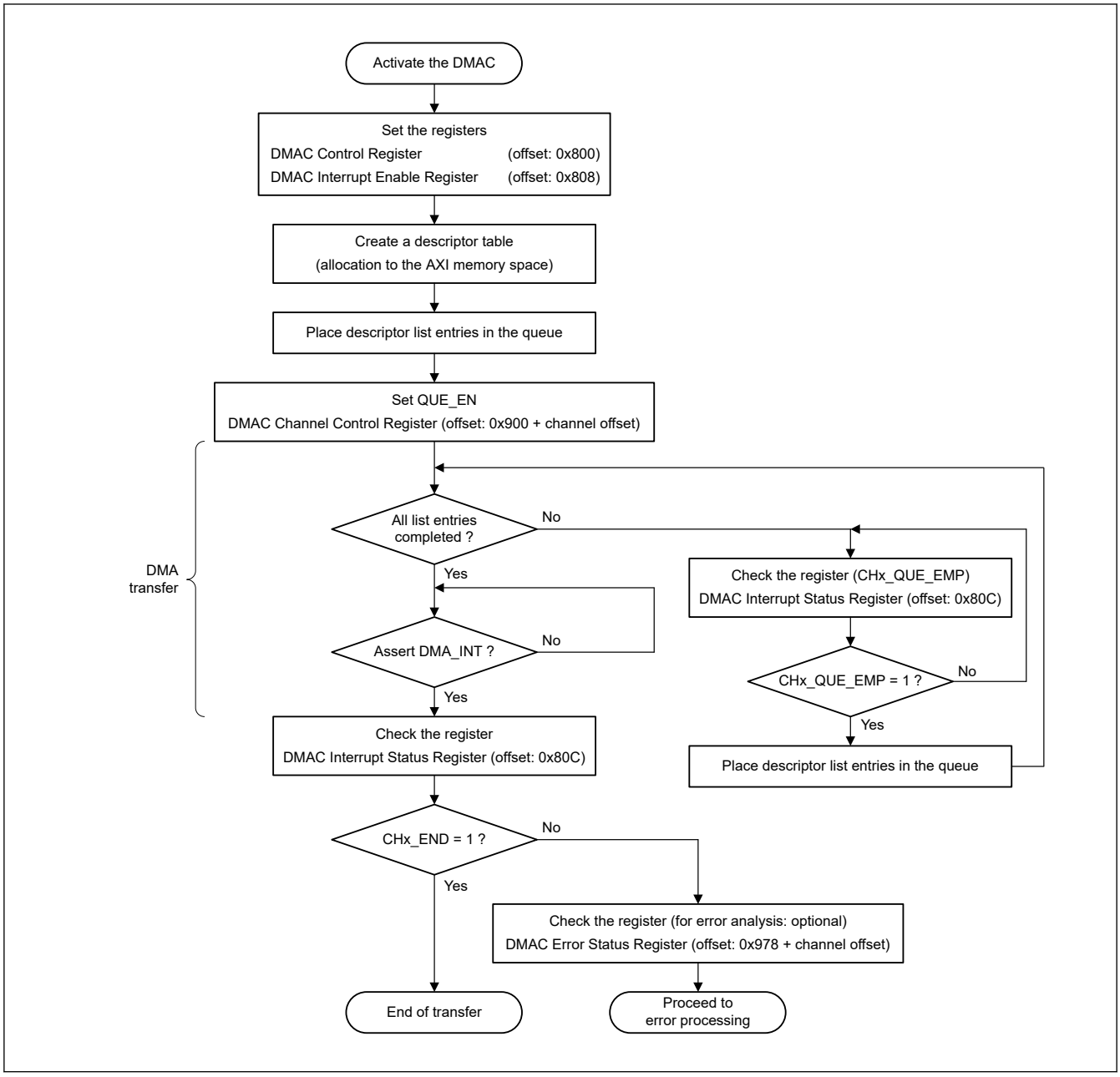


Figure 56.10 Flow of activation and normal operation

(b) Error Processing

If an error is detected in the AXI bus or a PCIe link while DMA transfer is in progress, the DMA_INT interrupt signal is asserted. Check the channel where the error was found (CHx_ERR = 1) by using the DMAC interrupt status register and also the type of error by using the DMAC error status register (offset: 0x978 + channel offset) as required.

Clear the CHx_ERR bit that is currently set in the DMAC interrupt status register (offset: 0x80C) to 0 by writing to it. The address to which transfer was most recently completed before it was stopped and the remaining number of bytes for transfer can be checked by reading the DMA Status registers (offset: 0x950 + channel offset to 0x978 + channel offset) for the given channel.

Based on that information, setting the registers as described in (a) Activation and Normal Operation and create a descriptor table again. At this time, write 1 to the QUE_CLR bit (bit [8]) of the DMAC channel control register (offset: 0x900 + channel offset). Writing 1 to it leads to clearing of the queue. The descriptor lists registered in the queue (those waiting for execution and the one currently being executed) are all cleared.

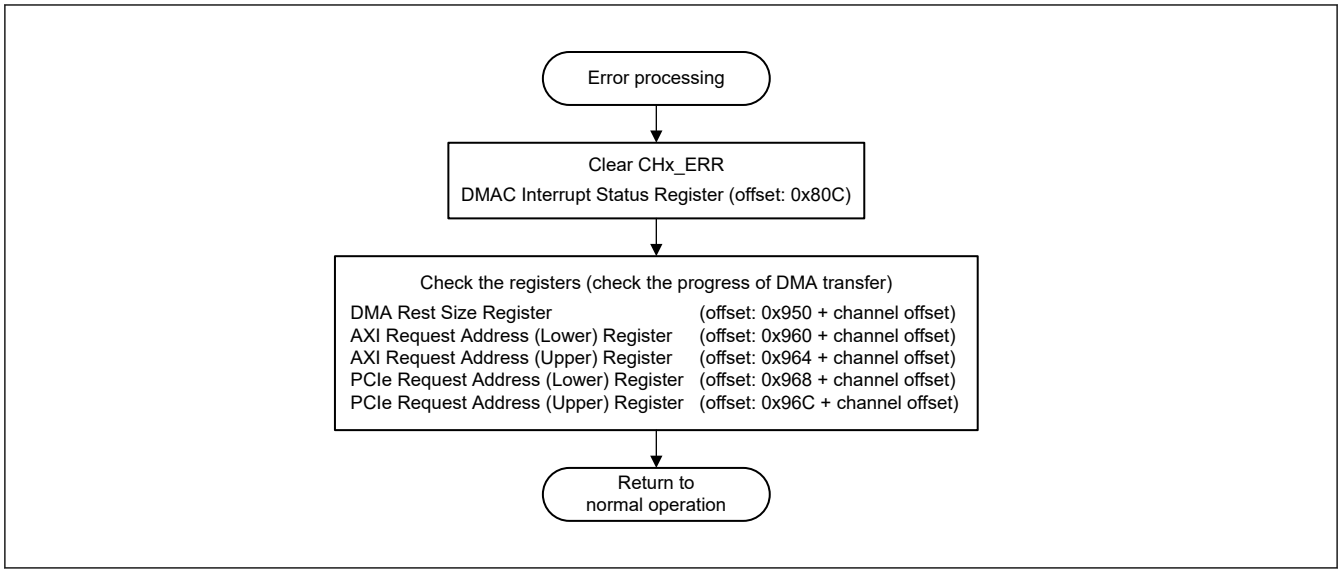


Figure 56.11 Flow of error processing

56.7.5.3 Method of Transfer

This section describes the issuing of requests to the AXI or PCIe and data transfer in the case of DMA transfer. In either direction of transfer (from AXI to PCIe or vice versa), a request is issued to the reading side. After confirming the completion of preparation (reception) of the data to be read, a request is issued to the writing side for data transfer.

(1) DMA Transfer from PCIe to AXI

An MRd is issued to the PCIe module and an AXI write request is transferred following the reception of CplD from the PCIe. MRd requests corresponding to the number of outstanding transfers can be issued regardless of the reception of CplD. The received CplD (read data) is stored in the data buffer (RAM) once and then transferred by the DMAC to the AXI.

(2) DMA Transfer from AXI to PCIe

After the DMAC is activated, a read request (address channel) is issued to the AXI. The reception of data for reading from the given AXI slave starts and an MWr is issued to the PCIe. Data are repeatedly transferred until the completion of DMA data transfer corresponding to the amount of data set in the DMA Size bits.

56.7.5.4 Inter-Channel Arbitration

The DMAC arbitrates the following three types of request between channels.

1. PCIe Read Request
2. AXI Read Request
3. AXI Write Request

PCIe write requests do not require arbitration because data read over the AXI are returned one by one.

In the case of arbitration on the AXI side, not only requests for DMA data, but also reading and writing of descriptor data are subject to arbitration (each channel has a single source of requests for arbitration since a channel will not output a descriptor and request for data at the same time).

(1) Method of Arbitration

An MRd is issued to the PCIe module and an AXI write request is transferred following the reception of CplD from the PCIe. MRd requests corresponding to the number of outstanding transfers (eight) can be issued regardless of the reception of CplD. The received CplD (read data) is stored in the data buffer (RAM) once and then transferred by the DMAC to the AXI.

Arbitration for each type of request 1) to 3) above proceeds in round-robin fashion. A channel for which execution has just been completed is given the lowest priority and each channel which had a lower priority than that channel is raised by one rank in the order of priority.

The initial order of priority is ch. 0 > ch. 1 > ch. 2 > ch. 3 > ch. 4 > ch. 5 > ch. 6 > ch. 7.

The order of priority is changed every time a single request is completed.

There are two types of request for the AXI, requests for DMA data and requests for descriptors, but they are not distinguished for purposes of arbitration.

56.7.5.5 DMA Completed Interrupt

The DMAC has interrupt functions. It supports two types of interrupt source: one set is for normal operation and the other is for errors. For details, see section 56.7.7.5. DMA Interrupt.

56.7.6 Reception of PCIe Commands

56.7.6.1 Reception of MSIs (Root Complex)

When MSI is received (see judgment conditions), it is forwarded to AXI as a Write Transaction and reflected in the MSI reception status (see status). An interrupt signal (INTMSI_RC) is asserted when the MSI reception interrupt generation condition (see Interrupt conditions) is met. S/W identifies MSI by the interrupt factor. If Message Data Enable = 0, the cause can be identified by reading the data on the memory.

[Related registers]

- MSI Receive Window Address (Lower) Registers (Offset: 0x100): Message Data Enable
- PCI INTx Receive Interrupt Enable Registers (Offset: 0x110): MSI Receive Interrupt Enable
- PCI INTx Receive Interrupt Status Registers (Offset: 0x114): MSI Receive Interrupt Status
- MSI Receive Mast n Registers (Offset: 0x6x8): MSI Mask
- MSI receive Status n Registers (Offset: 0x6xC): MSI Status

The conditions for judging a request to be an MSI:

When all of the following conditions are satisfied, a request is judged to be an MSI.

- The request is a memory write request.
- The request is entered for any area within an AXI window.
- The address of an MWr from the PCIe interface must be within the area set by the MSI Reception Window Address register and the MSI Reception Window Mask register.
  If a memory read request is received in the MSI reception area, the request is not judged to be an MSI so an interrupt is not generated.
- The length of memory write requests is 1 dword
- When MessageData matches (Only when Message Data Enable = 1)

[Status]

- When Message Data Enable = 0
  Set MSI Receive Interrupt Status (bit[4])
- When Message Data Enable = 1
  Set the corresponding Vector bit in MSI Status

[Interrupt condition]

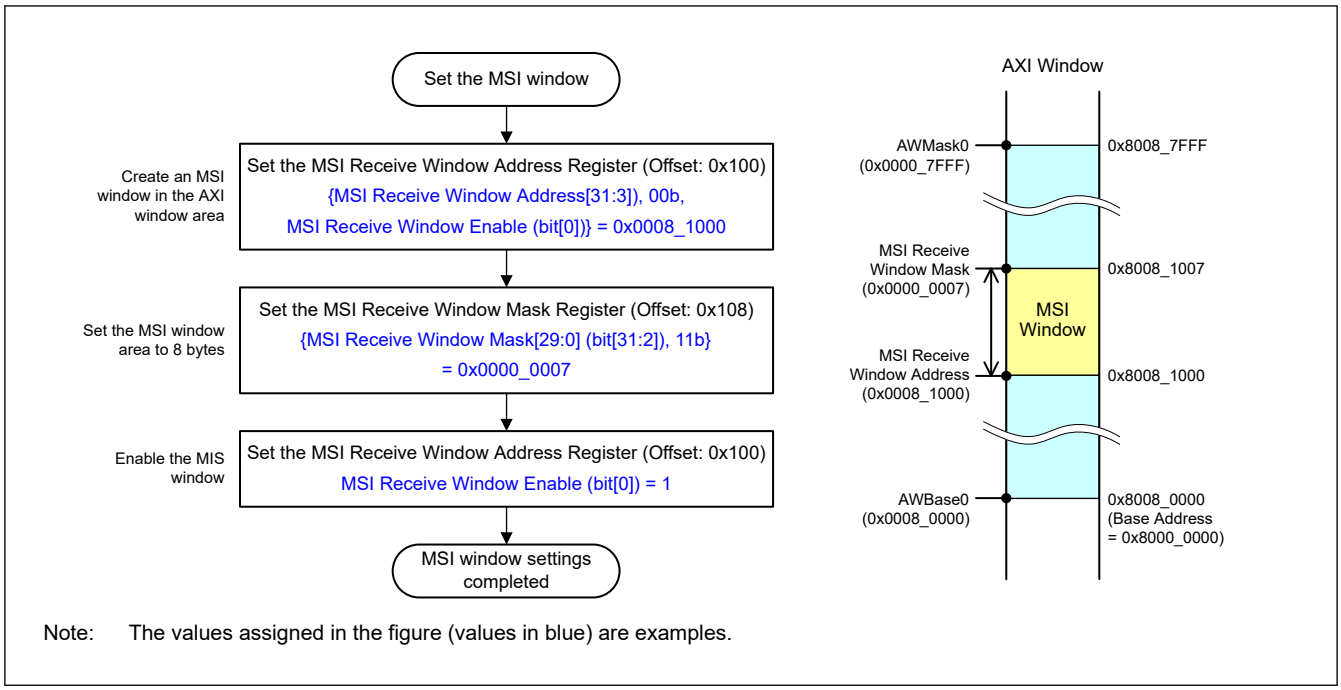
- When Message Data Enable = 0
  When MSI Receive Interrupt Enable (bit[4]) is set
- When Message Data Enable = 1
  When bit 0 of vector corresponding to MSI Mask

Notes on the MSI:

- When a response to the MSI write transaction is returned, this module judges execution of the MSI to have been completed and asserts an interrupt signal. Depending on the system, however, the MSI memory write transaction may not be completed due to the latency over the AXI to the actual memory even if the interrupt signal is asserted. To avoid this problem, buffering of AXI write transactions must be disabled (MAWCACHE[0] = 0) at the time an MSI is issued. For MAWCACHE[3:1], the setting of AWCACHE_L (bits [27:24]) of Mode Set 0 Register is used.
- If an MSI is received, the MSI write transaction is executed when MAWID[3:0] = 0x1. This is required for the unit to recognize the reception of the response to the MSI and for assertion of INTMSI_RC.

56.7.6.2 Setting the MSI Window

To enable MSI interrupts, the MSI window must be set. Figure 6.10-2 shows example settings. The MSI window can be allocated within any AXI window.



Note: The values assigned in the figure (values in blue) are examples.

Figure 56.12 Settings for the MSI window

56.7.6.3 Reception of an Interrupt in Respone to a Message Request (Root Complex)

When Assert_INTx is received in response to an Msg request from the PCIe interface, the corresponding interrupt bit (bits [3] to [0]) is set in the PCI INTx Receive Interrupt Status Register (offset: 0x114) and an interrupt (INTx_RC) is asserted. When Deassert_INTx is received in response to a message request, the corresponding interrupt register (bit) is cleared and INTx_RC is de-asserted. The PCI INTx Receive Interrupt Status Register (offset: 0x114) can be cleared by writing to it (RW1C) by software. In the PCIe, however, we do not recommend using software to clear an interrupt bit which has been set in response to an Msg in normal operation.

56.7.6.4 Reception of Message Requests

When a message is received through the PCIe interface, the relevant information is stored in the following registers.

Table 56.135 Message related registers (1 of 2)

	Related register
Message Code/Routing	Message Code Register (Offset: 0x130)
Message Data	Message Data Register (Offset: 0x134)

Table 56.135 Message related registers (2 of 2)

	Related register
Message 3rd Header	Message Header 3rdDW Register (Offset: 0x138)
Message 4th Header	Message Header 4thDW Register (Offset: 0x13C)

The corresponding bit in the Message Receive Interrupt Status Register is set at the same time as the reception of the following messages, and this is indicated by an interrupt.

Table 56.136

Received message		Corresponding bit in message receive interrupt status register
Type of message	Message code	
PM_Active_State_Nak	0001_0100	Bit [19]
PM_PME	0001_1000	Bit [18]
PME_Turn_Off	0001_1001	Bit [17]
PME_TO_Ack	0001_1011	Bit [16]

In the reception of PM messages other than the above, the relevant information is not written to the registers listed in Table 6.10-7.

- Unsupported messages:
  - Unlock
  - Vendor Defined Type 0
  - Vendor Defined Type 1

56.7.7 Interrupt

This module prepares the output of interrupt signals in the ways stated below. The states of all interrupt signals are indicated at a glance in the Interrupt Table Register.

56.7.7.1 Error and Event Interrupt Notification

Table 56.137 Error/event interrupt outputs

Interrupt source name	Active	Attribute	Description
DMA_INT	High	Level	DMA-related event
PCIE_EVT_INT	High	Level	PCIe-related event
MSG_INT	High	Level	Reception of messages
AXI_ERR_INT	High	Level	AXI-related error

Interrupt Output Related Status Registers

DMA_INT:	DMAC Interrupt Status Register (Offset: 0x80C) DMAC Error Status Register (Offset: 0x978 / + Channel Offset)
PCIE_EVT_INT:	PCIe Event Interrupt Status 0 Register (Offset: 0x204) PCIe Event Interrupt Status 1 Register (Offset: 0x20C) PCIe Event Interrupt Status 2 Register (Offset: 0x244)
MSG_INT:	Message Receive Interrupt Status Register (Offset: 0x124)
AXI_ERR_INT:	AXI Master Error Interrupt Status Register (Offset: 0x214) AXI Slave Error Interrupt Status 1 Register (Offset: 0x224)

(1) Unit-Internal RAM Parity Error Interrupt Notification

The detection of parity errors in the internal RAM of this module is indicated by the output of interrupt signals in the ways described below. The type of parity error detecting RAM can be confirmed from the corresponding register bit. A breakdown of the device is a possibility.

PCIE_EVT_INT:	PCIe Event Interrupt Status 0 Register (Offset: 0x204) PCIe Event Interrupt Status 1 Register (Offset: 0x20C)
DMA_INT:	DMAC Interrupt Status Register (Offset: 0x80C) DMAC Error Status Register (Offset: 0x968 / + Channel Offset)

Note: A RAM parity function is not available because it is not implemented in this core.

56.7.7.2 INTx/MSI Interrupt Notification (Root Complex)

Table 56.138 INTx/MSI interrupt outputs

Interrupt Source Name	Active	Attribute	Description
INTA_RC	High	Level	Set in response to the reception of an assert INTA message and cleared in response to the reception of a deassert INTA message.
INTB_RC	High	Level	Set in response to the reception of an assert INTB message and cleared in response to the reception of a deassert INTB message.
INTC_RC	High	Level	Set in response to the reception of an assert INTC message and cleared in response to the reception of a deassert INTC message.
INTD_RC	High	Level	Set in response to the reception of an assert INTD message and cleared in response to the reception of a deassert INTD message.
INTMSI_RC	High	Level	Set when a memory write request comes from the PCIe to the area set in the MSI reception window.
INT_LINK_BANDWIDTH	High	Level	Set when the link width has been changed.
INT_PM_PME	High	Level	Set in response to the reception of PME event notification (PM_PME message).
INT_SERR_COR/ INT_SERR_NONFATAL/ INT_SERR_FATAL/ INT_SERR	High	Level	Set in response to the reception of a correctable error message, non-fatal error message, fatal error message, logical OR of 3 above messages.
INT_LINK_EQUALIZATION_REQUEST	High	Level	Set in response to the reception of Link Equalization Request bit notification in Link Control 2/Status 2 Register.

56.7.7.3 Issuing INTx/MSI Interrupts (Endpoint)

An endpoint indicates two types of interrupt to the root complex.

1. Legacy interrupt (Assert INTx Message/Deassert INTx Message)
   - Issuing INTx Msg specified by PCI_EP_INT_Fi register by controlling INTX_EP_Fi bit in PCIE_INTX register (recommended)
   - Issuing Assert INTx and Deassert INTx messages through the issuing of special requests.
2. MSI
   - Issuing an MSI by generating a memory write transaction directly.

The MSI Enable bit in the MSI Capability Register (PCIe configuration Register) can be used to switch exclusive operation of two interrupts on or off.

To issue a legacy interrupt, the Interrupt Disable bit in the Command and Status Register (PCIe Configuration Register) must be cleared.

Furthermore, if writing proceeds in a power state other than D0 or a PME_TURN_OFF message is received, or if FLR_REQ is received, Deassert INTx Msg is automatically generated in response to the corresponding function asserting an INTx, and all active interrupts are withdrawn without waiting for the PME_Turn_Off Message reception/Non-D0 State transition request reception acknowledge/FLR_RESET response. However, when multiple functions share an INTx, the logical OR of the signals from the sharing functions is taken, a Deassert INTx Msg is generated, and the interrupts are withdrawn. If INTX_EP_F0/F1 is being asserted at the time of returning to the D0 state once more or an FLR sequence is completed (indicated by the de-assertion of FLR_REQ), Assert INTx is generated again in response.

Table 56.139 INTx interrupt inputs

Interrupt factor name	Active	Attribute	Description
INTX_EP_F0	High	Level	This is the trigger for Assert INTx and Deassert INTx messages.
INTX_EP_F1	High	Level	

[Related registers]

PCI INTx Out Status Register (Offset: 0x118)

(1) Issuing a Legacy PCI INTx Emulation Interrupt (when MSI Enable = 0)

A legacy PCI INTx emulation interrupt is issued by controlling INTX_EP_Fi bit in PCIE_INTX register.

The type of Assert INTx and Deassert INTx Msg to be issued by controlling INTX_EP_Fi bit in PCIE_INTX register is specified by PCI_EP_INT_Fi register provided for each of the functions. Assert INTx Msg is issued by setting INTX_EP_F* to 1, and Deassert INTx Msg is issued by clearing INTX_EP_F* to 0.

Make sure that a interrupt signal successfully arrives on the receiving side following the de-assertion of an interrupt signal. After completion of processing for an interrupt from a given source, negating the signal is recommended.

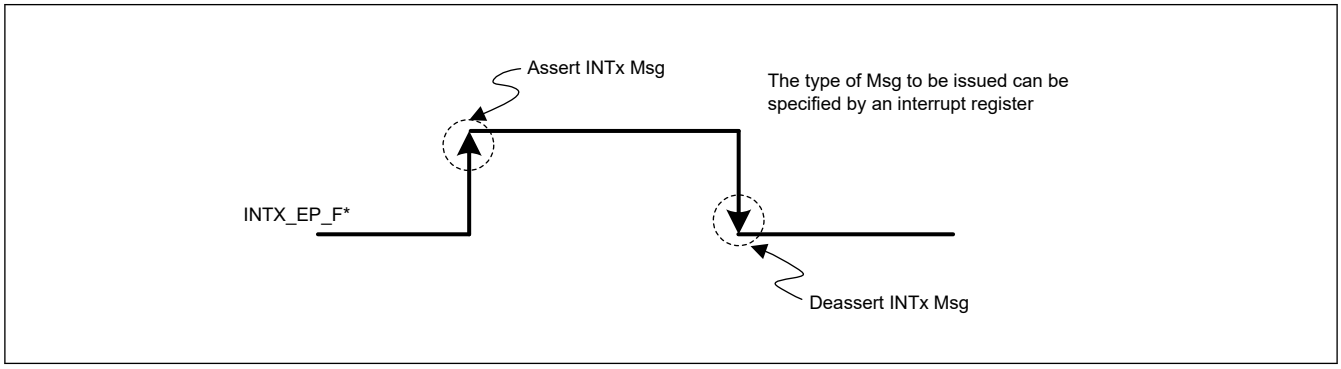


Figure 56.13 Legacy PCI INTx emulation interrupt

Changing the allocation of the legacy interrupt lines is possible by changing PCI_EP_INT_Fi register setting of the configuration register. If the same interrupt is shared by multiple functions, the logical OR of INTX_EP_F0 and INTX_EP_F1 is taken to issue an interrupt.

(2) MSI Transmission by Interupt Input (when MSI Enable = 1)

Five bits in PCIE_MSI1 register are used in specifying the interrupt (MSI). In the same way, each interrupt indication has its corresponding specified interrupt vector in PCIE_MSI2 and 3 registers.

Table 56.140 MSI interrupt notification

Register name	Interrupt notification name	Note
PCIE_MSI1	UI_EXTMSI_VAL0/1/2/3/4	Interrupt notification
PCIE MSI2 and 3	UI_EXTMSI_VEC0/1/2/3/4 [4:0]	Interrupt vector specification
PCIE MSI4 and 5	UI_EXTMSI_FUNC0/1/2/3/4 [4:0]	Interrupt function number specification

Note: Multiple Message Enable in the Base Spec.: Implemented as a specification that supports 32 (5 bits).
Multiple Message Enable on the RC side: Support for 32 is also required.

Note: Take care that the specified vector value does not impose a related restriction on the Multiple Message Enable register. For example, if the number set for Multiple Message Enable is 4, the two lower-order bits within the value set for the Message Data are overwritten by the value specified for the UI_EXTMSI_VEC[*] bits, and at that time, specify the value 0 for the higher-order bits that are not overwritten by UI_EXTMSI_VEC[*] bits. Convey interrupts in the state where the vector value and function number have been specified.

(3) Flow of Asserting MSI Enable

The following is the flow of asserting the MSI Enable field (bit 16) of the MSI Capability register.

1. Set 0 to INTX_EP_F0/1 (processing by the endpoint).
2. Wait until the Interrupt Status field (bit 19) of the Command and Status register becomes 0 (confirm that the device is not in the waiting state for INTX interrupt messages).
   (Processing by the root complex: CfgRd)
3. Write 1 to the MSI Enable field (bit 16) of the MSI Capability register.
   (Processing by the root complex: CfgWr)

56.7.7.4 Root Complex Interrupt Notification

When this module is in the root complex configuration, it has the following interrupt sources.

- INT_LINK_BAND_WIDTH
- INT_PM_PME
- INT_SERR
- INT_SERR_COR
- INT_SERR_FATAL
- INT_SERR_NONFATAL
- INT_LINK_EQUALIZATION_REQUEST

(1) Interrupt due to the Change to the Link Bandwidth (INT_LINK_BAND_WIDTH)

This interrupt source is conveyed in response to the change to the bandwidth in link negotiation of PCI Express.

Note that this interrupt source is enabled when the root complex in use and it is fixed to "0: Low" when the endpoint is in use.

This interrupt source is only enabled when the Link Bandwidth Notification Capability bit (bit 21) of the Link Capabilities register (offset: 0x60C) is set to 1. This interrupt signal can also be disabled by setting the Link Bandwidth Notification Capability bit to 0 even in operation as the root complex.

If the Link Bandwidth Notification Capability bit (bit 21) is set to 1, the Link Bandwidth Management Status bit (bit 30) and the Link Autonomous Bandwidth Status bit (bit 31) in the Link Status register (offset 0x70) serve as a source of the interrupt. These bits are set to 1 when the bandwidth is changed.

Whether to enable this interrupt source corresponds to the respective states of the Link Bandwidth Management Interrupt Enable bit (bit 10) and the Link Autonomous Bandwidth Interrupt Enable (bit 11) of the Link Control register (offset: 0x70).

Table 56.141 Related registers

Offset	Bit	Description
PCI Express Capability Structure: Link Control / Status		
0x06C	21	Link Bandwidth Notification Capability
0x070	10	Link Bandwidth Management Interrupt Enable
	11	Link Autonomous Bandwidth Interrupt Enable
	30	Link Bandwidth Management Status
	31	Link Autonomous Bandwidth Status

(2) PM-PME Reception Interrupt (INT_PM_PME)

This interrupt source is conveyed when the notification of a PME event (PM_PME message) is received from the other-party device.

Note that this interrupt source is enabled when the root complex in use and it is fixed to "0: Low" when the endpoint is in use.

This interrupt source is only conveyed when the PME Interrupt Enable bit (bit 3) of the Root Control/Capabilities Register (offset: 0x7C) is set to 1. This interrupt source is not conveyed when the PME Interrupt Enable bit (bit 3) is set to 0.

Note that the reception state of the received PM_PME message and the ID information, etc. are stored in the Root Status Register (offset: 0x80) regardless of the setting of the PME Interrupt Enable bit mentioned above.

Table 56.142 Related Registers

Offset	Bit	Description
PCI Express Capability Structure: Root Control Capabilities		
0x07C	3	PME Interrupt Enable
PCI Express Capability Structure: Root Status		
0x080	15:0	PME Requester ID
	16	PME Status
	17	PME Pending

(3) System Error Interrupt (INT_SERR_xxx)

This interrupt is conveyed when a correctable error message, non-fatal error message, or fatal error message is received.

INT_SERR:	The system error notification mentioned in the Base Spec. is obtained as the logical OR of the following three signals.
INT_SERR_COR:	System Error on Correctable Error
INT_SERR_FATAL:	System Error on Fatal Error
INT_SERR_NONFATAL:	System Error on Non-Fatal Error

These interrupt sources are enabled when the root complex is in use and these are fixed to "0: Low" when the endpoint is in use.

(a) Correctable Error Interrupt (INT_SERR_COR)

To proceed with interrupt control due to a correctable error, set the interrupt enable bits of the following registers to 1.

No.	Offset	Bit	Description
1	PCI Compatible Configuration: Bridge Control and Interrupt		
	0x03C	17	SERR# Enable
2	PCI Express Capability Structure: Device Control		
	0x068	0	Correctable Error Reporting Enable
3	Advanced Error Reporting (AER) Capability: Root Error Command		
	0x12C	0	Correctable Error Reporting Enable

<Interrupt Source>

(1) ERR_COR Message Reception

If an ERR_COR message is received from the other-party device (endpoint), the INT_SERR_COR interrupt will be asserted. The following status register will be set at the same time.

Offset	Bit	Description
Advanced Error Reporting (AER) Capability: Root Error Status		
0x130	0	ERR_COR Received

Note: Note: If the Enable bits in no. 1, 2, and 3 are not set, the interrupt source will not be generated.
If the Enable bits in no. 1 and 2 are not set, writing will be masked.

(2) Correctable Error Detection

Any of the following correctable errors being detected within the unit will be written to the corresponding status register.

Error	Offset	Bit	Description
Advanced Error Reporting (AER) Capability: Correctable Error Status			
8B10B Decode Error 8B10B RD Error	0x110	0	Receiver Error Status
Bad TLP		6	Bad TLP Status
Bad DLLP		7	Bad DLLP Status
REPLAY_NUM Roll over		8	REPLAY_NUM Rollover Status
Replay Timer Timeout		12	Replay Timer Timeout Status
Advisory Non-Fatal Error		13	Advisory Non-Fatal Error Status

Note: If the Mask bit (Correctable Error Mask Register: Offset 0x114) corresponding to each bit is set (masked), the corresponding error will not be detected and the interrupt source will not be generated.

At the same time, the INT_SERR_COR interrupt will be generated and the following status registers will be set to 1.

Offset	Bit	Description
PCI Express Capability Structure: Device Status		
0x068	16	Correctable Error Detect
Advanced Error Reporting (AER) Capability : Root Error Status		
0x130	0	ERR_COR Received

Note: If the Enable bit in no. 2 is not set, the interrupt source will not be generated.
If the Enable bit in no. 2 is not set, writing to the Root Error Status Register will be masked.

If the Correctable Error Reporting Enable bit listed in no. 3 is not set, the source due to (1) and (2) stated above will be generated and INT_SERR_COR will not be asserted even if writing to the Root Error Register has proceeded.

(b) Fatal Error Interrupt (INT_SERR_FATAL)

To proceed with interrupt control due to a fatal error, set the interrupt enable bits of the following registers to 1.

No.	Offset	Bit	Description
1	PCI Compatible Configuration: Bridge Control and Interrupt		
	0x03C	17	SERR# Enable
2	PCI Express Capability Structure: Device Control		
	0x068	2	Fatal Error Reporting Enable
3	Common Configuration Space: Command and Status		
	0x004	8	SERR# Enable
4	Advanced Error Reporting (AER) Capability: Root Error Command		
	0x12C	2	Fatal Error Reporting Enable

<Interrupt Source>

(1) ERR_FATAL Message Reception

If an ERR_FATAL message is received from the other-party device (endpoint), the INT_SERR_FATAL interrupt will be asserted (if the Enable bits in no.1 and 2 or no.3 and 4 are not set, the interrupt source will not be generated). The following status registers will be set at the same time.

Offset	Bit	Description	Remarks
Type1 Configuration Space: Secondary Status			*1
0x01C	30	Received System Error	
Common Configuration Space: Status Register			*2
0x004	30	Signaled System Error	

Offset	Bit	Description	Remarks
Advanced Error Reporting (AER) Capability: Root Error Status			*3
0x130	6	Fatal Error Message Received	

Note 1. Writing to this status register proceeds at the point a message is received regardless of the settings of the Enable bits.
Note 2. If the Enable bits in no. 1 and 2 are not set, writing will be masked.
Note 3. If the Enable bit in no. 1, 2, or 3 is not set, writing will be masked.

(2) Fatal Error Detection

Any of the following fatal errors being detected within the unit will be written to the corresponding status register.

Error	Offset	Bit	Description
Advanced Error Reporting (AER) Capability: Uncorrectable Error Status			
Data Link Protocol Error	0x104	4	Data Link Protocol Error Status
Poisoned TLP		12	Poisoned TLP Status
Completion Timeout		14	Completion Timeout Status
Completer Abort		15	Completer Abort Status
Unexpected Completion		16	Unexpected Completion Status
Receiver Overflow		17	Receiver Overflow Status
Malformed TLP		18	Malformed TLP Status
ECRC Error		19	ECRC Error Status
Unsupported Request		20	Unsupported Request Error Status

Note: If the Mask bit (Uncorrectable Error Mask Register: Offset 0x108) corresponding to each bit is set (masked), the corresponding error will not be detected and the interrupt source will not be generated.
Also, the setting of the corresponding Severity bit (Uncorrectable Error Severity: Offset 0x10C) must be 1 (Fatal).

The INT_SERR_FATAL interrupt will be generated at the same time (if the Enable bits in no. 2 or no.3 and 4 are not set, the interrupt source will not be generated). The following status registers will be set at the same time.

Offset	Bit	Description	Remarks
PCI Express Capability Structure: Device Status			
0x068	18	Fatal Error Detect	
Common Configuration Space: Status Register			*1
0x004	30	Signaled System Error	
Advanced Error Reporting (AER) Capability: Root Error Status			*2
0x130	6	Fatal Error Message Received	

Note 1. If the Enable bit in no. 3 is not set, writing will be masked.
Note 2. If the Enable bit in no. 2 or 3 is not set, writing will be masked.

If the Fatal Error Reporting Enable bit listed in no. 4 is not set, the source due to (1) and (2) stated above will be generated and INT_SERR_FATAL will not be generated even if writing to the Root Error Status Register has proceeded.

(c) Non-Fatal Error Interrupt (INT_SERR_NONFATAL)

To proceed with interrupt control due to a non-fatal error, set the interrupt enable bits of the following registers to 1.

No.	Offset	Bit	Description
1	PCI Compatible Configuration: Bridge Control and Interrupt		
	0x03C	17	SERR# Enable
2	PCI Express Capability Structure: Device Control		
	0x068	1	Non-Fatal Error Reporting Enable
3	Common Configuration Space: Command and Status		
	0x004	8	SERR# Enable

No.	Offset	Bit	Description
4	Advanced Error Reporting (AER) Capability: Root Error Command		
	0x12C	1	Non-Fatal Error Reporting Enable

<Interrupt Source>

(1) ERR_NONFATAL Message Reception

If an ERR_NONFATAL message is received from the other-party device (endpoint), the INT_SERR_NONFATAL interrupt will be generated (if the Enable bits in no.1 and 2 or no.3 and 4 are not set, the interrupt source will not be generated). The following status registers will be set at the same time.

Offset	Bit	Description	Remarks
Type1 Configuration Space: Secondary Status			*1
0x01C	30	Received System Error	
Common Configuration Space: Status Register			*2
0x004	30	Signaled System Error	
Advanced Error Reporting (AER) Capability: Root Error Status			*3
0x130	5	Non-Fatal Error Message Received	

Note 1. Writing to this status register proceeds at the point a message is received regardless of the settings of the Enable bits.
Note 2. If the Enable bits in no. 1 and 3 are not set, writing will be masked.
Note 3. If the Enable bit in no. 1, 2, or 3 is not set, writing will be masked.

(2) Non-Fatal Error Detection

Any of the following non-fatal errors being detected within the unit will be written to the corresponding status register.

Error	Offset	Bit	Description
Advanced Error Reporting (AER) Capability: Uncorrectable Error Status			
Data Link Protocol Error	0x104	4	Data Link Protocol Error Status
Poisoned TLP		12	Poisoned TLP Status
Completion Timeout		14	Completion Timeout Status
Completer Abort		15	Completer Abort Status
Unexpected Completion		16	Unexpected Completion Status
Receiver Overflow		17	Receiver Overflow Status
Malformed TLP		18	Malformed TLP Status
ECRC Error		19	ECRC Error Status
Unsupported Request		20	Unsupported Request Error Status

Note: If the Mask bit (Uncorrectable Error Mask Register: Offset 0x108) corresponding to each bit is set (masked), the corresponding error will not be detected and the interrupt source will not be generated.
Also, the setting of the corresponding Severity bit (Uncorrectable Error Severity: Offset 0x10C) must be 0 (Non-fatal).

The INT_SERR_NONFATAL interrupt will be generated at the same time (if the Enable bits in no. 2 or no.3 and 4 are not set, the interrupt source will not be generated). The following status registers will be set at the same time.

Offset	Bit	Description	Remarks
PCI Express Capability Structure: Device Status			*1
0x068	17	Non-Fatal Error Detect	
Common Configuration Space: Status Register			*1
0x004	30	Signaled System Error	
Advanced Error Reporting (AER) Capability: Root Error Status			*2
0x130	5	Non-Fatal Error Message Received	

Note 1. If the Enable bit in no. 3 is not set, writing will be masked.
Note 2. If the Enable bit in no. 2 or 3 is not set, writing will be masked.

If the Non-Fatal Error Reporting Enable bit listed in no. 4 is not set, the source due to (1) and (2) stated above will be generated and INT_SERR_NONFATAL will not be generated even if writing to the Root Error Status Register has proceeded.

(4) Equalization interrupt (INT_LINK_EQUALIZATION_REQUEST)

This interrupt source is conveyed when PCIe module starts equalization.

Table 56.143 Related registers

Offset	Bit	Description	Remarks
PCI Express Capability Structure : Link Control 2 / Status 2			—
0x090	21	Link Equalization Request	
Secondary PCI Express Extended Capability Structure : Link Control 3 / Status 3			*1
0x1B4	2	Link Equalization Request Interrupt Enable	

Note 1. If the Enable bit is not set, interrupt notification will be masked.

56.7.7.5 DMA Interrupt

The DMAC has interrupt functions. It supports two types of interrupt source: one set is for normal operation and the other is for errors. Use the DMAC Interrupt Enable Register to select the interrupt sources and control masking. DMA interrupt notification includes notification by DMA_INT in the direction of the AXI bus (local bus direction) and notification by issuing an MSI to an other-party RC.

On completion of DMA transfer, either conveying a DMA_INT interrupt or transmitting an MSI to the RC, but not both, is selected per-channel by a register setting (transmitting an MSI is prohibited in RC mode).

The DMA_CH*_MSI_EN bits of the DMA Interrupt Vector 0 Register and the DMA Interrupt Vector 1 Register are used to switch the setting ("*" corresponds to the channel number of a given DMA and per-channel control is possible). If an MSI is to be issued, the value of its traffic class (TC) bits is fixed to 0. The vector value can be specified by using the DMA_CH*_vec bits of these registers.

Note: Take care that the specified vector value does not impose a related restriction on the Multiple Message Enable register.

For example, if the number set for Multiple Message Enable is 4, the two lower-order bits within the value set for the Message Data are overwritten by the value specified for the DMA_CH*_vec bits, and at that time, specify the value 0 for the higher-order bits that are not overwritten by DMA_CH*_vec bits.

56.7.8 Power Management (Root Complex Mode)

56.7.8.1 PCI Power Management (PCI-PM)

An endpoint is placed in non-D0 (the D3 hot state) by changing the setting of the Power State field register of the endpoint by the root complex., which causes a transition of the link state of both RC and EP to L1.

This module does not support the AUX power supply function. D3 cold is not supported. Implementing the L2 function goes to implementation of L2/L3 ready.

(1) Flow of Transition to the L2/L3 Ready State (RC)

< For transition via PCIPM L1 >

1. A root complex (RC) issues a config write to change the D-state of an endpoint (EP) to D3 (D3 hot).
2. The RC is automatically placed in L1 by automatic response of the other-party EC.
3. The RC transmits a PME_Turn_Off message.
4. Confirm the reception of a PME_TO_Ack Message from the EP by reading the value of the PME_TO_Ack Receive Interrupt bit (bit 16) in the Message Receive Interrupt Status Register of the RC.
5. Set UI_ENTER_L2 of the PCIe Core Control 1 Registers to 1.
6. Confirm the transition to the L2/L3 ready state by reading the value of the LTSSM_STATE bits (bits [14:8]) in the PCIe Core Status 1 Register (5 higher-order bits [14:10] = 0111xb: L2).

< For direct transition to the L2/L3 ready state >

Steps 3 to 6 above apply.

(2) Flow of Return from the L2/L3 Ready State (RC)

< Return from the L2/L3 ready state by the RC >

1. Assert the following reset signals.
   RST_B, RST_GP_B, RST_PS_B, RST_CFG_B
2. Deassert the reset signals at a desired time and wait for return to L0 (wait for linkup).
3. Clear UI_ENTER_L2 of the PCIe Core Control 1 Registers to 0.
4. If the link is not up, repeat steps 1 and 2.

< Return from the L2/L3 ready state by the EP >

1. Wait for the reception of a beacon by the EP. Read the LINKDN bit (0x0204, bit 9) in the PCIe Event Interrupt Status register regularly to check whether ELECTRICAL_IDLE_BROKEN has been generated.
2. Assert the following reset signals.
   RST_B, RST_GP_B, RST_PS_B, RST_CFG_B
3. Deassert the reset signals at a desired time and wait for it to return to L0 (wait for linkup).
4. Clear UI_ENTER_L2 of the PCIe Core Control 1 Registers to 0.
5. If the link is not up, repeat steps 2 and 3.

56.7.8.2 Active State Power Management (ASPM)

The ASPM L0s and L1 states can be used by the setting of the Active State PM Control bits (bits [1:0]) in the Link Control/Status Register (PCIe Configuration Register: 0x6070).

ASPM L0s:	The module is automatically placed in the ASPM L0s state following setting of the Active State PM Control bits to 01b or 11b. When to initiate a transition to this state is determined by judgment of the idle state by the PCIe core.
ASPM L1:	An RC is placed in the ASPM L1 state in response a request for transition to the ASPM L1 state from an EP following setting of the Active State PM Control bit to 11b. When to initiate a transition to this state is determined by judgment of the idle state by the PCIe core. A transition to the ASPM L1 state is enabled by setting the UI_RC_REJECT_ASPM_L1 bit (bit [19]) of the PCIe Core Control 1 Register (offset: 0x404) to 0. To reject a transition to the ASPM L1 state on the RC side, set the UI_RC_REJECT_ASPM_L1 bit to 1.

Return to the L0 state from the L1 state:

Resuming access from the AXI master side:

Return to the L0 state is initiated in response to the other-party PCIe device resuming access.

Resuming access from the AXI slave side:

The AXI can be returned to the L0 state in response to the AXI starting access to a PCIe device. Access to an AXI configuration register also returns the AXI to the L0 state.

56.7.9 Power Management (Endpoint Mode)

56.7.9.1 PCI Power Management (PCI-PM)

The endpoint unit can be placed in the D3 hot state by changing the setting of the Power State field register by the root complex. A PME_Turn_Off message can be received even after the transition to the D3 state. The turning-off procedure described below starts in such cases.

D0:	This is a full-power state in which no restrictions apply. The link state is basically L0. However, if the ASPM generates a source for a transition, the latter takes priority.

D3hot: The link state basically changes to L1. By changing the value of the Power State bits of the PM Status/Control register in response to the reception of CfgWr, the unit sets D3_EVENT_Fi interrupt to 1 when a transition to the non-D0 state is requested per function, and issues Deassert_INTx in response to the assertion of any of the INTX interrupts. The EP should set D3_EVENT_ACK_Fi bit in PCIE_ACK register to 1 after preparation for low power consumption. After D3_EVENT_ACK_Fi bit in PCIE_ACK register is set to 1, issuing of the AXI transaction to the PCIe is prohibited.(this does not apply to access to internal registers which include the configuration register).

This module does not support the AUX power supply function. D3 cold is not supported. Implementing the L2 function goes to implementation of L2/L3 ready.

(1) Flow of Transition to the L2/L3 Ready State (EP)

< For transition via PCIPM L1 >

1. The D-state of an endpoint (EP) is changed to D3 (D3 hot) in response to the reception of a config write from a root complex (RC).
2. Check that D3_EVENT_Fi interrupt has been set to 1, and clear D3_EVENT_ACK_Fi bit in PCIE_ACK register to 0 after setting it to 1.
3. A transition to the L1 state is automatically initiated.
4. The EC receives a PME_Turn_Off message from the RC.
5. Check that TURN_OFF_EVENT interrupt has been set to 1, and clear TURN_OFF_EVENT_ACK bit in PCIE_ACK register to 0 after setting it to 1.
6. A transition to the L2/L3 ready state is automatically initiated.

< For direct transition to the L2/L3 ready state >

Steps 4 to 6 above apply.

(2) Flow of Return from the L2/L3 Ready State (EP)

< Return from to the L2/L3 ready state by the RC >

1. Wait for an Electrical Idle exit from the RC.
2. After that, follow the instruction by the RC to return to L0 (the following is an example).
   - Example 2-1. Follow the instruction by the RC to reset the EP (assert RST_B, RST_GP_B, RST_PS_B, and RST_CFG_B).
   - Example 2-2. Follow the instruction by the RC to release the EP from the reset state at a desired time.
   - Example 2-3. Return to L0.

< Return from to the L2/L3 ready state by the EP >

1. Check that the value of the LTSSM_STATE bits (bits [14:8]) in the PCIe Core Status 1 Register is 0x3A.

   Note: As a criterion for judgment at this time, checking the value read from this register requires consecutively checking the above value several times. Since the value of the LTSSM bits is checked by direct reference to the internal circuit state information, an undefined state value may be read depending on the timing of the operation for transition. To make sure that the current state is the L2 idle state, we recommend checking the state by access more than once. Only check all seven bits of LTSSM_STATE (bits [14:8]) of the PCIe Core Status 1 Register at this time in the flow of return.
2. Assert CFG_PMCSR_PME_STATUS_Fi bit in PCIE_PME register to 1.
3. Assert RST_B (the EP transmits a beacon).
4. After that, follow the instruction by the RC to return to L0 (the following is an example).
   - Example 4-1. The RC detects the reception of a beacon from the EP and initiates return to L0 on the RC side.
   - Example 4-2. Follow the instruction by the RC to reset the EP (assert RST_B, RST_GP_B, and RST_PS_B, RST_CFG_B).
   - Example 4-3. Follow the instruction by the RC to release the EP from the reset state at a desired time.
   - Example 4-4. Return to L0.

56.7.9.2 Active State Power Management (ASPM)

The ASPM L0 and L1 states can be used by setting the Active State PM Control bits (bits [1:0]) of the Link Control/Status Register (PCIe Configuration Register: 0x6070).

Permit a transition to ASPM L1 by setting ALLOW_ENTER_L1 bit in PCIE_MISC register to 1.

The ASPM L1 Idle Time bits of the registers must be set in advance to suit the AXI bus specifications (the range of settings is for from 256 to 65536 cycles of ACLK). If transfer does not proceed for the time set in the ASPM L1 Idle Time bits within the unit, the module is automatically placed in the ASPM L1 state.

ASPM L0s: A transition to the ASPM L0s state is automatically initiated by setting the Active State PM Control bits to 01b or 11b. When to initiate a transition to this state is determined by judgment of the idle state by the PCIe core, so your operation will not be especially required.
A transition to L0s can be controlled by setting the UI_ENTER_TXL0S bit (bit [16]) of the PCIe Core Control 1 Register.

ASPM L1: A transition to the ASPM L1 state can be initiated by setting the Active State PM Control bits to 10b or 11b. When to initiate a transition to this state is determined by judgment of the idle state by the PCIe core. A transition to the L1 state is enabled by setting ALLOW_ENTER_L1 bit in PCIE_MISC register to 1.
If the Root Complex denies the L1 transition, this macro receives PM_Active_State_Nak and a MSG receive interrupt occurs. After that, if the ALLOWE_ENTER_L1 bit in the PCIE_MISC register is 1, it will issue L1 transition request again. If you want to cancel L1 transition request and perform CFG access, etc when the transition is denied, clear the ALLOWE_ENTER_L1 bit in the PCIE_MISC register and confirm UI_ENTER_L1_STATUS (bit [21]) of the PCIe Core Control 1 Registers is 0. If it is not 0, write 1 to the bit to clear it.

Return from the L1 state to the L0 state

Resuming access from the AXI master side:

Return to the L0 state is initiated in response to the other-party PCIe device resuming access.

Resuming access from the AXI slave side:

The AXI can be returned to the L0 state in response to the AXI starting access to a PCIe device. Access to an AXI configuration register also returns the AXI to the L0 state.

(1) ASPM L1 Idle Time Setting

The Mode Set 3 registers (offset: 0x380) are defined as the ASPM L1 Idle Time registers. These registers are used to set the counter for monitoring the state of the AXI bus or internal transactions in terms of remaining activity when this module is placed in the ASPM L1 state. The counter monitors whether the AXI bus and transactions within this module are idle based on one of the following settings and uses this as the condition for triggering the transition to the ASPM L1 state.

```
ASPM L1 Idle Time[7:0] = 0x00: 256 cycles of ACLK
ASPM L1 Idle Time[7:0] = 0x01: 512 cycles of ACLK
        ⋮
ASPM L1 Idle Time[7:0] = 0xFF: 65536 cycles of ACLK
```

56.7.10 Power Management (Common)

56.7.10.1 Turn Off (PME Turn Off Message Recept)

Upon the reception of a PME_Turn_Off message, the TURN_OFF_EVENT interrupt is asserted and Deassert_INTx is issued for all of the INTx that are asserted. Moreover, preparation for sending PME_TO_Ack message is automatically started.

After receiving TURN_OFF_EVENT interrupt and preparing power down, set TURN_OFF_EVENT_ACK bit in PCIE_ACK register. This allows PME_TO_Ack Message transmission.

After that, issue of AXI transaction to PCIe is prohibited. Module is ready for assertion of ARESETn or DL_Down state.

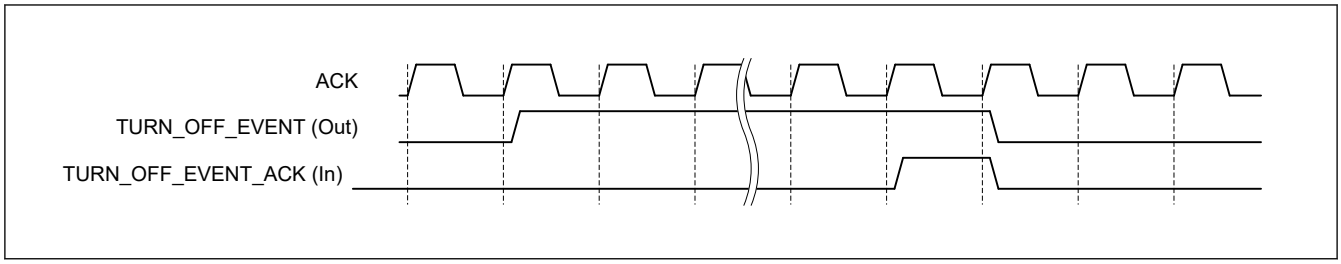


Figure 56.14 Relationship between TURN_OFF_EVENT and TURN_OFF_EVENT_ACK

56.7.10.2 Wake Up (PME Message transmit)

When CFG_PMCSR_PME_STATUS_Fi bit in PCIE_PME register is 1, preparation for automatic transmission of PME message is made. The transmission timing is set by the edge detection after the toggle timing of the PME_TIM bit in PCIE_PME register is synchronized by ACLK.

Note that when a PME Turn Off message is received, automatic transmission of PME message will be stopped until ARESETn is generated or the DL_Up state is reached once after the DL_Down state is reached.

<Control procedure>

1. Among the configuration registers, when PME Enable = 1 in the PM Status/Control register, a PM_PME message can be sent from the EP to the RC by setting CFG_PMCSR_PME_STATUS_Fi bit in PCIE_PME register to 1 and toggling PME_TIM bit in PCIE_PME register.
2. The RC issues CfgWr to write 1 to and clear the PME_STATUS field of the device that has received the PM_PME message. At the same time, it sends CfgWr0 to change the Power State bit to D0.
3. When the EP enters the D0 state, the CFG_PMCSR_PME_STATUS_WRITECLEAR interrupt of the EP is asserted.
4. By toggling PME_TIM bit in PCIE_PME register every 100 to 150 ms, the PM_PME Msg retransmission can be instructed in accordance with the following base spec. regulations.

Note that the PME_TIM, CFG_PMCSR_PME_STATUS bits in PCIE_PME register and CFG_PMCSR_PME_STATUS_WRITECLEAR interrupt are not used when there is no need to request the endpoint side to change the power state to D0.

56.7.11 Error Processing

Two error reporting paradigms are defined for PCI Express: baseline error reporting capability, which is the minimal requirement, and advanced error reporting (AER) capability, which can provide greater stability, as an optional error reporting facility. This module supports both error reporting functions.

56.7.11.1 Error Classification

PCI Express errors are of two types, correctable and uncorrectable. Uncorrectable errors are further classified into two types, non-fatal and fatal.

Error type	Description
Correctable Error	An error which cannot be recovered by hardware
Uncorrectable Error (Non-Fatal)	An error which causes a particular transaction in PCI Express to be unreliable (but the PCI Express link itself is functional).
Uncorrectable Error (Fatal)	An error which causes the PCI Express link itself to be unreliable

In operation as a root complex, the above types of error are indicated by the individual interrupt names.

- INT_SERR_COR
- INT_SERR_NONFATAL
- INT_SERR_FATAL

56.7.11.2 Error Checking Mechanism

1. Physical Layer Error List
   The following type of error is to be detected in the physical layer.
   - Receiver Error
2. Data Link Layer Error List
   The following type of error is to be detected in the link layer.
   - Bad TLP Error
   - Bad DLLP Error
   - REPLAY_NUM Rollover
   - Replay Timer Timeout
   - Receiver Overflow Error
3. Transaction Layer Error List
   The following type of error is to be detected in the transaction layer.
   - Completion Timeout
   - Completer Abort
   - Unsupported Request
   - Unexpected Completion
   - Malformed TLP
   - Poisoned TLP

Note: The specifications of the receiver error detection function in each LTSSM state are as listed below.

LTSSM state	Unit specification	Base spec specifications
Configuration	Support	Gen1/Gen3 (Must)
Recovery	Non-Support	Option
L0	Support	Must
Disabled	Support	Option
Hot Reset	Support	Option

56.7.11.3 Error Message

The PCI Express Base Specification defines error messages as one of the mechanisms for notifying the system or another device of an error when it is detected by a PCI Express agent.

Error message	Description
ERR_COR	Used when a correctable error is detected
ERR_NONFATAL	Used when a non-fatal, uncorrectable error is detected
ERR_FATAL	Used when a fatal, uncorrectable error is detected

Advisory Non-Fatal Error

The PCI Express Base Specification states that when a PCI Express agent as the detecting agent detects a non-fatal error, it handles the error as an advisory non-fatal error whether it does or does not support AER. In handling a non-fatal error as an advisory non-fatal error, the agent sends an ERR_COR message instead of an ERR_NONFATAL message and sends an advisory notification to the software. At this time, the advisory non-fatal error status bit of the correctable error status register is set to indicate the error state. Note that subsequent setting of the first error pointer register, logging of the header, and message transmission only proceed if the Advisory Non-Fatal Error Mask bit of the Correctable Error Mask Register is clear (no masking). They do not proceed if the bit is set.

The error cases which are handled as advisory non-fatal errors are as follows.

- Reception of unsupported non-posted requests

- Reception of a non-posted request with a completer abort completion response
- Reception of an unexpected completion
- Reception of a poisoned TLP (this is not handled as an advisory non-fatal error in this core)
- Detection of a completion timeout (this is not handled as an advisory non-fatal error in this core)

This module may merge multiple error messages with the same ID. This would occur when multiple errors are detected during the wait for the unit to be ready for the transmission of messages after an error condition is detected. However, messages will not always be merged in such cases.

56.8 Operation

56.8.1 Setting Up

Set the internal registers of the unit including the configuration registers.

This section explains the procedure for setting up until the PCI Express link is up (the module is ready for data transfer).

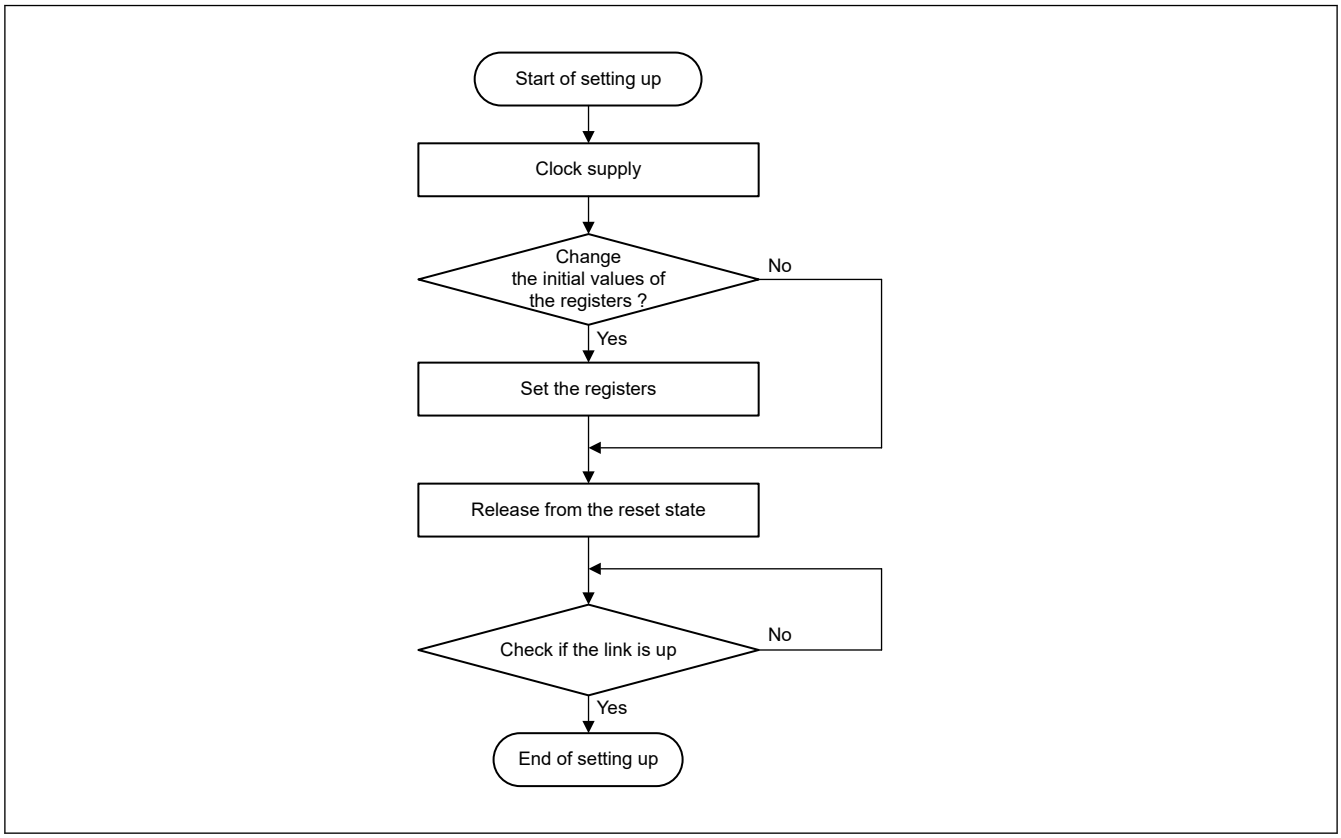


Figure 56.15 Setting up

56.8.1.1 Changing the Initial Values of the Registers

Of the internal registers, the initial values of the configuration registers can be changed via the AXI slave interface.

1. AXI Bridge Registers
   De-asserting ARESETn following supply of CLK allows access to these registers. In the case of resetting of the PCIe core, write to the registers while the reset signal is being asserted.
2. PCIe Configuration Register
   De-asserting ARESETn following supply of CLK allows access to these registers. Of the reset signals of the PCIe core, those related to the configuration register must be de-asserted.
3. Phisical Layer Control/Monitor Registers

De-asserting ARESETn following supply of CLK allows access to these registers. In the case of resetting of the PCIe core, write to the registers while the reset signal is being asserted. Access to the PIPE PHY Register should be permitted by setting bit 1 of the Permission Register (Offset: 0x300) to 1 in advance.

Note that the initial values of the registers listed below are 0. Set appropriate values in the registers before the start of link up.

- Device ID
- Vendor ID
- Class Code (base class/sub-class/programming interface)
- Revision ID
- Subsystem ID
- Subsystem Vendor ID

56.8.1.2 De-asserting the Reset

In this module, software reset control by the register is possible.

De-asserting of the reset of the core (the reset register in the case of register control) automatically starts operations such as receiver detection and the training sequence with the other-party device.

Note that this module requires a wait of at least 5 ms from power-on to de-assertion of the reset.

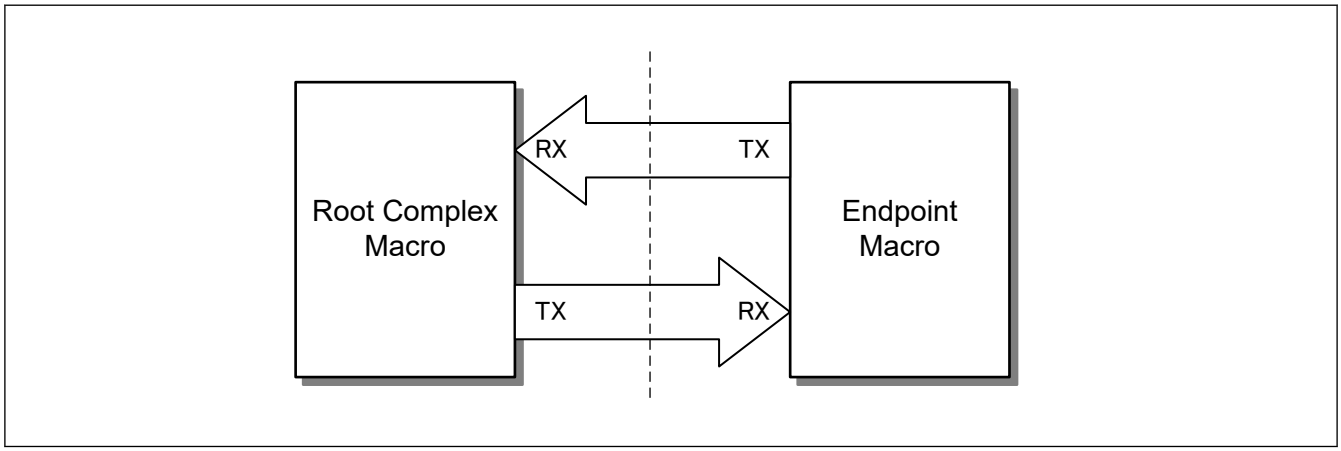


Figure 56.16 Link negotiation

Check that link negotiation with the other party has been completed and the link with the unit is up.

56.8.1.3 Checking If the Link Is Up

Checking if the link is up can be done through either of the following methods. This is usually done by the root complex.

1. Polling
   Have the CPU of the chip poll the DL_Down Status bit (bit [0]) of the PCIe core Status 1 Register in AXI bridge registers until the value of the bit is 0 (indicating the DL_Up state).
2. Interrupt
   After assertion of the PCIE_EVT_INT interrupt, check the interrupt source by reading the DL_UpDown bit (bit [9]) in the PCIe Event Interrupt Status 0 register in AXI Bridge Registers. After that, check that the DL_Down Status bit (bit [0]) of the PCIe Core Status 1 Register is 0 (indicating the DL_Up state)

The settings above make the unit ready for transfer with the other party device.

However, data transfer such as reading or writing memory is not yet possible in this state. This requires subsequent window settings.

56.8.1.4 Link Equalization

This macro automatically executes link equalization in the link up flow when both EP and RC support and enable Gen3 rate, and links up at Gen3 rate.

Gen3 data transfer has TX (RX side is not supported) link equalization function to ensure and improve signal quality. Link equalization adjusts both TX settings between EP and RC. Link equalization has four phases. Recovery Equalization Phase 0-3 exist as LTSSM state, and phase transits according to the progress of Gen3 equalization.

Phase 2 and 3 of link equalization are optional in Base Spec, but this macro has the function, and whether to use it or not can be controlled by the MODE_EQ_PHASE23_ENABLE register. However, due to the constraints of the installed PHY, although it can receive FB from the opposite macro, it cannot perform FB (direction change) from this macro. In addition, if phase 2 and 3 are skipped, the link training speed will be faster, but the bit error rate of the link may not be optimized.

If equalization fails due to a timeout or other reason in Root Complex mode, perform the following to rerun equalization.

1. Set the PE bit of the Link Control 3 Register Perform Equalization to 1
2. Set the TLS[3:0] bits of the Link Control 2 Register to Gen3
3. Set the RL bit of the Link Control Register to 1

56.8.2 Reset

A software reset by the register can be used for this module. Table 56.144 lists the software resets.

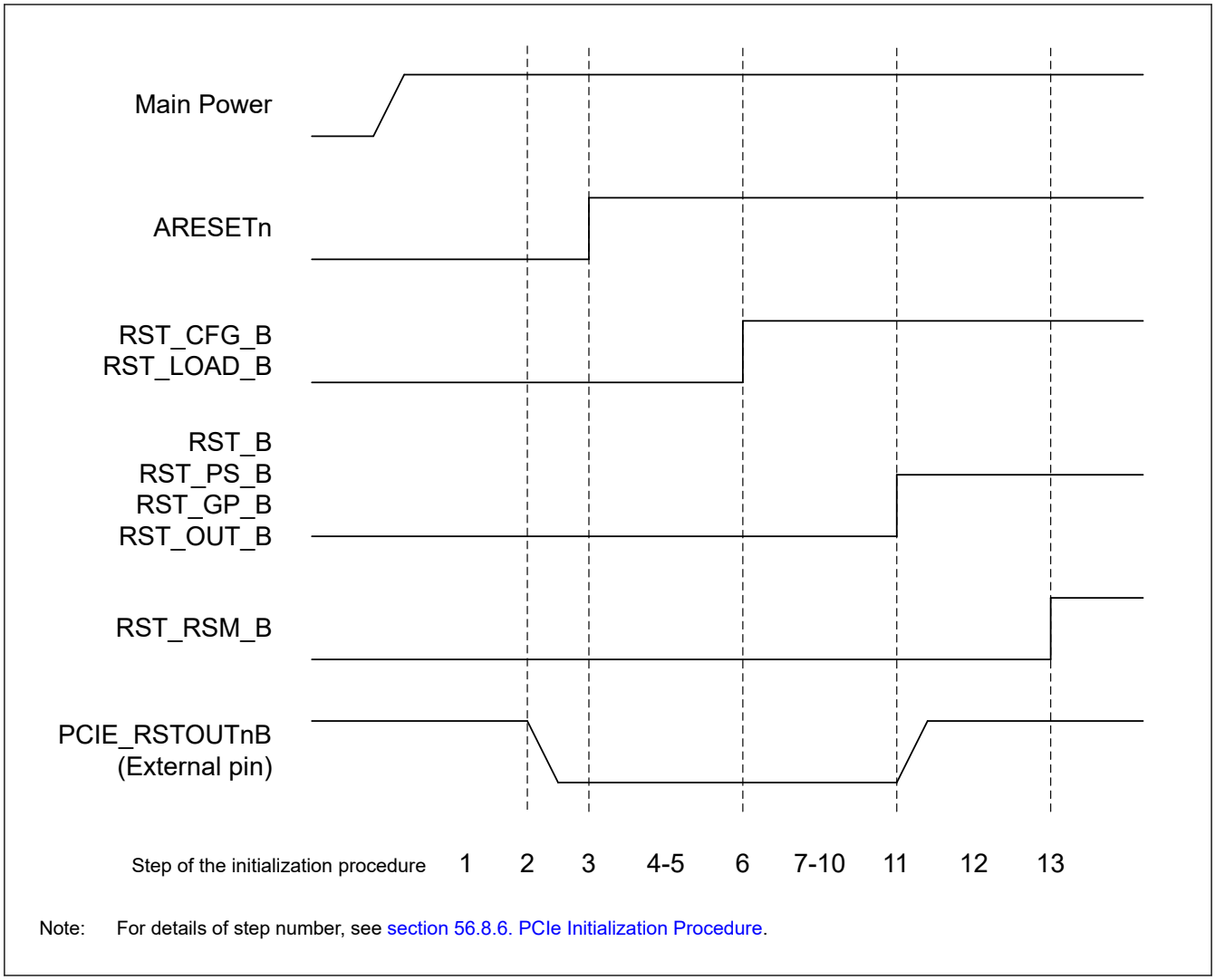
Table 56.144 List of software reset

Reset name	Reset control register	Reset target
ARESETn	Bit 8 in MRCTLM register	AXI2PCIe Bridge
RST_B	Bit 0 in PCI_RC_RESET/PCI_EP_RESET register	Transaction Layer, Data Link Layer, Physical Layer
RST_GP_B	Bit 1 in PCI_RC_RESET/PCI_EP_RESET register	Transaction Layer
RST_RSM_B	Bit 2 in PCI_RC_RESET/PCI_EP_RESET register	Configuration Registers, Transaction Layer, Physical Layer
RST_CFG_B	Bit 3 in PCI_RC_RESET/PCI_EP_RESET register	Configuration Registers
RST_LOAD_B	Bit 4 in PCI_RC_RESET/PCI_EP_RESET register	Configuration Registers
RST_PS_B	Bit 5 in PCI_RC_RESET/PCI_EP_RESET register	Transaction Layer
RST_OUT_B	Bit 6 in PCI_RC_RESET/PCI_EP_RESET register	Other Endpoint device via external PCIE_RSTOUTnB pin when using RC mode. In case of EP mode, it is not available.

Note: Follow the reset sequence defined in section 56.8.2.1. Reset Sequence after power-on. Do not assert the reset during operation.

56.8.2.1 Reset Sequence

Figure 56.17 shows the reset sequence at power-on.



Note: For details of step number, see section 56.8.6. PCIe Initialization Procedure.

Figure 56.17 Reset sequence at power-on

56.8.2.2 Function Level Reset

Control procedure of the Function Level Reset (FLR) is shown below. This control is available in EP mode only.

1. In order to make FLR request from the RC to the EP, CfgWr transaction is issued from the RC to Initiate Function Level Reset (bit 15) in the Device Control/Status register (PCI_EP_DEVCS_Fi) in the EP.
2. When 1 is written in the above bit, FLR_REQ_Fi interrupt is asserted in the EP.
3. After FLR_REQ assertion is detected, the entire EP side system is checked to determine whether FLR can be performed or to prepare a state where FLR can be executed. Preparations related to PCIe require that all ongoing data transfers have been completed and that the issuance of new transactions must be stopped.
4. FLR is performed by writing 1 to FLR_RESETi bit in PCIE_MISC register.

Note: Access to the Configuration Register is prohibited after the FLR request from RC (No.1 above). Before making FLR request, the necessary actions (such as stopping DMA) must be performed in advance. In addition, the completion notification to the RC after FLR execution by the EP (No.4 above) should be performed by using the interrupt function (MSI notification).

The FRS Message feature is not supported.

Configuration Registers and corresponding AXI Bridge Registers bits are cleared by FLR each Function unit.

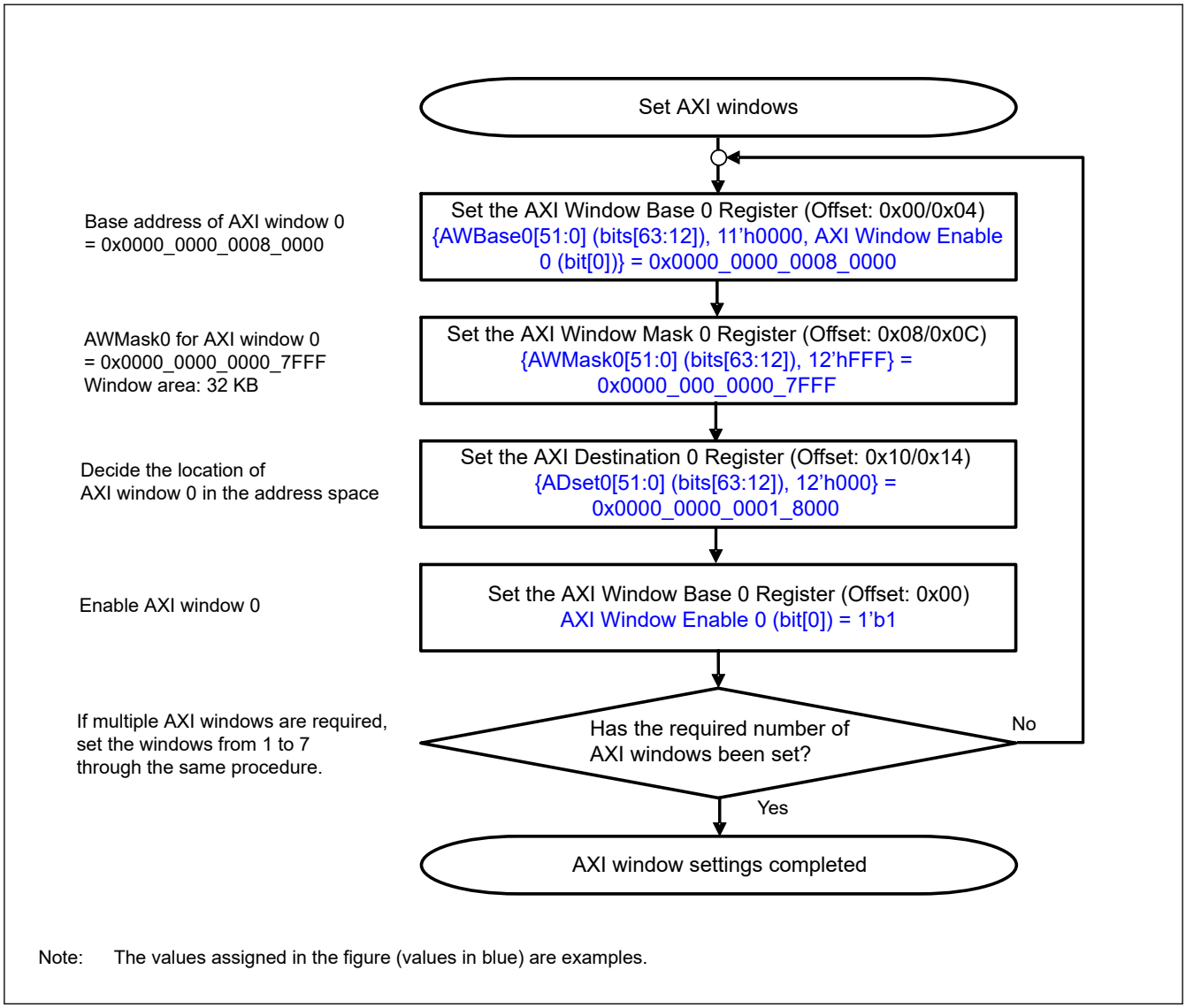
56.8.3 Setting the Windows (Root Complex Mode)

56.8.3.1 Setting the AXI Windows

AXI Window is used when transferring MWR/MRd data from PCIe (RX side) to the AXI bus.

It can be set by write access to the registers below from the AXI Slave I/F. Up to 8 windows can be set. In that case, the following register settings are required for the number of windows required.

- AXI Window Base Lower Register n (n = 0 to 7)
- AXI Window Base Upper Register n (n = 0 to 7)
- AXI Window Mask Lower Register n (n = 0 to 7)
- AXI Window Mask Upper Register n (n = 0 to 7)
- AXI Destination Lower Register n (n = 0 to 7)
- AXI Destination Upper Register n (n = 0 to 7)



Note: The values assigned in the figure (values in blue) are examples.

Figure 56.18 Example settings of the AXI windows

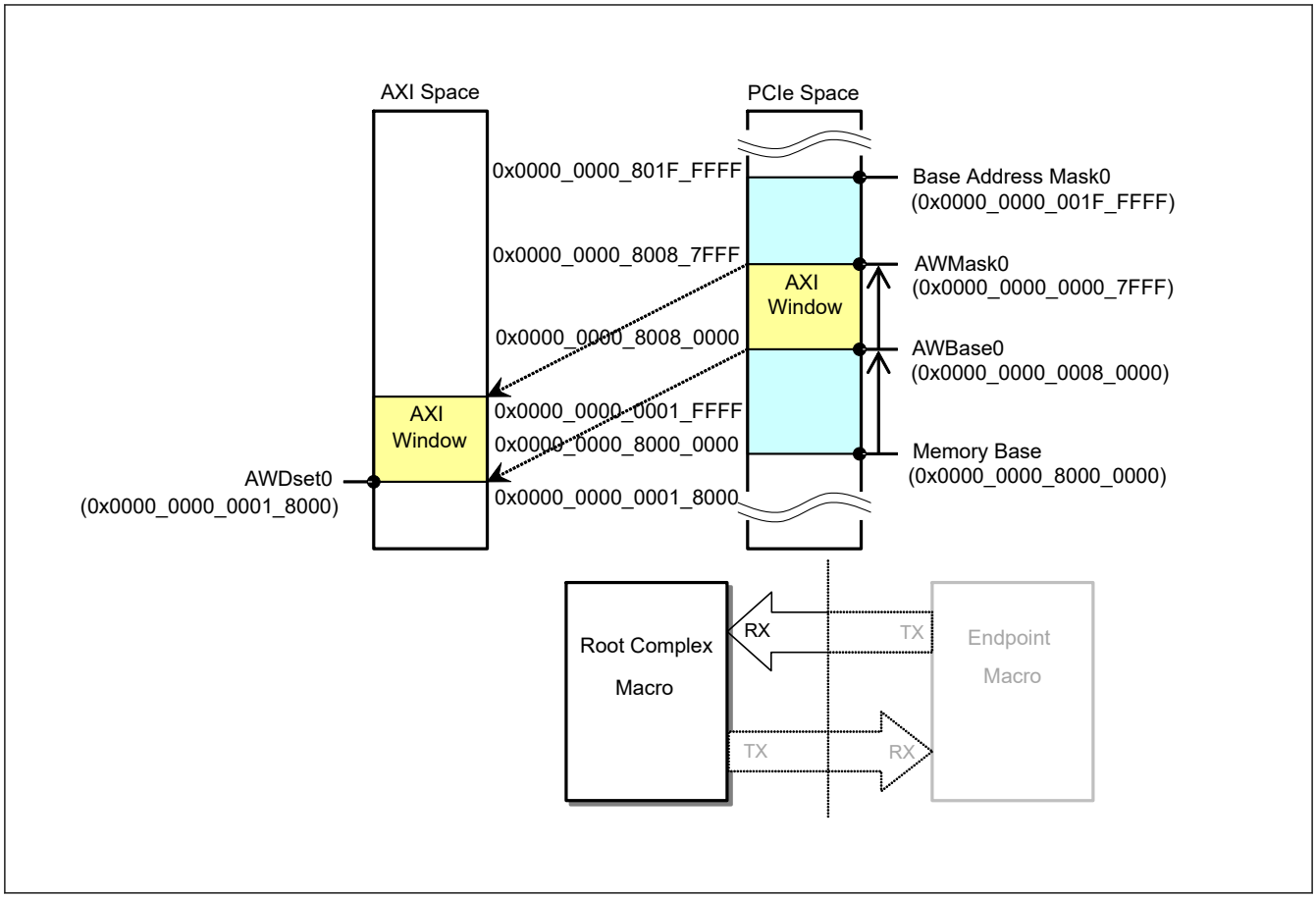


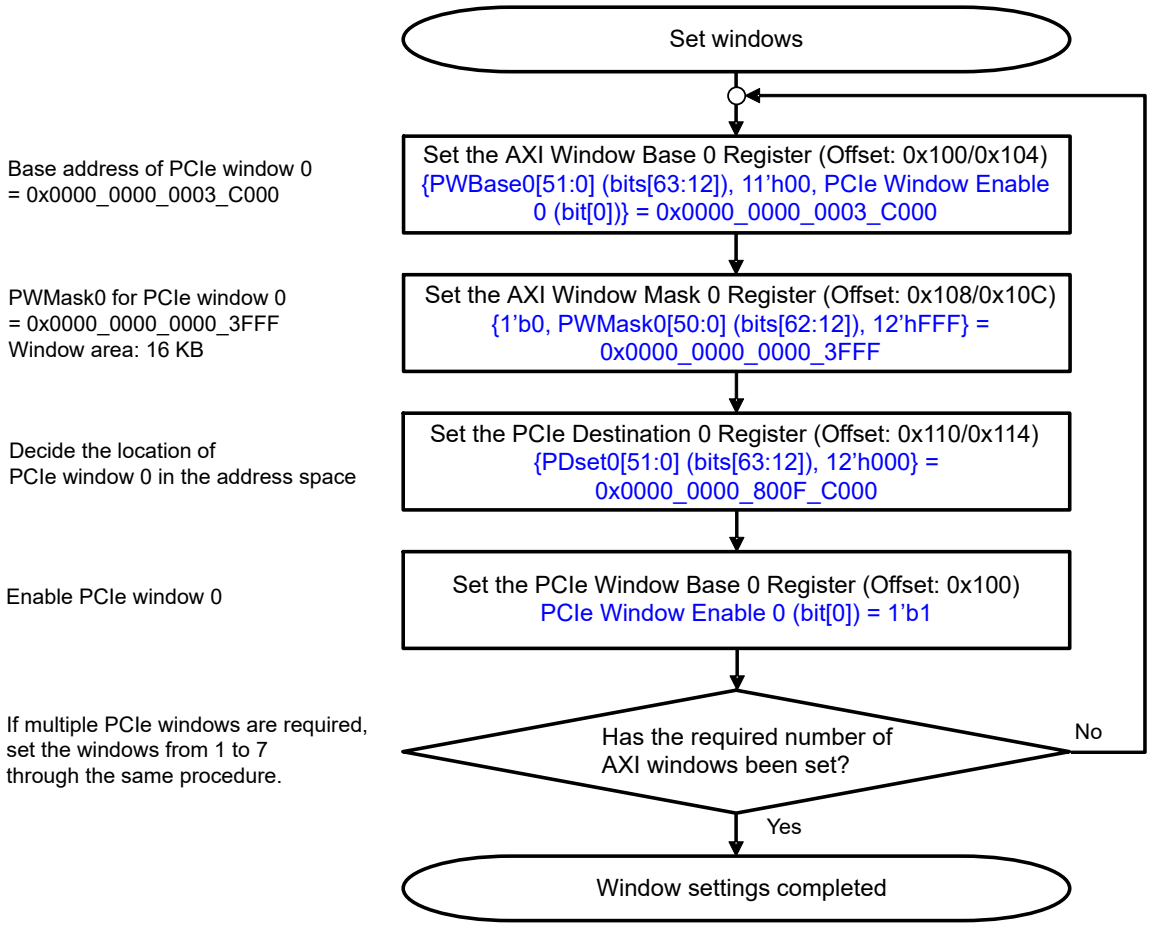
Figure 56.19 Mapping of the AXI windows

56.8.3.2 Setting the PCIe Windows

PCIe Window is used when transferring data from the AXI bus to PCIe (TX side) as MWr/MRd.

It can be set by write access to the registers below from the AXI Slave I/F. Up to 8 windows can be set. In that case, the following register settings are required for the number of windows required.

- PCIe Window Base Lower Register n (n = 0 to 7)
- PCIe Window Base Upper Register n (n = 0 to 7)
- PCIe Window Mask Lower Register n (n = 0 to 7)
- PCIe Window Mask Upper Register n (n = 0 to 7)
- PCIe Destination Lower Register n (n = 0 to 7)
- PCIe Destination Upper Register n (n = 0 to 7)



Note: The values assigned in the figure (values in blue) are examples.

Figure 56.20 Example settings of the PCIe windows

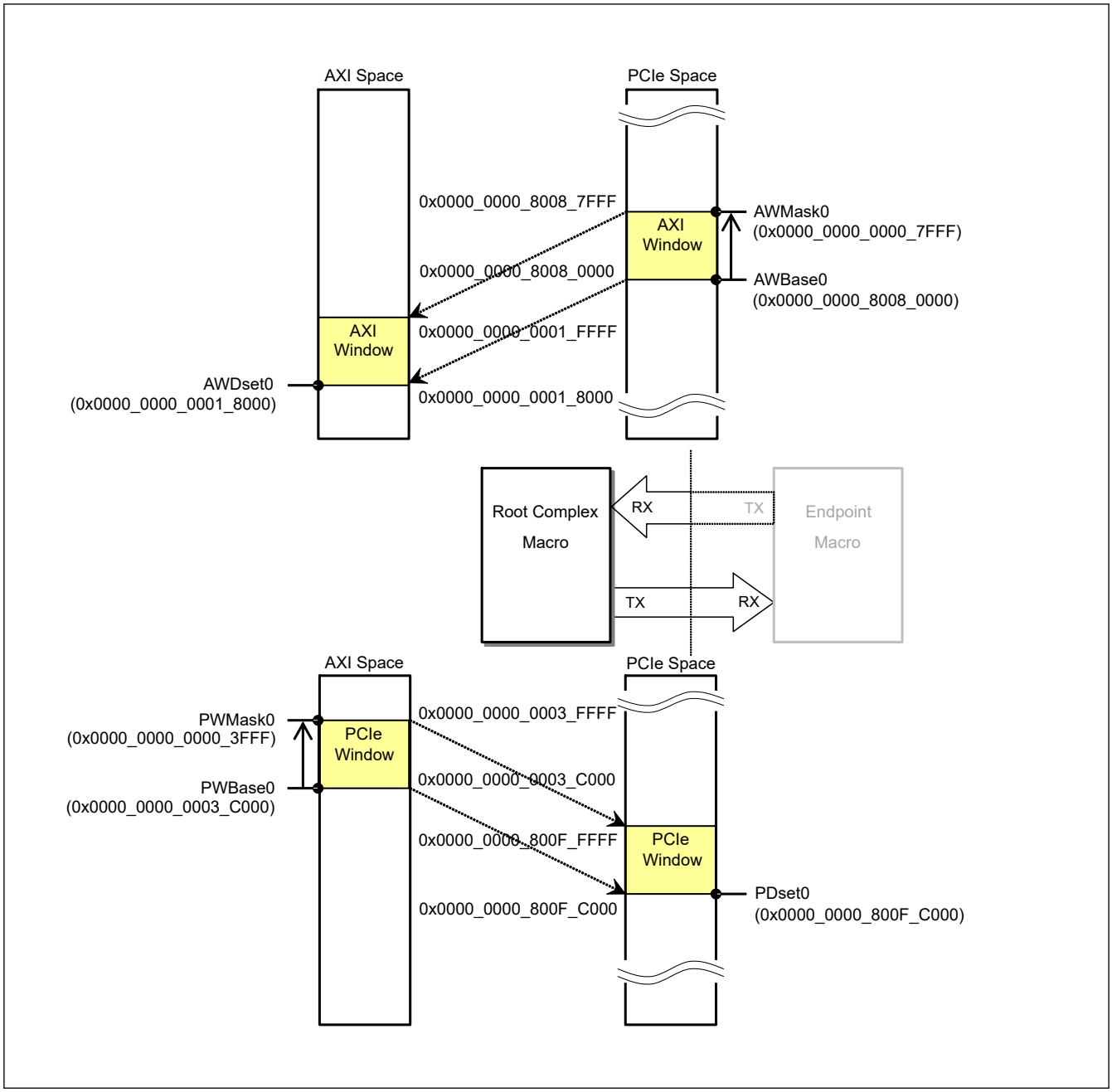


Figure 56.21 Mapping of the PCIe windows

After setting the window, activating Bus Master Enable/Memory Space Enable bits enables memory read/write requests from the Root Complex to the endpoint. If this setting is not made, the endpoint's Configuration Register cannot be set.

Note that this macro performs address conversion within the AXI Bridge, so set Memory Base/Limit (Offset: 0x6020) and Base Address Register Mask00 (Offset: 0x60A0/0x60A4) in the Configuration Register to All 1.

In addition, to allow writing to the Hwinit attribute register, CFG_HWINIT_EN bit (bit 2) in the Permission Register (Offset: 0x300) must be set to 1 before setting the Configuration Register below. It is recommended to rewrite (reset) CFG_HWINIT_EN to 0 after setting the following. This is to prevent the Hwinit attribute register from being easily (by mistake) rewritten.

[Configuration Register Settings]

- Memory Base/Limit Register (Offset: 0x6020): set all 1
- Base Address Register Mask00 (Lower) (Offset: 0x60A0): set all 1
- Base Address Register Mask00 (Upper) (Offset: 0x60A4): set all 1
- Status Command Register (Offset: 0x6004): set Bus Master Enable (bit 2) to 1

- Status Command Register (Offset: 0x6004): set Memory Enable (bit 1) to 1

56.8.3.3 Device Search

In typical PC systems, the Root Complex (Host CPU) can determine what product is connected as the opposing device by reading the Vendor ID/Device ID and Header Type in the opposing device's Configuration Register. For details, see the Base Spec.

In embedded systems and other cases where it is clear that the opposing device is an Endpoint device, there is no need to perform a device search. However, Bus Numbers must be assigned from the Root Complex to the Endpoint.

A simple setup method is described below. Please refer to the information below and take appropriate action depending on your device configuration (number of installed macros, and so on.) and system configuration.

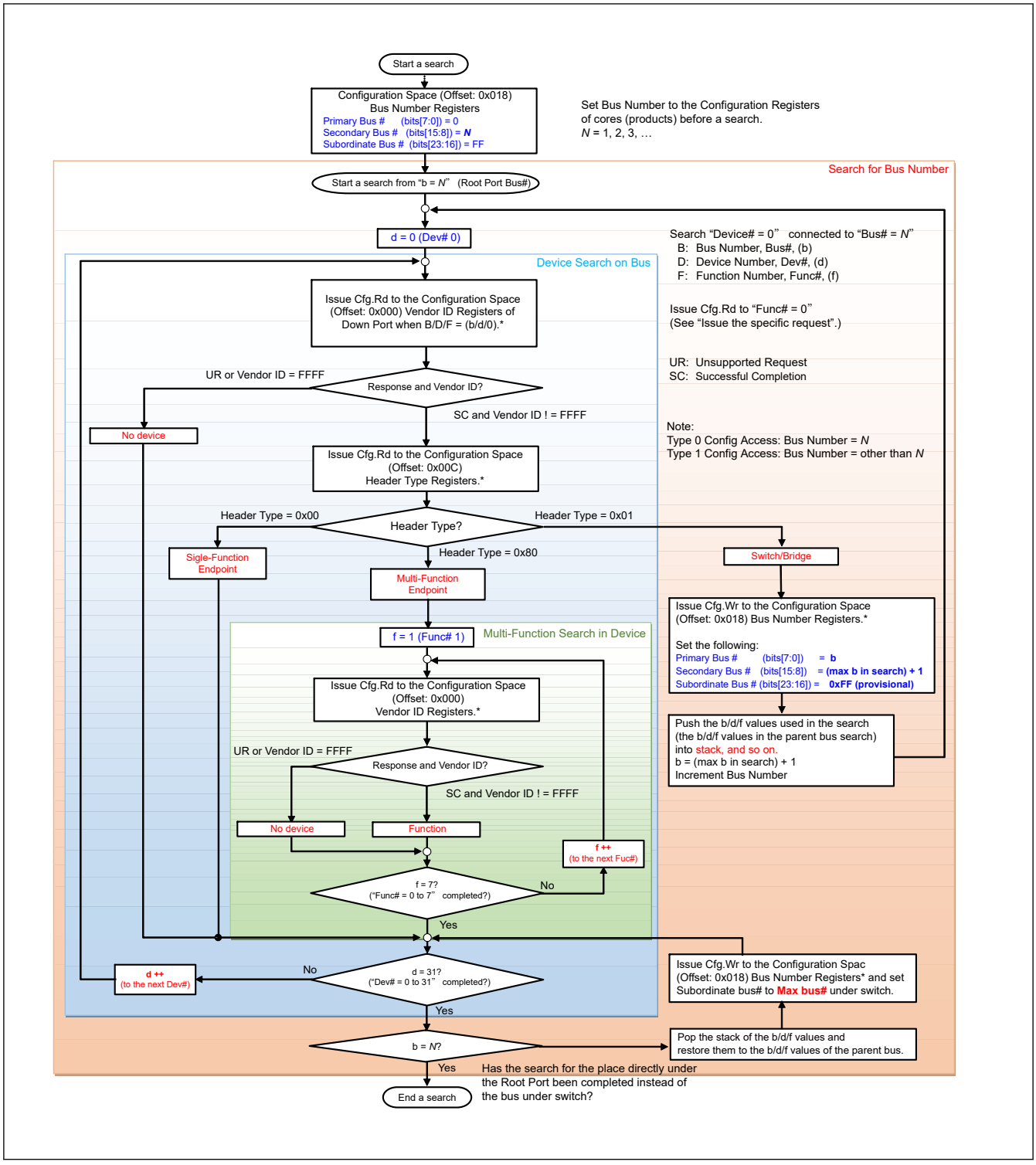


Figure 56.22 Example flow of device search

56.8.4 Setting the Windows (Endpoint Mode)

All PCI Express commands are issued via the set memory spaces or register spaces. After making the window settings and the configuration register settings, data such as PCIe MRd and MWr can be transferred. All PCI Express commands are issued via the set memory spaces or register spaces. After making the window settings and the configuration register settings, data can be transferred in response to PCIe MRd and MWr requests and so on.

56.8.4.1 BAR Specification (Two-Function Configuration)

This module has functionality for dividing the 64-bit memory space into window spaces for AXI access. Windows for access from PCIe to AXI and from AXI to PCIe are respectively referred to as AXI windows and PCI windows.

The AXI Bridge Registers (unit-specific registers) are accessible regardless of the function being used.

The 64-bit memory spaces must be set exclusively of each other and have no overlaps.

Set the AXI window spaces such that they do not overlap.

Table 56.145 Example BAR settings (64 bits × 3, function = 2)

Function	BAR (Memory space)	AXI window	Note
Function #0	64-bit memory space (1) {BAR1, BAR0}	AXI #0	
		AXI #1	
		AXI #2	
		AXI #3	
	64-bit memory space (2) {BAR3, BAR2}	AXI #4	
		AXI #5	
		AXI #6	
		AXI #7	
	64-bit memory space (3) {BAR5, BAR4}	—	AXI Bridge Registers
Function #1	64-bit memory space (1) {BAR1, BAR0}	AXI #0	
		AXI #1	
		AXI #2	
		AXI #3	
	64-bit memory space (2) {BAR3, BAR2}	AXI #4	
		AXI #5	
		AXI #6	
		AXI #7	
	64-bit memory space (3) {BAR5, BAR4}	—	AXI Bridge Registers

Note: BAR: Abbreviation of Base Address Register
Note: The address spaces are invalid when the corresponding BAR values are all 0s.
The BAR settings are required for each function.

The base addresses of the 64-bit memory spaces (1), (2), and (3) above are set by using the following configuration registers.

64-bit memory space (1): Configuration registers {BAR1, BAR0}

64-bit memory space (2): Configuration registers {BAR3, BAR2}

64-bit memory space (3): Configuration registers {BAR5, BAR4}

Note:
- Up to eight AXI windows can be set by dividing the 64-bit memory space (1).
- Up to eight AXI windows can be set by dividing the 64-bit memory space (2).
- The number of AXI windows allocated to each of the BAR areas is as follows.
  64-bit memory space (1): 64-bit memory space (2) = 8:0 or 4:4;
  Select either of the above settings. Note that all of the allocated AXI windows do not need to be used effectively.
- The 64-bit memory space (3) is dedicated for access to the AXI Bridge Registers.
- The base addresses of each of the AXI windows are limited within the 4-Gbyte space from the BAR base address.

56.8.4.2 Setting the AXI Windows

In operation as an endpoint device, the AXI windows can be set in the following two ways.

1. The root complex issuing memory read and memory write requests.
2. Setting from the CPU of the endpoint under software control.

The following shows the procedure for making the settings by method 1.

Set the AXI windows on the endpoint side by issuing memory read (MRd) and memory write (MWr) requests to the AXI Bridge Register access space set in the previous section.

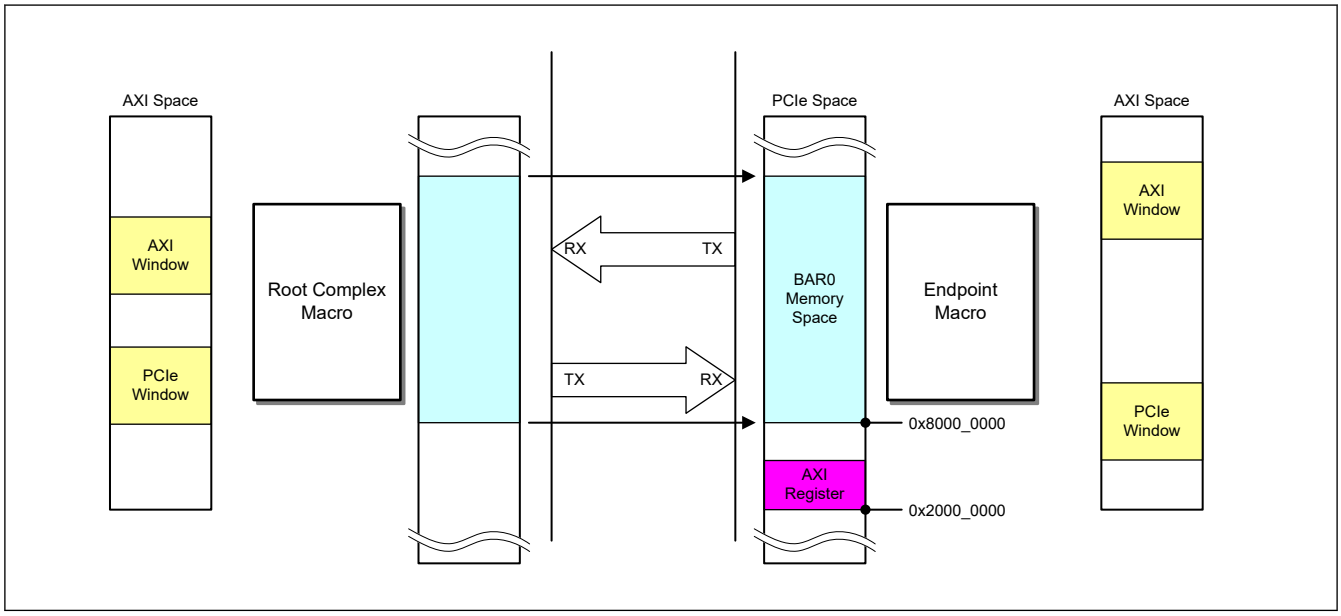
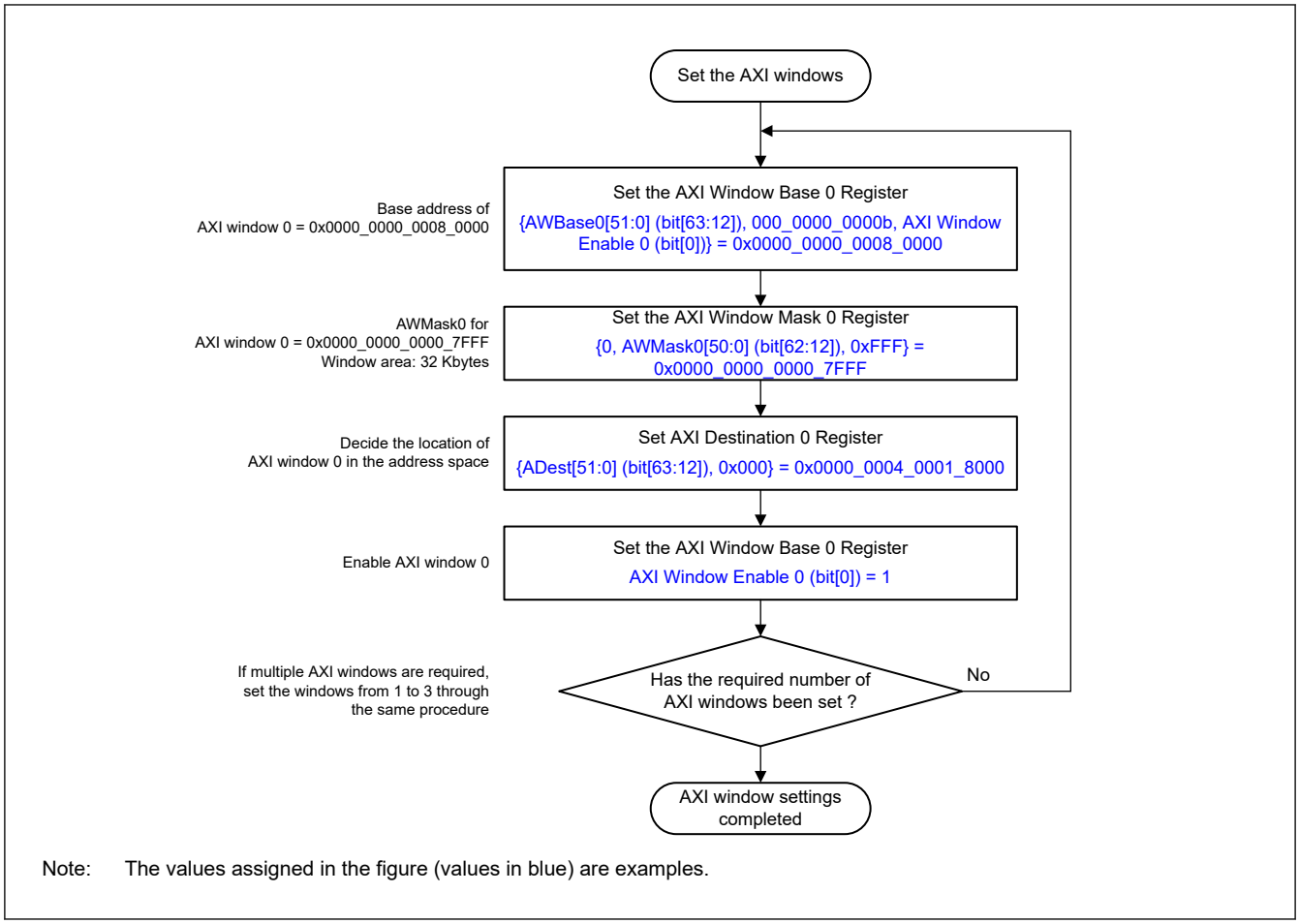


Figure 56.23 Setting of AXI windows on the endpoint by issuing memory read and memory write from root complex

Data transfer is possible when setting of the windows is completed and the Bus Master Enable bit and the Memory Space Enable bit of the given configuration register are set to the enabled state.

The following shows the procedure for making the settings by method 2.

The AXI windows can be set by write access to the registers through the AXI slave interface.



Note: The values assigned in the figure (values in blue) are examples.

Figure 56.24 Example settings of the AXI windows

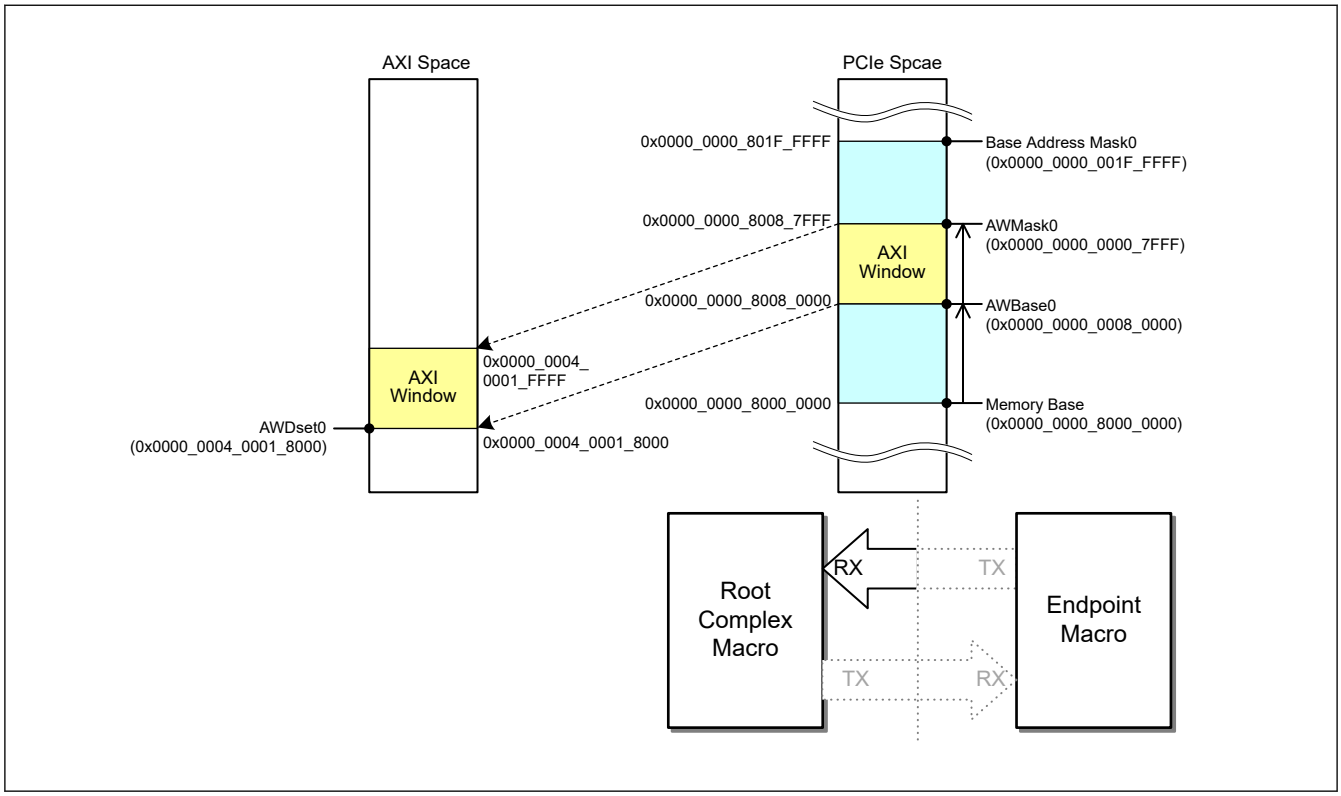


Figure 56.25 Mapping of the AXI windows

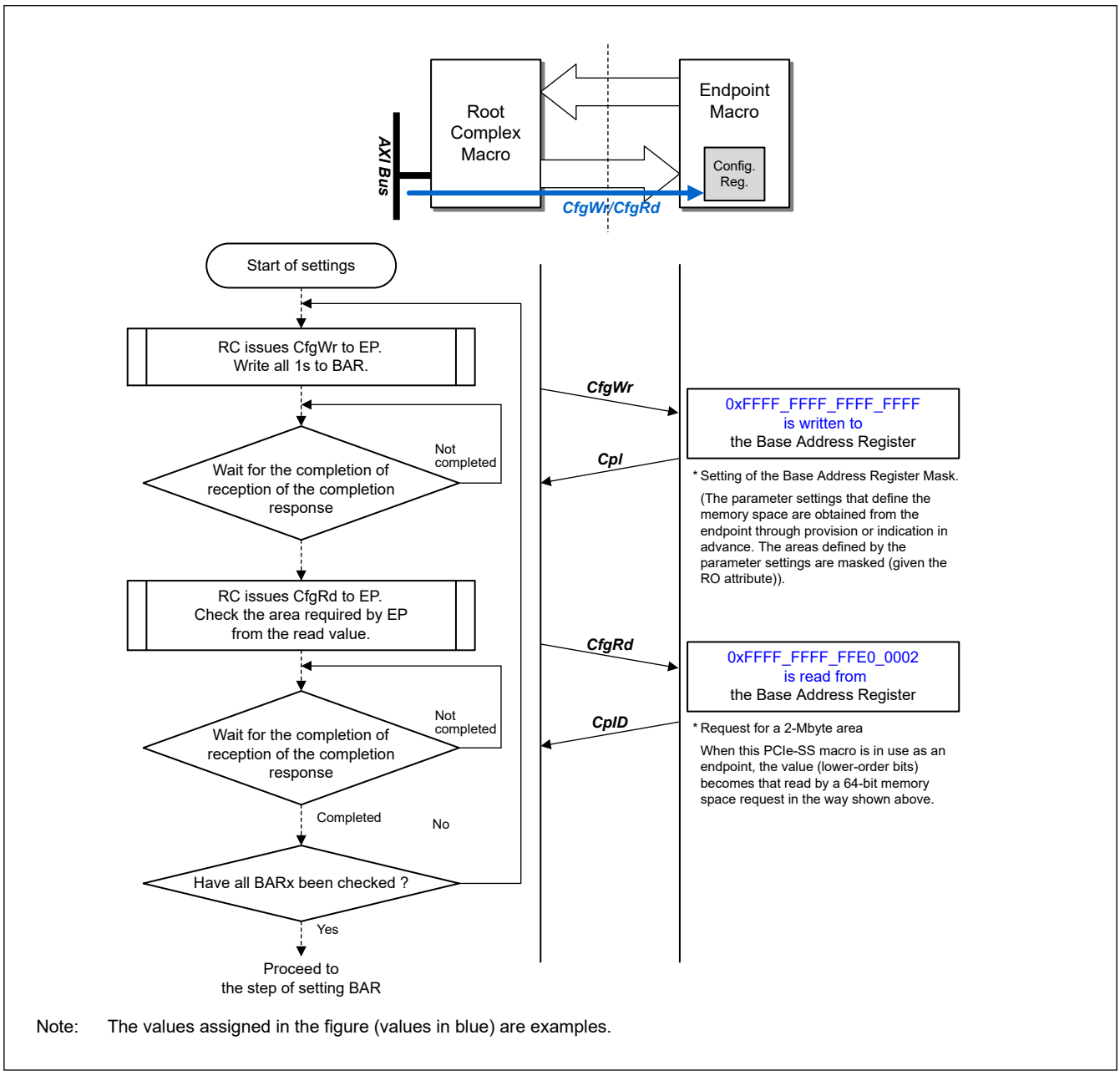
At this time, three address space settings, namely the AXI address space for the root complex device, the AXI address space for the endpoint device, and the PCI address space between the root complex and the endpoint, must be consistent within the system. If they are not, data cannot be transferred. Take care with the settings.

56.8.4.3 Setting the PCIe Address Space

A PC or similar system recognizes information on the areas of the memory space which are required by other-party endpoint devices by access to the configuration registers of the endpoints through software processing by the CPU on the root-complex side. In the case of embedded systems, etc., the areas required by the other-party devices will generally be known in advance, so this processing is not usually required.

The following is an example procedure for setting of the PCIe memory space for an other-party endpoint by the root complex.

Configuration requests are made to be issued to the other-party endpoint devices by register access through the AXI slave interface.



Note: The values assigned in the figure (values in blue) are examples.

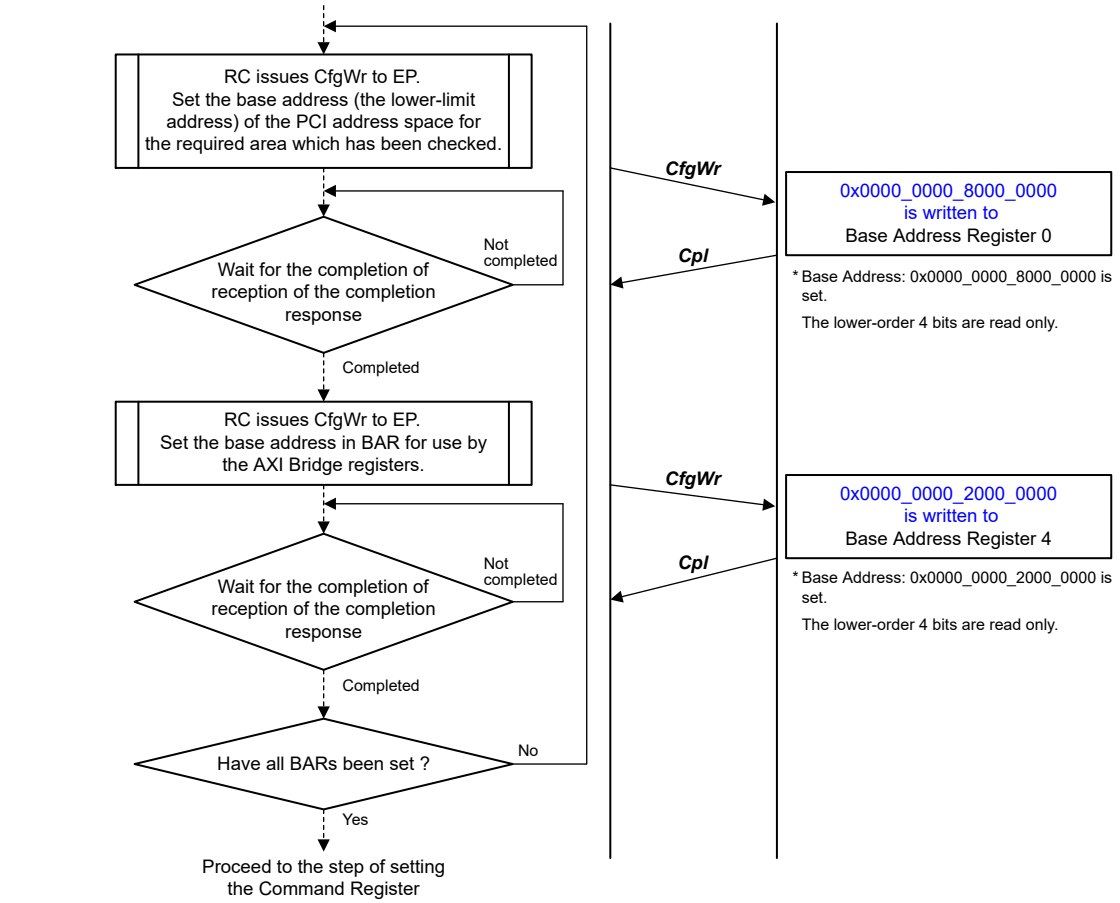
Figure 56.26 Example settings for the PCIe address space (1)

After issuing a configuration write request for writing all 1s (0xFFFF_FFFF_FFFF_FFFF), a configuration read request is issued and if the result of reading is all 0s (0x0000_0000_0000_0000), the given base address register (BAR) is judged to be reserved and the corresponding space is considered to be unused.

From the result of reading, software on the root complex side writes the base address (the lower-limit address) of the PCI address space to be allocated to the endpoint device by issuing a configuration write request.*1 At this time, setting of a BAR for which the area requested by the endpoint device was found to be reserved and thus unused is not required.

Also, if an address space which includes the other party has been prepared in advance, the above confirmation is not required. Set the base address directly.

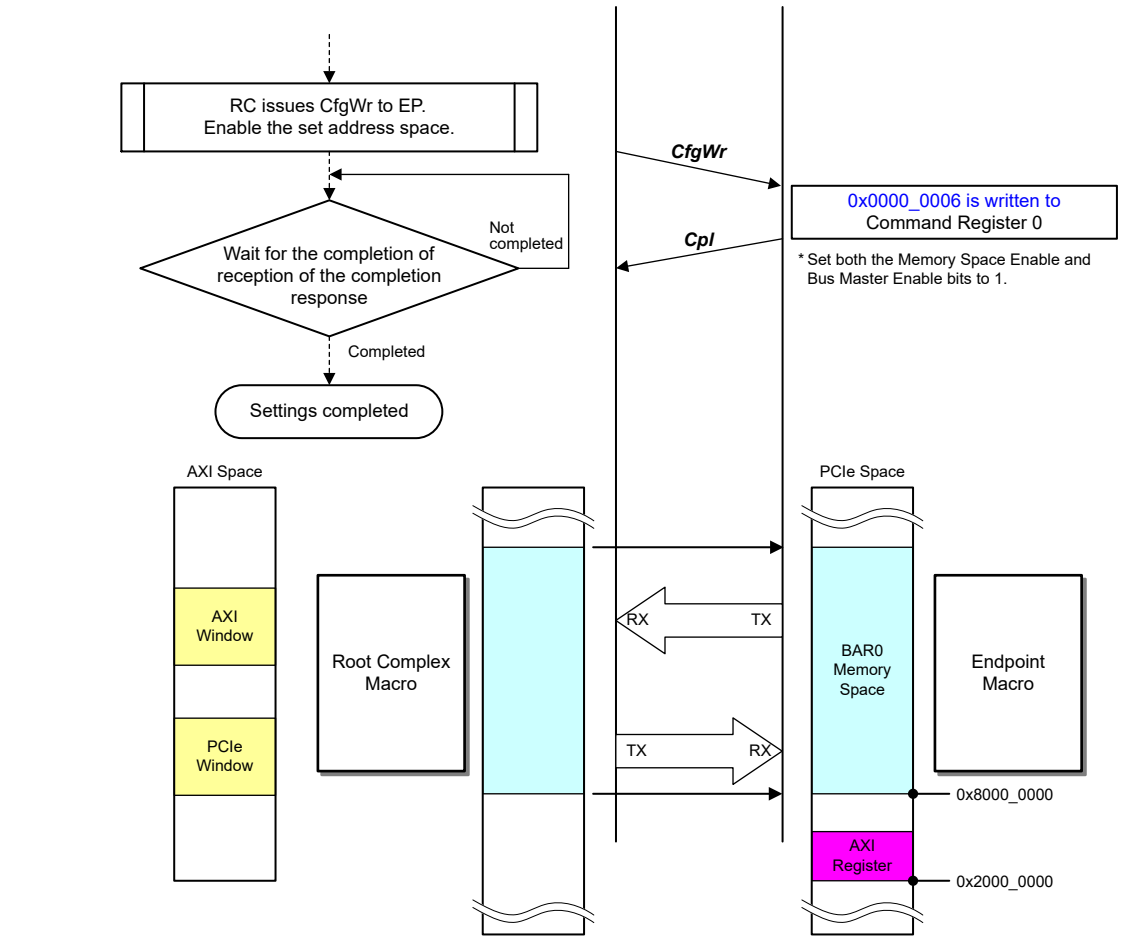
Note: Setting a base address register to all 0s is prohibited.



Note: The values assigned in the figure (values in blue) are examples.

Figure 56.27 Example settings for the PCIe address space (2)

The completion of settings for the PCIe address space means that the root complex is ready for data transfer to the corresponding endpoint. Finally, set the Bus Master Enable bit and the Memory Space Enable bit of the given command register to enable the memory space.



Note: The values assigned in the figure (values in blue) are examples.

Figure 56.28 Example settings for the PCIe address space (3)

56.8.5 Data Transfer

This module has one port each for master and slave operation as an AXI interface. For PCIe requests which can be issued through the master and slave ports, see section 56.7.3. Issuing of PCIe Requests and Register Access (by AXI) and section 56.7.4. Initiation of AXI Transactions and Register Access (by PCIe).

The focus of this explanation is on normal memory data transfer.

56.8.5.1 PCIe MWr (When the AXI Slave Is in Use)

A write transaction from the AXI bus (AXI-WR) via a window set as a PCIe window is converted into an MWr command (PCIe MWr) and then issued. If a PCIe MWr request is to be issued through the AXI interface slave port of the unit by using a DMAC, etc. external to the unit, the number of write transactions which can be accepted at a time is one, so the operation is as follows.

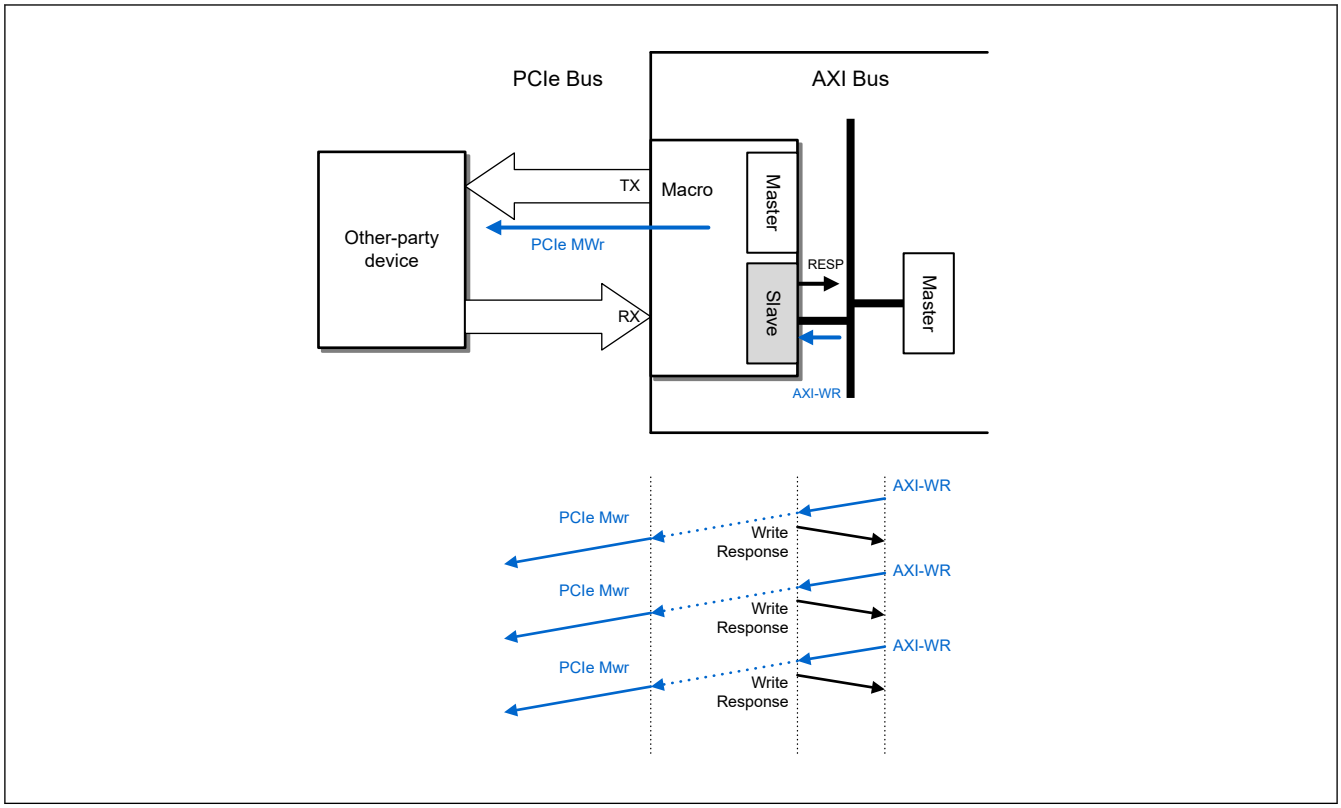


Figure 56.29 Data transfer by PCIe MWr (AXI Slave)

1. A write transaction from the AXI bus is issued through the AXI slave interface (write address channel, write data channel).
2. The transaction is converted into a PCIe MWr command and data are transferred to the other-party device via the PCIe bus.
3. After a wait for a response from the AXI slave interface (write response channel), a next AXI write transaction is issued.

A write response is issued after writing data to the transmission buffer (SRAM) within the unit. If this transmission buffer is full, the unit receives as much data as it can capture and then places the READY signal at the low level. Accordingly, the AXI bus may be placed in the hold state during transfer.

Since this depends not only on the specifications of the PCIe module (the number of lanes, rate, maximum payload size, etc.) but also on the size of the transmission buffer, the size of the reception buffer of the other-party device, and the external system configuration (the time for access to the external DRAM, etc.), care should be taken when considering the system configuration in general.

56.8.5.2 PCIe MWr (When the DMAC Is in Use)

If a PCIe MWr request is to be issued through the AXI master port by using the DMAC within the unit, the number of requests which can be read by the internal DMAC is one transaction per channel, so the operation is as follows.

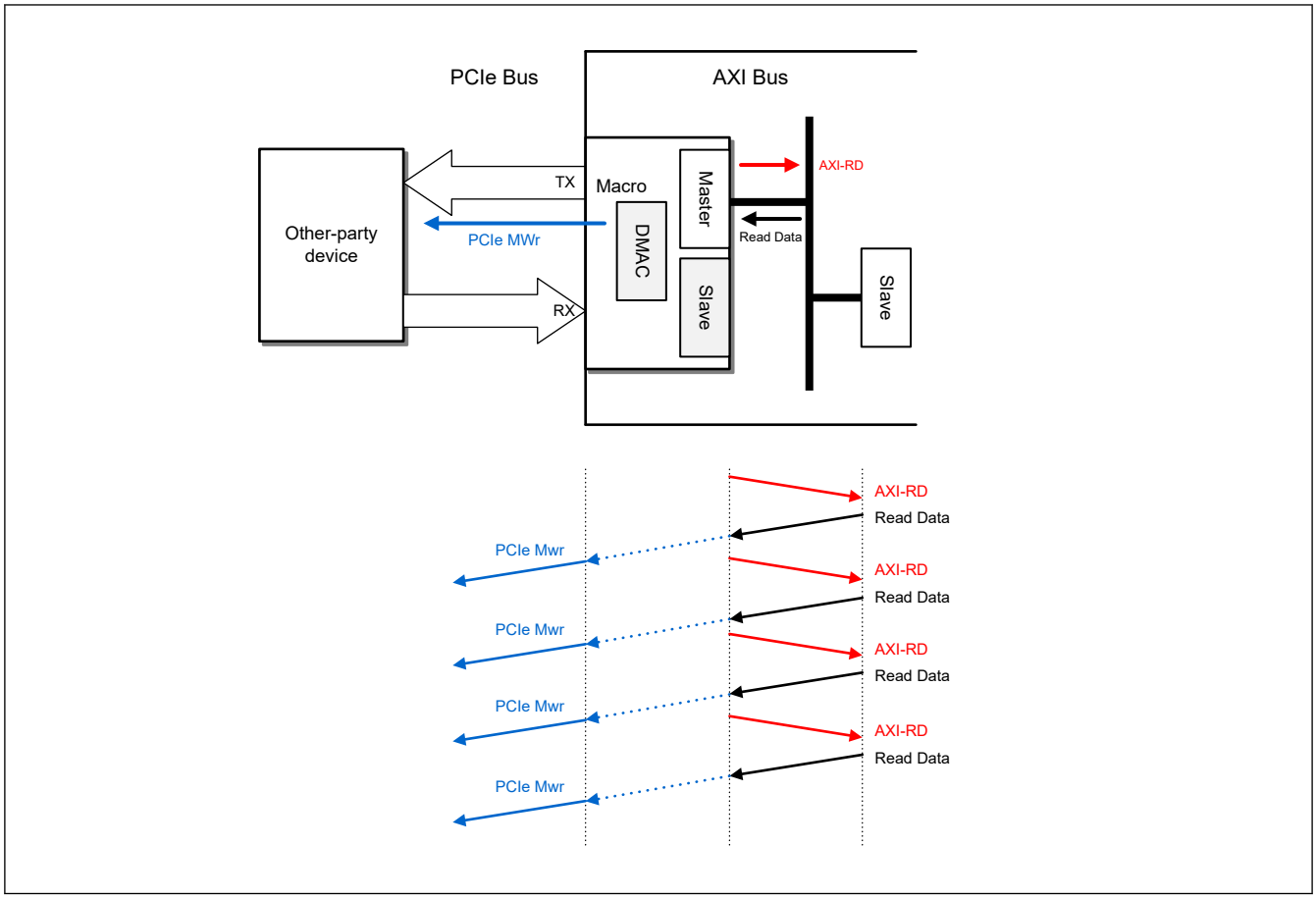


Figure 56.30 Data transfer by PCIe MWr (DMAC with 1 channel)

1. An AXI read (AXI-RD) request is issued through the AXI master interface (read address channel).
2. After the AXI master interface (read data channel) receives read data, this is converted into a PCIe MWr command and data are transferred to the other-party device via the PCIe bus.
3. Following the completion of the reception of read data, a next AXI-RD is issued.
4. The above steps are repeated until transfer of all bytes set in the DMA Size bits is completed.

The read data channel checks that the data that have been read are valid and indicates the readiness of the data for reception. Even when the transmission buffer (SRAM) within the unit is full, the unit issues read requests through the read address channel. Also, it receives as much data as it can capture and places subsequent data in the non-receivable state. Accordingly, depending on the state of the transmission buffer, data may not be transferred regardless of a request having been issued, or the AXI bus may be placed in the hold state during transfer. This lengthens the period of waiting for data that have been read, leading to the deterioration of transfer performance as well as the deterioration of overall system performance.

Likewise, contention with write access by the slave interface and contention in write access between channels may decrease performance in transfer.

Since this depends not only on the specifications of the PCIe module (the number of lanes, rate, maximum payload size, etc.) but also on the size of the transmission buffer, the size of the reception buffer of the other-party device, and the external system configuration (the time for access to the external DRAM, etc.), care should be taken when considering the system configuration in general.

The following is an example where the number of requests which can be read by the AXI master interfaced through four DMAC channels is 4.

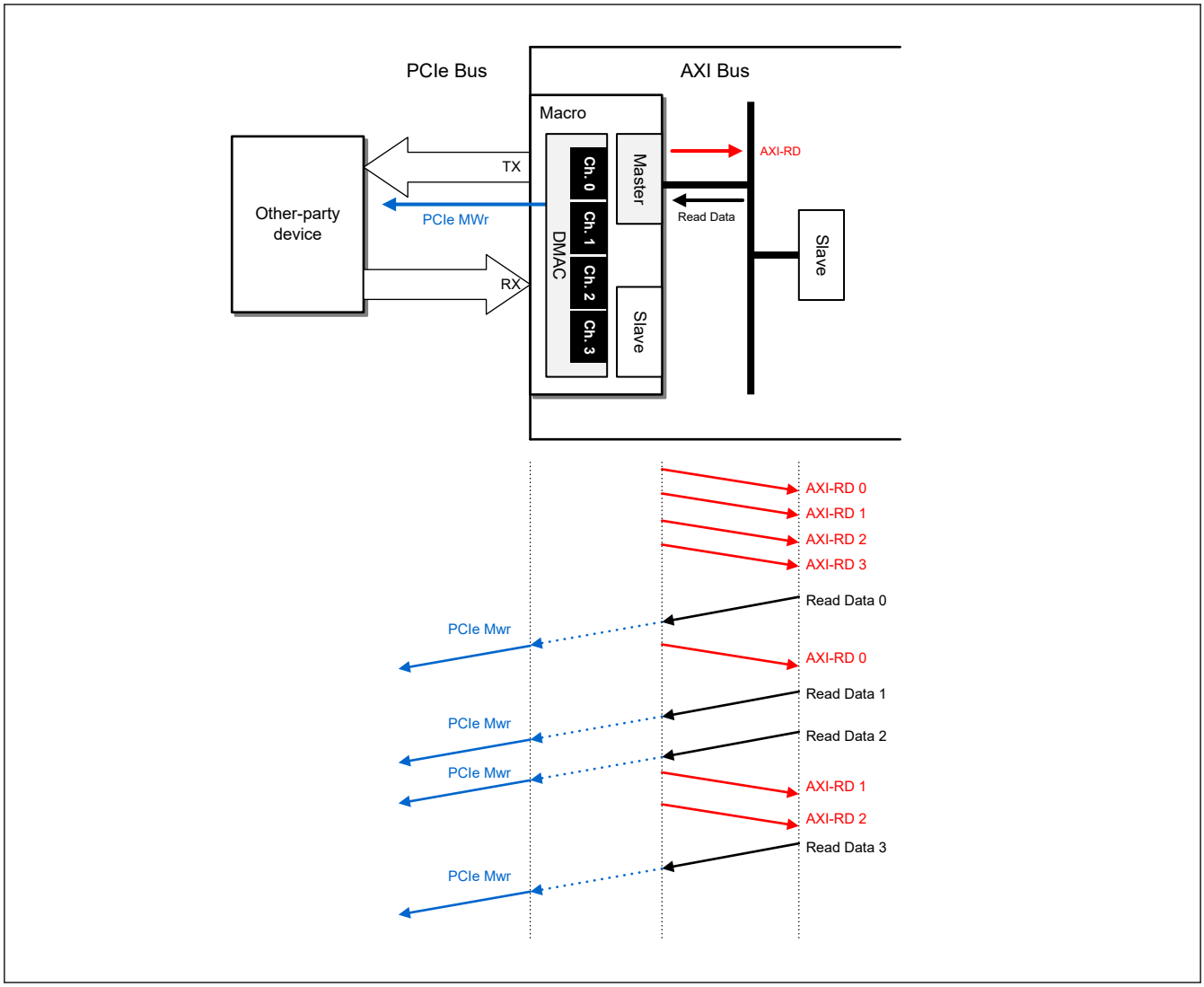


Figure 56.31 Data transfer by PCIe MWr (DMAC with 4 channels)

1. An AXI read (AXI-RD0: DMAC ch. 0) request is issued through the AXI master interface (read address channel).
2. Since the number of requests which can be read by the unit = 4, AXI read requests are subsequently issued through DMAC ch. 1, ch. 2, and ch. 3.
   No order of priority applies to the issuing of requests through these channels.
3. Following the completion of the reception of data read in response to the read request through ch. 0, ch. 0 is able to issue a next AXI-RD0.
4. The above steps are repeated until transfer of all bytes set in the DMA Size bits for each channel is completed.

Configuring data transfer as described above allows increased performance, although such a configuration complicates software control.

56.8.5.3 PCIe MRd (When the AXI Slave Is in Use)

In usage as an AXI slave, memory read requests (PCIe MRd) are issued to the PCIe interface, a completion (CplD) is received from the PCIe interface, and an AXI read transfer (AXI-RD) is initiated (this module supports eight outstanding transfers). Therefore, only the number of PCIe MRd requests that corresponds to this set number of outstanding transfers can be issued first, regardless of the reception of CplD. The received CplDs are stored in a data buffer (RAM) and then transferred through the AXI bus.

If a PCIe MRd is issued through the AXI interface slave port of the unit, the number of read requests which can be accepted by the AXI slave at a time = 1 to 8 corresponding to the PCIe section.

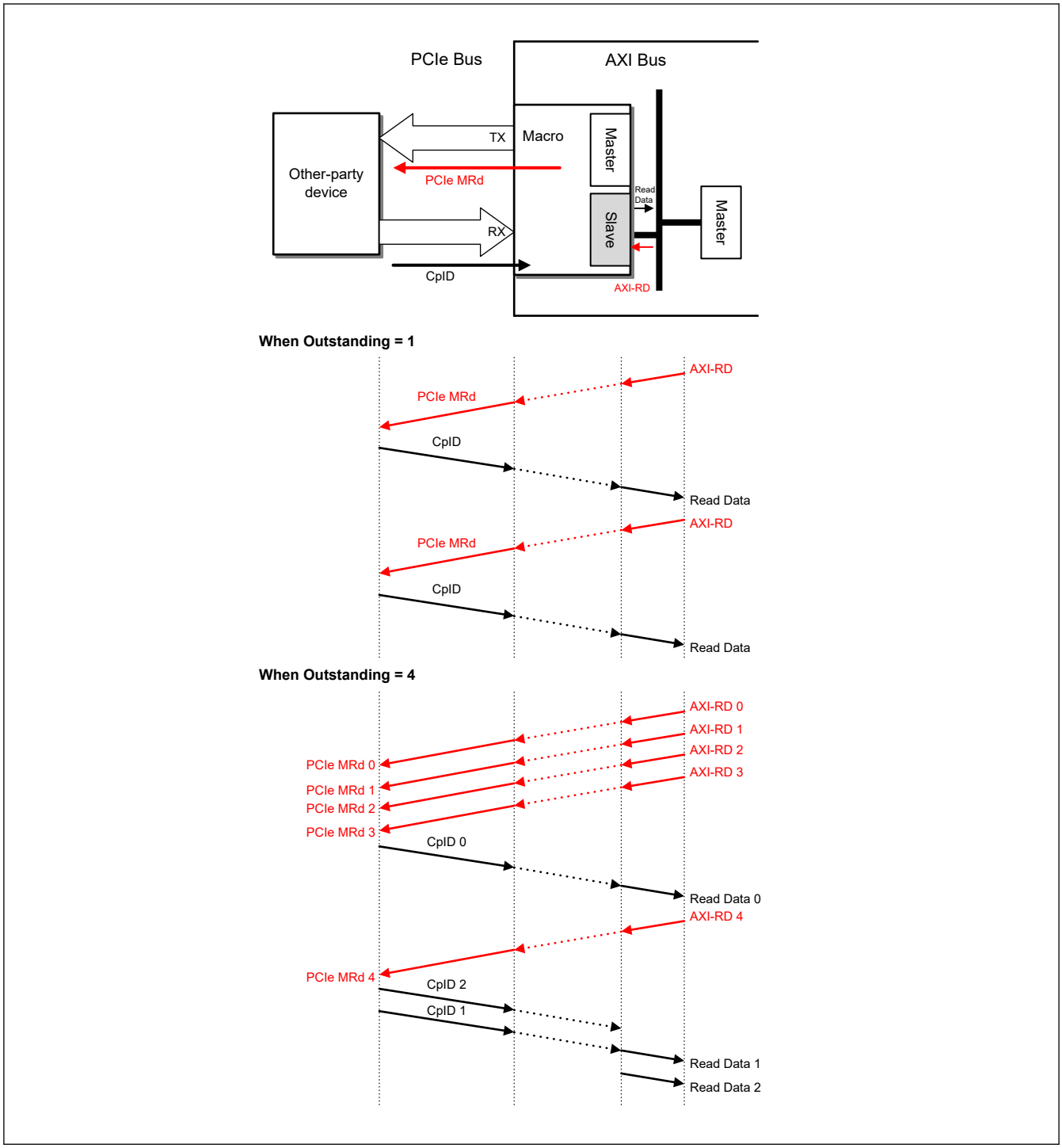


Figure 56.32 Data transfer by PCIe MRd (AXI Slave)

When outstanding = 1

1. An AXI read transaction is issued through the AXI slave interface (read address channel).
2. A transaction is converted into a PCIe MRd command and the request for data reading is issued to the other-party device via the PCIe bus.
3. CplD is received from the other-party device and an AXI read response is issued through the AXI slave interface (read data channel).
4. Steps 1 to 3 are repeated.

When outstanding = 4

1. An AXI read transaction is issued through the AXI slave interface (read address channel).
2. A transaction is converted into a PCIe MRd command and the request for data reading is issued to the other-party device via the PCIe bus.
   Steps 1 and 2 can be repeated until up to four consecutive requests have been issued.
3. CplD is received from the other-party device and an AXI read response is issued through the AXI slave interface (read data channel).
   At this time, the PCIe module can operate with out-of-order transactions, so the order of CplDs from the otherparty device may not be preserved. The unit also includes a buffer to support in-order transactions with the AXI bus. If this is used and the CplD for a preceding PCIe MRd is never returned, however, the next request cannot be issued and performance deteriorates accordingly.
   After that, steps 1 to 3 are repeated.

56.8.5.4 PCIe MRd (When the DMAC Is in Use)

If a PCIe MRd request is to be issued by using the DMAC within the unit, the overall number of requests is also 1 to 8 when the DMAC is incorporated depending on the number of outstanding transfers.

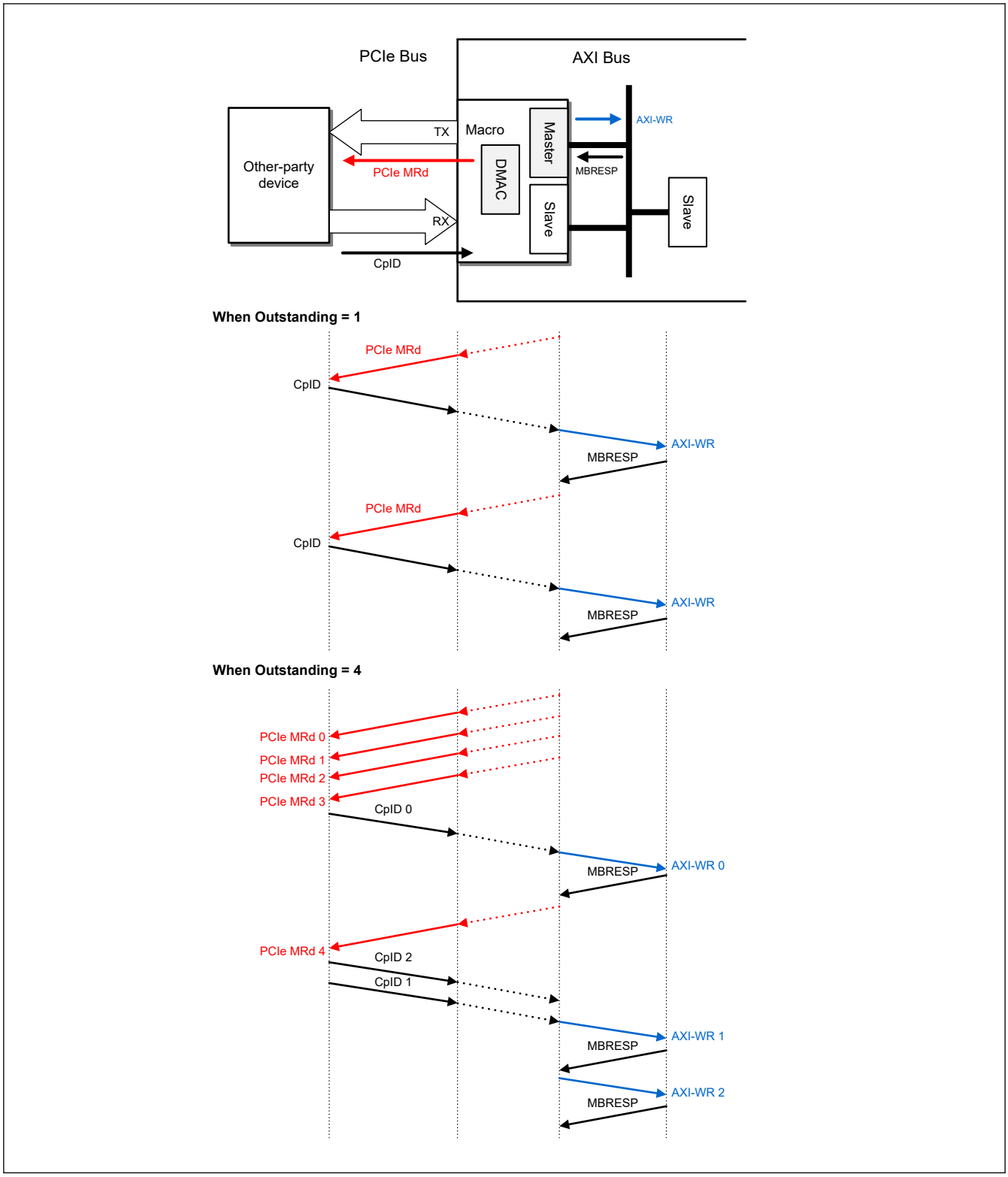


Figure 56.33 Data transfer by PCIe MRd (DMAC)

When outstanding = 1

1. Activate the internal DMAC through the AXI slave interface or from an other-party device.
2. The DMAC issues a read request and this is converted to a PCIe MRd command within the unit. The request for data reading is issued to the other-party device via the PCIe bus.
3. CplD is received from the other-party device and an AXI write transaction is issued through the AXI master interface (write address channel, write data channel).

4. Following the completion of the AXI write transfer, a next request is issued.
5. Steps 2 to 4 are repeated until transfer of all bytes set in the DMA Size bits is complete.

When outstanding = 4

1. Activate the internal DMAC through the AXI slave interface or from an other-party device.
2. The DMAC issues a read request and this is converted to a PCIe MRd command within the unit. The request for data reading is issued to the other-party device via the PCIe bus. Up to four consecutive requests can be issued.
3. CplD is received from the other-party device and an AXI write transaction is issued through the AXI master interface (write address channel, write data channel).
   At this time, the PCIe module can operate with out-of-order transactions, so the order of CplDs from the otherparty device may not be preserved. The unit also includes a buffer to support in-order transactions with the AXI bus. If this is used and the CplD for a preceding PCIe MRd is never returned, however, the next request cannot be issued and performance deteriorates accordingly.
4. Following the completion of the AXI write transfer, a next request is issued.
5. Steps 2 to 4 are repeated until transfer of all bytes set in the DMA Size bits is completed.

56.8.6 PCIe Initialization Procedure

56.8.6.1 Root Complex Mode

An example of the initialization procedure for PCIe Root Complex mode is described in steps 1 to 13.

Table 56.146 Initialization Procedure (RC) (1 of 2)

Step	Procedure
1	Set the TZC-400-2 register to grant proper transaction. For details, see section 13.4.4.3. TZC-400. Set the IO Port register to enable PCIE_RSTOUTnB pin. (when used) For details, see section 17, I/O Ports.
2	Set the Root Complex mode and Channel Configuration. Set the MODE_PORT bit in the PCIE_MODE register to 1. Set LINK_MASTER[1:0] bits in the PCIE_LINKMODE register.
3	Release the reset (ARESETn). Set the MRTCLM08 bit in the MRCTLM register to 0.
4	Release the module stop state. Set the MSTPCRM08 bit in the MSTPCRM register to 0.
5	Release Slave Stop Function. For details, see section 13.4.3. Slave Stop Function.
6	Release the reset (RST_LOAD_B, RST_CFG_B). Set RST_LOAD_B (bit4) and RST_CFG_B (bit3) in the PCI_RC_RESET to 1.
7	Setting of the PHY. Set the PHY_REG_CLK_EN bit in the PCI_RC_PERM register. Set the PCI_PHY_XCFGDn register (n = 0 to 26). Set the PCI_PHY_XCFGA_CMNn register (n = 0 to 15). Set the PCI_PHY_XCFGA_L0n register (n = 0 to 5). Set the PCI_PHY_XCFGA_L1n register (n = 0 to 5). For details, see section 56.5.1. PCI_PHY_XCFGDn : XCFGD Setting Register n (n = 0 to 26) to section 56.5.4. PCI_PHY_XCFGA_L1n : XCFGA_LN1 Setting Register n (n = 0 to 5).
8*1	Setting of HWINT related registers Set the CFG_HWINT_EN bit in the PCI_RC_PERM register. Set the PCI_RC_VID register to Device ID and Vendor ID. Set the PCI_RC_RID_CC register to Revision ID and Class Code. Set the PCI_RC_BARMSK00L and PCI_RC_BARMSK00U registers to Base Address Register Mask. Set the PCI_RC_BSIZE00_01 register to Base Size. Set the UPTP0, UPTP1, DPTP0, and DPTP1 bits in the PCI_RC_LEQCTL register (m = 0) to 0x8 each for 2 lanes × 1 channel configuration. Set the UPTP0 and DPTP0 in the PCI_RC_LEQCTL register (m = 0 and 1) to 0x8 for 1 lane × 2 channels configuration. Set the MODE_EQ_AUTONOMOUS bit in the PCI_RC_PCCTRL1 register to 1.

Table 56.146 Initialization Procedure (RC) (2 of 2)

Step	Procedure
9	Permit ASPM L1 State Transition Set ALLOW_ENTER_L1 bit in the PCIE_MISC register to 1.
10*2	Interrupt settings Set the PCI_RC_PEIS0 register to 0x0000_1200. Set the PCI_RC_PEIE0 register to 0x6000_1200. Set the PCI_RC_PEIS1 register to 0x0003_0303. Set the PCI_RC_PEIE1 register to 0x0003_0303. Set the PCI_RC_AMEIS register to 0x0000_0F0F. Set the PCI_RC_AMEIE register to 0x0000_0F0F. Set the PCI_RC_ASEIS1 register to 0x0000_0F03. Set the PCI_RC_ASEIE1 register to 0x0000_0F03. Set the PCI_RC_MSGRCVIS register to 0x010F_0000. Set the PCI_RC_MSGRCVIE register to 0x0105_0000.
11	Release the reset (RST_PS_B, RST_GP_B, RST_B, RST_OUT_B). Set RST_PS_B, RST_GP_B, RST_B and RST_OUT_B in the PCI_RC_RESET to 1.
12	Wait for 500 µs or more
13	Release the reset (RST_RSM_B). Set RST_RSM_B in the PCI_RC_RESET to 1.

Note 1. Change the setting values according to the operating conditions.
Note 2. Set the corresponding interrupt handler prior to using this function.

56.8.6.2 Endpoint Mode

An example of the initialization procedure for PCIe End Point mode is described in steps 1 to 17.

Table 56.147 Initialization Procedure (EP) (1 of 2)

Step	Procedure
1	Set the TZC-400-2 register to grant proper transaction. For details, see section 13.4.4.3. TZC-400. (PCIE_RSTOUTnB is not used for Endpoint.)
2	Set the Endpoint mode and Channel Configuration. Set the MODE_PORT bit in the PCIE_MODE register to 0. Set the LINK_MASTER[1:0] bits in the PCIE_LINKMODE register.
3	Release the reset (ARESETn). Set the MRTCLM08 bit in MRCTLM register to 0.
4	Release the module stop state. Set the MSTPCRM08 bit in MSTPCRM register to 0.
5	Release Slave Stop Function. For details, see section 13.4.3. Slave Stop Function.
6	Release the reset (RST_LOAD_B, RST_CFG_B). Set RST_LOAD_B (bit4) and RST_CFG_B (bit3) in the PCI_EP_RESET to 1.
7	Setting of the PHY. Set the PHY_REG_CLK_EN bit in PCI_EP_PERM register. Set the PCI_PHY_XCFGDn register (n = 0 to 26). Set the PCI_PHY_XCFGA_CMNn register (n = 0 to 15). Set the PCI_PHY_XCFGA_L0n register (n = 0 to 5). Set the PCI_PHY_XCFGA_L1n register (n = 0 to 5). For details, see section 56.5.1. PCI_PHY_XCFGDn : XCFGD Setting Register n (n = 0 to 26) to section 56.5.4. PCI_PHY_XCFGA_L1n : XCFGA_LN1 Setting Register n (n = 0 to 5).

Table 56.147 Initialization Procedure (EP) (2 of 2)

Step	Procedure
8*1	Setting of HW related registers (Function #i, i = 0, 1) Set Device ID and Vendor ID to the PCI_EP_VID_Fi registers. Set Revision ID and Class Code to the PCI_EP_RID_CC_Fi registers. Set Subsystem ID and Subsystem Vendor ID to the PCI_EP_SSID_Fi registers. Set Base Address Register Mask to the PCI_EP_BARMSK00L_Fi and PCI_EP_BARMSK00U_Fi registers. Set Base Size to the PCI_EP_BSIZE00_01_Fi registers. Set Base Address Register Mask to the PCI_EP_BARMSK01L_Fi and PCI_EP_BARMSK01U_Fi registers. Set Base Address Register Mask to the PCI_EP_BARMSK02L_Fi and PCI_EP_BARMSK02U_Fi registers. Set Base Size to the PCI_EP_BSIZE02_03_Fi registers. Set Base Size to the PCI_EP_BSIZE04_05_Fi registers. Set Base Size to the PCI_EP_BSIZE06_Fi registers.
9	Permit ASPM L1 State Transition Set ALLOW_ENTER_L1 bit in the PCIE_MISC register to 1.
10*2	Interrupt settings Set the PCI_EP_PEIS0 register to 0x0000_1200. Set the PCI_EP_PEIE0 register to 0x6000_1200. Set the PCI_EP_PEIS1 register to 0x0003_0303. Set the PCI_EP_PEIE1 register to 0x0003_0303. Set the PCI_EP_AMEIS register to 0x0000_0F0F. Set the PCI_EP_AMEIE register to 0x0000_0F0F. Set the PCI_EP_ASEIS1 register to 0x0000_0F03. Set the PCI_EP_ASEIE1 register to 0x0000_0F03. Set the PCI_EP_MSGRCVIS register to 0x010F_0000. Set the PCI_EP_MSGRCVIE register to 0x010A_0000.
11	Release the reset (RST_PS_B, RST_GP_B, RST_B, RST_OUT_B). Set RST_PS_B, RST_GP_B, RST_B and RST_OUT_B in the PCI_EP_RESET to 1.
12	Wait for 500 µs or more
13	Release the reset (RST_RSM_B). Set RST_RSM_B in the PCI_EP_RESET to 1.

Note 1. Change the setting values according to the operating conditions.
Note 2. Set the corresponding interrupt handler prior to using this function.

56.9 Points for Caution and Restrictions

56.9.1 Points for Caution and Prohibited Items in the Issuing of Requests

The following describes the restrictions in the issuing of various requests.

1. Prohibition of fixed bursts of 2 or more beats
   Fixed bursts for 2 or more beats are prohibited. If an attempt to use these is made, an OKAY response is returned to the AXI bus, but this may lead to issuing of the unexpected requests to the PCIe side, or unexpected register access which will change register values, and so on.
2. Points for caution in the issuing of special requests
   The requests listed below are issued by access to the internal registers. The registers are only accessible from the AXI side and writing to them while requests are being processed is prohibited. Attempted access from the PCIe side will be ignored.
   [Special Requests]
   - Zero-Length Memory Read Request
   - Configuration Read
   - Configuration Write
   - Message Request
   - Message Request with data payload

Note: Do not issue a special request for which issuing of the request is prohibited. Operation is not guaranteed if this is done.

56.9.2 Ordering Specifications of Received Non-Posted and Posted Requests

In the specifications of this module, a non-posted request is not overtaken by a posted request on the receiving side.

Requests received from another party are output to the higher-level bus in the order of reception.

56.9.3 Caution when Changing the Speed Spontaneously by the EP Unit

Follow the procedure below. The following control bits are present in the PCI Core Control 2 Register and PCI Core Status 2 Register among the AXI-Bridge registers.

1. Wait until the home node is placed in the L0 state.
2. Read the value of the STATE_DATA_RATE_IDENTIFIER_RECEIVED bit to check the supported speed of the other-party node.
3. If the home node supports the speed to which the other-party node wants to change, set the UI_LINK_SPEED_CHANGE[1:0] bits and assert the UI_LINK_SPEED_CHANGE_REQ bit.
4. Wait for the UI_LINK_SPEED_CHANGE_DONE bit to be asserted (wait for the completion).

Note: UI_LINK_SPEED_CHANGE_REQ is retained until UI_LINK_SPEED_CHANGE_DONE is asserted. It should be de-asserted after checking the assertion of UI_LINK_SPEED_CHANGE_DONE.

56.9.4 Error Processing of Unsupported Requests

The following describes the flow of processing for error reporting and error logging.

1. Access that straddles a 4-KB boundary
   Data transfer to a memory space which straddles a 4-KB boundary cannot proceed (this is stipulated by the PCI Express Base Specification). In this module, the reception of a memory write or read request for a memory space which straddles a 4-KB boundary is handled as a malformed TLP.
2. RCB violations
   The PCI Express Base Specification states that error processing in response to violations of the read completion boundary (RCB) is optional. The PCIe module of this LSI chip does not support the detection of RCB violations at the time of the reception of completion responses. Support for this function should be handled by the user logic.
3. Error processing in response to byte enable fields
   The PCI Express Base Specification states that error processing in response to Byte Enable fields is optional. This module does not support the detection of errors in the form of violations of Byte Enable rules. Support for this function should be handled by the user logic.
4. Request that extends beyond the base address boundary
   This core does not detect a request that starts within the range set in the Base Address Register but extends beyond the boundary from the base address as an error. Such a request should be handled by the user logic.
5. Processing when a TPL that is ready to be transferred by the transmitter is a malformed TLP
   Even if malformation of a TLP that is ready to be transferred by the transmitter is detected, due to the detection of parity errors when it is read from the FIFO buffer and so on, this module does not suspend the transmission of such a TLP. Received malformed TLPs must be handled appropriately by other-party devices.

56.9.5 Processing on Reception of the Message

The following lists the messages to be detected as a silent drop or UR on reception of the given message.

Table 56.148 Operations on message reception (1 of 2)

Received message	Root complex	Endpoint
Assert_INTx	Normal processing	Silent Drop
Deassert_INTx	Normal processing	Silent Drop
ERR_COR	Normal processing	Silent Drop
ERR_NONFATAL	Normal processing	Silent Drop
ERR_FATAL	Normal processing	Silent Drop

Table 56.148 Operations on message reception (2 of 2)

Received message	Root complex	Endpoint
UNLOCK	Silent Drop	Silent Drop
Set_Slot_Power_Limit	Silent Drop	Normal processing
Vendor_Defined_Type0	UR	UR
Vendor_Defined_Type1	Silent Drop	Silent Drop
Ignore	Silent Drop	Silent Drop
LTR	UR	UR
OBFF	UR	UR
PM_PME	Normal processing	Silent Drop
PME_TO_Ack	Normal processing	Silent Drop
PM_Active_State_Nak	Silent Drop	Normal processing
PME_Turn_Off	Silent Drop	Normal processing
PTM_Request	UR	UR
PTM_Response	UR	UR
PTM_ResponseD	UR	UR
Invalidate_Request	Silent Drop	UR
Invalidate_Completion	Normal processing	Silent Drop
Page_Request	UR	UR
Page_Response	UR	UR

Note: Silent Drop: Normal completion, data are not reflected
UR: Unsupported Request

56.9.6 Other Points for Caution

1. Access after the de-assertion of the reset signal
   When a cycle of writing starts before the value of the SAWREADY bit has become 1 following the de-assertion of the reset signal, access in the second and subsequent cycles produces slave errors.
   After the reset signal has been de-asserted, do not assert the SAWVALID signal until SAWREADY has become 1.
2. Point for caution at times of register writing
   In some cases of writing 2DW or more and skipping over with the use of byte enable to a register from the AXI side, writing might not proceed as expected.
   Using a pin reset or register reset from the AXI side as a non-consecutive SWSTRB, restricts writing to no more than 1DW (32 bits).
3. Point for caution on register reading
   In the case of reading a register from the AXI side, values read from invalid byte lanes are from undefined outputs (meaningless garbage data).
4. Generation of unexpected correctable errors
   A correctable error may be detected at the time of EIOS reception following low power state transitions of this module such as from L0s to L1. If this happens, processing to send a message or assert an interrupt flag signal is to proceed. Take care on this point and respond appropriately so that a correctable error is not handled as a fatal error (a mask setting by a register to switch the notification of unexpected correctable errors off is recommended).
5. Reset interval in transitions from a hot reset to detection in RC mode
   In generations of PCI Express after Gen1, at the time of transitions from a hot reset to detection in accord with the operating rate, the base specification prescribes securing a 1-ms waiting interval for changes to the rate, so secure a reset interval of 1 ms.
6. Enter Compliance Pattern (gen3)
   Lane information included in compliance pattern corresponds to physical lane number.
7. Use of Retimer
   One retimer is supported.

8. Note for Window setting
   The following caution and limitation should be considered for address coversion for the Window.
   - Do not generate carry bit when BAR and BARmask are added.
   - Do not generate carry bit when BaseX and MaskX are added.
   - Do not generate carry bit when MaskX and DestX are added.
   - Do not set the bit data in MaskX intermittently (Setting value must be 2^{n-1}).
   - Minimum area provided by BAR and BARMask is 4 KB.
   - Each window should have a single memory size that is a power of two.
   - Do not overlap window and AXI Bridge Register each other.
9. Clock setting in case of single link mode
   When using this module in single link mode, only common clock is supported, no separate clock.
   Since initial value of bit 6 in Link Control/Status register is separate clock, setting should be changed.
   When using this module in multi link mode, both common clock and separate clock are supported.
10. In case of using down config in single link mode, when LTSSM (Link Training Status State Machine) transits to Detect State by Link Down and Hot Reset after Link up, the state may not transit to other state.
    Assert reset by following procedure to recover from that state.

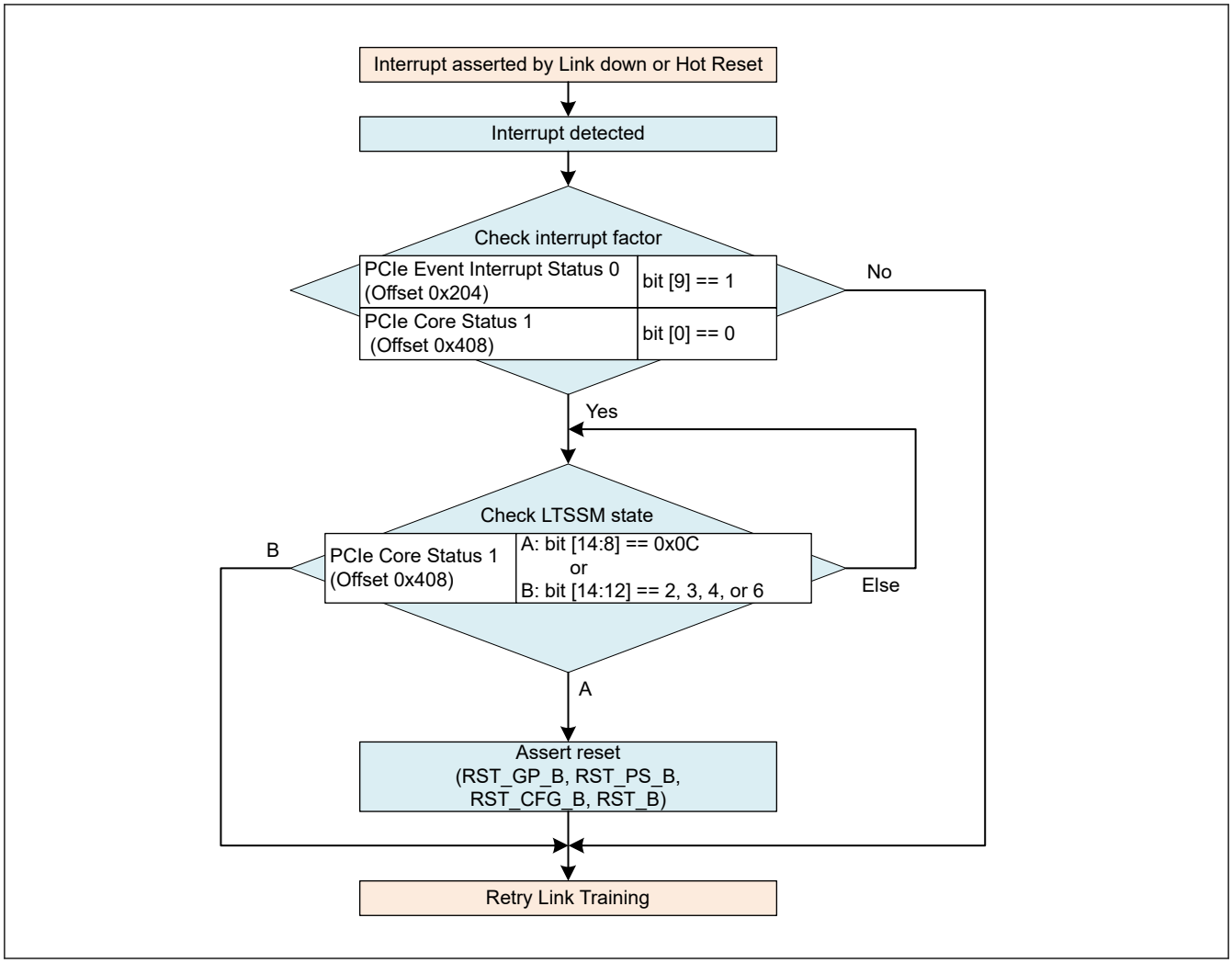


Figure 56.34 Recovery procedure

57. LPDDR4 SDRAM Subsystem (DDRSS)

57.1 Overview

The LPDDR4 SDRAM Subsystem consists of Memory Controller (MC) and PHY. The DDRSS supports the following specifications:

- LPDDR4 (JEDEC STANDARD JESD209-4D)

Table 57.1 Specifications of DDRSS

Item	Specification
Number of units	• One unit
DRAM IF	• LPDDR4: 3200 Mbps (1600 MHz) • Width: 32 bits (16 bits per channel) • Rank: 1, 2 • Density: up to 64 Gb (byte mode not supported)
MC	• Fully pipelined command, read and write data interfaces to the controller. • Advanced bank look-ahead features for high memory throughput. • A programmable register interface to control memory parameters and protocols including auto pre-charge. • Full initialization of memory on controller reset. • Supports the Weighted Round-Robin arbitration schema for arbitrating request from the ports. • ECC function for single-bit and double-bit error reporting, single-bit error correction, and programmable removal of ECC storage. • Built-in Self test (BIST) for external DRAM memories.
PHY	• Command Bus eye training relative to CK • Write leveling to compensate for CK-DQS timing skew • Write training to deskew DQs, DM, and DQS for write: – Command-based FIFO WR/RD with user patterns – An internal DQS clock tree oscillator, to determine the need for, and the magnitude of, required periodic training – Data bus VREFDQ training • Read training to deskew DQs, DM, and DQS for read: – DQ bit deskew training using DRAM mode registers – DQS to DQ eye centering training using DRAM array • Automatic periodic retraining through PHY master interface • LVSTL I/O calibration and ODT calibration • Support for a SW controllable DQ bit and AC bit swizzling
Low power	• Multiple low power states (power-down, self-refresh) • Automatic or software interface

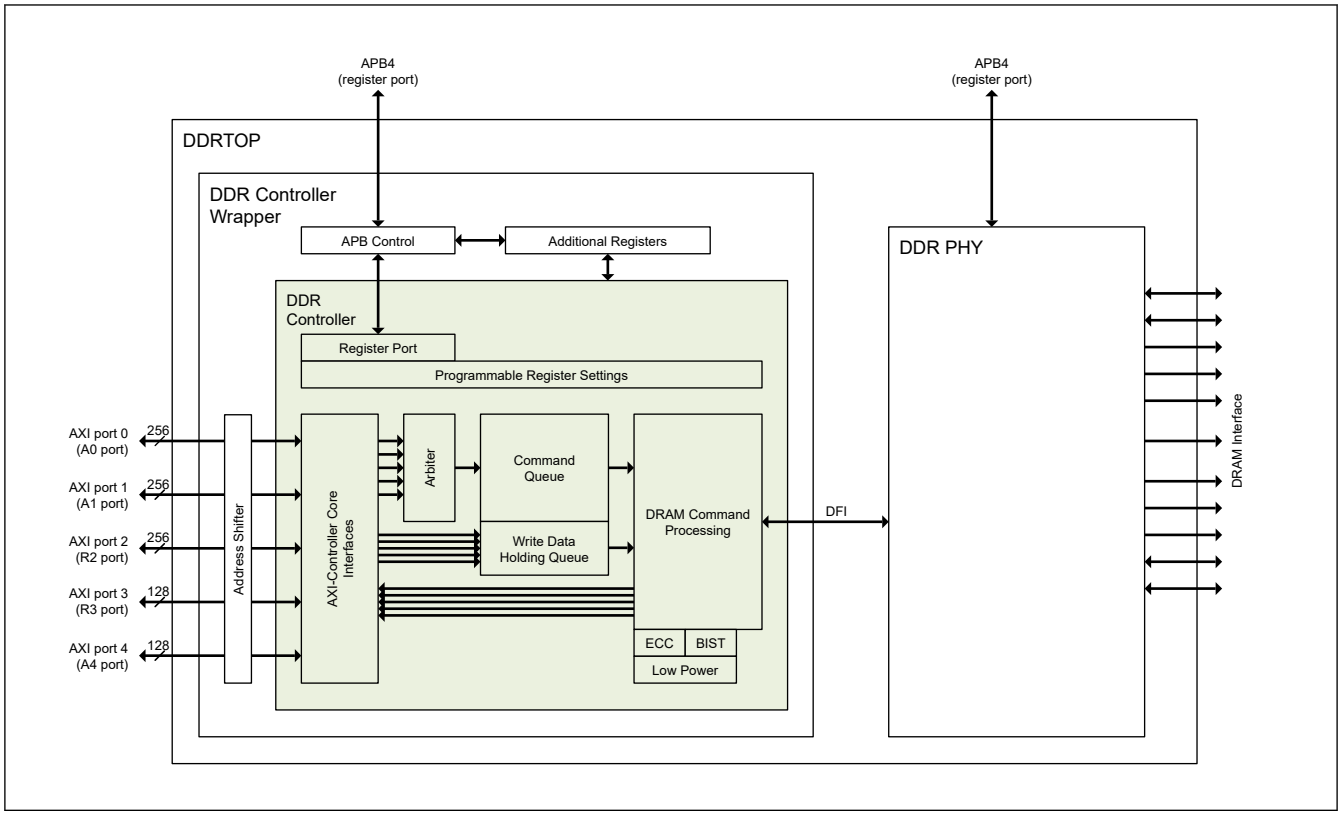


Figure 57.1 Block diagram of the DDRSS

Table 57.2 Input/output pins of the DDRSS (1 of 2)

Pin name	I/O	Function
DDR_ZN	Output	DDRSS calibration external reference resistor
DDR_DTEST	I/O	DDRSS Digital test point for debug. Keep open on the PCB.
DDR_ATEST	I/O	DDRSS Analog test point for debug. Keep open on the PCB.
DDR_RESET_N	Output	DDRSS DRAM reset
DDR_CKA_T	I/O	DDRSS DRAM Ch A Clock (positive)
DDR_CKA_C	I/O	DDRSS DRAM Ch A Clock (negative)
DDR_CKEA[1:0]	I/O	DDRSS DRAM Ch A Clock Enable
DDR_CSA[1:0]	I/O	DDRSS DRAM Ch A Chip Select
DDR_CAA[5:0]	I/O	DDRSS DRAM Ch A Command/Address
DDR_CKB_T	I/O	DDRSS DRAM Ch B Clock (positive)
DDR_CKB_C	I/O	DDRSS DRAM Ch B Clock (negative)
DDR_CKEB[1:0]	I/O	DDRSS DRAM Ch B Clock Enable
DDR_CSB[1:0]	I/O	DDRSS DRAM Ch B Chip Select
DDR_CAB[5:0]	I/O	DDRSS DRAM Ch B Command/Address
DDR_DQA[15:0]	I/O	DDRSS DRAM Ch A Data
DDR_DMIA[1:0]	I/O	DDRSS DRAM Ch A Data Mask Inversion
DDR_DQSA_T[1:0]	I/O	DDRSS DRAM Ch A Data Strobe (positive)
DDR_DQSA_C[1:0]	I/O	DDRSS DRAM Ch A Data Strobe (negative)
DDR_DQB[15:0]	I/O	DDRSS DRAM Ch B Data
DDR_DMIB[1:0]	I/O	DDRSS DRAM Ch B Data Mask Inversion
DDR_DQSB_T[1:0]	I/O	DDRSS DRAM Ch B Data Strobe (positive)

Table 57.2 Input/output pins of the DDRSS (2 of 2)

Pin name	I/O	Function
DDR_DQSB_C[1:0]	I/O	DDRSS DRAM Ch B Data Strobe (negative)

57.2 Register Map

Table 57.3 DDRSS register map (1 of 3)

Address	Register symbol	Register name	Write protection
0x8030_0000	DDR_MEMC_DENALI_CTL_00	DENALI_CTL_00 Register	—
0x8030_036C	DDR_MEMC_DENALI_CTL_219	DENALI_CTL_219 Register	—
0x8030_0370	DDR_MEMC_DENALI_CTL_220	DENALI_CTL_220 Register	—
0x8030_0374	DDR_MEMC_DENALI_CTL_221	DENALI_CTL_221 Register	—
0x8030_0378	DDR_MEMC_DENALI_CTL_222	DENALI_CTL_222 Register	—
0x8030_0384	DDR_MEMC_DENALI_CTL_225	DENALI_CTL_225 Register	—
0x8030_0388	DDR_MEMC_DENALI_CTL_226	DENALI_CTL_226 Register	—
0x8030_038C	DDR_MEMC_DENALI_CTL_227	DENALI_CTL_227 Register	—
0x8030_0390	DDR_MEMC_DENALI_CTL_228	DENALI_CTL_228 Register	—
0x8030_0394	DDR_MEMC_DENALI_CTL_229	DENALI_CTL_229 Register	—
0x8030_039C	DDR_MEMC_DENALI_CTL_231	DENALI_CTL_231 Register	—
0x8030_03A0	DDR_MEMC_DENALI_CTL_232	DENALI_CTL_232 Register	—
0x8030_03A4	DDR_MEMC_DENALI_CTL_233	DENALI_CTL_233 Register	—
0x8030_03A8	DDR_MEMC_DENALI_CTL_234	DENALI_CTL_234 Register	—
0x8030_03AC	DDR_MEMC_DENALI_CTL_235	DENALI_CTL_235 Register	—
0x8030_03B0	DDR_MEMC_DENALI_CTL_236	DENALI_CTL_236 Register	—
0x8030_03B4	DDR_MEMC_DENALI_CTL_237	DENALI_CTL_237 Register	—
0x8030_03B8	DDR_MEMC_DENALI_CTL_238	DENALI_CTL_238 Register	—
0x8030_03BC	DDR_MEMC_DENALI_CTL_239	DENALI_CTL_239 Register	—
0x8030_03C0	DDR_MEMC_DENALI_CTL_240	DENALI_CTL_240 Register	—
0x8030_03C4	DDR_MEMC_DENALI_CTL_241	DENALI_CTL_241 Register	—
0x8030_03C8	DDR_MEMC_DENALI_CTL_242	DENALI_CTL_242 Register	—
0x8030_03D0	DDR_MEMC_DENALI_CTL_244	DENALI_CTL_244 Register	—
0x8030_03D4	DDR_MEMC_DENALI_CTL_245	DENALI_CTL_245 Register	—
0x8030_03D8	DDR_MEMC_DENALI_CTL_246	DENALI_CTL_246 Register	—
0x8030_03DC	DDR_MEMC_DENALI_CTL_247	DENALI_CTL_247 Register	—
0x8030_03E0	DDR_MEMC_DENALI_CTL_248	DENALI_CTL_248 Register	—
0x8030_03E4	DDR_MEMC_DENALI_CTL_249	DENALI_CTL_249 Register	—
0x8030_03E8	DDR_MEMC_DENALI_CTL_250	DENALI_CTL_250 Register	—
0x8030_03EC	DDR_MEMC_DENALI_CTL_251	DENALI_CTL_251 Register	—
0x8030_0434	DDR_MEMC_DENALI_CTL_269	DENALI_CTL_269 Register	—
0x8030_0438	DDR_MEMC_DENALI_CTL_270	DENALI_CTL_270 Register	—
0x8030_043C	DDR_MEMC_DENALI_CTL_271	DENALI_CTL_271 Register	—
0x8030_0440	DDR_MEMC_DENALI_CTL_272	DENALI_CTL_272 Register	—
0x8030_0444	DDR_MEMC_DENALI_CTL_273	DENALI_CTL_273 Register	—

Table 57.3 DDRSS register map (2 of 3)

Address	Register symbol	Register name	Write protection
0x8030_0448	DDR_MEMC_DENALI_CTL_274	DENALI_CTL_274 Register	—
0x8030_044C	DDR_MEMC_DENALI_CTL_275	DENALI_CTL_275 Register	—
0x8030_04F8	DDR_MEMC_DENALI_CTL_318	DENALI_CTL_318 Register	—
0x8030_04FC	DDR_MEMC_DENALI_CTL_319	DENALI_CTL_319 Register	—
0x8030_0518	DDR_MEMC_DENALI_CTL_326	DENALI_CTL_326 Register	—
0x8030_051C	DDR_MEMC_DENALI_CTL_327	DENALI_CTL_327 Register	—
0x8030_0520	DDR_MEMC_DENALI_CTL_328	DENALI_CTL_328 Register	—
0x8030_0524	DDR_MEMC_DENALI_CTL_329	DENALI_CTL_329 Register	—
0x8030_0528	DDR_MEMC_DENALI_CTL_330	DENALI_CTL_330 Register	—
0x8030_0530	DDR_MEMC_DENALI_CTL_332	DENALI_CTL_332 Register	—
0x8030_0534	DDR_MEMC_DENALI_CTL_333	DENALI_CTL_333 Register	—
0x8030_0538	DDR_MEMC_DENALI_CTL_334	DENALI_CTL_334 Register	—
0x8030_053C	DDR_MEMC_DENALI_CTL_335	DENALI_CTL_335 Register	—
0x8030_0540	DDR_MEMC_DENALI_CTL_336	DENALI_CTL_336 Register	—
0x8030_0544	DDR_MEMC_DENALI_CTL_337	DENALI_CTL_337 Register	—
0x8030_0548	DDR_MEMC_DENALI_CTL_338	DENALI_CTL_338 Register	—
0x8030_054C	DDR_MEMC_DENALI_CTL_339	DENALI_CTL_339 Register	—
0x8030_0554	DDR_MEMC_DENALI_CTL_341	DENALI_CTL_341 Register	—
0x8030_0558	DDR_MEMC_DENALI_CTL_342	DENALI_CTL_342 Register	—
0x8030_055C	DDR_MEMC_DENALI_CTL_343	DENALI_CTL_343 Register	—
0x8030_0560	DDR_MEMC_DENALI_CTL_344	DENALI_CTL_344 Register	—
0x8030_0564	DDR_MEMC_DENALI_CTL_345	DENALI_CTL_345 Register	—
0x8030_0568	DDR_MEMC_DENALI_CTL_346	DENALI_CTL_346 Register	—
0x8030_056C	DDR_MEMC_DENALI_CTL_347	DENALI_CTL_347 Register	—
0x8030_0570	DDR_MEMC_DENALI_CTL_348	DENALI_CTL_348 Register	—
0x8030_0578	DDR_MEMC_DENALI_CTL_350	DENALI_CTL_350 Register	—
0x8030_057C	DDR_MEMC_DENALI_CTL_351	DENALI_CTL_351 Register	—
0x8030_0580	DDR_MEMC_DENALI_CTL_352	DENALI_CTL_352 Register	—
0x8030_0584	DDR_MEMC_DENALI_CTL_353	DENALI_CTL_353 Register	—
0x8030_0588	DDR_MEMC_DENALI_CTL_354	DENALI_CTL_354 Register	—
0x8030_058C	DDR_MEMC_DENALI_CTL_355	DENALI_CTL_355 Register	—
0x8030_0590	DDR_MEMC_DENALI_CTL_356	DENALI_CTL_356 Register	—
0x8030_05C0	DDR_MEMC_DENALI_CTL_368	DENALI_CTL_368 Register	—
0x8030_05C4	DDR_MEMC_DENALI_CTL_369	DENALI_CTL_369 Register	—
0x8030_05F8	DDR_MEMC_DENALI_CTL_382	DENALI_CTL_382 Register	—
0x8030_05FC	DDR_MEMC_DENALI_CTL_383	DENALI_CTL_383 Register	—
0x8030_0600	DDR_MEMC_DENALI_CTL_384	DENALI_CTL_384 Register	—
0x8030_0604	DDR_MEMC_DENALI_CTL_385	DENALI_CTL_385 Register	—
0x8030_0608	DDR_MEMC_DENALI_CTL_386	DENALI_CTL_386 Register	—
0x8030_060C	DDR_MEMC_DENALI_CTL_387	DENALI_CTL_387 Register	—
0x8030_062C	DDR_MEMC_DENALI_CTL_395	DENALI_CTL_395 Register	—

Table 57.3 DDRSS register map (3 of 3)

Address	Register symbol	Register name	Write protection
0x8030_0630 + 0x10 × n	DDR_MEMC_DENALI_CTL_396+4n	DENALI_CTL_396+4n Register (n = 0 to 15)	—
0x8030_0634 + 0x10 × n	DDR_MEMC_DENALI_CTL_397+4n	DENALI_CTL_397+4n Register (n = 0 to 15)	—
0x8030_0730 + 0x10 × n	DDR_MEMC_DENALI_CTL_460+4n	DENALI_CTL_460+4n Register (n = 0 to 15)	—
0x8030_0734 + 0x10 × n	DDR_MEMC_DENALI_CTL_461+4n	DENALI_CTL_461+4n Register (n = 0 to 15)	—
0x8030_0830 + 0x10 × n	DDR_MEMC_DENALI_CTL_524+4n	DENALI_CTL_524+4n Register (n = 0 to 15)	—
0x8030_0834 + 0x10 × n	DDR_MEMC_DENALI_CTL_525+4n	DENALI_CTL_525+4n Register (n = 0 to 15)	—
0x8030_0930 + 0x10 × n	DDR_MEMC_DENALI_CTL_588+4n	DENALI_CTL_588+4n Register (n = 0 to 15)	—
0x8030_0934 + 0x10 × n	DDR_MEMC_DENALI_CTL_589+4n	DENALI_CTL_589+4n Register (n = 0 to 15)	—
0x8030_0A30 + 0x10 × n	DDR_MEMC_DENALI_CTL_652+4n	DENALI_CTL_652+4n Register (n = 0 to 15)	—
0x8030_0A34 + 0x10 × n	DDR_MEMC_DENALI_CTL_653+4n	DENALI_CTL_653+4n Register (n = 0 to 15)	—
0x8030_0B30	DDR_MEMC_DENALI_CTL_716	DENALI_CTL_716 Register	—
0x8030_0B34	DDR_MEMC_DENALI_CTL_717	DENALI_CTL_717 Register	—
0x8030_0B38	DDR_MEMC_DENALI_CTL_718	DENALI_CTL_718 Register	—
0x8030_0B3C	DDR_MEMC_DENALI_CTL_719	DENALI_CTL_719 Register	—
0x8030_0B40	DDR_MEMC_DENALI_CTL_720	DENALI_CTL_720 Register	—
0x8030_0B44	DDR_MEMC_DENALI_CTL_721	DENALI_CTL_721 Register	—
0x8030_0B48	DDR_MEMC_DENALI_CTL_722	DENALI_CTL_722 Register	—
0x8030_0B4C	DDR_MEMC_DENALI_CTL_723	DENALI_CTL_723 Register	—
0x8030_0B50	DDR_MEMC_DENALI_CTL_724	DENALI_CTL_724 Register	—
0x8030_0B54	DDR_MEMC_DENALI_CTL_725	DENALI_CTL_725 Register	—
0x8030_0B58	DDR_MEMC_DENALI_CTL_726	DENALI_CTL_726 Register	—
0x8030_0B5C	DDR_MEMC_DENALI_CTL_727	DENALI_CTL_727 Register	—
0x8030_0B60	DDR_MEMC_DENALI_CTL_728	DENALI_CTL_728 Register	—
0x8030_0B64	DDR_MEMC_DENALI_CTL_729	DENALI_CTL_729 Register	—
0x8030_0B68	DDR_MEMC_DENALI_CTL_730	DENALI_CTL_730 Register	—
0x8030_0B6C	DDR_MEMC_DENALI_CTL_731	DENALI_CTL_731 Register	—
0x8030_2000	DDR_MEMC_MCAR_CTL	MCAR_CTL Register	—
0x8030_2004	DDR_MEMC_MCAR_MON	MCAR_MON Register	—
0x8030_2008	DDR_MEMC_MCAR_AXCTL0	MCAR_AXCTL0 Register	—
0x8030_200C	DDR_MEMC_MCAR_AXCTL1	MCAR_AXCTL1 Register	—
0x8030_2010	DDR_MEMC_MCAR_AXCTL2	MCAR_AXCTL2 Register	—
0x8030_2014	DDR_MEMC_MCAR_AXCTL3	MCAR_AXCTL3 Register	—
0x8030_2018	DDR_MEMC_MCAR_AXCTL4	MCAR_AXCTL4 Register	—

Table 57.4 DDRSS related system control register

Registers	Module Reset Control Register	Module Stop Control Register	Slave Access Control Register
All the registers	MRCTLM.MRCTLM[25:16]	MSTPCRM.MSTPCRM00	SLVACCCTL8.DDRCTRL_SL

57.3 Register Description

Table 57.5 lists the R/W definitions of registers.

Table 57.5 Register R/W definitions

R/W	Description
RW	A register that can be read and written.
R	A register that can be read. The written value will be ignored.
W	A register that can be written. Reading is invalid.
RW0	A register that can be read and to which 0 can be written. Writing 1 is invalid.
RW1	A register that can be read and to which 1 can be written. Writing 0 is invalid.
R0W	A register that can be read and written. Whenever it is read, the value read is 0.
R1W	A register that can be read and written. Whenever it is read, the value read is 1.
R0W0	A register that can be read and to which 0 can be written. Writing 1 is invalid. Whenever it is read, the value read is 0.
R0W1	A register that can be read and to which 1 can be written. Writing 0 is invalid. Whenever it is read, the value read is 0.
R1W0	A register that can be read and to which 0 can be written. Writing 1 is invalid. Whenever it is read, the value read is 1.
R1W1	A register that can be read and to which 1 can be written. Writing 0 is invalid. Whenever it is read, the value read is 1.
W0	A register to which 0 can be written. Writing 1 and reading are invalid.
W1	A register to which 1 can be written. Writing 0 and reading are invalid.

57.3.1 DDR_MEMC_DENALI_CTL_00 : DENALI_CTL_00 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	start
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	start	Triggers controller initialization and normal operation. Prior to setting this parameter to 1, the controller will not issue any commands to the DRAM memories or respond to any signal activity except for reading and writing parameters and accepting traffic that the customer may send into the Controller core queues. Once this parameter is set to 1, the controller will respond to inputs from the CPU and begin to process memory access commands. Note that resetting this parameter to 0 will not shutoff traffic. Note: Until the initialization complete interrupt (bit [1]) will be set to 1 in the int_status_init parameter and the dfi_init_complete signal is asserted from the PHY, commands will not be accepted into the Controller core command queue. Note: It is imperative that the controller is driving the dfi_reset_n_pZ and dfi_cke_pZ signals before this parameter is asserted. 0: Controller is not in active mode. 1: Initiate active mode for the controller.	R/W
31:1	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.2 DDR_MEMC_DENALI_CTL_219 : DENALI_CTL_219 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x036C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	ADDR_SPACE						—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BIST_GO	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	The read values are undefined.	R/W
8	BIST_GO	Initiate a BIST operation if the bist_data_check parameter is set to 1. Set to 1 to trigger. This parameter is write-only and will always read back as 0.	R/W
23:9	—	The read values are undefined.	R/W
29:24	ADDR_SPACE	Sets the number of address bits to check during BIST operation. Used in BIST data checking to define the address space in bytes from 0 to 2^{addr_space} that the BIST logic will check. As an example, if the addr_space parameter was programmed to 0x1c, then the BIST logic would check 2^{28} bytes = 256 MB.	R/W
31:30	—	The read values are undefined.	R/W

57.3.3 DDR_MEMC_DENALI_CTL_220 : DENALI_CTL_220 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0370

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BIST_DATA_CHECK
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BIST_DATA_CHECK	Enable data checking with BIST operation. Set to 1 to enable.	R/W
31:1	—	The read values are undefined.	R/W

57.3.4 DDR_MEMC_DENALI_CTL_221 : DENALI_CTL_221 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0374

Bit position:	31	0
Bit field:	BIST_START_ADDRESS[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	BIST_START_ADDRESS[31:0]	Start BIST checking at this address. (bit[31:0] of address)	R/W

57.3.5 DDR_MEMC_DENALI_CTL_222 : DENALI_CTL_222 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0378

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BIST_START_ADDRESS32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BIST_START_ADDRESS32	Start BIST checking at this address. (bit32 of address)	R/W
31:1	—	The read values are undefined.	R/W

57.3.6 DDR_MEMC_DENALI_CTL_225 : DENALI_CTL_225 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0384

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

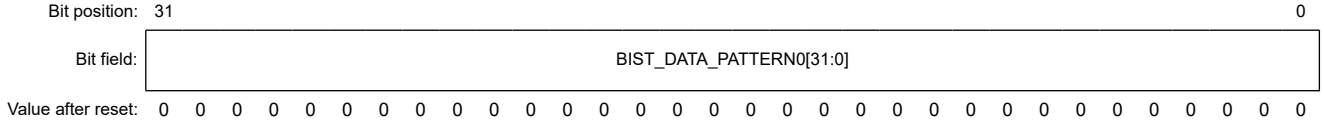
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BIST_TEST_MODE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	BIST_TEST_MODE[2:0]	Sets the BIST test mode. For memory initialization function, set 100b.	R/W
31:3	—	The read values are undefined.	R/W

57.3.7 DDR_MEMC_DENALI_CTL_226 : DENALI_CTL_226 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0388

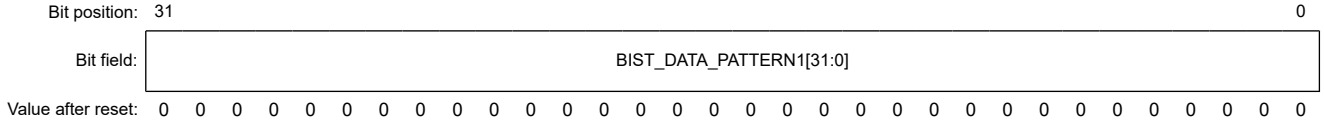


Bit	Symbol	Function	R/W
31:0	BIST_DATA_PATTERN0[31:0]	Defines the data pattern to be used for memory initialization. (Bit[31:0] of 128-bit data) Only data corresponding to active portion of core word will be used while inactive portion will be ignored.	R/W

57.3.8 DDR_MEMC_DENALI_CTL_227 : DENALI_CTL_227 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x038C

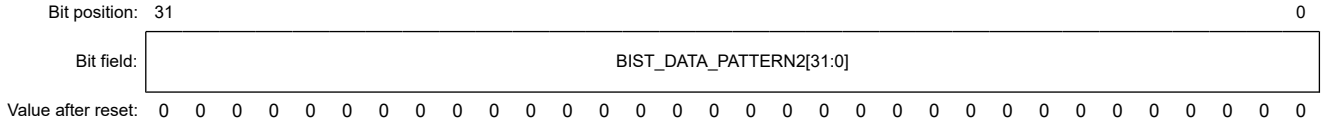


Bit	Symbol	Function	R/W
31:0	BIST_DATA_PATTERN1[31:0]	Defines the data pattern to be used for memory initialization. (Bit[63:32] of 128-bit data) Only data corresponding to active portion of core word will be used while inactive portion will be ignored.	R/W

57.3.9 DDR_MEMC_DENALI_CTL_228 : DENALI_CTL_228 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0390

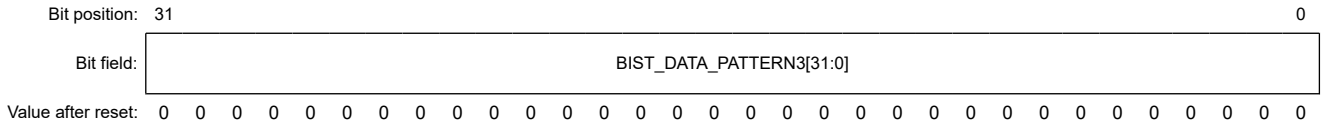


Bit	Symbol	Function	R/W
31:0	BIST_DATA_PATTERN2[31:0]	Defines the data pattern to be used for memory initialization. (Bit[95:64] of 128-bit data) Only data corresponding to active portion of core word will be used while inactive portion will be ignored.	R/W

57.3.10 DDR_MEMC_DENALI_CTL_229 : DENALI_CTL_229 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0394



Bit	Symbol	Function	R/W
31:0	BIST_DATA_PATTERN3[31:0]	Defines the data pattern to be used for memory initialization. (Bit[127:96] of 128-bit data) Only data corresponding to active portion of core word will be used while inactive portion will be ignored.	R/W

57.3.11 DDR_MEMC_DENALI_CTL_231 : DENALI_CTL_231 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x039C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ECC_ENABLE[1:0]		—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	—	The read values are undefined.	R/W
25:24	ECC_ENABLE[1:0]	Controls ECC error reporting (single-bit and double-bit errors) and correcting (single-bit errors). 0 0: ECC disabled, no ECC error reporting, no ECC error correcting 0 1: ECC enabled, no ECC error reporting, no ECC error correcting 1 0: ECC enabled, ECC error reporting, no ECC error correcting 1 1: ECC enabled, ECC error reporting, ECC error correcting	R/W
31:26	—	The read values are undefined.	R/W

57.3.12 DDR_MEMC_DENALI_CTL_232 : DENALI_CTL_232 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ECC_DISABLE_W_UC_ERR	XOR_CHECK_BITS[15:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	XOR_CHECK_BITS[15:0]								—	—	—	—	—	—	—	FWC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FWC	Force a write check. Xor the XOR_CHECK_BITS parameter with the ECC code and write to memory. Set to 1 to trigger.	R/W
7:1	—	The read values are undefined.	R/W
23:8	XOR_CHECK_BITS[15:0]	Value to xor with generated ECC codes for forced write check	R/W
24	ECC_DISABLE_W_UC_ERR	Controls auto-corruption of ECC when un-correctable errors occur in R/M/W operations. Set to 1 to disable corruption.	R/W

Bit	Symbol	Function	R/W
31:25	—	The read values are undefined.	R/W

57.3.13 DDR_MEMC_DENALI_CTL_233 : DENALI_CTL_233 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ECC_READ_CACHING_EN	—	—	—	—	—	INLINE_ECC_BANK_OFFSET[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	INLINE_ECC_SAME_PAGE	—	—	—	—	—	—	—	ECC_WRITEBACK_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECC_WRITEBACK_EN	ECC writeback will occur on detection of single bit errors for reads. The ECC_ENABLE parameter must be programmed to 3 for this to take any effect. Note that no writebacks will be issued during BIST.	R/W
7:1	—	The read values are undefined.	R/W
8	INLINE_ECC_SAME_PAGE	Inline ECC Configurations will store the ECC in top 1/8th of page as data; otherwise it will store ECC in top 1/8th of rows.	R/W
15:9	—	The read values are undefined.	R/W
18:16	INLINE_ECC_BANK_OFFSET[2:0]	Inline ECC Bank Offset defines the bank shift between data and ECC commands associated with the same sequence; the bank is offset to prevent inefficiencies due to opening a closing the pages on the same bank during the transition between data and ECC commands.	R/W
23:19	—	The read values are undefined.	R/W
24	ECC_READ_CACHING_EN	Allows ECC read data already in one of the ECC buffers to be used when possible in place of issuing an ECC read command to memory. This implies that some ECC read commands will be dropped in the command queue when it can pull the read data from the buffer instead.	R/W
31:25	—	The read values are undefined.	R/W

57.3.14 DDR_MEMC_DENALI_CTL_234 : DENALI_CTL_234 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03A8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC_WRITE_COMBINING_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECC_WRITE_COMBINING_EN	Allows ECC write data within a given ECC buffer to be combined across commands so that in certain cases where we see multiple ECC writes to the same ECC address, the controller may end up only issuing one final ECC write command to memory.	R/W
31:1	—	The read values are undefined.	R/W

57.3.15 DDR_MEMC_DENALI_CTL_235 : DENALI_CTL_235 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03AC

Bit position:	31																													0
Bit field:	ECC_U_ADDR[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ECC_U_ADDR[31:0]	Address of uncorrectable ECC event. (bit[31:0] of address)	R

57.3.16 DDR_MEMC_DENALI_CTL_236 : DENALI_CTL_236 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ECC_U_SYND[7:0]								—	—	—	—	—	—	—	ECC_U_ADDR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECC_U_ADDR32	Address of uncorrectable ECC event. (bit32 of address)	R
7:1	—	The read values are undefined.	R

Bit	Symbol	Function	R/W
15:8	ECC_U_SYND[7:0]	Syndrome for uncorrectable ECC event.	R
31:16	—	The read values are undefined.	R

57.3.17 DDR_MEMC_DENALI_CTL_237 : DENALI_CTL_237 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03B4

Bit position:	31																												0
Bit field:	ECC_U_DATA0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ECC_U_DATA0[31:0]	Data associated with uncorrectable ECC event. (bit[31:0] of data)	R

57.3.18 DDR_MEMC_DENALI_CTL_238 : DENALI_CTL_238 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03B8

Bit position:	31																												0
Bit field:	ECC_U_DATA1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ECC_U_DATA1[31:0]	Data associated with uncorrectable ECC event. (bit[63:32] of data)	R

57.3.19 DDR_MEMC_DENALI_CTL_239 : DENALI_CTL_239 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03BC

Bit position:	31																												0
Bit field:	ECC_C_ADDR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ECC_C_ADDR[31:0]	Address of correctable ECC event. (bit[31:0] of address)	R

57.3.20 DDR_MEMC_DENALI_CTL_240 : DENALI_CTL_240 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ECC_C_SYND[7:0]								—	—	—	—	—	—	—	ECC_C_ADDR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECC_C_ADDR32	Address of correctable ECC event. (bit32 of address)	R
7:1	—	The read values are undefined.	R
15:8	ECC_C_SYND[7:0]	Syndrome for correctable ECC event.	R
31:16	—	The read values are undefined.	R

57.3.21 DDR_MEMC_DENALI_CTL_241 : DENALI_CTL_241 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03C4

Bit position:	31	0
Bit field:	ECC_C_DATA0[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	ECC_C_DATA0[31:0]	Data associated with correctable ECC event. (bit[31:0] of data)	R

57.3.22 DDR_MEMC_DENALI_CTL_242 : DENALI_CTL_242 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03C8

Bit position:	31	0
Bit field:	ECC_C_DATA1[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	ECC_C_DATA1[31:0]	Data associated with correctable ECC event. (bit[63:32] of data)	R

57.3.23 DDR_MEMC_DENALI_CTL_244 : DENALI_CTL_244 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03D0

Bit position:	31																												0
Bit field:	RMODW_ECC_U_ADDR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RMODW_ECC_U_ADDR[31:0]	Address of uncorrectable read-modify-write ECC event. (bit[31:0] of address)	R

57.3.24 DDR_MEMC_DENALI_CTL_245 : DENALI_CTL_245 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03D4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RMODW_ECC_U_SYND[7:0]								—	—	—	—	—	—	—	RMODW_ECC_U_ADDR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMODW_ECC_U_ADDR32	Address of uncorrectable read-modify-write ECC event. (bit32 of address)	R
7:1	—	The read values are undefined.	R
15:8	RMODW_ECC_U_SYND[7:0]	Syndrome for uncorrectable read-modify-write ECC event.	R
31:16	—	The read values are undefined.	R

57.3.25 DDR_MEMC_DENALI_CTL_246 : DENALI_CTL_246 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03D8

Bit position:	31																												0
Bit field:	RMODW_ECC_U_DATA0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RMODW_ECC_U_DATA0[31:0]	Data associated with uncorrectable read-modify-write ECC event. (bit[31:0] of data)	R

57.3.26 DDR_MEMC_DENALI_CTL_247 : DENALI_CTL_247 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03DC

Bit position:	31																												0
Bit field:	RMODW_ECC_U_DATA1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RMODW_ECC_U_DATA1[31:0]	Data associated with uncorrectable read-modify-write ECC event. (bit[63:32] of data)	R

57.3.27 DDR_MEMC_DENALI_CTL_248 : DENALI_CTL_248 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03E0

Bit position:	31																												0
Bit field:	RMODW_ECC_C_ADDR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RMODW_ECC_C_ADDR[31:0]	Address of correctable read-modify-write ECC event. (bit[31:0] of address)	R

57.3.28 DDR_MEMC_DENALI_CTL_249 : DENALI_CTL_249 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03E4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

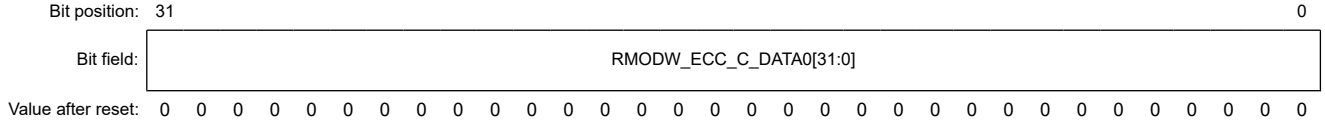
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RMODW_ECC_C_SYND[7:0]								—	—	—	—	—	—	—	RMODW_ECC_C_ADDR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMODW_ECC_C_ADDR32	Address of correctable read-modify-write ECC event. (bit32 of address)	R
7:1	—	The read values are undefined.	R
15:8	RMODW_ECC_C_SYND[7:0]	Syndrome for correctable read-modify-write ECC event.	R
31:16	—	The read values are undefined.	R

57.3.29 DDR_MEMC_DENALI_CTL_250 : DENALI_CTL_250 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03E8

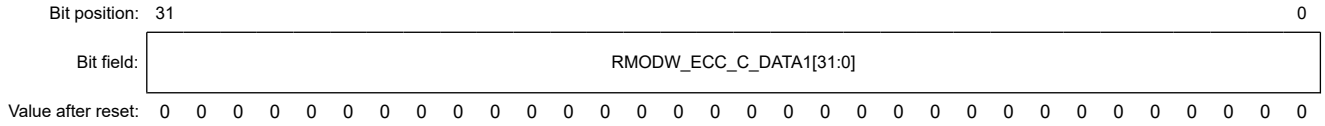


Bit	Symbol	Function	R/W
31:0	RMODW_ECC_C_DATA0[31:0]	Data associated with correctable read-modify-write ECC event. (bit[31:0] of data)	R

57.3.30 DDR_MEMC_DENALI_CTL_251 : DENALI_CTL_251 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x03EC

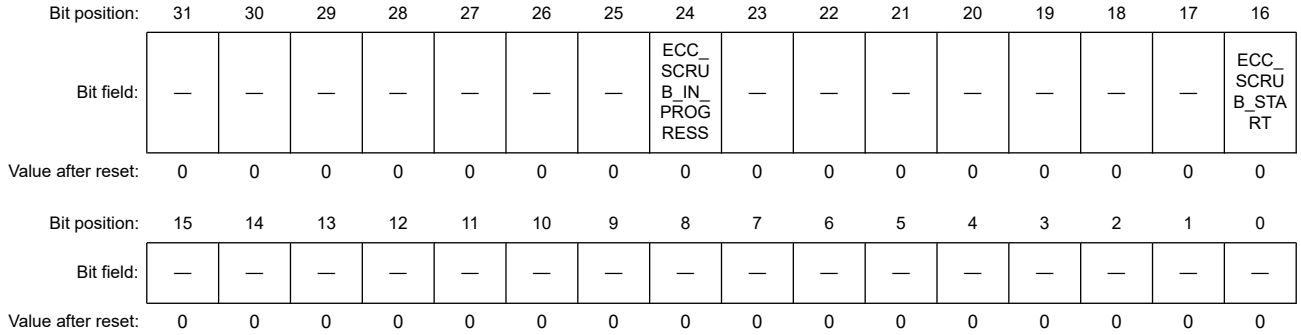


Bit	Symbol	Function	R/W
31:0	RMODW_ECC_C_DATA1[31:0]	Data associated with correctable read-modify-write ECC event. (bit[63:32] of data)	R

57.3.31 DDR_MEMC_DENALI_CTL_269 : DENALI_CTL_269 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0434



Bit	Symbol	Function	R/W
15:0	—	The read values are undefined.	R/W
16	ECC_SCRUB_START	ECC scrubbing control. Set to 1 to kick start the scrubbing operation.	W
23:17	—	The read values are undefined.	R/W
24	ECC_SCRUB_IN_PROGRESS	Reports the scrubbing operation status. A value of 1 indicates that the controller is in the process of performing a scrubbing operation.	R
31:25	—	The read values are undefined.	R/W

57.3.32 DDR_MEMC_DENALI_CTL_270 : DENALI_CTL_270 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0438

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC_SCRUB_MODE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	ECC_SCRUB_LEN[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	ECC_SCRUB_LEN[12:0]	Defines the length of the ECC scrubbing read command that the controller will issue.	R/W
15:13	—	The read values are undefined.	R/W
16	ECC_SCRUB_MODE	Defines how often ECC scrubbing operations will occur. Clear to 0 to scrub at regular intervals as dictated by the ECC_SCRUB_INTERVAL parameter, or set to 1 to scrub only when the controller is idle.	R/W
31:17	—	The read values are undefined.	R/W

57.3.33 DDR_MEMC_DENALI_CTL_271 : DENALI_CTL_271 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x043C

Bit position:	31	0
Bit field:	ECC_SCRUB_IDLE_CNT[15:0]	ECC_SCRUB_INTERVAL[15:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	ECC_SCRUB_INTERVAL[15:0]	The minimum interval between two ECC scrubbing commands in number of controller clock cycles. Valid when the ECC_SCRUB_MODE parameter is cleared to 0.	R/W
31:16	ECC_SCRUB_IDLE_CNT[15:0]	The number of controller clock cycles that the scrubbing engine will wait in controller idle state before starting scrubbing operations. Valid when the ECC_SCRUB_MODE parameter is set to 1.	R/W

57.3.34 DDR_MEMC_DENALI_CTL_272 : DENALI_CTL_272 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0440

Bit position:	31	0
Bit field:	ECC_SCRUB_START_ADDR[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	ECC_SCRUB_START_ADDR[31:0]	The starting address from where scrubbing operations will begin. (bit[31:0] of address)	R/W

57.3.35 DDR_MEMC_DENALI_CTL_273 : DENALI_CTL_273 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0444

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC_SCRUB_START_ADDR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECC_SCRUB_START_ADDR32	The starting address from where scrubbing operations will begin. (bit32 of address)	R/W
31:1	—	The read values are undefined.	R/W

57.3.36 DDR_MEMC_DENALI_CTL_274 : DENALI_CTL_274 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0448

Bit position:	31	0
Bit field:	ECC_SCRUB_END_ADDR[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	ECC_SCRUB_END_ADDR[31:0]	The end address where scrubbing operations will wrap around to the start address. (bit[31:0] of address) If this parameter is cleared to 0, the maximum physical address will be considered as the end address.	R/W

57.3.37 DDR_MEMC_DENALI_CTL_275 : DENALI_CTL_275 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x044C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC_SCRUB_END_ADDR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECC_SCRUB_END_ADDR32	The end address where scrubbing operations will wrap around to the start address. (bit32 of address) If this parameter is cleared to 0, the maximum physical address will be considered as the end address.	R/W
31:1	—	The read values are undefined.	R/W

57.3.38 DDR_MEMC_DENALI_CTL_318 : DENALI_CTL_318 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x04F8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	device3_byte0_cs0[3:0]				—	—	—	—	device2_byte0_cs0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	device1_byte0_cs0[3:0]				—	—	—	—	device0_byte0_cs0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	device0_byte0_cs0[3:0]	Defines the byte location of byte0 in the memory data path for device 0 for chip select 0. Used for MRRs to identify where data will be returned. Bit [0] correlates to byte [0] of the memory data path, bit [1] correlates to byte [1] of the memory data path, etc.*1 For each bit: 0: This bit does NOT correlate to byte 0 for device 0. 1: This bit DOES correlate to byte 0 for device 0.	R/W
7:4	—	Reserved*2	R/W
11:8	device1_byte0_cs0[3:0]	Defines the byte location of byte0 in the memory data path for device 1 for chip select 0. Used for MRRs to identify where data will be returned. Bit [0] correlates to byte [0] of the memory data path, bit [1] correlates to byte [1] of the memory data path, etc.*1 For each bit: 0: This bit does NOT correlate to byte 0 for device 1. 1: This bit DOES correlate to byte 0 for device 1.	R/W
15:12	—	Reserved*2	R/W
19:16	device2_byte0_cs0[3:0]	Defines the byte location of byte0 in the memory data path for device 2 for chip select 0. Used for MRRs to identify where data will be returned. Bit [0] correlates to byte [0] of the memory data path, bit [1] correlates to byte [1] of the memory data path, etc.*1 For each bit: 0: This bit does NOT correlate to byte 0 for device 2. 1: This bit DOES correlate to byte 0 for device 2.	R/W
23:20	—	Reserved*2	R/W
27:24	device3_byte0_cs0[3:0]	Defines the byte location of byte0 in the memory data path for device 3 for chip select 0. Used for MRRs to identify where data will be returned. Bit [0] correlates to byte [0] of the memory data path, bit [1] correlates to byte [1] of the memory data path, etc.*1 For each bit: 0: This bit does NOT correlate to byte 0 for device 3. 1: This bit DOES correlate to byte 0 for device 3.	R/W
31:28	—	Reserved*2	R/W

Note 1. Only 1 bit may be set to 1 at any time to indicate which bit holds the byte 0 information. If this device is not relevant, no bits should be set to 1.

Note 2. The write value should follow those generated by DDR parameter generation tool.

57.3.39 DDR_MEMC_DENALI_CTL_319 : DENALI_CTL_319 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x04FC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	device3_byte0_cs1[3:0]				—	—	—	—	device2_byte0_cs1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	device1_byte0_cs1[3:0]				—	—	—	—	device0_byte0_cs1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	device0_byte0_cs1[3:0]	Defines the byte location of byte0 in the memory data path for device 0 for chip select 1. Used for MRRs to identify where data will be returned. Bit [0] correlates to byte [0] of the memory data path, bit [1] correlates to byte [1] of the memory data path, etc.*1 For each bit: 0: This bit does NOT correlate to byte 0 for device 0. 1: This bit DOES correlate to byte 0 for device 0.	R/W
7:4	—	Reserved*2	R/W
11:8	device1_byte0_cs1[3:0]	Defines the byte location of byte0 in the memory data path for device 1 for chip select 1. Used for MRRs to identify where data will be returned. Bit [0] correlates to byte [0] of the memory data path, bit [1] correlates to byte [1] of the memory data path, etc.*1 For each bit: 0: This bit does NOT correlate to byte 0 for device 1. 1: This bit DOES correlate to byte 0 for device 1.	R/W
15:12	—	Reserved*2	R/W
19:16	device2_byte0_cs1[3:0]	Defines the byte location of byte0 in the memory data path for device 2 for chip select 1. Used for MRRs to identify where data will be returned. Bit [0] correlates to byte [0] of the memory data path, bit [1] correlates to byte [1] of the memory data path, etc.*1 For each bit: 0: This bit does NOT correlate to byte 0 for device 2. 1: This bit DOES correlate to byte 0 for device 2.	R/W
23:20	—	Reserved*2	R/W
27:24	device3_byte0_cs1[3:0]	Defines the byte location of byte0 in the memory data path for device 3 for chip select 1. Used for MRRs to identify where data will be returned. Bit [0] correlates to byte [0] of the memory data path, bit [1] correlates to byte [1] of the memory data path, etc.*1 For each bit: 0: This bit does NOT correlate to byte 0 for device 3. 1: This bit DOES correlate to byte 0 for device 3.	R/W
31:28	—	Reserved*2	R/W

Note 1. Only 1 bit may be set to 1 at any time to indicate which bit holds the byte 0 information. If this device is not relevant, no bits should be set to 1.

Note 2. The write value should follow those generated by DDR parameter generation tool.

57.3.40 DDR_MEMC_DENALI_CTL_326 : DENALI_CTL_326 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0518

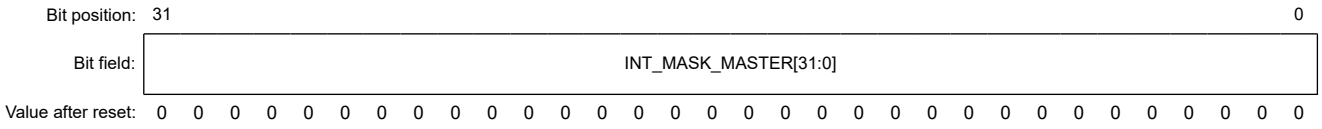
Bit position:	31																													0
Bit field:	INT_STATUS_MASTER[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	INT_STATUS_MASTER[31:0]	Master status reporting register for interrupt status groups	R

57.3.41 DDR_MEMC_DENALI_CTL_327 : DENALI_CTL_327 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x051C

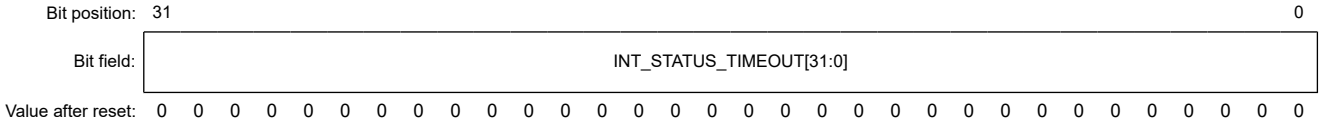


Bit	Symbol	Function	R/W
31:0	INT_MASK_MASTER[31:0]	Master mask register for interrupt status groups	W

57.3.42 DDR_MEMC_DENALI_CTL_328 : DENALI_CTL_328 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0520

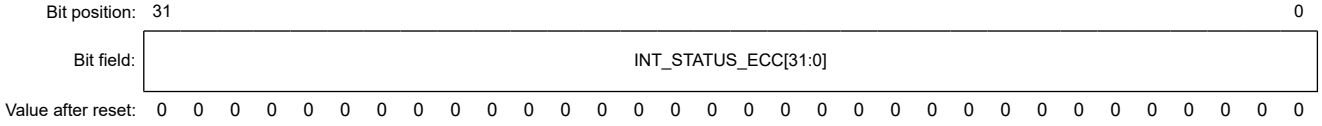


Bit	Symbol	Function	R/W
31:0	INT_STATUS_TIMEOUT[31:0]	Status of interrupts in the controller related to Timeout monitors	R

57.3.43 DDR_MEMC_DENALI_CTL_329 : DENALI_CTL_329 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0524

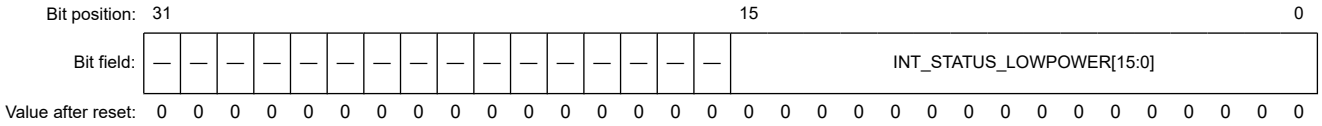


Bit	Symbol	Function	R/W
31:0	INT_STATUS_ECC[31:0]	Status of interrupts in the controller related to ECC	R

57.3.44 DDR_MEMC_DENALI_CTL_330 : DENALI_CTL_330 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0528



Bit	Symbol	Function	R/W
15:0	INT_STATUS_LOWPOWER[15:0]	Status of interrupts in the controller related to Low Power	R
31:16	—	The read values are undefined.	R

57.3.45 DDR_MEMC_DENALI_CTL_332 : DENALI_CTL_332 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0530

Bit position:	31																												0
Bit field:	INT_STATUS_TRAINING[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	INT_STATUS_TRAINING[31:0]	Status of interrupts in the controller related to Training/Calibration	R

57.3.46 DDR_MEMC_DENALI_CTL_333 : DENALI_CTL_333 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0534

Bit position:	31																												0
Bit field:	INT_STATUS_USERIF[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	INT_STATUS_USERIF[31:0]	Status of interrupts in the controller related to CPU to Controller Interface	R

57.3.47 DDR_MEMC_DENALI_CTL_334 : DENALI_CTL_334 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0538

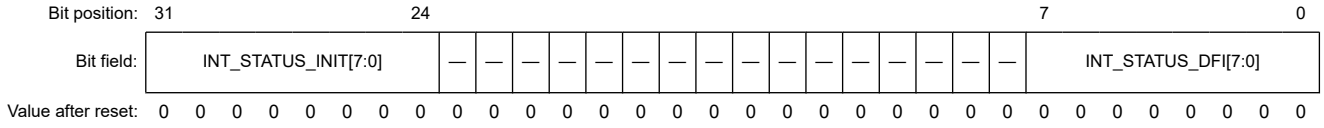
Bit position:	31								23							16	15												0
Bit field:	—	—	—	—	—	—	—	—	INT_STATUS_BIST[7:0]								INT_STATUS_MISC[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	INT_STATUS_MISC[15:0]	Status of interrupts in the controller related to Miscellaneous features	R
23:16	INT_STATUS_BIST[7:0]	Status of interrupts in the controller related to BIST	R
31:24	—	The read values are undefined.	R

57.3.48 DDR_MEMC_DENALI_CTL_335 : DENALI_CTL_335 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x053C

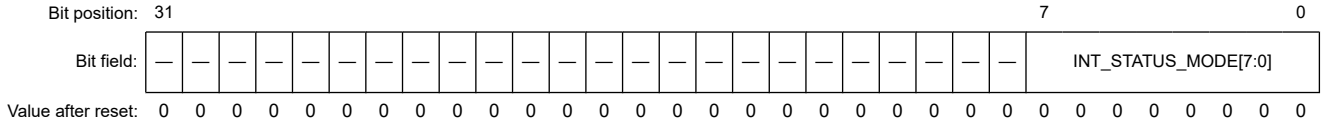


Bit	Symbol	Function	R/W
7:0	INT_STATUS_DFI[7:0]	Status of interrupts in the controller related to DFI	R
23:8	—	The read values are undefined.	R
31:24	INT_STATUS_INIT[7:0]	Status of interrupts in the controller related to Initialization	R

57.3.49 DDR_MEMC_DENALI_CTL_336 : DENALI_CTL_336 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0540

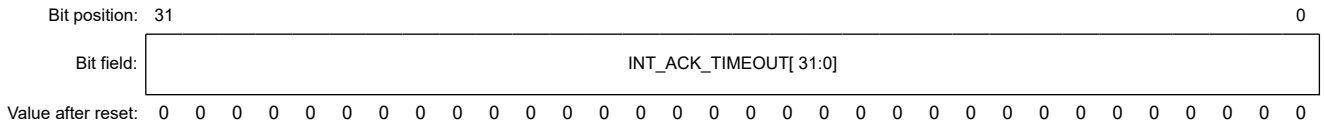


Bit	Symbol	Function	R/W
7:0	INT_STATUS_MODE[7:0]	Status of interrupts in the controller related to Memory Mode Settings	R
31:8	—	The read values are undefined.	R

57.3.50 DDR_MEMC_DENALI_CTL_337 : DENALI_CTL_337 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0544

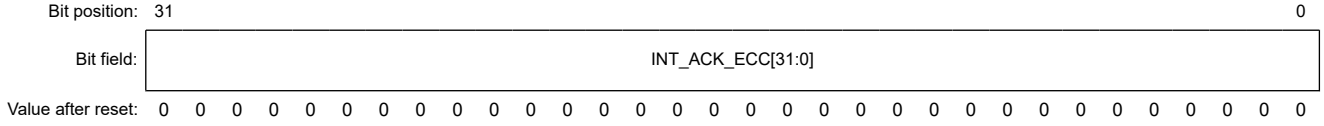


Bit	Symbol	Function	R/W
31:0	INT_ACK_TIMEOUT[31:0]	Clear status of the INT_STATUS_TIMEOUT parameter	W

57.3.51 DDR_MEMC_DENALI_CTL_338 : DENALI_CTL_338 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0548

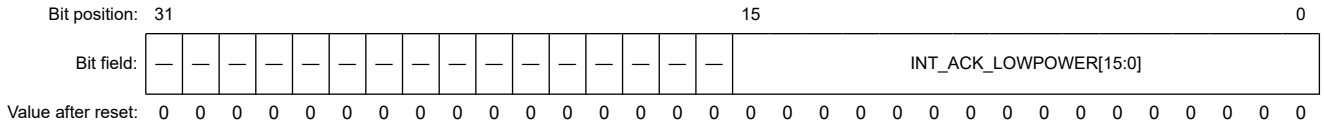


Bit	Symbol	Function	R/W
31:0	INT_ACK_ECC[31:0]	Clear status of the INT_STATUS_ECC parameter	W

57.3.52 DDR_MEMC_DENALI_CTL_339 : DENALI_CTL_339 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x054C

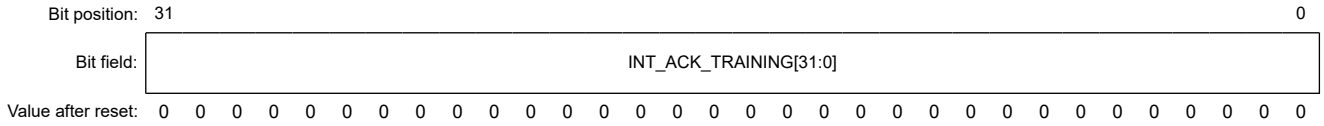


Bit	Symbol	Function	R/W
15:0	INT_ACK_LOWPOWER[15:0]	Clear status of the INT_STATUS_LOWPOWER parameter	W
31:16	—	The read values are undefined.	W

57.3.53 DDR_MEMC_DENALI_CTL_341 : DENALI_CTL_341 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0554

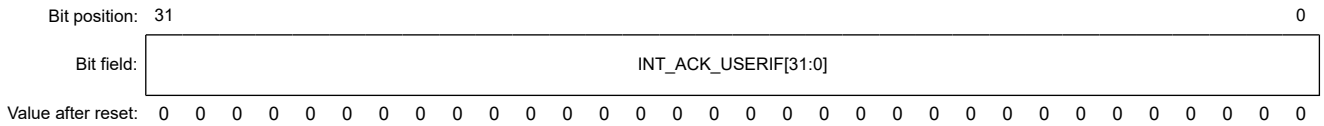


Bit	Symbol	Function	R/W
31:0	INT_ACK_TRAINING[31:0]	Clear status of the INT_STATUS_TRAINING parameter	W

57.3.54 DDR_MEMC_DENALI_CTL_342 : DENALI_CTL_342 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0558

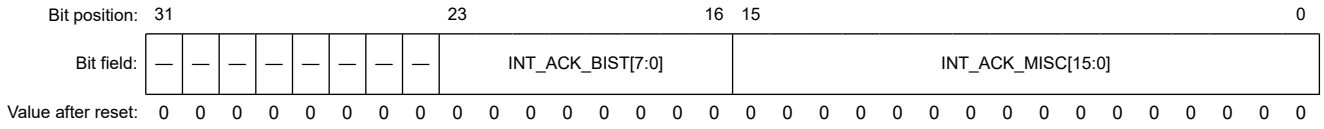


Bit	Symbol	Function	R/W
31:0	INT_ACK_USERIF[31:0]	Clear status of the INT_STATUS_USERIF parameter	W

57.3.55 DDR_MEMC_DENALI_CTL_343 : DENALI_CTL_343 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x055C

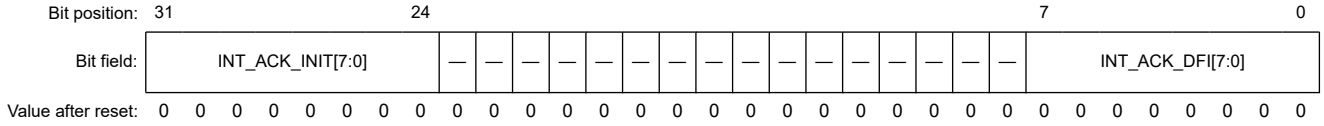


Bit	Symbol	Function	R/W
15:0	INT_ACK_MISC[15:0]	Clear status of the INT_STATUS_MISC parameter	W
23:16	INT_ACK_BIST[7:0]	Clear status of the INT_STATUS_BIST parameter	W
31:24	—	The read values are undefined.	W

57.3.56 DDR_MEMC_DENALI_CTL_344 : DENALI_CTL_344 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0560

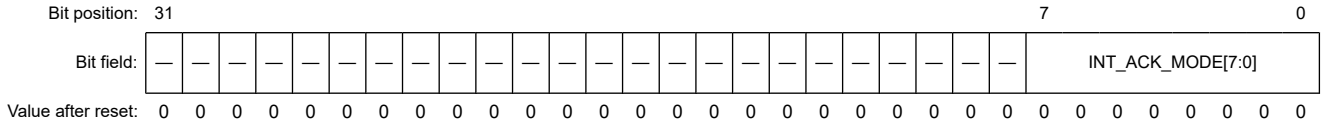


Bit	Symbol	Function	R/W
7:0	INT_ACK_DFI[7:0]	Clear status of the INT_STATUS_DFI parameter	W
23:8	—	The read values are undefined.	W
31:24	INT_ACK_INIT[7:0]	Clear status of the INT_STATUS_INIT parameter	W

57.3.57 DDR_MEMC_DENALI_CTL_345 : DENALI_CTL_345 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0564

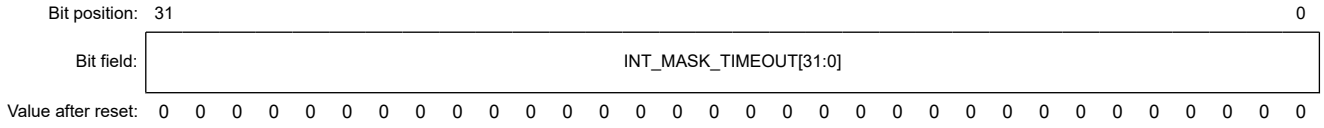


Bit	Symbol	Function	R/W
7:0	INT_ACK_MODE[7:0]	Clear status of the INT_STATUS_MODE parameter	W
31:8	—	The read values are undefined.	W

57.3.58 DDR_MEMC_DENALI_CTL_346 : DENALI_CTL_346 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0568

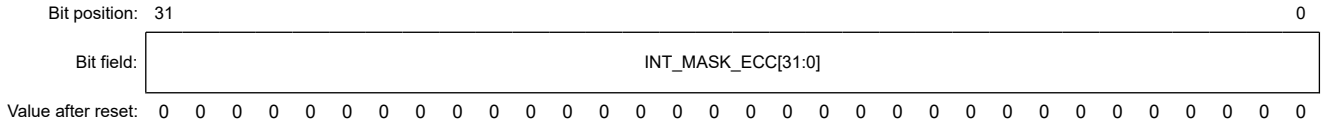


Bit	Symbol	Function	R/W
31:0	INT_MASK_TIMEOUT[31:0]	Mask for the DDRC_INT signal from the INT_MASK_TIMEOUT parameter	R/W

57.3.59 DDR_MEMC_DENALI_CTL_347 : DENALI_CTL_347 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x056C

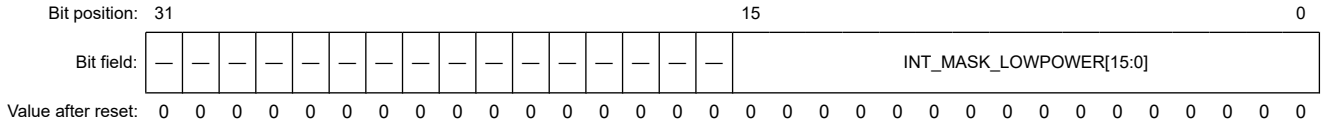


Bit	Symbol	Function	R/W
31:0	INT_MASK_ECC[31:0]	Mask for the DDRC_INT signal from the INT_MASK_ECC parameter	R/W

57.3.60 DDR_MEMC_DENALI_CTL_348 : DENALI_CTL_348 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0570

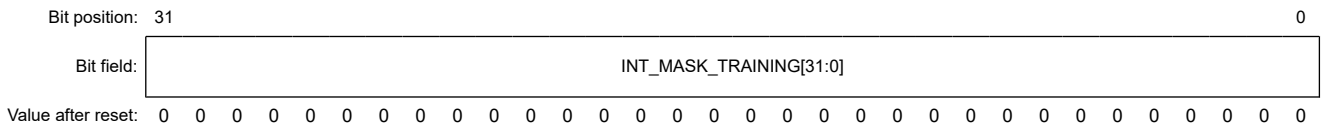


Bit	Symbol	Function	R/W
15:0	INT_MASK_LOWPOWER[15:0]	Mask for the DDRC_INT signal from the INT_MASK_LOWPOWER parameter	R/W
31:16	—	The read values are undefined.	R/W

57.3.61 DDR_MEMC_DENALI_CTL_350 : DENALI_CTL_350 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0578

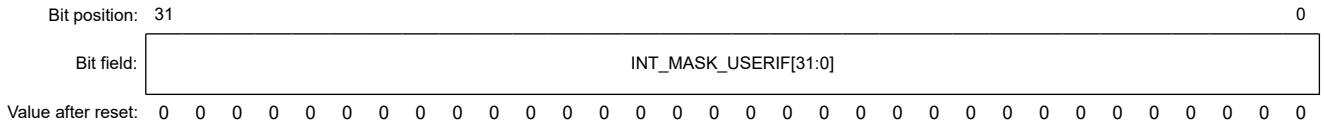


Bit	Symbol	Function	R/W
31:0	INT_MASK_TRAINING[31:0]	Mask for the DDRC_INT signal from the INT_MASK_TRAINING parameter	R/W

57.3.62 DDR_MEMC_DENALI_CTL_351 : DENALI_CTL_351 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x057C

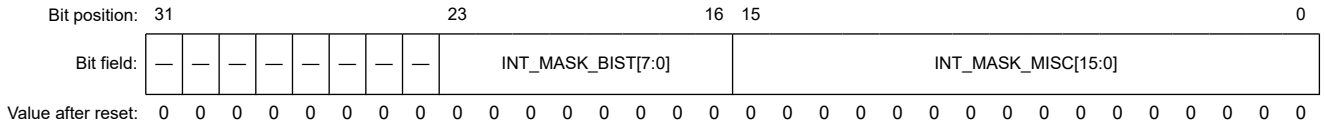


Bit	Symbol	Function	R/W
31:0	INT_MASK_USERIF[31:0]	Mask for the DDRC_INT signal from the INT_MASK_USERIF parameter	R/W

57.3.63 DDR_MEMC_DENALI_CTL_352 : DENALI_CTL_352 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0580

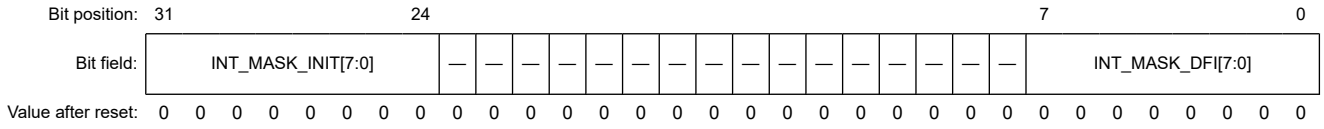


Bit	Symbol	Function	R/W
15:0	INT_MASK_MISC[15:0]	Mask for the DDRC_INT signal from the INT_MASK_MISC parameter	R/W
23:16	INT_MASK_BIST[7:0]	Mask for the DDRC_INT signal from the INT_MASK_BIST parameter	R/W
31:24	—	The read values are undefined.	R/W

57.3.64 DDR_MEMC_DENALI_CTL_353 : DENALI_CTL_353 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0584

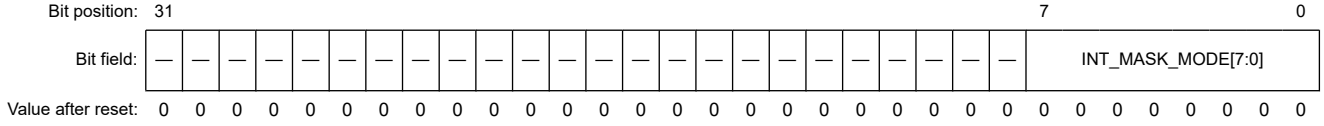


Bit	Symbol	Function	R/W
7:0	INT_MASK_DFI[7:0]	Mask for the DDRC_INT signal from the INT_MASK_DFI parameter	R/W
23:8	—	The read values are undefined.	R/W
31:24	INT_MASK_INIT[7:0]	Mask for the DDRC_INT signal from the INT_MASK_INIT parameter	R/W

57.3.65 DDR_MEMC_DENALI_CTL_354 : DENALI_CTL_354 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0588

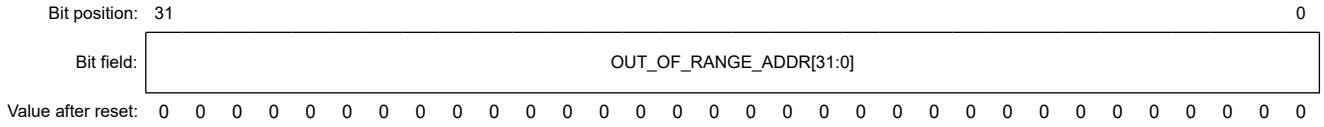


Bit	Symbol	Function	R/W
7:0	INT_MASK_MODE[7:0]	Mask for the DDRC_INT signal from the INT_MASK_MODE parameter	R/W
31:8	—	The read values are undefined.	R/W

57.3.66 DDR_MEMC_DENALI_CTL_355 : DENALI_CTL_355 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x058C

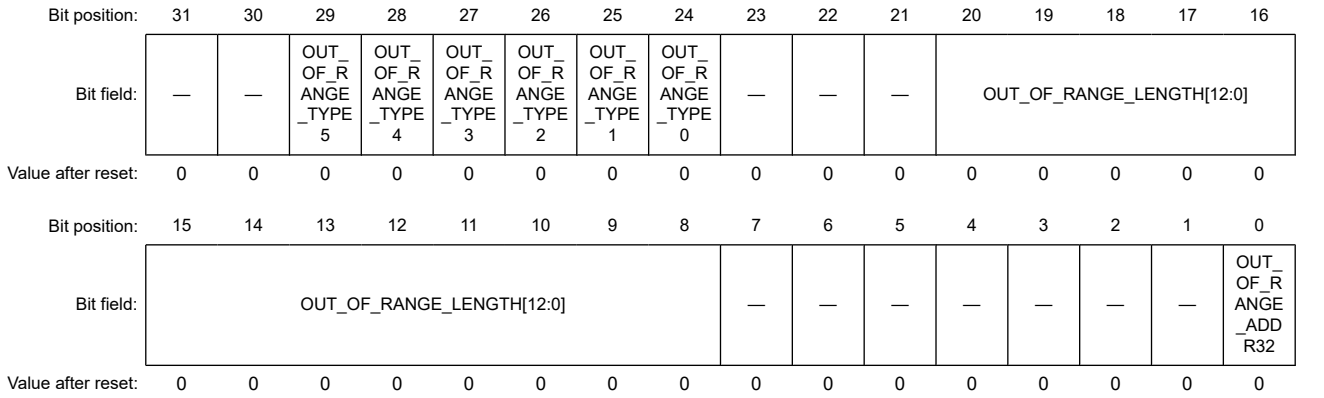


Bit	Symbol	Function	R/W
31:0	OUT_OF_RANGE_ADDR[31:0]	Address of command that caused an out-of-range interrupt (address of bits[31:0])	R

57.3.67 DDR_MEMC_DENALI_CTL_356 : DENALI_CTL_356 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0590



Bit	Symbol	Function	R/W
0	OUT_OF_RANGE_ADDR32	Address of command that caused an out-of-range interrupt (address of bit 32)	R
7:1	—	The read values are undefined.	R
20:8	OUT_OF_RANGE_LENGTH[12:0]	Length of command that caused an out-of-range interrupt	R
23:21	—	The read values are undefined.	R

Bit	Symbol	Function	R/W
24	OUT_OF_RANGE_TYPE0	Type of command that caused an out-of-range interrupt 0: Write 1: Read	R
25	OUT_OF_RANGE_TYPE1	Type of command that caused an out-of-range interrupt 0: Non-masked write 1: Masked write	R
26	OUT_OF_RANGE_TYPE2	Type of command that caused an out-of-range interrupt 0: No wrapping 1: Wrapping	R
27	OUT_OF_RANGE_TYPE3	Type of command that caused an out-of-range interrupt 0: Non-exclusive 1: Exclusive	R
28	OUT_OF_RANGE_TYPE4	Type of command that caused an out-of-range interrupt 0: Non-flushed 1: Flushed	R
29	OUT_OF_RANGE_TYPE5	Type of command that caused an out-of-range interrupt 0: Without auto-precharge 1: With auto-precharge	R
31:30	—	The read values are undefined.	R

57.3.68 DDR_MEMC_DENALI_CTL_368 : DENALI_CTL_368 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x05C0

Bit position:	31																												0
Bit field:	PORT_CMD_ERROR_ADDR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PORT_CMD_ERROR_ADDR[31:0]	Address of command that caused the PORT command error (address of bits[31:0])	R

57.3.69 DDR_MEMC_DENALI_CTL_369 : DENALI_CTL_369 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x05C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	PORT_CMD_ERROR_TYPE	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PORT_CMD_ERROR_ADDR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PORT_CMD_ERROR_ADDR32	Address of command that caused the PORT command error (address of bit 32)	R
25:1	—	The read values are undefined.	R
26	PORT_CMD_ERROR_TYPE	This bit indicates that a protection error occurred. The error may relate to the address, access type, and transaction type.	R
31:27	—	The read values are undefined.	R

57.3.70 DDR_MEMC_DENALI_CTL_382 : DENALI_CTL_382 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x05F8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	axi0_fixed_port_priority_enable	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	—	Reserved*1	R/W
24	axi0_fixed_port_priority_enable	Defines the priority scheme used by AXI port 0 (A0 I/F). 0: Priority is defined per-command by the value of the axi0_ARQOS / axi0_AWQOS incoming signals. In this LSI, both axi0_ARQOS and axi0_AWQOS are fixed to 000b for all the command. 1: Priority is defined per-port by the axi0_r_priority and axi0_w_priority parameters. The priority value should be defined during initialization by programming the priority parameters. The user may change either/both of the priority values during normal operation when the Controller is idle. To do so, the user should de-assert the axi0_fixed_port_priority_enable parameter, change the axi0_r_priority and/or axi0_w_priority parameters, and then assert the axi0_fixed_port_priority_enable parameter.	R/W
31:25	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.71 DDR_MEMC_DENALI_CTL_383 : DENALI_CTL_383 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x05FC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	axi1_fixed_port_priority_enable	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	axi0_w_priority[2:0]			—	—	—	—	—	axi0_r_priority[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	axi0_r_priority[2:0]	Sets the priority of read commands from AXI port 0. According to the AXI specification, a value of 0 on the axi0_ARQOS signal is the lowest priority, and a value of 7 is the highest priority. However, this unit will invert this signal inside the Controller core and use 0 as the highest priority. If the axi0_fixed_port_priority_enable parameter is set to 1, this parameter will be used to define the priority for read commands from port 0. If the axi0_fixed_port_priority_enable parameter is cleared to 0, this parameter will be ignored and the priority will be defined by the value of the axi0_ARQOS incoming signal.*1	R/W
7:3	—	Reserved*2	R/W
10:8	axi0_w_priority[2:0]	Sets the priority of write commands from AXI port 0. According to the AXI specification, a value of 0 on the axi0_AWQOS signal is the lowest priority, and a value of 7 is the highest priority. However, this unit will invert this signal inside the Controller core and use 0 as the highest priority. If the axi0_fixed_port_priority_enable parameter is set to 1, this parameter will be used to define the priority for write commands from port 0. If the axi0_fixed_port_priority_enable parameter is cleared to 0, this parameter will be ignored and the priority will be defined by the value of the axi0_AWQOS incoming signal.*1	R/W
23:11	—	Reserved*2	R/W
24	axi1_fixed_port_priority_enable	Defines the priority scheme used by AXI port 1 (A1 I/F). 0: Priority is defined per-command by the value of the axi1_ARQOS / axi1_AWQOS incoming signals. In this LSI, both axi1_ARQOS and axi1_AWQOS signals are fixed to 000b for all the command. 1: Priority is defined per-port by the axi1_r_priority and axi1_w_priority parameters. The priority value should be defined during initialization by programming the priority parameters. The user may change either/both of the priority values during normal operation when the Controller is idle. To do so, the user should de-assert the axi1_fixed_port_priority_enable parameter, change the axi1_r_priority and/or axi1_w_priority parameters, and then assert the axi1_fixed_port_priority_enable parameter.	R/W
31:25	—	Reserved*2	R/W

Note 1. This parameter may only be changed during initialization or when the Controller is idle with no data in the port FIFOs and the axi0_fixed_port_priority_enable parameter is cleared to 0. The Controller will use a rising edge of the axi0_fixed_port_priority_enable parameter to capture the new value of this parameter into AXI port 0.

Note 2. The write value should follow those generated by DDR parameter generation tool.

57.3.72 DDR_MEMC_DENALI_CTL_384 : DENALI_CTL_384 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0600

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	axi2_fixed_port_priority_enable	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	axi1_w_priority[2:0]			—	—	—	—	—	axi1_r_priority[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	axi1_r_priority[2:0]	Sets the priority of read commands from AXI port 1. According to the AXI specification, a value of 0 on the axi1_ARQOS signal is the lowest priority, and a value of 7 is the highest priority. However, this unit will invert this signal inside the Controller core and use 0 as the highest priority. If the axi1_fixed_port_priority_enable parameter is set to 1, this parameter will be used to define the priority for read commands from port 1. If the axi1_fixed_port_priority_enable parameter is cleared to 0, this parameter will be ignored and the priority will be defined by the value of the axi1_ARQOS incoming signal.*1	R/W
7:3	—	Reserved*2	R/W
10:8	axi1_w_priority[2:0]	Sets the priority of write commands from AXI port 1. According to the AXI specification, a value of 0 on the axi1_AWQOS signal is the lowest priority, and a value of 7 is the highest priority. However, this unit will invert this signal inside the Controller core and use 0 as the highest priority. If the axi1_fixed_port_priority_enable parameter is set to 1, this parameter will be used to define the priority for write commands from port 1. If the axi1_fixed_port_priority_enable parameter is cleared to 0, this parameter will be ignored and the priority will be defined by the value of the axi1_AWQOS incoming signal.*1	R/W
23:11	—	Reserved*2	R/W
24	axi2_fixed_port_priority_enable	Defines the priority scheme used by AXI port 2 (R2 I/F). 0: Priority is defined per-command by the value of the axi2_ARQOS / axi2_AWQOS incoming signals. In this LSI, both axi2_ARQOS and axi2_AWQOS signals are fixed to 000b for all the command. 1: Priority is defined per-port by the axi2_r_priority and axi2_w_priority parameters. The priority value should be defined during initialization by programming the priority parameters. The user may change either/both of the priority values during normal operation when the Controller is idle. To do so, the user should de-assert the axi2_fixed_port_priority_enable parameter, change the axi2_r_priority and/or axi2_w_priority parameters, and then assert the axi2_fixed_port_priority_enable parameter.	R/W
31:25	—	Reserved*2	R/W

Note 1. This parameter may only be changed during initialization or when the Controller is idle with no data in the port FIFOs and the axi1_fixed_port_priority_enable parameter is cleared to 0. The Controller will use a rising edge of the axi1_fixed_port_priority_enable parameter to capture the new value of this parameter into AXI port 1.

Note 2. The write value should follow those generated by DDR parameter generation tool.

57.3.73 DDR_MEMC_DENALI_CTL_385 : DENALI_CTL_385 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0604

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	axi3_fixed_port_priority_enable	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	axi2_w_priority[2:0]			—	—	—	—	—	axi2_r_priority[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	axi2_r_priority[2:0]	Sets the priority of read commands from AXI port 2. According to the AXI specification, a value of 0 on the axi2_ARQOS signal is the lowest priority, and a value of 7 is the highest priority. However, this unit will invert this signal inside the Controller core and use 0 as the highest priority. If the axi2_fixed_port_priority_enable parameter is set to 1, this parameter will be used to define the priority for read commands from port 2. If the axi2_fixed_port_priority_enable parameter is cleared to 0, this parameter will be ignored and the priority will be defined by the value of the axi2_ARQOS incoming signal.*1	R/W
7:3	—	Reserved*2	R/W
10:8	axi2_w_priority[2:0]	Sets the priority of write commands from AXI port 2. According to the AXI specification, a value of 0 on the axi2_AWQOS signal is the lowest priority, and a value of 7 is the highest priority. However, this unit will invert this signal inside the Controller core and use 0 as the highest priority. If the axi2_fixed_port_priority_enable parameter is set to 1, this parameter will be used to define the priority for write commands from port 2. If the axi2_fixed_port_priority_enable parameter is cleared to 0, this parameter will be ignored and the priority will be defined by the value of the axi2_AWQOS incoming signal.*1	R/W
23:11	—	Reserved*2	R/W
24	axi3_fixed_port_priority_enable	Defines the priority scheme used by AXI port 3 (R3 I/F). 0: Priority is defined per-command by the value of the axi3_ARQOS / axi3_AWQOS incoming signals. In this LSI, both axi3_ARQOS and axi3_AWQOS signals are fixed to 000b for all the command. 1: Priority is defined per-port by the axi3_r_priority and axi3_w_priority parameters. The priority value should be defined during initialization by programming the priority parameters. The user may change either/both of the priority values during normal operation when the Controller is idle. To do so, the user should de-assert the axi3_fixed_port_priority_enable parameter, change the axi3_r_priority and/or axi3_w_priority parameters, and then assert the axi3_fixed_port_priority_enable parameter.	R/W
31:25	—	Reserved*2	R/W

Note 1. This parameter may only be changed during initialization or when the Controller is idle with no data in the port FIFOs and the axi2_fixed_port_priority_enable parameter is cleared to 0. The Controller will use a rising edge of the axi2_fixed_port_priority_enable parameter to capture the new value of this parameter into AXI port 2.

Note 2. The write value should follow those generated by DDR parameter generation tool.

57.3.74 DDR_MEMC_DENALI_CTL_386 : DENALI_CTL_386 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0608

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	axi4_fixed_port_priority_enable	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	axi3_w_priority[2:0]			—	—	—	—	—	axi3_r_priority[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	axi3_r_priority[2:0]	Sets the priority of read commands from AXI port 3. According to the AXI specification, a value of 0 on the axi3_ARQOS signal is the lowest priority, and a value of 7 is the highest priority. However, this unit will invert this signal inside the Controller core and use 0 as the highest priority. If the axi3_fixed_port_priority_enable parameter is set to 1, this parameter will be used to define the priority for read commands from port 3. If the axi3_fixed_port_priority_enable parameter is cleared to 0, this parameter will be ignored and the priority will be defined by the value of the axi3_ARQOS incoming signal.*1	R/W
7:3	—	Reserved*2	R/W
10:8	axi3_w_priority[2:0]	Sets the priority of write commands from AXI port 3. According to the AXI specification, a value of 0 on the axi3_AWQOS signal is the lowest priority, and a value of 7 is the highest priority. However, this unit will invert this signal inside the Controller core and use 0 as the highest priority. If the axi3_fixed_port_priority_enable parameter is set to 1, this parameter will be used to define the priority for write commands from port 3. If the axi3_fixed_port_priority_enable parameter is cleared to 0, this parameter will be ignored and the priority will be defined by the value of the axi3_AWQOS incoming signal.*1	R/W
23:11	—	Reserved*2	R/W
24	axi4_fixed_port_priority_enable	Defines the priority scheme used by AXI port 4 (A4 I/F). 0: Priority is defined per-command by the value of the axi4_ARQOS / axi4_AWQOS incoming signals. In this LSI, both axi4_ARQOS and axi4_AWQOS signals are fixed to 000b for all the command. 1: Priority is defined per-port by the axi4_r_priority and axi4_w_priority parameters. The priority value should be defined during initialization by programming the priority parameters. The user may change either/both of the priority values during normal operation when the Controller is idle. To do so, the user should de-assert the axi4_fixed_port_priority_enable parameter, change the axi4_r_priority and/or axi4_w_priority parameters, and then assert the axi4_fixed_port_priority_enable parameter.	R/W
31:25	—	Reserved*2	R/W

Note 1. This parameter may only be changed during initialization or when the Controller is idle with no data in the port FIFOs and the axi3_fixed_port_priority_enable parameter is cleared to 0. The Controller will use a rising edge of the axi3_fixed_port_priority_enable parameter to capture the new value of this parameter into AXI port 3.
Note 2. The write value should follow those generated by DDR parameter generation tool.

57.3.75 DDR_MEMC_DENALI_CTL_387 : DENALI_CTL_387 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x060C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	axi4_w_priority[2:0]			—	—	—	—	—	axi4_r_priority[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	axi4_r_priority[2:0]	Sets the priority of read commands from AXI port 4. According to the AXI specification, a value of 0 on the axi4_ARQOS signal is the lowest priority, and a value of 7 is the highest priority. However, this unit will invert this signal inside the Controller core and use 0 as the highest priority. If the axi4_fixed_port_priority_enable parameter is set to 1, this parameter will be used to define the priority for read commands from port 4. If the axi4_fixed_port_priority_enable parameter is cleared to 0, this parameter will be ignored and the priority will be defined by the value of the axi4_ARQOS incoming signal.*1	R/W

Bit	Symbol	Function	R/W
7:3	—	Reserved*2	R/W
10:8	axi4_w_priority[2:0]	Sets the priority of write commands from AXI port 4. According to the AXI specification, a value of 0 on the axi4_AWQOS signal is the lowest priority, and a value of 7 is the highest priority. However, this unit will invert this signal inside the Controller core and use 0 as the highest priority. If the axi4_fixed_port_priority_enable parameter is set to 1, this parameter will be used to define the priority for write commands from port 4. If the axi4_fixed_port_priority_enable parameter is cleared to 0, this parameter will be ignored and the priority will be defined by the value of the axi4_AWQOS incoming signal.*1	R/W
31:11	—	Reserved*2	R/W

Note 1. This parameter may only be changed during initialization or when the Controller is idle with no data in the port FIFOs and the axi4_fixed_port_priority_enable parameter is cleared to 0. The Controller will use a rising edge of the axi4_fixed_port_priority_enable parameter to capture the new value of this parameter into AXI port 4.

Note 2. The write value should follow those generated by DDR parameter generation tool.

57.3.76 DDR_MEMC_DENALI_CTL_395 : DENALI_CTL_395 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x062C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	AXI0_ADDRESS_RANGE_ENABLE	—	—	—	—	—	—	—	PORT_ADDR_PROTECTION_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	The read values are undefined.	R/W
16	PORT_ADDR_PROTECTION_EN	Enable port address range protection logic and interrupt generation. Set to 1 to enable.	R/W
23:17	—	The read values are undefined.	R/W
24	AXI0_ADDRESS_RANGE_ENABLE	Determines if port 0 will use the axi0_arregion / axi0_awregion signals or the AXI0_START_ADDR_Z / AXI0_END_ADDR_Z parameters to define the region/range. Set to 1 for parameter control for this LSI.	R/W
31:25	—	The read values are undefined.	R/W

57.3.77 DDR_MEMC_DENALI_CTL_396+4n : DENALI_CTL_396+4n Register (n = 0 to 15)

Base address: DDRSS = 0x8030_0000

Offset address: 0x0630 + 0x10 × n

Bit position:	31													18																	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	AXI0_START_ADDR[18:0]																	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
18:0	AXI0_START_ADDR[18:0]	Start address of port 0 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
31:19	—	The read values are undefined.	R/W

57.3.78 DDR_MEMC_DENALI_CTL_397+4n : DENALI_CTL_397+4n Register (n = 0 to 15)

Base address: DDRSS = 0x8030_0000

Offset address: 0x0634 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	AXI0_RANGE_PROT_BITS[1:0]		—	—	—	—	—	AXI0_END_ADDR[18:16]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AXI0_END_ADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
18:0	AXI0_END_ADDR[18:0]	End address of port 0 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
23:19	—	The read values are undefined.	R/W
25:24	AXI0_RANGE_PROT_BITS[1:0]	Allowed transaction types for port 0 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low. 0 0: Privileged and Secure Access Only 0 1: Secure Access, Privileged or Non-Privileged 1 0: Privileged Access, Secure or Non-Secure 1 1: Full Access	R/W
31:26	—	The read values are undefined.	R/W

57.3.79 DDR_MEMC_DENALI_CTL_460 : DENALI_CTL_460 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0730

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	AXI1_START_ADDR[18:8]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AXI1_START_ADDR[7:0]								—	—	—	—	—	—	—	AXI1_ADDRESS_RANGE_ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AXI1_ADDRESS_RANGE_ENABLE	Determines if port 1 will use the axi1_arregion / axi1_awregion signals or the AXI1_START_ADDR_Z / AXI1_END_ADDR_Z parameters to define the region/range. Set to 1 for parameter control for this LSI.	R/W
7:1	—	The read values are undefined.	R/W
26:8	AXI1_START_ADDR[18:0]	Start address of port 1 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
31:27	—	The read values are undefined.	R/W

57.3.80 DDR_MEMC_DENALI_CTL_460+4n : DENALI_CTL_460+4n Register (n = 1 to 15)

Base address: DDRSS = 0x8030_0000

Offset address: 0x0730 + 0x10 × n

Bit position:	31													18																0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	AXI1_START_ADDR[18:0]																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
18:0	AXI1_START_ADDR[18:0]	Start address of port 1 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
31:19	—	The read values are undefined.	R/W

57.3.81 DDR_MEMC_DENALI_CTL_461+4n : DENALI_CTL_461+4n Register (n = 0 to 15)

Base address: DDRSS = 0x8030_0000

Offset address: 0x0734 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	AXI1_RANGE_PROT_BITS[1:0]		—	—	—	—	—	AXI1_END_ADDR[18:16]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AXI1_END_ADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
18:0	AXI1_END_ADDR[18:0]	End address of port 1 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
23:19	—	The read values are undefined.	R/W
25:24	AXI1_RANGE_PROT_BITS[1:0]	Allowed transaction types for port 1 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low. 0 0: Privileged and Secure Access Only 0 1: Secure Access, Privileged or Non-Privileged 1 0: Privileged Access, Secure or Non-Secure 1 1: Full Access	R/W

Bit	Symbol	Function	R/W
31:26	—	The read values are undefined.	R/W

57.3.82 DDR_MEMC_DENALI_CTL_524 : DENALI_CTL_524 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0830

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	AXI2_START_ADDR[18:8]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AXI2_START_ADDR[7:0]								—	—	—	—	—	—	—	AXI2_ADDRESS_RANGE_ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AXI2_ADDRESS_RANGE_ENABLE	Determines if port 2 will use the axi2_arregion / axi2_awregion signals or the AXI2_START_ADDR_Z / AXI2_END_ADDR_Z parameters to define the region/range. Set to 1 for parameter control for this LSI.	R/W
7:1	—	The read values are undefined.	R/W
26:8	AXI2_START_ADDR[18:0]	Start address of port 2 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
31:27	—	The read values are undefined.	R/W

57.3.83 DDR_MEMC_DENALI_CTL_524+4n : DENALI_CTL_524+4n Register (n = 1 to 15)

Base address: DDRSS = 0x8030_0000

Offset address: 0x0830 + 0x10 × n

Bit position:	31												18																	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	AXI2_START_ADDR[18:0]																	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
18:0	AXI2_START_ADDR[18:0]	Start address of port 2 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
31:19	—	The read values are undefined.	R/W

57.3.84 DDR_MEMC_DENALI_CTL_525+4n : DENALI_CTL_525+4n Register (n = 0 to 15)

Base address: DDRSS = 0x8030_0000

Offset address: 0x0834 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	AXI2_RANGE_PROT_BITS[1:0]		—	—	—	—	—	AXI2_END_ADDR[18:16]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AXI2_END_ADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
18:0	AXI2_END_ADDR[18:0]	End address of port 2 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
23:19	—	The read values are undefined.	R/W
25:24	AXI2_RANGE_PROT_BITS[1:0]	Allowed transaction types for port 2 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low. 0 0: Privileged and Secure Access Only 0 1: Secure Access, Privileged or Non-Privileged 1 0: Privileged Access, Secure or Non-Secure 1 1: Full Access	R/W
31:26	—	The read values are undefined.	R/W

57.3.85 DDR_MEMC_DENALI_CTL_588 : DENALI_CTL_588 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0930

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	AXI3_START_ADDR[18:8]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AXI3_START_ADDR[7:0]								—	—	—	—	—	—	—	AXI3_ADDRESS_RANGE_ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

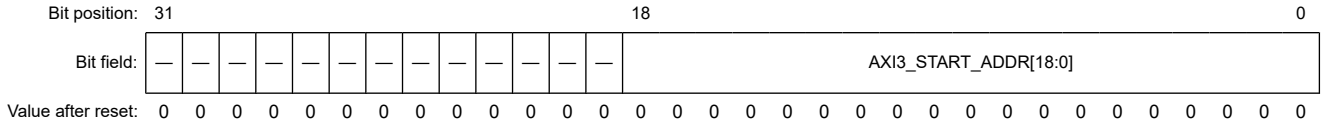
Bit	Symbol	Function	R/W
0	AXI3_ADDRESS_RANGE_ENABLE	Determines if port 3 will use the axi3_arregion / axi3_awregion signals or the AXI3_START_ADDR_Z / AXI3_END_ADDR_Z parameters to define the region/range. Set to 1 for parameter control for this LSI.	R/W
7:1	—	The read values are undefined.	R/W
26:8	AXI3_START_ADDR[18:0]	Start address of port 3 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W

Bit	Symbol	Function	R/W
31:27	—	The read values are undefined.	R/W

57.3.86 DDR_MEMC_DENALI_CTL_588+4n : DENALI_CTL_588+4n Register (n = 1 to 15)

Base address: DDRSS = 0x8030_0000

Offset address: 0x0930 + 0x10 × n

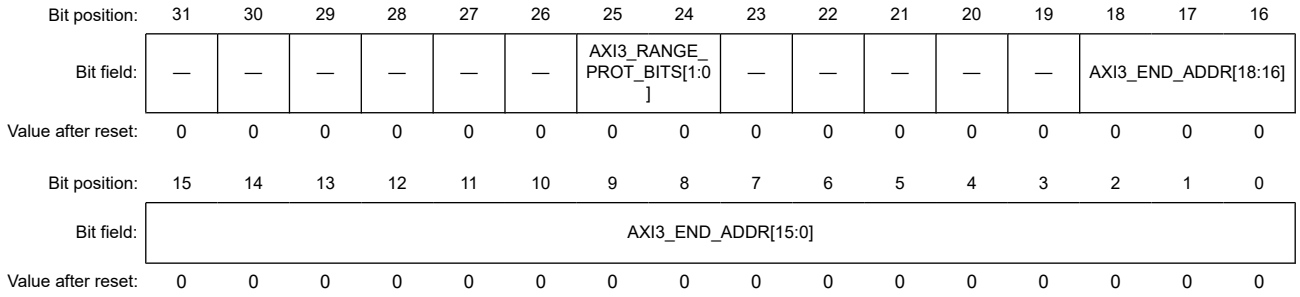


Bit	Symbol	Function	R/W
18:0	AXI3_START_ADDR[18:0]	Start address of port 3 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
31:19	—	The read values are undefined.	R/W

57.3.87 DDR_MEMC_DENALI_CTL_589+4n : DENALI_CTL_589+4n Register (n = 0 to 15)

Base address: DDRSS = 0x8030_0000

Offset address: 0x0934 + 0x10 × n



Bit	Symbol	Function	R/W
18:0	AXI3_END_ADDR[18:0]	End address of port 3 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
23:19	—	The read values are undefined.	R/W
25:24	AXI3_RANGE_PROT_BITS[1:0]	Allowed transaction types for port 3 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low. 0 0: Privileged and Secure Access Only 0 1: Secure Access, Privileged or Non-Privileged 1 0: Privileged Access, Secure or Non-Secure 1 1: Full Access	R/W
31:26	—	The read values are undefined.	R/W

57.3.88 DDR_MEMC_DENALI_CTL_652 : DENALI_CTL_652 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0A30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	AXI4_START_ADDR[18:8]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AXI4_START_ADDR[7:0]								—	—	—	—	—	—	—	AXI4_ADDRESS_RANGE_ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AXI4_ADDRESS_RANGE_ENABLE	Determines if port 4 will use the axi4_arregion / axi4_awregion signals or the AXI4_START_ADDR_Z / AXI4_END_ADDR_Z parameters to define the region/range. Set to 1 for parameter control for this LSI.	R/W
7:1	—	The read values are undefined.	R/W
26:8	AXI4_START_ADDR[18:0]	Start address of port 4 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
31:27	—	The read values are undefined.	R/W

57.3.89 DDR_MEMC_DENALI_CTL_652+4n : DENALI_CTL_652+4n Register (n = 1 to 15)

Base address: DDRSS = 0x8030_0000

Offset address: 0x0A30 + 0x10 × n

Bit position:	31													18																	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	AXI4_START_ADDR[18:0]																	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
18:0	AXI4_START_ADDR[18:0]	Start address of port 4 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
31:19	—	The read values are undefined.	R/W

57.3.90 DDR_MEMC_DENALI_CTL_653+4n : DENALI_CTL_653+4n Register (n = 0 to 15)

Base address: DDRSS = 0x8030_0000

Offset address: 0x0A34 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	AXI4_RANGE_PROT_BITS[1:0]		—	—	—	—	—	AXI4_END_ADDR[18:16]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AXI4_END_ADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
18:0	AXI4_END_ADDR[18:0]	End address of port 4 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low.	R/W
23:19	—	The read values are undefined.	R/W
25:24	AXI4_RANGE_PROT_BITS[1:0]	Allowed transaction types for port 4 address range n. This may only be changed before initialization begins or when the controller is quiescent, there is no data in the port FIFOs, and PORT_ADDR_PROTECTION_EN is low. 0 0: Privileged and Secure Access Only 0 1: Secure Access, Privileged or Non-Privileged 1 0: Privileged Access, Secure or Non-Secure 1 1: Full Access	R/W
31:26	—	The read values are undefined.	R/W

57.3.91 DDR_MEMC_DENALI_CTL_716 : DENALI_CTL_716 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	axi0_priority0_relative_priority[3:0]				—	—	—	—	wrr_param_value_err[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	weighted_round_robin_weight_sharing[1:0]		—	—	—	—	—	—	—	weighted_round_robin_latency_control
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	weighted_round_robin_latency_control	Controls the weighted round-robin latency option. 0: Counters only count when their port has a command waiting to be processed. 1: Counters are always running.	R/W
7:1	—	Reserved*1	R/W

Bit	Symbol	Function	R/W
9:8	weighted_round_robin_weight_sharing[1:0]	Indicates that the port pair is tied together in arbitration decisions in weighted round-robin arbitration. Bit [0] controls ports 0 and 1, bit [1] controls port 2 and 3. For each port pair: 0: The represented ports are treated independently in arbitration. 1: The represented ports are tied together for arbitration.	R/W
15:10	—	Reserved*1	R/W
19:16	wrr_param_value_err[3:0]	Reports errors/warnings from the weighted round-robin arbitration. Bit [3] = The port ordering parameter values for paired ports is not sequential. Bit [2] = The relative priority values for any of the ports paired through the weighted_round_robin_weight_sharing parameter are not identical. Bit [1] = Any of the relative priority parameters have been programmed to a zero value. Bit [0] = The port ordering parameters do not all contain unique values. For each bit: 0: No error 1: Error	R
23:20	—	Reserved*1	R/W
27:24	axi0_priority0_relative_priority[3:0]	Holds the relative priority of AXI port 0 for priority 0 commands in weighted round robin arbitration.	R/W
31:28	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.92 DDR_MEMC_DENALI_CTL_717 : DENALI_CTL_717 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	axi0_priority4_relative_priority[3:0]				—	—	—	—	axi0_priority3_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	axi0_priority2_relative_priority[3:0]				—	—	—	—	axi0_priority1_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	axi0_priority1_relative_priority[3:0]	Holds the relative priority of AXI port 0 for priority 1 commands in weighted round robin arbitration.	R/W
7:4	—	Reserved*1	R/W
11:8	axi0_priority2_relative_priority[3:0]	Holds the relative priority of AXI port 0 for priority 2 commands in weighted round robin arbitration.	R/W
15:12	—	Reserved*1	R/W
19:16	axi0_priority3_relative_priority[3:0]	Holds the relative priority of AXI port 0 for priority 3 commands in weighted round robin arbitration.	R/W
23:20	—	Reserved*1	R/W
27:24	axi0_priority4_relative_priority[3:0]	Holds the relative priority of AXI port 0 for priority 4 commands in weighted round robin arbitration.	R/W
31:28	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.93 DDR_MEMC_DENALI_CTL_718 : DENALI_CTL_718 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	axi0_port_ordering[2:0]			—	—	—	—	axi0_priority7_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	axi0_priority6_relative_priority[3:0]				—	—	—	—	axi0_priority5_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	axi0_priority5_relative_priority[3:0]	Holds the relative priority of AXI port 0 for priority 5 commands in weighted round robin arbitration.	R/W
7:4	—	Reserved*1	R/W
11:8	axi0_priority6_relative_priority[3:0]	Holds the relative priority of AXI port 0 for priority 6 commands in weighted round robin arbitration.	R/W
15:12	—	Reserved*1	R/W
19:16	axi0_priority7_relative_priority[3:0]	Holds the relative priority of AXI port 0 for priority 7 commands in weighted round robin arbitration.	R/W
23:20	—	Reserved*1	R/W
26:24	axi0_port_ordering[2:0]	Reassigned port order for AXI port 0 Used in weighted round-robin arbitration to modify the order than the ports are scanned when multiple commands are at the same priority level and have the same relative priorities.	R/W
31:27	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.94 DDR_MEMC_DENALI_CTL_719 : DENALI_CTL_719 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	axi1_priority1_relative_priority[3:0]				—	—	—	—	axi1_priority0_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	axi0_priority_relax[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	axi0_priority_relax[9:0]	Holds the counter value for AXI port 0 at which the priority relax condition is triggered in weighted round robin arbitration.	R/W
15:10	—	Reserved*1	R/W
19:16	axi1_priority0_relative_priority[3:0]	Holds the relative priority of AXI port 1 for priority 0 commands in weighted round robin arbitration.	R/W
23:20	—	Reserved*1	R/W

Bit	Symbol	Function	R/W
27:24	axi1_priority1_relative_priority[3:0]	Holds the relative priority of AXI port 1 for priority 1 commands in weighted round robin arbitration.	R/W
31:28	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.95 DDR_MEMC_DENALI_CTL_720 : DENALI_CTL_720 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	axi1_priority5_relative_priority[3:0]				—	—	—	—	axi1_priority4_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	axi1_priority3_relative_priority[3:0]				—	—	—	—	axi1_priority2_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	axi1_priority2_relative_priority[3:0]	Holds the relative priority of AXI port 1 for priority 2 commands in weighted round robin arbitration.	R/W
7:4	—	Reserved*1	R/W
11:8	axi1_priority3_relative_priority[3:0]	Holds the relative priority of AXI port 1 for priority 3 commands in weighted round robin arbitration.	R/W
15:12	—	Reserved*1	R/W
19:16	axi1_priority4_relative_priority[3:0]	Holds the relative priority of AXI port 1 for priority 4 commands in weighted round robin arbitration.	R/W
23:20	—	Reserved*1	R/W
27:24	axi1_priority5_relative_priority[3:0]	Holds the relative priority of AXI port 1 for priority 5 commands in weighted round robin arbitration.	R/W
31:28	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.96 DDR_MEMC_DENALI_CTL_721 : DENALI_CTL_721 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B44

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	axi1_port_ordering[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	axi1_priority7_relative_priority[3:0]				—	—	—	—	axi1_priority6_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	axi1_priority6_relative_priority[3:0]	Holds the relative priority of AXI port 1 for priority 6 commands in weighted round robin arbitration.	R/W

Bit	Symbol	Function	R/W
7:4	—	Reserved*1	R/W
11:8	axi1_priority7_relative_priority[3:0]	Holds the relative priority of AXI port 1 for priority 7 commands in weighted round robin arbitration.	R/W
15:12	—	Reserved*1	R/W
18:16	axi1_port_ordering[2:0]	Reassigned port order for AXI port 1 Used in weighted round-robin arbitration to modify the order than the ports are scanned when multiple commands are at the same priority level and have the same relative priorities.	R/W
31:19	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.97 DDR_MEMC_DENALI_CTL_722 : DENALI_CTL_722 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B48

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	axi2_priority1_relative_priority[3:0]				—	—	—	—	axi2_priority0_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	axi1_priority_relax[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	axi1_priority_relax[9:0]	Holds the counter value for AXI port 1 at which the priority relax condition is triggered in weighted round robin arbitration.	R/W
15:10	—	Reserved*1	R/W
19:16	axi2_priority0_relative_priority[3:0]	Holds the relative priority of AXI port 2 for priority 0 commands in weighted round robin arbitration.	R/W
23:20	—	Reserved*1	R/W
27:24	axi2_priority1_relative_priority[3:0]	Holds the relative priority of AXI port 2 for priority 1 commands in weighted round robin arbitration.	R/W
31:28	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.98 DDR_MEMC_DENALI_CTL_723 : DENALI_CTL_723 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B4C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	axi2_priority5_relative_priority[3:0]				—	—	—	—	axi2_priority4_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	axi2_priority3_relative_priority[3:0]				—	—	—	—	axi2_priority2_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	axi2_priority2_relative_priority[3:0]	Holds the relative priority of AXI port 2 for priority 2 commands in weighted round robin arbitration.	R/W
7:4	—	Reserved*1	R/W
11:8	axi2_priority3_relative_priority[3:0]	Holds the relative priority of AXI port 2 for priority 3 commands in weighted round robin arbitration.	R/W
15:12	—	Reserved*1	R/W
19:16	axi2_priority4_relative_priority[3:0]	Holds the relative priority of AXI port 2 for priority 4 commands in weighted round robin arbitration.	R/W
23:20	—	Reserved*1	R/W
27:24	axi2_priority5_relative_priority[3:0]	Holds the relative priority of AXI port 2 for priority 5 commands in weighted round robin arbitration.	R/W
31:28	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.99 DDR_MEMC_DENALI_CTL_724 : DENALI_CTL_724 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	axi2_port_ordering[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	axi2_priority7_relative_priority[3:0]				—	—	—	—	axi2_priority6_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	axi2_priority6_relative_priority[3:0]	Holds the relative priority of AXI port 2 for priority 6 commands in weighted round robin arbitration.	R/W
7:4	—	Reserved*1	R/W
11:8	axi2_priority7_relative_priority[3:0]	Holds the relative priority of AXI port 2 for priority 7 commands in weighted round robin arbitration.	R/W
15:12	—	Reserved*1	R/W
18:16	axi2_port_ordering[2:0]	Reassigned port order for AXI port 2 Used in weighted round-robin arbitration to modify the order than the ports are scanned when multiple commands are at the same priority level and have the same relative priorities.	R/W
31:19	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.100 DDR_MEMC_DENALI_CTL_725 : DENALI_CTL_725 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B54

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	axi3_priority1_relative_priority[3:0]				—	—	—	—	axi3_priority0_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	axi2_priority_relax[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	axi2_priority_relax[9:0]	Holds the counter value for AXI port 2 at which the priority relax condition is triggered in weighted round robin arbitration.	R/W
15:10	—	Reserved*1	R/W
19:16	axi3_priority0_relative_priority[3:0]	Holds the relative priority of AXI port 3 for priority 0 commands in weighted round robin arbitration.	R/W
23:20	—	Reserved*1	R/W
27:24	axi3_priority1_relative_priority[3:0]	Holds the relative priority of AXI port 3 for priority 1 commands in weighted round robin arbitration.	R/W
31:28	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.101 DDR_MEMC_DENALI_CTL_726 : DENALI_CTL_726 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B58

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	axi3_priority5_relative_priority[3:0]				—	—	—	—	axi3_priority4_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	axi3_priority3_relative_priority[3:0]				—	—	—	—	axi3_priority2_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	axi3_priority2_relative_priority[3:0]	Holds the relative priority of AXI port 3 for priority 2 commands in weighted round robin arbitration.	R/W
7:4	—	Reserved*1	R/W
11:8	axi3_priority3_relative_priority[3:0]	Holds the relative priority of AXI port 3 for priority 3 commands in weighted round robin arbitration.	R/W
15:12	—	Reserved*1	R/W
19:16	axi3_priority4_relative_priority[3:0]	Holds the relative priority of AXI port 3 for priority 4 commands in weighted round robin arbitration.	R/W
23:20	—	Reserved*1	R/W
27:24	axi3_priority5_relative_priority[3:0]	Holds the relative priority of AXI port 3 for priority 5 commands in weighted round robin arbitration.	R/W

Bit	Symbol	Function	R/W
31:28	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.102 DDR_MEMC_DENALI_CTL_727 : DENALI_CTL_727 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B5C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	axi3_port_ordering[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	axi3_priority7_relative_priority[3:0]				—	—	—	—	axi3_priority6_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	axi3_priority6_relative_priority[3:0]	Holds the relative priority of AXI port 3 for priority 6 commands in weighted round robin arbitration.	R/W
7:4	—	Reserved*1	R/W
11:8	axi3_priority7_relative_priority[3:0]	Holds the relative priority of AXI port 3 for priority 7 commands in weighted round robin arbitration.	R/W
15:12	—	Reserved*1	R/W
18:16	axi3_port_ordering[2:0]	Reassigned port order for AXI port 3 Used in weighted round-robin arbitration to modify the order than the ports are scanned when multiple commands are at the same priority level and have the same relative priorities.	R/W
31:19	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.103 DDR_MEMC_DENALI_CTL_728 : DENALI_CTL_728 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B60

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	axi4_priority1_relative_priority[3:0]				—	—	—	—	axi4_priority0_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	axi3_priority_relax[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	axi3_priority_relax[9:0]	Holds the counter value for AXI port 3 at which the priority relax condition is triggered in weighted round robin arbitration.	R/W
15:10	—	Reserved*1	R/W
19:16	axi4_priority0_relative_priority[3:0]	Holds the relative priority of AXI port 4 for priority 0 commands in weighted round robin arbitration.	R/W

Bit	Symbol	Function	R/W
23:20	—	Reserved*1	R/W
27:24	axi4_priority1_relative_priority[3:0]	Holds the relative priority of AXI port 4 for priority 1 commands in weighted round robin arbitration.	R/W
31:28	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.104 DDR_MEMC_DENALI_CTL_729 : DENALI_CTL_729 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B64

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	axi4_priority5_relative_priority[3:0]				—	—	—	—	axi4_priority4_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	axi4_priority3_relative_priority[3:0]				—	—	—	—	axi4_priority2_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	axi4_priority2_relative_priority[3:0]	Holds the relative priority of AXI port 4 for priority 2 commands in weighted round robin arbitration.	R/W
7:4	—	Reserved*1	R/W
11:8	axi4_priority3_relative_priority[3:0]	Holds the relative priority of AXI port 4 for priority 3 commands in weighted round robin arbitration.	R/W
15:12	—	Reserved*1	R/W
19:16	axi4_priority4_relative_priority[3:0]	Holds the relative priority of AXI port 4 for priority 4 commands in weighted round robin arbitration.	R/W
23:20	—	Reserved*1	R/W
27:24	axi4_priority5_relative_priority[3:0]	Holds the relative priority of AXI port 4 for priority 5 commands in weighted round robin arbitration.	R/W
31:28	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.105 DDR_MEMC_DENALI_CTL_730 : DENALI_CTL_730 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	axi4_port_ordering[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	axi4_priority7_relative_priority[3:0]				—	—	—	—	axi4_priority6_relative_priority[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	axi4_priority6_relative_priority[3:0]	Holds the relative priority of AXI port 4 for priority 6 commands in weighted round robin arbitration.	R/W
7:4	—	Reserved*1	R/W
11:8	axi4_priority7_relative_priority[3:0]	Holds the relative priority of AXI port 4 for priority 7 commands in weighted round robin arbitration.	R/W
15:12	—	Reserved*1	R/W
18:16	axi4_port_ordering[2:0]	Reassigned port order for AXI port 4 Used in weighted round-robin arbitration to modify the order than the ports are scanned when multiple commands are at the same priority level and have the same relative priorities.	R/W
31:19	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.106 DDR_MEMC_DENALI_CTL_731 : DENALI_CTL_731 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x0B6C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	axi4_priority_relax[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	axi4_priority_relax[9:0]	Holds the counter value for AXI port 4 at which the priority relax condition is triggered in weighted round robin arbitration.	R/W
31:10	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.107 DDR_MEMC_MCAR_CTL : MCAR_CTL Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x2000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	dfi_init_start
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	set_axctl	—	—	—	—	—	—	—	block_fm_norm_req
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	block_fm_norm_req	Controls the FM arbiter. If the fm_override_control parameter is set to 1, this bit will limit the FM to only read/write commands.	R/W

Bit	Symbol	Function	R/W
7:1	—	Reserved*1	R/W
8	set_axctl	The read value is undefined. Load the settings of MCAR_AXCTL* registers to MC. This bit is read as 0.	W
15:9	—	Reserved*1	R/W
16	dfi_init_start	Override dfi_init_start signal.	R/W
31:17	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.108 DDR_MEMC_MCAR_MON : MCAR_MON Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x2004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	refresh_in_process	q_almost_full
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	port_busy[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	port_busy[4:0]	This bits contain 1 bit per port. This bits will only be low when the controller is not reading data, writing data, or processing a command for that port.	R
15:5	—	Reserved*1	R/W
16	q_almost_full	Indicates that the command queue has reached the value set in the q_fullness parameter.	R
17	refresh_in_process	Active-high bit that indicates that the controller is executing a refresh command. This bit is asserted when a refresh command is sent to the DRAM memories and remains asserted until the refresh command has completed.	R
31:18	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.109 DDR_MEMC_MCAR_AXCTL0 : MCAR_AXCTL0 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x2008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	axi0_awcobuf	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	Reserved*1	R/W

Bit	Symbol	Function	R/W
8	axi0_awcobuf	AXI port 0 coherent bufferable selection If the axi0_AWCACHE signal is set to 1 for Bufferable operation, this signal determines exactly what type of bufferable response is returned to the user interface. For guaranteed data coherency across all ports, the user should send all commands with axi0_AWCOBUF set to 1. 0: Response returned when command and data have been received by the port. 1: Response returned when command accepted into the MC core command queue and all associated data has been received by the AXI data port.	R/W
31:9	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.110 DDR_MEMC_MCAR_AXCTL1 : MCAR_AXCTL1 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x200C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	axi1_awcobuf	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	Reserved*1	R/W
8	axi1_awcobuf	AXI port 1 coherent bufferable selection If the axi1_AWCACHE signal is set to 1 for Bufferable operation, this signal determines exactly what type of bufferable response is returned to the user interface. For guaranteed data coherency across all ports, the user should send all commands with axi1_AWCOBUF set to 1. 0: Response returned when command and data have been received by the port. 1: Response returned when command accepted into the MC core command queue and all associated data has been received by the AXI data port.	R/W
31:9	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.111 DDR_MEMC_MCAR_AXCTL2 : MCAR_AXCTL2 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x2010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	axi2_awcobuf	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	Reserved*1	R/W

Bit	Symbol	Function	R/W
8	axi2_awcobuf	AXI port 2 coherent bufferable selection If the axi2_AWCACHE signal is set to 1 for Bufferable operation, this signal determines exactly what type of bufferable response is returned to the user interface. For guaranteed data coherency across all ports, the user should send all commands with axi2_AWCOBUF set to 1. 0: Response returned when command and data have been received by the port. 1: Response returned when command accepted into the MC core command queue and all associated data has been received by the AXI data port.	R/W
31:9	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.112 DDR_MEMC_MCAR_AXCTL3 : MCAR_AXCTL3 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x2014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	axi3_awcobuf	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	Reserved*1	R/W
8	axi3_awcobuf	AXI port 3 coherent bufferable selection If the axi3_AWCACHE signal is set to 1 for Bufferable operation, this signal determines exactly what type of bufferable response is returned to the user interface. For guaranteed data coherency across all ports, the user should send all commands with axi3_AWCOBUF set to 1. 0: Response returned when command and data have been received by the port. 1: Response returned when command accepted into the MC core command queue and all associated data has been received by the AXI data port.	R/W
31:9	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.3.113 DDR_MEMC_MCAR_AXCTL4 : MCAR_AXCTL4 Register

Base address: DDRSS = 0x8030_0000

Offset address: 0x2018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	axi4_awcobuf	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	Reserved*1	R/W

Bit	Symbol	Function	R/W
8	axi4_awcobuf	AXI port 4 coherent bufferable selection If the axi4_AWCACHE signal is set to 1 for Bufferable operation, this signal determines exactly what type of bufferable response is returned to the user interface. For guaranteed data coherency across all ports, the user should send all commands with axi4_AWCOBUF set to 1. 0: Response returned when command and data have been received by the port. 1: Response returned when command accepted into the MC core command queue and all associated data has been received by the AXI data port.	R/W
31:9	—	Reserved*1	R/W

Note 1. The write value should follow those generated by DDR parameter generation tool.

57.4 Function Description

57.4.1 External Pin Swizzling

This unit has the feature to remap the functions assigned to the external pins, making the PCB wiring design easier.

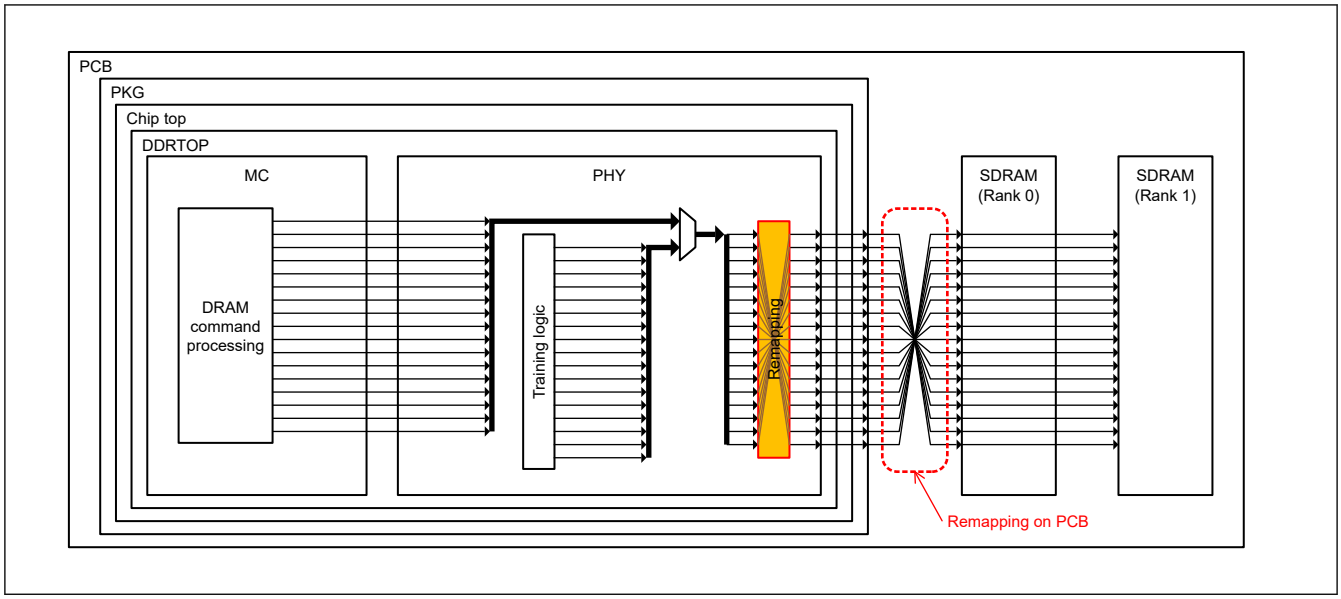


Figure 57.2 External pin swizzling

57.4.1.1 Command/Address Remapping

This feature statically remaps the functions assigned to the Command/Address pins (DDR_CA[AB][0-5]) of this unit.

The pin functions that can be remapped are as follows.

- LPDDR4: CA[0-5]_A, CA[0-5]_B

57.4.1.2 Rank Swapping

This feature statically swaps the Rank assigned to the CS/CKE pins (DDR_CS[AB][01], DDR_CKE[AB][01]) of this unit. It can swap between Group-0A (CS0_A, CKE0_A) and Group-1A (CS1_A, CKE1_A), and/or between Group-0B (CS0_B, CKE0_B) and Group-1B (CS1_B, CKE1_B). No register settings are required.

57.4.1.3 Data Swizzling

(1) Data Remapping in Byte Lane

This feature statically remaps the functions assigned to the DQ pins (DDR_DQ[AB][0-15]) of this unit within same Byte Lanes.

(2) Data Byte Lane Swapping

This feature statically swaps the DQ/DMI/DQS pins (DDR_DQ[AB][0-15], DDR_DMI[AB][01], DDR_DQS[AB]_[TC][01]) of this unit in Byte Lane units.

If byte lane swapping is used or not, the user must program the deviceN_byte0_csX parameters, where N is the device number and X is the chip select on that device. Unique parameters are provided for each chip select to allow for different byte swapping on each chip select. Each parameter contains one bit for each byte of the memory data bus. If a device is not being used, the deviceN_byte0_csX parameter should be programmed to 0x0, but if the device is being used, then one and only 1 bit of the deviceN_byte0_csX parameter should be set to 1.

57.4.1.4 Channel Swapping

This feature statically swaps the Channels of the external pins of this unit. No register settings are required for this feature itself.

(1) Example of the external pin swizzling

(a) Example of Channel/Lane swizzling

Channel/Lane swizzling is available only for the case shown in Figure 57.3 to Figure 57.10. Any swizzling not shown in figures is not possible.

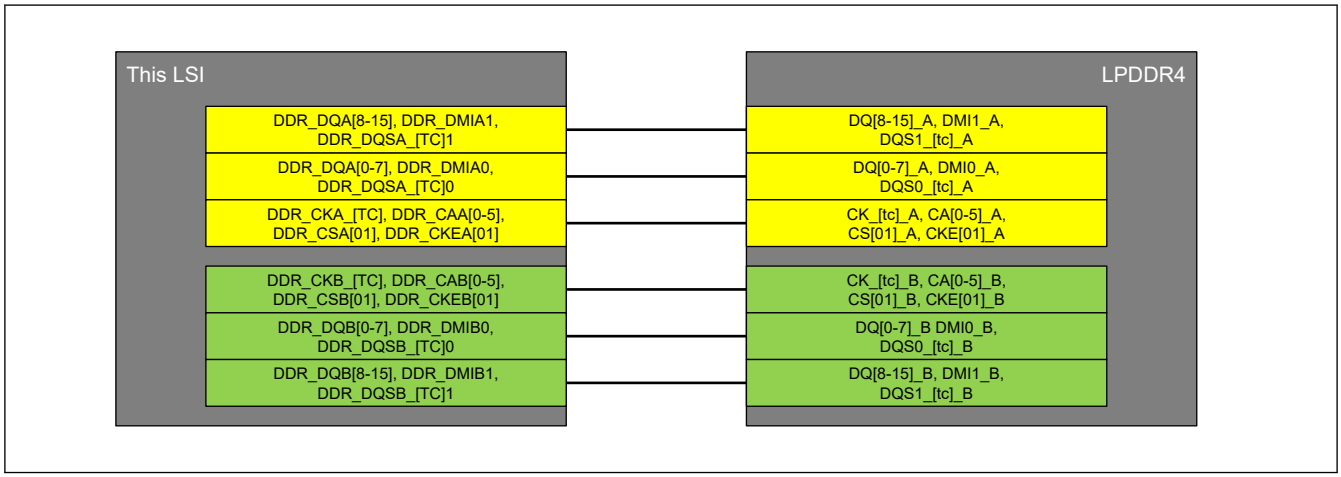


Figure 57.3 Case 1, No Swizzling

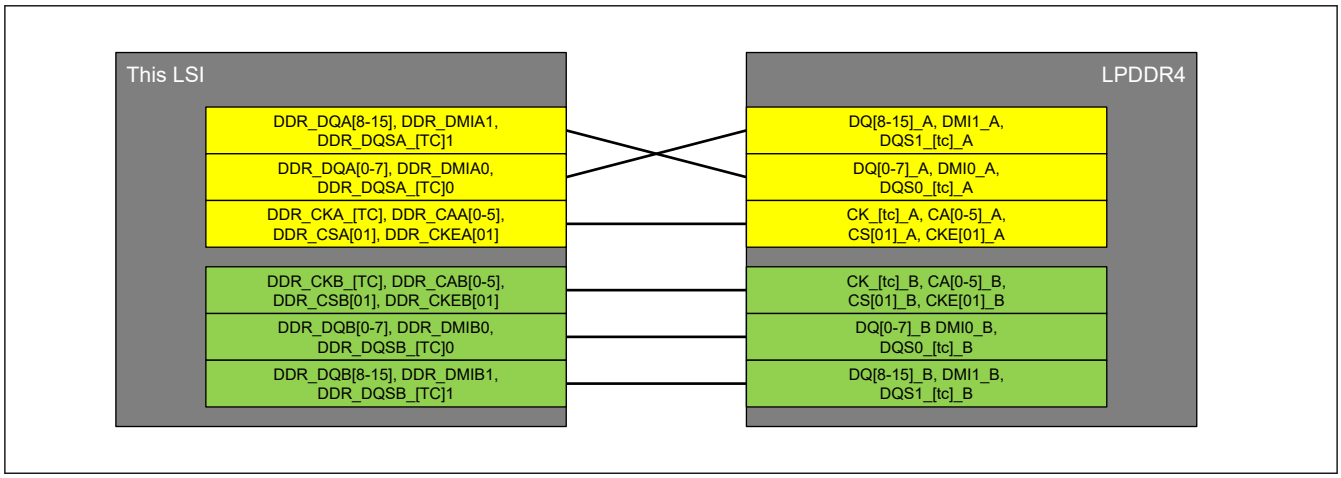


Figure 57.4 Case 2-1, Swapping Upper/Lower of DQ/DMI/DQS in Ch-A

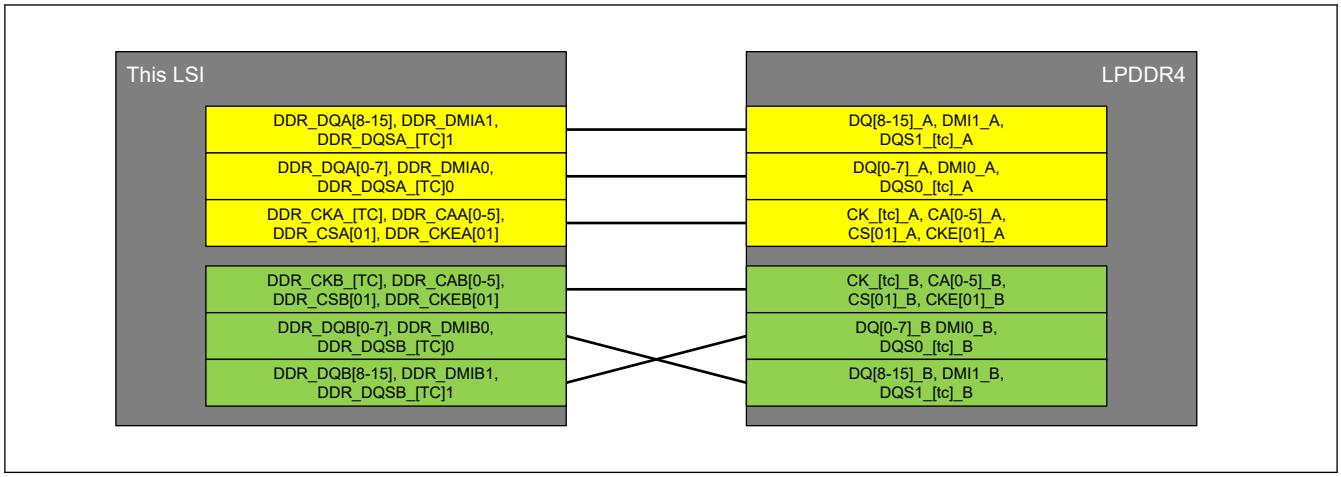


Figure 57.5 Case 2-2, Swapping Upper/Lower of DQ/DMI/DQS in Ch-B

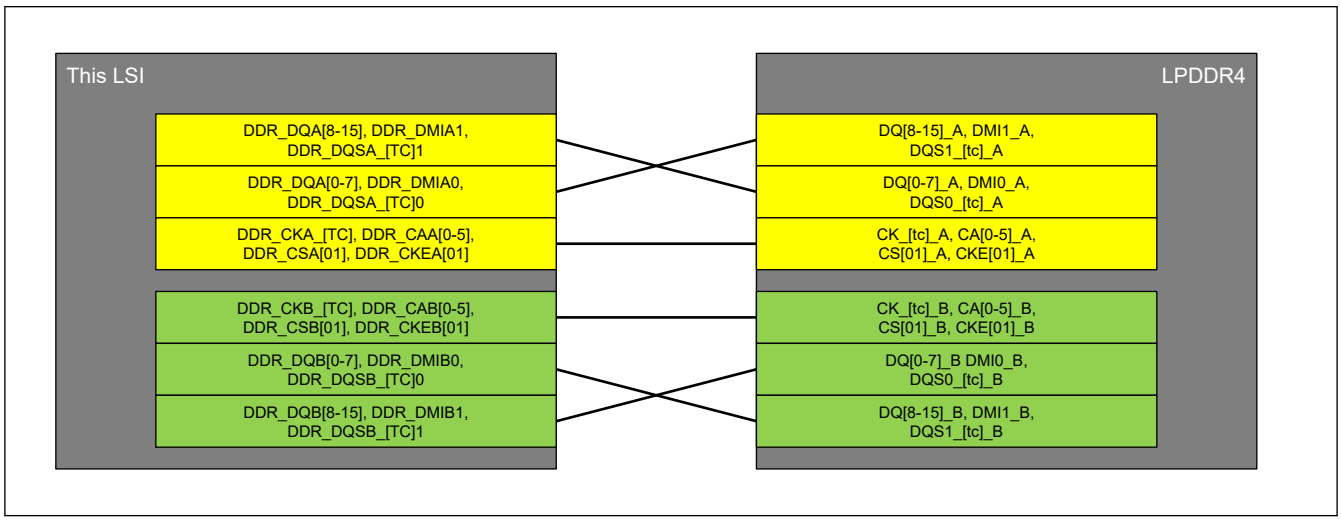


Figure 57.6 Case 2-3, Swapping Upper/Lower of DQ/DMI/DQS in Ch-A/B

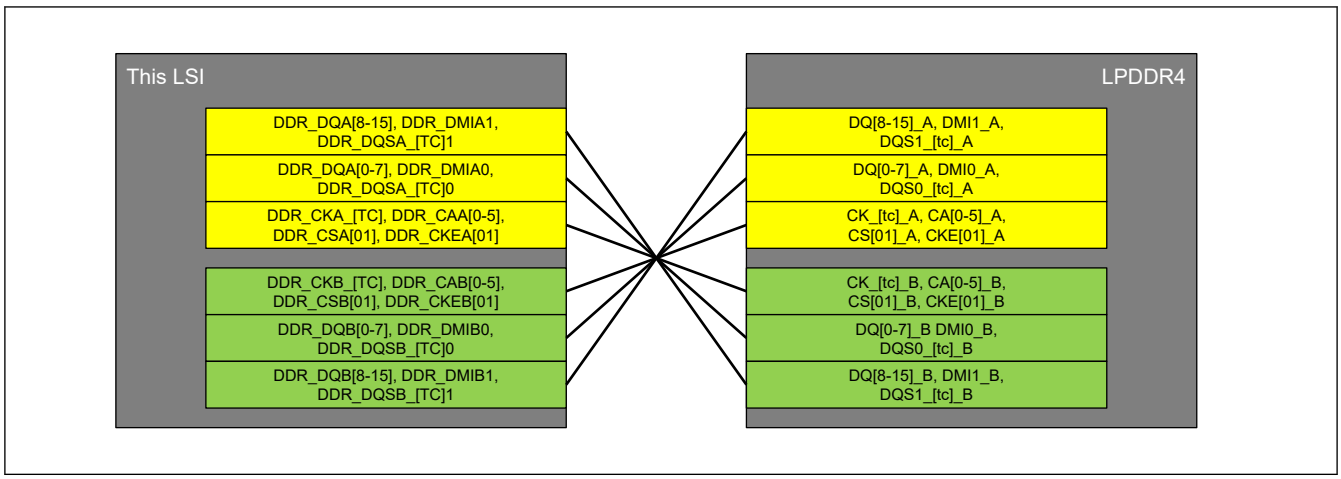


Figure 57.7 Case 3, Swapping Ch-A/B

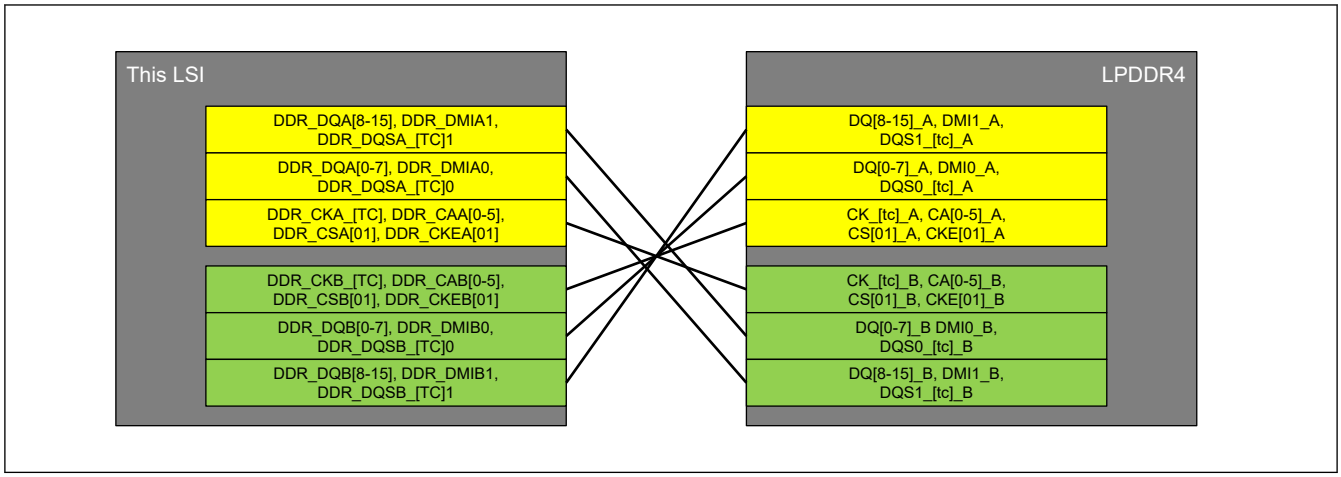


Figure 57.8 Case 4-1, Swapping Ch-A/B and Swapping Upper/Lower of DQ/DMI/DQS in Ch-A

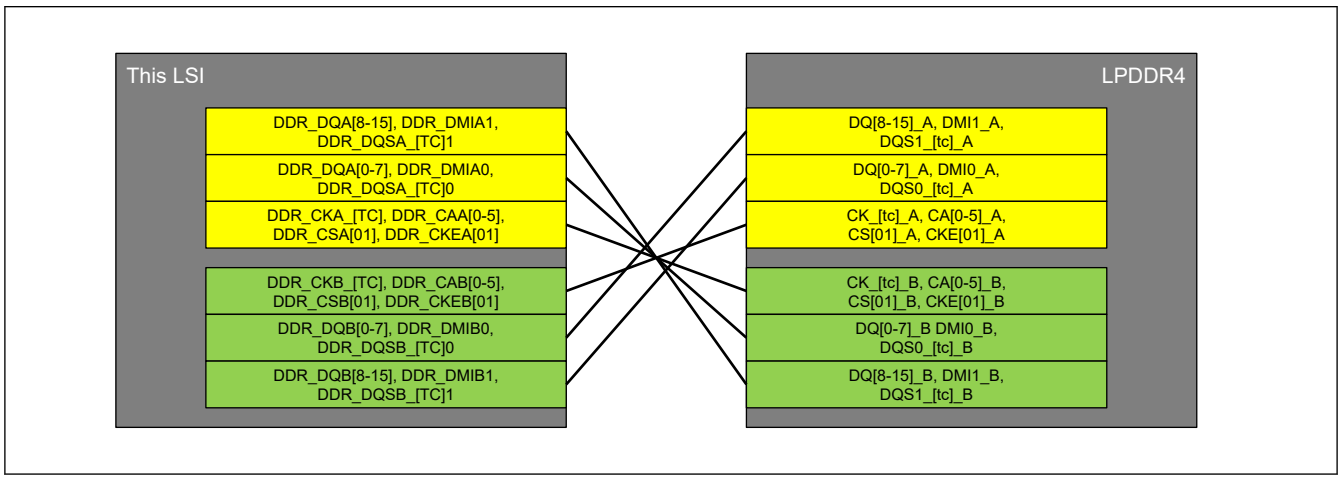


Figure 57.9 Case 4-2, Swapping Ch-A/B and Swapping Upper/Lower of DQ/DMI/DQS in Ch-B

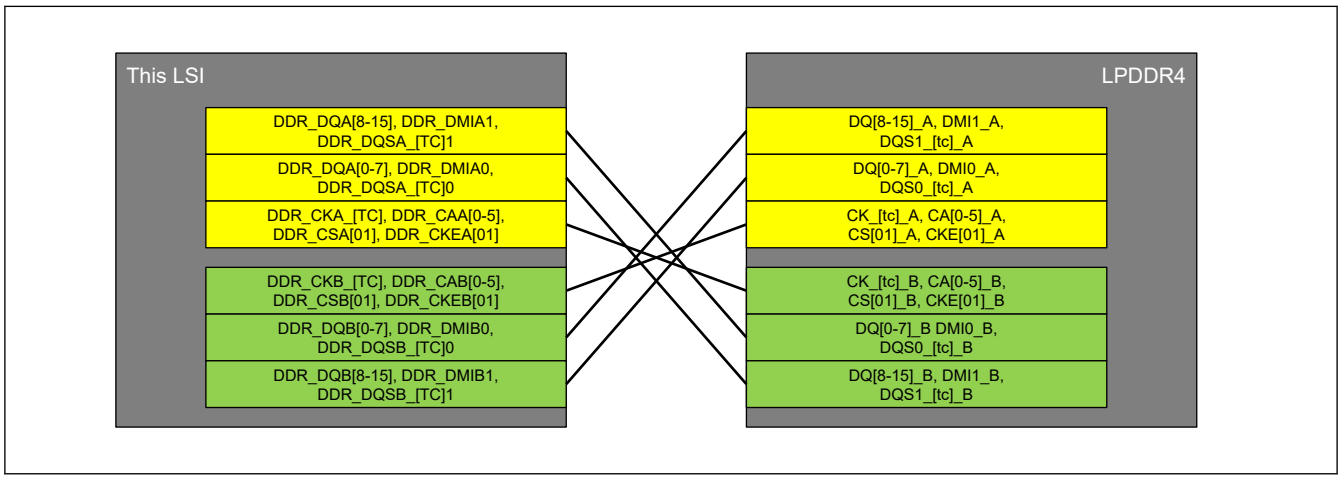


Figure 57.10 Case 4-3, Swapping Ch-A/B and Swapping Upper/Lower of DQ/DMI/DQS in Ch-A/B

(b) Example of Bit swizzling

Examples of DQ bits swizzling are shown. It is able to swizzle DQ bits within the same lane (Figure 57.11), but it is not able to swizzle DQ bits across lanes (Figure 57.12).

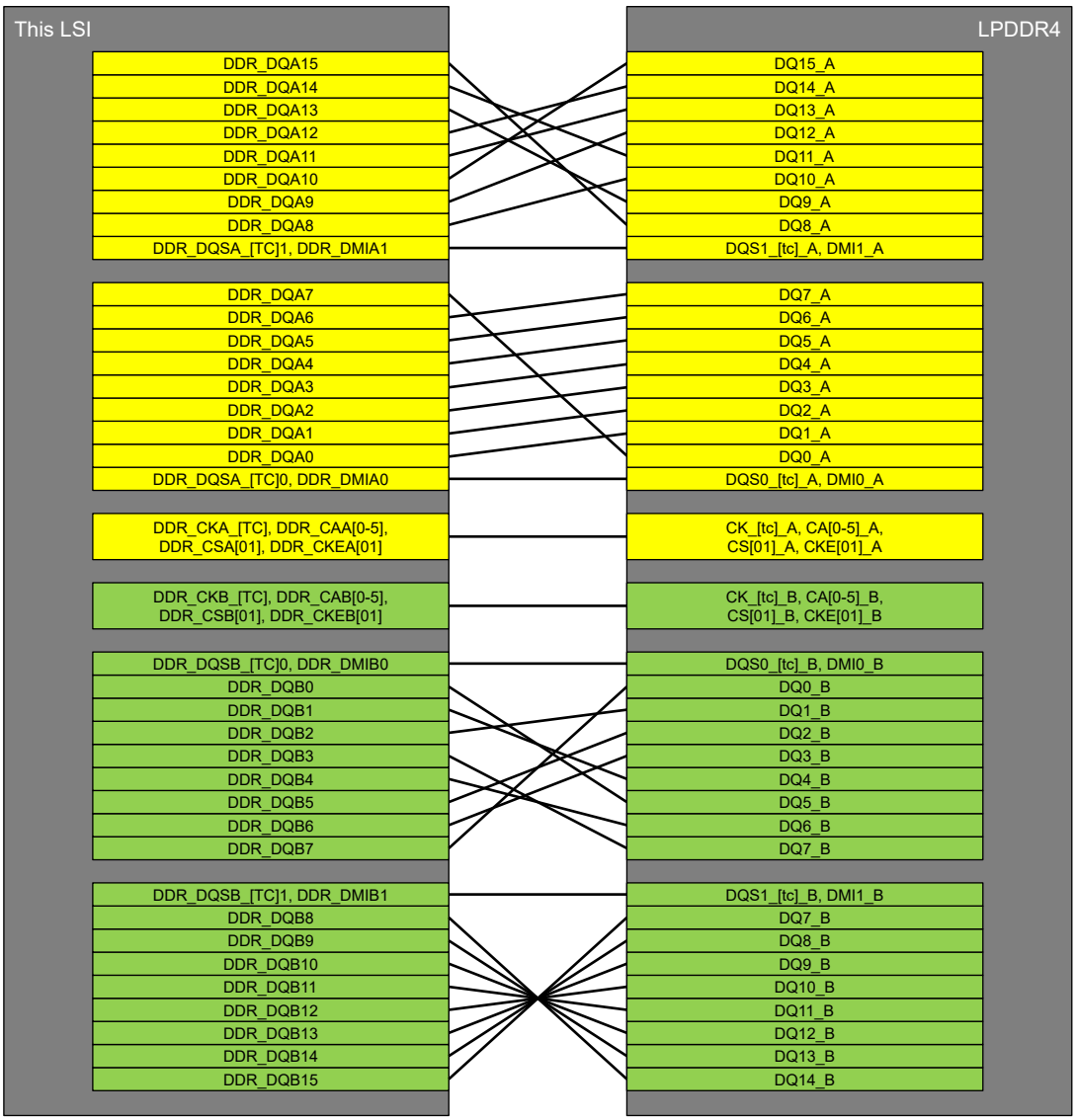


Figure 57.11 DQ Bits Swizzling within Same Lane (Possible)

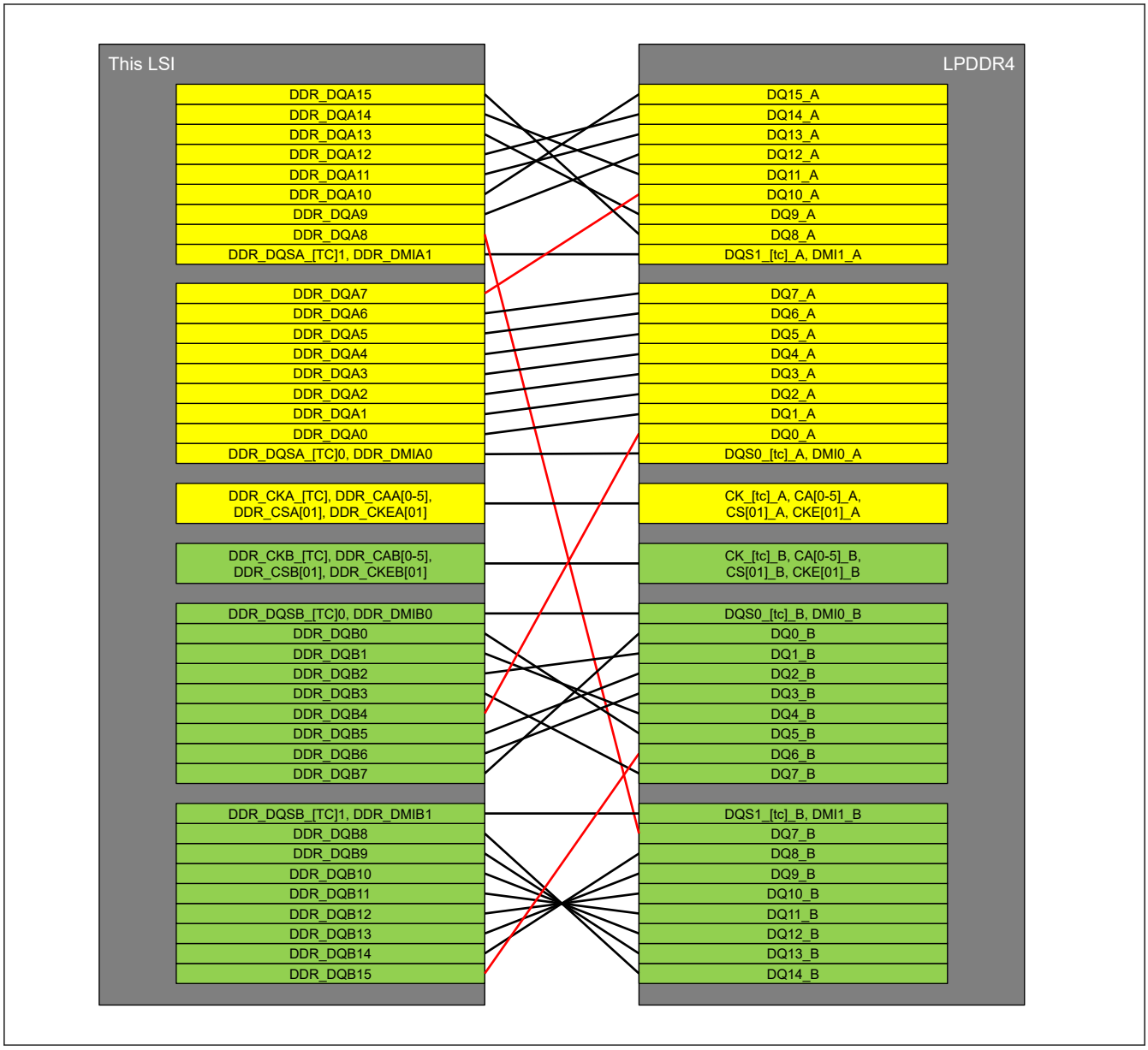


Figure 57.12 DQ bits Swizzling across Lanes (Impossible)

Examples of CAA/CAB bits swizzling are shown. It is able to swizzle CAA/CAB bits within the same channel (Figure 57.13), but it is not able to swizzle CAA/CAB bits across channels (Figure 57.14).

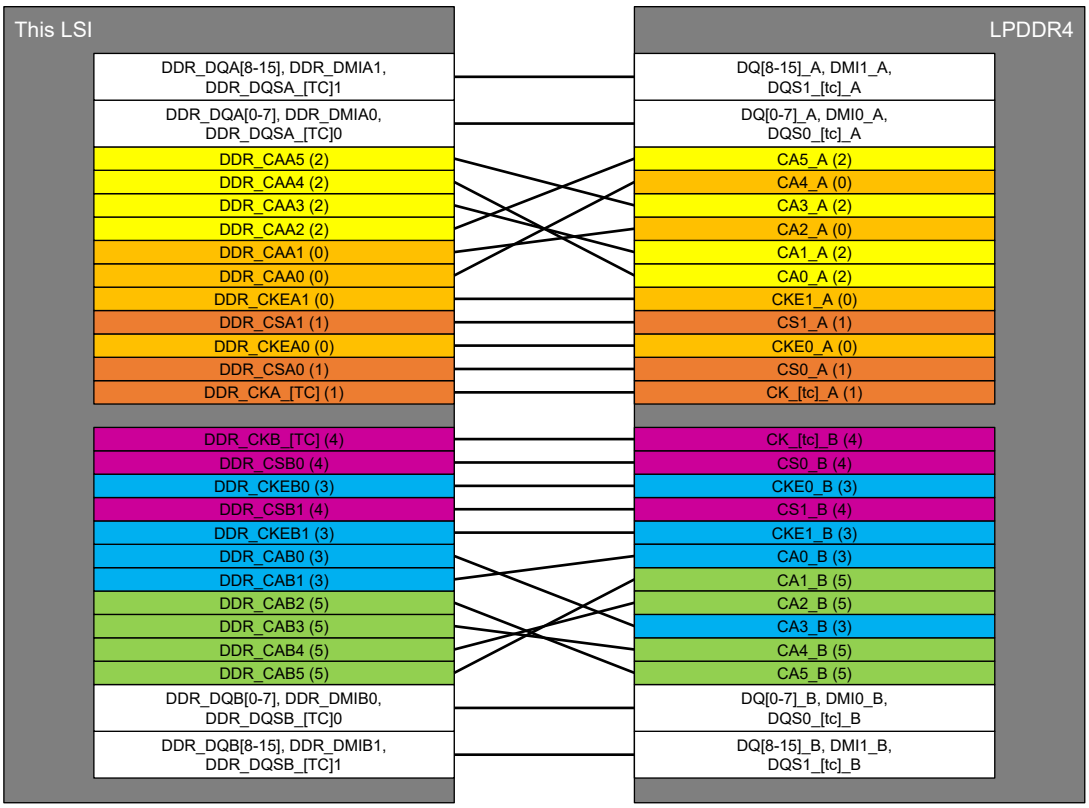


Figure 57.13 CAA/CAB Bits Swizzling within Same Channel (Possible)

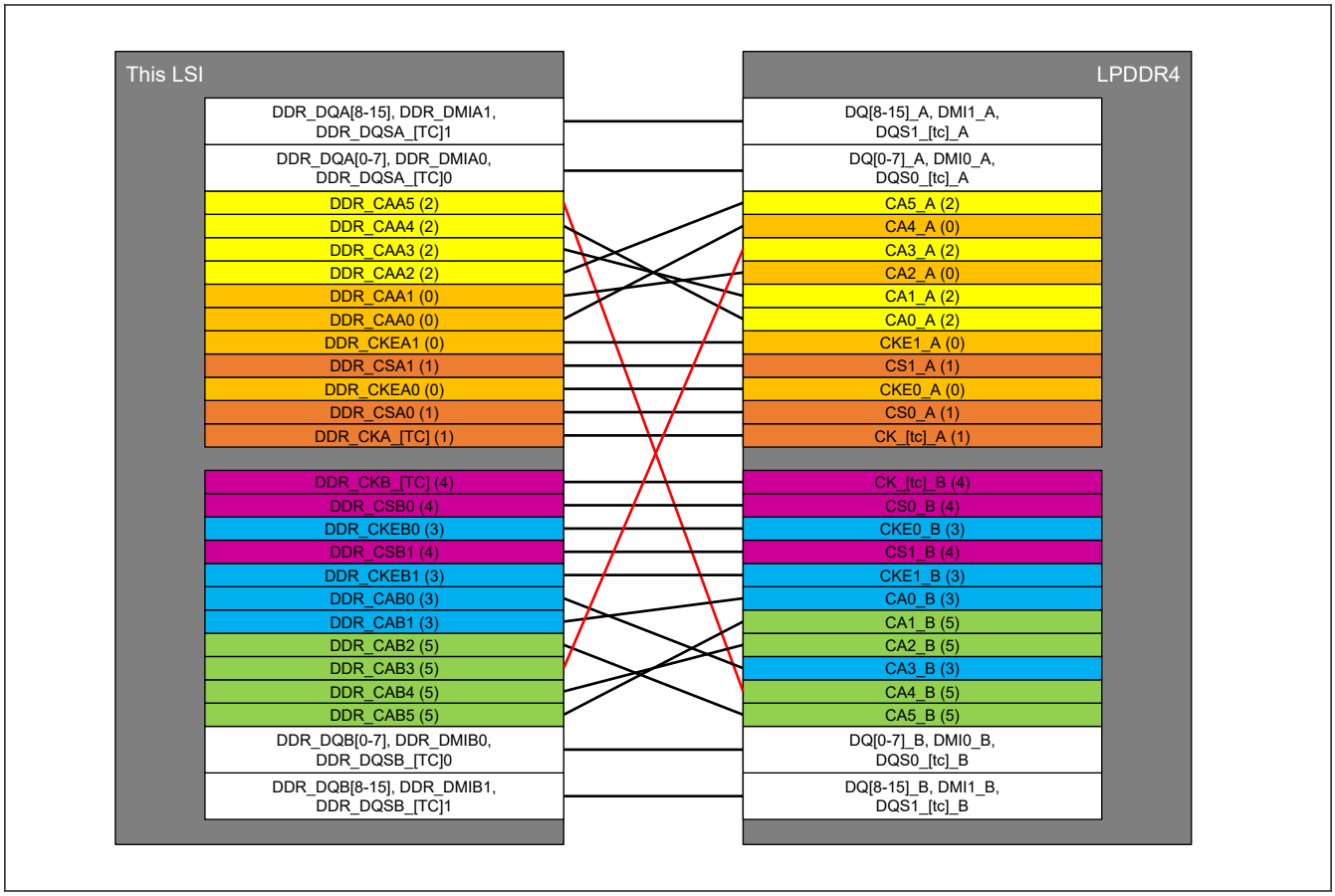


Figure 57.14 CAA/CAB Bits Swizzling across Channels (Impossible)

CSA/CKEA and CSB/CKEB can only be swizzled as shown in Figure 57.15.

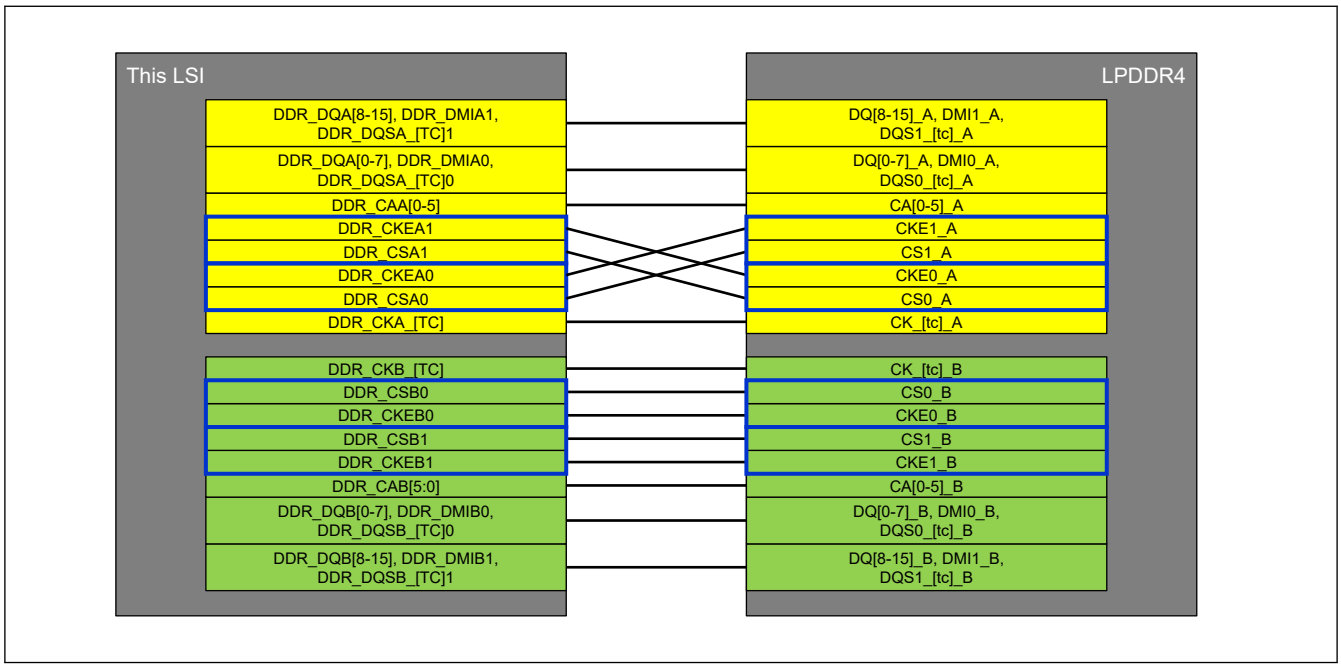


Figure 57.15 CSA/CKEA and CSB/CKEB Bits Swizzling (Possible)

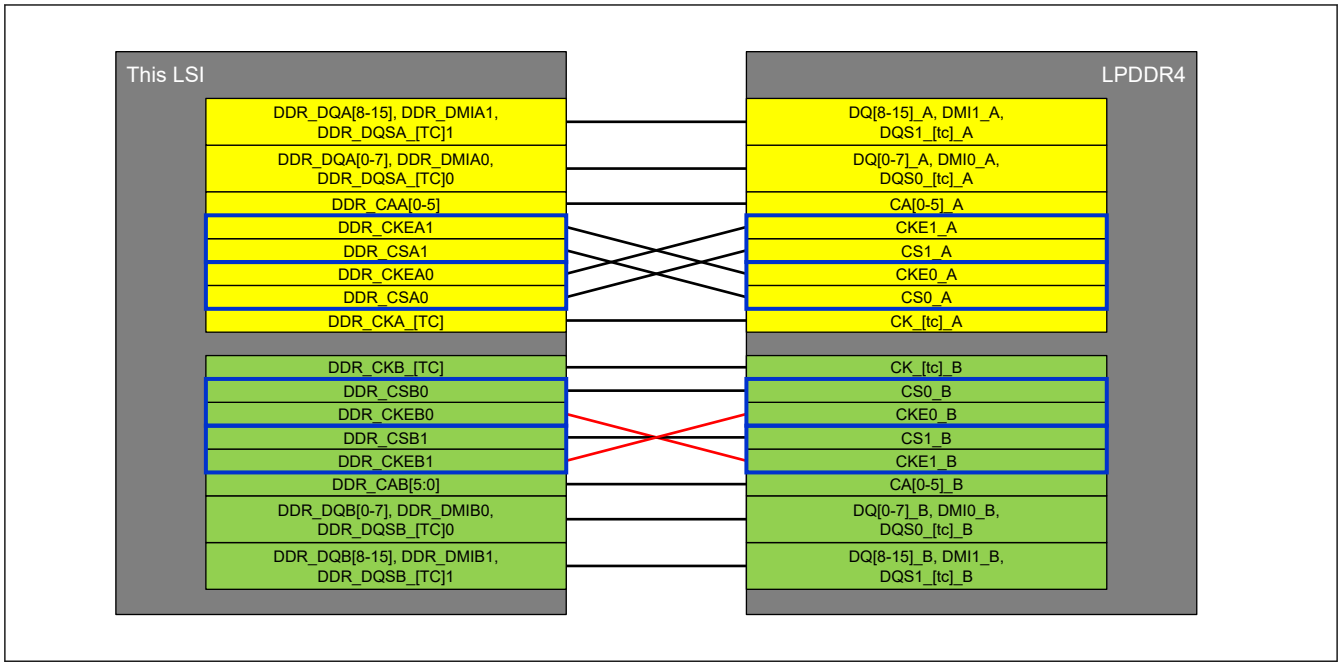


Figure 57.16 CSA/CKEA and CSB/CKEB Bits Swizzling (Impossible)

57.4.2 Multi-Port Arbiter

The Arbiter is responsible for arbitrating requests from the ports and sending requests to the Controller core. Each transaction received at the Arbiter logic has an associated priority, which works with each port's arbitration logic to determine how ports issue requests to the Controller core. This unit supports the Weighted Round-Robin arbitration scheme.

The Arbiter logic routes read data from the Controller core to the appropriate port. The requesting port is assumed able to receive the data. Write data from each port is connected directly to its own write data interface in the Controller core, allowing the ports to independently pass write data to the Controller core buffers.

57.4.2.1 Arbitration Overview

The weighted round-robin arbitration scheme is a three-step arbitration system. All commands are routed into priority groups based on the priority of the requests. Then, within each priority group, requests are serviced according to the "weight" (relative priority) of each port. Finally, each priority group presents a single command to the priority select module, which passes the highest priority command on to the Controller core.

This arbitration scheme also supports two additional features. For situations where the priority and the relative priority for multiple commands are identical, a port ordering system in included whereby the user may adjust the order in which the ports are considered. Secondly, for situations where two ports may be related, a mechanism is included which allows a pair of ports to share arbitration bandwidth for bandwidth efficiency.

Weighted round-robin arbitration is a complex arbitration scheme. To understand the operation, each concept must be first understood individually. section 57.4.2.2. Understanding Round-Robin Operation through section 57.4.2.5. Understanding Port Ordering describe the various components of weighted round-robin arbitration. Note that the examples may utilize a greater number of ports and a larger number of priority levels than are available in this unit. This is done intentionally for explanation.

57.4.2.2 Understanding Round-Robin Operation

Round-robin operation is the simplest form of arbitration and is ideal for systems that do not require requests to be treated preferentially to maintain bandwidth or minimize latency. This scheme uses a counter that rotates through the port numbers, incrementing every time a port request is granted.

If the port that the counter is referencing has an active request, and the Controller core command queue is not full, then this request will be sent to the Controller core. If there is not an active request for that port, then the port will be skipped and the next port will be checked. The counter will increment by one whenever any request has been processed, regardless of which port's request was arbitrated.

Round-robin arbitration ensures that each port's requests can be successfully arbitrated into the Controller core every N cycles, where N is the number of ports in this unit. No port will ever be locked out, and any port can have its requests serviced on every cycle as long as all other ports are quiet and the command queue is not full.

An example of the round-robin scheme is shown in Table 57.6. Cycles 0, 2, and 6 show the system behavior when the command queue is full. Cycles 8 and 11 show the system behavior when the port addressed by the arbitration counter does not have an active request. In particular, note cycle 11: The port addressed by the arbitration counter (0) is not requesting, so the counter scans through the other ports, in incrementing order, to find an active request. Port 2 is requesting and therefore wins arbitration, but the counter only increments to port 1 which was the next port in the sequence. All other cycles show normal behavior.

Table 57.6 Round-Robin Operation example

Cycle	Port Addressed by the Arbitration Counter	Ports Requesting				Command Queue Full?	Arbitration Winner	Value of Counter at Next Cycle
		P0	P1	P2	P3			
0	0	Y	Y	Y	Y	Yes	None	0
1	0	Y	Y	Y	Y	No	P0	1
2	1	—	Y	Y	Y	Yes	None	1
3	1	Y	Y	Y	Y	No	P1	2
4	2	Y	—	Y	Y	No	P2	3
5	3	Y	—	—	Y	No	P3	0
6	0	Y	—	Y	—	Yes	None	0
7	0	Y	—	Y	—	No	P0	1
8	1	—	—	Y	—	No	P2	2
9	2	—	—	Y	Y	No	P2	3
10	3	Y	—	—	Y	No	P3	0
11	0	—	—	Y	—	No	P2	1

57.4.2.3 Understanding Transaction Priority

Priorities may be associated with each command or with the requesting port, depending on the value of the axiY_fixed_port_priority_enable parameter. If the axiY_fixed_port_priority_enable parameter is set to 1, all commands from port Y will use the priority defined in the axiY_r_priority or axiY_w_priority parameters. If the axiY_fixed_port_priority_enable parameter is cleared to 0, the priority will be specified with each incoming command through the axiY_ARQOS and axiY_AWQOS signals. Internally, all commands are organized into priority groups based on their priority setting. All commands within a priority group are treated equally for arbitration unless a port has exceeded its allocated bandwidth. The priority value is also used by the placement logic inside the Controller core when filling the command queue.

Note: The axiY_ARQOS and axiY_AWQOS signals are fixed to 000b in this LSI.

To use per-port priorities, the user must program the axiY_r_priority and axiY_w_priority parameters and assert the axiY_fixed_port_priority_enable parameter to 1 before the start parameter is set. If the user wants to change these values when the system is idle, the axiY_fixed_port_priority_enable parameter must be cleared to 0, the parameters may be changed, and then the axiY_fixed_port_priority_enable parameter must be set to 1 again.

A priority value of 0 is the lowest priority, and a priority value of 7 is the highest priority in this unit. The user may program at priority level 7; however, it is best to reserve this priority value so that the placement queue can elevate to this level through aging.

57.4.2.4 Understanding Relative Priority

Inside each priority group, the relative priority is used to determine arbitration. This unit contains 8 identical priority groups with logic that selects between the requests from all commands at that priority level. The relative priority parameters axiY_priorityZ_relative_priority (where Y is the port number and Z is the priority group) "weight" the ports for each level and determine how the priority group will be arbitrated. Figure 57.17 shows this type of arbitration system.

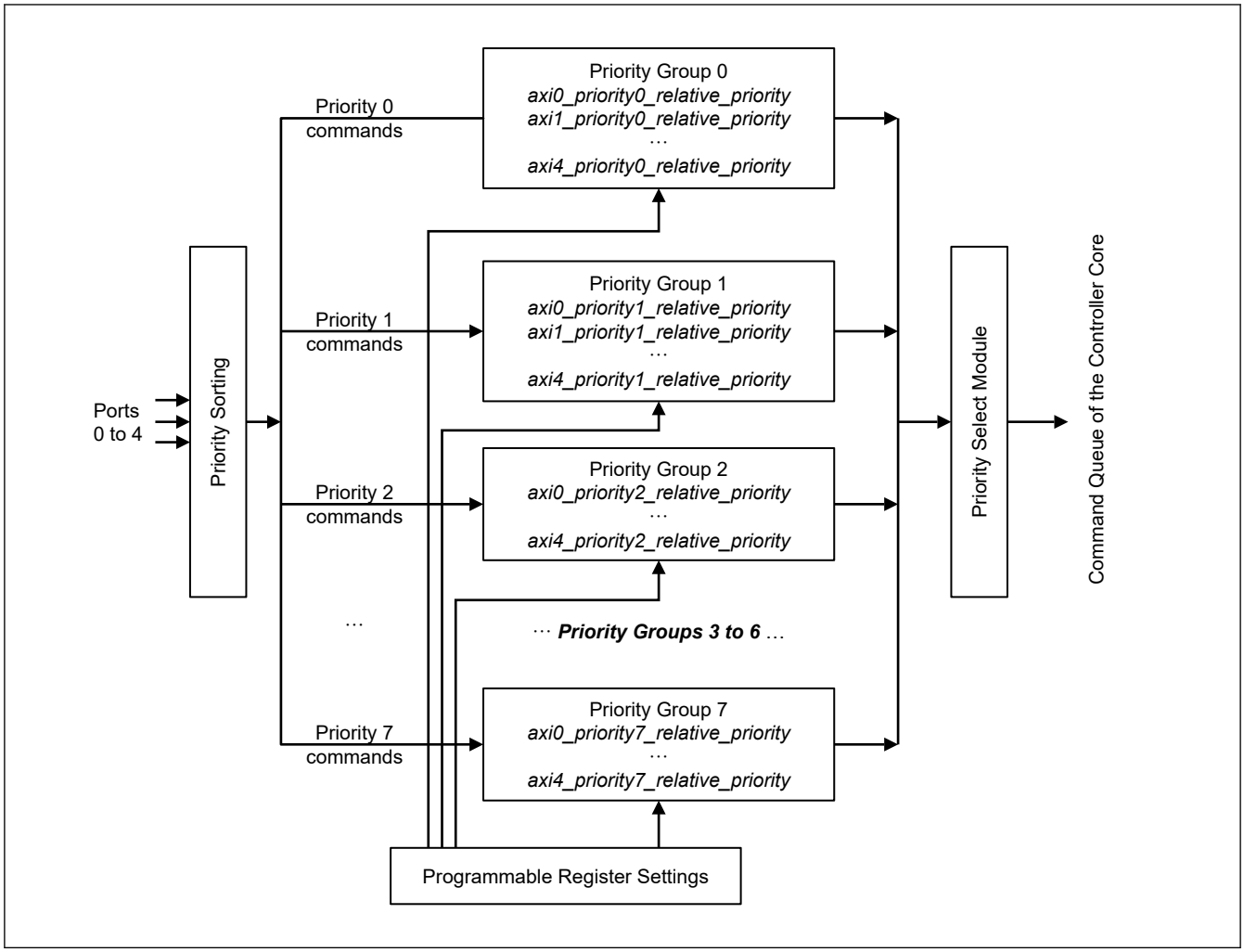


Figure 57.17 Weighted Round-Robin Priority Group structure

By using the relative priority concept, the arbitration is skewed in favor of certain ports based on user programming.

Note that the relative priority parameters have a minimum acceptable value of 1 to prevent port lockout. A 0 value will cause an error condition.

If the relative priorities are all programmed to the same value within any priority group, then the arbitration will mimic a version of simple round-robin scheme within that priority group. Instead of incrementing whenever any request is processed, the simple round-robin counter will only increment to the next port after the value in the axiY_priorityZ_relative_priority parameter number of requests are processed.

Each port Y for priority level Z will be allocated the ratio of that port's relative priority parameter (axiY_priorityZ_relative_priority) to the sum of all requesting port's relative priority values. If a particular port is not requesting, then it is not included in the sum calculation, which means that the arbitration will be split with relative proportions among the requesting ports.

As an example, consider a system with 5 ports where all requests are at priority 0. This system is described in Table 57.7.

Table 57.7 Relative Priority example

Controller parameter	Value
axi0_priority_relative_priority	1
axi1_priority_relative_priority	2
axi2_priority_relative_priority	3
axi3_priority_relative_priority	4
axi4_priority_relative_priority	5

For this system, port 0 will be serviced 1/(1+2+3+4+5) = 1/15 of the time and Port 3 will be serviced 4/(1+2+3+4+5) = 4/15 of the time. However, if Port 2 is not actively requesting, then port 0 will be serviced 1/(1+2+4+5) = 1/12 of the time and port 3 will be serviced 4/(1+2+4+5) = 4/12 of the time.

In order to ensure that relative priorities are maintained, there is a weight counter for each port within each priority group. These counters track the number of transactions accepted for that port in that priority group. When any counter value reaches the programmed relative port priority, the scan order for that priority group will be internally modified. The port that has met its relative priority will be dynamically positioned to the bottom of the scan order (and its counter will be reset), allowing other ports a preferential position.

For ports that are not expected to issue requests at a certain priority level, the associated relative priority parameter should be programmed to 0x1. This allows for minimum allocation without the risk of lockout in case a command appears.

57.4.2.5 Understanding Port Ordering

With simple round-robin arbitration, the ports are scanned based on their port number in incrementing order in the system. Assuming that the command queue is not full, the port referenced by the counter is examined for valid incoming transactions. If there is an active request, it will be accepted. Otherwise, the next port in the scan order will be checked, and its request accepted.

For this unit with weighted round-robin arbitration, the user has the option of adjusting the order that the ports are scanned. This is useful if requests from certain ports are more critical, or if a specific order may reduce contention between ports.

The axiY_port_ordering parameters are used to set this new scan order. A value of 000b gives the highest listing in the scan order, and a value of 111b is the lowest listing in the scan order.

If the 5 axiY_port_ordering parameters are programmed with unique values, then the scan order will be modified to proceed sequentially in this new order. If any of the port ordering parameters have the same value, then those ports will still be equal in the arbitration test. In this case, the port number will select between these ports, with the lower numbered port automatically being selected first.

To demonstrate this concept, consider a system with 5 ports and the two port orders as shown in Table 57.8. For Value 1 of the example, the port ordering parameters all contain unique values, so the resulting order is entirely based on the values of the parameters. For Value 2 of the example, three ports have the same programmed values for the port order. For these three ports, the port number sets the order. Remaining ports follow the port ordering parameters.

Table 57.8 Port Ordering example

Controller parameter	Value 1	Value 2
axi0_port_ordering	3	3
axi1_port_ordering	4	0
axi2_port_ordering	5	5
axi3_port_ordering	1	6
axi4_port_ordering	0	0
Port Scan Order	P4-P3-P0-P1-P2	P1-P4-P0-P2-P3

If all of the port ordering parameters are programmed with the same value, then the scan order will default to the numbered port order.

57.4.2.6 Weighted Round-Robin Arbitration Summary

The Controller weighted round-robin arbitration system combines the concepts of round-robin operation, priority, relative priority and port ordering. The incoming commands are separated into priority groups based on their priority.

Within each priority group, the relative priority values are examined to determine the arbitration winner. If the relative priority values are identical and no individual command can be selected, then the scan order is used to select between the requests. In the end, the highest priority command, from the highest relative priority port, with the highest location in the scan order will be selected and sent to the Controller core.

As an example, consider the system described in Table 57.9. The counters refer to the counters that exist for each port within each priority group to ensure that relative priorities are maintained. For simplification, the command queue is considered to never be full and commands are only received at priority level 0. The behavior is shown in Table 57.10. The

highest port in the scan order that is requesting always wins arbitration, and the scan order is dynamically modified when any port counter reaches its allocated relative priority value. Note that if the command queue was considered, then cycles where the command queue was full would not have any arbitration winner and therefore, the counter values and scan order would not change on that cycle.

Table 57.9 Weighted Round-Robin Arbitration example 1

Controller parameter	Port 0	Port 1	Port 2	Port 3	Port 4
axiY_priority0_relative_priority	4	3	2	1	1
axiY_port_ordering	0	1	2	3	4

Table 57.10 Weighted Round-Robin Arbitration operation 1

Cycle	Ports requesting					Arbitration winner	Next counter					Next scan order
	P0	P1	P2	P3	P4		P0	P1	P2	P3	P4	
	—	—	—	—	—	—	—	—	—	—	—	P0-P1-P2-P3-P4
0	Y	—		Y	—	P0	1	0	0	0	0	P0-P1-P2-P3-P4
1	Y	—	Y	Y	—	P0	2	0	0	0	0	P0-P1-P2-P3-P4
2	Y	Y	Y	Y	—	P0	3	0	0	0	0	P0-P1-P2-P3-P4
3	Y	Y	Y	Y	—	P0	4	0	0	0	0	P1-P2-P3-P4-P0
4	Y	Y	Y	Y	—	P1	0	1	0	0	0	P1-P2-P3-P4-P0
5	Y	Y	Y	Y	Y	P1	0	2	0	0	0	P1-P2-P3-P4-P0
6	Y	Y	Y	Y	Y	P1	0	3	0	0	0	P2-P3-P4-P0-P1
7	Y	—	Y	Y	Y	P2	0	0	1	0	0	P2-P3-P4-P0-P1
8	Y	—	Y	Y	Y	P2	0	0	2	0	0	P3-P4-P0-P1-P2
9	Y	—	—	Y	Y	P3	0	0	0	1	0	P4-P0-P1-P2-P3
10	Y	—	—	Y	Y	P4	0	0	0	0	1	P0-P1-P2-P3-P4
11	Y	—	Y	Y	—	P0	1	0	0	0	0	P0-P1-P2-P3-P4
12	—	—	Y	Y	—	P2	1	0	1	0	0	P0-P1-P2-P3-P4
13	—	—	Y	Y	—	P2	1	0	2	0	0	P0-P1-P3-P4-P2

If the same system also contains two ports that only request at priority level 1, then the system behavior will be slightly altered. These 2 ports creates the second priority group structure that adds to the arbitration complexity. Table 57.11 describes this system.

Again, for simplification, the command queue is considered to never be full and it is assumed that commands from ports 0, 1, and 2 are only received at priority level 0. The behavior is shown in Table 57.12. Note that if any of the priority 0 ports (P0, P1, P2) are requesting, the system behavior will match the behavior when there is only one priority group, as in Table 57.10. Ports 3 and 4 can only win arbitration when no higher priority commands exist.

Table 57.11 Weighted Round-Robin Arbitration example 2

Controller parameter	Port 0	Port 1	Port 2	Port 3	Port 4
axiY_priority0_relative_priority	4	3	2	1	1
axiY_priority1_relative_priority	1	1	1	3	2
axiY_port_ordering	0	1	2	3	4

Table 57.12 Weighted Round-Robin Arbitration operation 2 (1 of 2)

Cycle	Ports requesting					Arbitration winner	Next counter					Next scan order
	P0	P1	P2	P3	P4		P0	P1	P2	P3	P4	
	—	—	—	—	—	—	—	—	—	—	—	Priority 0: P0-P1-P2 Priority 1: P3-P4

Table 57.12 Weighted Round-Robin Arbitration operation 2 (2 of 2)

Cycle	Ports requesting					Arbitration winner	Next counter					Next scan order
	P0	P1	P2	P3	P4		P0	P1	P2	P3	P4	
0	—	—	Y	—	Y	P2	0	0	1	0	0	P0-P1-P2 P3-P4
1	Y	—	Y	—	Y	P0	1	0	1	0	0	P0-P1-P2 P3-P4
2	—	—	Y	—	Y	P2	1	0	2	0	0	P0-P1-P2 P3-P4
3	Y	—	Y	Y	Y	P0	2	0	0	0	0	P0-P1-P2 P3-P4
4	—	—	Y	Y	Y	P2	2	0	1	0	0	P0-P1-P2 P3-P4
5	—	—	—	Y	Y	P3	2	0	1	1	0	P0-P1-P2 P3-P4
6	—	Y	—	Y	Y	P1	2	1	1	1	0	P0-P1-P2 P3-P4
7	—	—	—	Y	Y	P3	2	1	1	2	0	P0-P1-P2 P3-P4
8	—	—	—	Y	Y	P3	2	1	1	3	0	P0-P1-P2 P4-P3
9	—	—	—	Y	Y	P4	2	1	1	0	1	P0-P1-P2 P4-P3
10	—	—	—	Y	—	P3	2	1	1	1	1	P0-P1-P2 P4-P3

57.4.2.7 Priority Relaxing

From Table 57.12, it is evident that commands at lower priority levels will not win arbitration in weighted round-robin arbitration unless there are no higher priority requests. This could mean that, in a situation where high-priority requests are being received continuously, lower-priority requests could be locked out indefinitely. To avoid this scenario and control the arbitration latency for lower-priority commands, it is possible to disable priority groups temporarily. This is known as priority relaxing, and it is a time-controlled function.

Each higher priority group will be temporarily disabled when the pre-set counter value for the lower priority group has been reached and a request is waiting. The axiY_priority_relax parameters set the counter value for port Y at which the priority relax condition will be triggered.

The timing counters inside each port are controlled by the weighted_round_robin_latency_control parameter. When the latency control bit is set to 1, the timing counters are free-running. Any timing counter may hit its axiY_priority_relax parameter value at any point. When this occurs, higher-priority groups are disabled to allow a waiting request for this port to be processed. This results in a random latency for each port, but the maximum latency is fixed at the axiY_priority_relax parameter value. If the current port does not have any commands waiting when the timing counter hits the relax value, then the counter will be reset and the Arbiter will function normally.

When the weighted_round_robin_latency_control parameter is cleared to 0, the timing counters only count while that port has a waiting request that is not being processed. In this case, when the port's axiY_priority_relax parameter value is reached, all priority groups at priority levels higher than the waiting request are disabled. This port's command is granted arbitration and is moved through to the Controller core.

Since the priority relax parameters and counters are associated with individual ports, it is possible that multiple priority relax counters could reach their specified value simultaneously. In this case, the lower priority command will be arbitrated first and then the higher priority command. This situation could alter the arbitration latency slightly, causing it to be longer than the expected value in the priority relax parameter.

Consider the system example as described in Table 57.13. The same conditions apply as for the previous example. The command queue is considered to never be full, commands from ports 0, 1, and 2 are only received at priority level 0, and commands from ports 3 and 4 are only received at priority 1.

Table 57.13 Priority Relaxing example

Controller parameter	Port 0	Port 1	Port 2	Port 3	Port 4
axiY_priority0_relative_priority	4	3	2	1	1
axiY_priority1_relative_priority	1	1	1	2	1
axiY_port_ordering	0	1	2	4	3

Table 57.14 shows the system behavior. The exact settings of the latency control and priority relax parameters are not shown. Instead, the "Relaxed Ports" column indicates which, if any, ports have hit their priority relax values. The following cycles are important to observe:

- Cycles 1 and 7: A port relaxes while a higher priority request and a higher scan order request are both present. The relaxed port still wins arbitration.
- Cycle 4: Two ports of the same priority relax. The higher scan order request wins arbitration.
- Cycle 5: Two ports of different priorities relax. The lower priority port that relaxed wins arbitration. The higher priority port that relaxed will maintain its relax condition, and win arbitration in the next cycle.

Table 57.14 Priority Relaxing operation

Cycle	Ports requesting					Relaxed ports	Arbitration winner	Next counter					Next scan order
	P0	P1	P2	P3	P4			P0	P1	P2	P3	P4	
	—	—	—	—	—	—	—	—	—	—	—	—	Priority 0: P0-P1-P2 Priority 1: P4-P3
0	—	—	Y	Y	Y	—	P2	0	0	1	0	0	P0-P1-P2 P4-P3
1	—	—	Y	Y	Y	P4	P4	0	0	1	0	1	P0-P1-P2 P3-P4
2	—	—	Y	Y	Y	—	P2	0	0	2	0	0	P0-P1-P2 P3-P4
3	—	Y	—	Y	Y	—	P1	0	1	0	0	0	P0-P1-P2 P3-P4
4	Y	—	—	Y	Y	P3, P4	P3	0	1	0	1	0.	P0-P1-P2 P3-P4
5	Y	—	—	Y	Y	P0, P4	P4	0	1	0	1	1	P0-P1-P2 P3-P4
6	Y	—	—	Y	—	P0	P0	1	1	0	1	0	P0-P1-P2 P3-P4
7	Y	—	—	Y	Y	P3	P3	1	1	0	2	0	P0-P1-P2 P4-P3
8	Y	Y	Y	—	Y	—	P0	2	1	0	0	0	P0-P1-P2 P4-P3
9	—	Y	Y	—	Y	—	P1	2	2	0	0	0	P0-P1-P2 P4-P3
10	—	Y	Y	—	Y	P2	P2	2	2	1	0	0	P0-P1-P2 P4-P3

Priority relaxing allows low-priority commands to be able to move through the Arbiter to the Controller core. This will ensure that the system can meet maximum latency requirements.

57.4.2.8 Port Pairing

This unit Arbiter incorporates a feature that allows adjacent ports to be grouped together and considered jointly for arbitration. The weighted_round_robin_weight_sharing parameter controls this function, with 1 bit per pair of ports in this unit. Bit [0] controls ports 0 and 1, bit [1] controls ports 2 and 3. If this unit interfaces to an odd number of ports, the highest-numbered port (port 4) is excluded from the port pairing system.

Since the ports are grouped together, their relative priorities are not considered separately. Referring to section 57.4.2.4. Understanding Relative Priority, the general formula for port priority allocation is the ratio of that port's relative priority parameter (axiY_priorityZ_relative_priority) to the sum of all requesting port's relative priority values. In this case, the relative priority value of only one of the paired ports is used for the sum calculation. This means that the bandwidth will be divided differently among the ports.

If the port pair is at the top of the scan order, and either of the ports is requesting, then the requesting port will win arbitration. If both are requesting, port ordering is used to determine which port wins arbitration. Note that when the ports are paired, their scan order can never be altered and they will always remain together in the scan order. Their counters increment together, and so when they reach their relative priority value, the port pair will dynamically be placed at the bottom of the scan order for that priority group.

In order for port weight sharing to be used, the relative priority parameters for the port pair must be programmed to the same value and the port order of the paired ports should be sequential. If either condition is not followed, an error bit will be set to 1.

Consider the system example as described in Table 57.15. Again, for simplification, the command queue is considered to never be full, commands from ports 0, 1, and 2 are only received at priority level 0 and commands from ports 3 and 4 are always at priority 1. However, now ports 0 and 1 are paired.

Table 57.15 Port Pairing example

Controller parameter	Port 0	Port 1	Port 2	Port 3	Port 4
axiY_priority0_relative_priority	3	3	2	1	1
axiY_priority1_relative_priority	1	1	1	2	2
axiY_port_ordering	0	1	2	4	3
weighted_round_robin_weight_sharing	1 (Paired)		0 (Not Paired)		—

Table 57.16 shows the system behavior with port pairing. Since ports 3 and 4 are still at a lower priority, they will be ignored unless none of the higher priority ports (P0, P1, and P2) are requesting. Note the following points:

- When either port of a port pair wins arbitration, the counters for both ports of the pair increment.
- In Cycle 3, the port pair P0/P1 reaches its allocated relative priority. Note that the port pair dynamically moves to the bottom of the scan order.

Table 57.16 Port Pairing operation (1 of 2)

Cycle	Ports requesting					Arbitration winner	Next counter					Next scan order
	P0	P1	P2	P3	P4		P0	P1	P2	P3	P4	
	—	—	—	—	—	—	—	—	—	—	—	Priority 0: P0-P1-P2 Priority 1: P4-P3
0	Y	—	Y	—	—	P0	1	1	0	0	0	P0-P1-P2 P4-P3
1	Y	—	Y	—	Y	P0	2	2	0	0	0	P0-P1-P2 P4-P3
2	—	—	Y	—	Y	P2	2	2	1	0	0	P0-P1-P2 P4-P3
3	Y	Y	—	—	Y	P0	3	3	1	0	0	P2-P0-P1 P4-P3
4	—	Y	—	—	Y	P1	1	1	1	0	0	P2-P0-P1 P4-P3
5	—	—	—	—	Y	P4	1	1	1	0	1	P2-P0-P1 P4-P3
6	—	—	—	Y	—	P3	1	1	1	1	1	P2-P0-P1 P4-P3
7	—	—	—	Y	Y	P4	1	1	1	1	2	P2-P0-P1 P3-P4

Table 57.16 Port Pairing operation (2 of 2)

Cycle	Ports requesting					Arbitration winner	Next counter					Next scan order
	P0	P1	P2	P3	P4		P0	P1	P2	P3	P4	
8	—	Y	Y	Y	Y	P2	1	1	2	1	0	P0-P1-P2 P3-P4
9	—	Y	—	Y	Y	P1	2	2	0	1	0	P0-P1-P2 P3-P4

57.4.2.9 Error Conditions

With the programming complexities of the weighted round-robin arbitration scheme, an error reporting mechanism is included to notify users of illegal programming scenarios. These error conditions will each set a bit in the wrr_param_value_err parameter to 1. The potential error conditions are:

- Bit [0] = The 5 axiY_port_ordering parameters do not all contain unique values.
- Bit [1] = Any of the axiY_priorityZ_relative_priority parameters have been programmed with a zero value. A 0 value leads to unknown behavior. The minimum allowable value is 1.
- Bit [2] = Any ports, whose related bit of the weighted_round_robin_weight_sharing parameter is set to 1, do not have the same values in their axiY_priorityZ_relative_priority parameters.
- Bit [3] = For ports whose related bit of the weighted_round_robin_weight_sharing parameter is set to 1, the values of the axiY_port_ordering parameters are not sequential.

If bit [0], [2], or [3] is set to 1 in the wrr_param_value_err parameter, and any of the ports are paired in the weighted_round_robin_weight_sharing parameter, then all weight-sharing data will be ignored during this unit initialization and the ports will be prioritized by port number. If port pairing is not being used, but the bit [0] error condition is set to 1, then ports with a non-unique port ordering are prioritized by port number.

Remark: The user is strongly cautioned against modifying the values of the port ordering or relative priority parameters during active port usage.

57.4.2.10 Programmable Options for Weighted Round Robin Arbitration

This unit's weighted round-robin arbitration scheme provides a great deal of programmable control for the user. The parameters are referenced throughout this section and are summarized here for clarity:

- axiY_fixed_port_priority_enable (5 parameters, 1 for each port Y)
- axiY_r_priority (5 parameters, 1 for each port Y)
- axiY_w_priority (5 parameters, 1 for each port Y)
- axiY_priorityZ_relative_priority (8 parameters for each port Y)
- axiY_port_ordering (5 parameters, 1 for each port Y)
- axiY_priority_relax (5 parameters, 1 for each port Y)
- weighted_round_robin_latency_control
- weighted_round_robin_weight_sharing
- wrr_param_value_err

57.4.3 ECC

The Controller provide an optional error reporting and correcting circuitry that can be used to verify data in memory and correct memory errors if they occur. The logic will check for errors in both the data and the check code on all read transactions.

ECC, or error checking and correcting, is the process of detecting bit errors in the memory data and if possible, correcting them. This function can confirm the accuracy of data and remove or at least identify bit errors.

ECC works by creating unique "check codes" which are a mathematical description of the information in an aligned segment of data known as an "ECC data word". The check code is always related to the entire ECC data word, and is used inside the Controller to control data accuracy. These check codes are never input from, or output to, the user interface.

An ECC data word cannot start and end at any random address; these words are memory aligned to their size. The starting addresses of ECC data words are defined as ECC word boundaries and the alignment of user transactions to these boundaries determines how transactions are processed inside the Controller.

The Controller supports a 64-bit ECC data word size. An 8-bit check code is maintained for each 64-bit memory area. ECC word boundaries fall on each 8-byte address (0xN0, 0xN8).

57.4.3.1 ECC Error Types

An ECC error is defined in the Controller as "correctable" or "un-correctable". A correctable error is a single-bit error in the check code or the data. The Controller uses the syndrome to determine which bit is erroneous and is able to correct the error. An un-correctable error is a double-bit error in the check code and/or the data. In this case, the Controller is able to identify that 2 bits in the check code and/or the data are incorrect, but cannot determine exactly which 2 bits are erroneous, and therefore cannot correct the error.

Note: The Controller can detect single bit and double bit errors. Errors in more than 2 bits of the check code and/or the data will cause the system to behave unpredictably.

57.4.3.2 In-line ECC

Traditionally, there is a dedicated memory device to hold ECC codes, requiring an extra memory to be placed and routed, and the data bus to memory to be wider. As an alternative, in order to reduce the memory data bus and eliminate the place and route issues of a separate device, this Controller supports in-line ECC in which a section (1/8th) of the memory device used to store data is used for ECC storage. However, this does mean that use of ECC generates additional memory operations since a separate write to the ECC area will be required after a data write, and a separate read will be required to pull the stored ECC information on read operations.

Use of ECC generation and correction is controlled through the ecc_enable parameter. If this parameter is programmed to a non-zero value, ECC is computed and stored in memory on all write transactions, and read from memory, computed, compared and optionally corrected on all read transactions. If this parameter is cleared to 00b, ECC generation, checking, correction and storage will not occur, and therefore the extra memory operations will not be required.

When using in-line ECC, the bank information may be adjusted to account for the location of the ECC data. The user may define the bank offset in the inline_ecc_bank_offset parameter. The ECC information will be stored in the bank defined by the formula = bank + inline_ecc_bank_offset. This parameter may be cleared to 0x0, or programmed to a non-zero value for better performance.

57.4.3.3 In-line ECC Compatibilities

In-line ECC is enabled when the ecc_enable parameter is programmed to 01b, 10b, or 11b. If in-line ECC is enabled, several parameters in the Controller have required or recommended programming values. See Table 57.17 for a summary of these values. The user must also program the cs_msk_X and cs_val_lower_X / cs_val_upper_X parameters correctly.

Table 57.17 In-line ECC parameter programming

Parameter name	Supported values
addr_cmp_en	should be set to 1 (enabled)
addr_collision_mpm_dis	must be set to 1 (disabled)
disable_memory_masked_write	should be cleared to 0 (allow)
disable_rd_interleave	must be cleared to 0 (allow)
ecc_writeback_en	should be set to 1 (enabled)
in_order_accept	out-of-order (0 setting) can only be enabled if the inline_ecc_same_page parameter = 1
inline_ecc_bank_offset	only valid if the inline_ecc_same_page parameter = 0
inline_ecc_same_page	both bg/bank-based and same-page in-line ECC are supported
rw_same_page_en	must be cleared to 0 (disabled)
swap_en	must be cleared to 0 (disabled)

57.4.3.4 Features of the In-line ECC Logic

The Controller ECC implementation provides the following features:

- Internal controls to disable ECC usage
  The user may disable the ECC usage completely by clearing the ecc_enable parameter to 00b. In this mode, the Controller will not compute and compare check codes, nor attempt to store the check codes to memory.
- Register storage of the ECC signature on ECC errors
  Two sets of parameters (one set for correctable errors, one set for un-correctable errors) store the address, data, and syndrome associated with the ECC error on a read command from a port transaction that generates a first correctable or first un-correctable error interrupt if the ecc_enable parameter is programmed for detection (10b or 11b).
- Interrupt generation
  Four interrupt bits in the register array communicate ECC status to the user for read commands or read/modify/write commands if the ecc_enable parameter is programmed for detection (10b or 11b).
- User interface signaling when an ECC error occurs on a read operation
  Two output signals per port transmit ECC error status with the associated read data when an un-correctable error is detected if the ecc_enable parameter is programmed for detection (10b or 11b).
- Automatic correction of single-bit errors
  If single-bit error occurs and the ecc_enable parameter is programmed for detection and correction (11b), the erroneous bit will be flipped automatically and accurate data/check codes will be written to memory (write transactions) or returned to the user interface (read transactions). Note that corrections made on read data will not be reflected to memory.
- Memory integrity preservation through ECC scrubbing
  On write commands, ECC scrubbing is used to ensure that the entire combined ECC data word will be written to memory to preserve memory integrity. On read commands, if the ecc_enable parameter is programmed for detection and correction (11b), and a correctable error is detected, the Controller will fix the data prior to returning it to the user interface and flag the existence of the error in memory. Since this operation does not include a write, the error will still exist in the memory. If the ecc_enable parameter is only programmed for detection (10b), the Controller will NOT fix the data but will flag the error. The user has two options to handle these read errors, if removing the error in memory is desired. If the ecc_enable parameter is programmed to 11b, the user may enable the ECC writeback option (by setting the ecc_writeback_en parameter to 1) which will automatically trigger a masked write to write the newly corrected bit to memory. Or, if the option is not enabled, there will be no automatic command generated, but the user may respond to a correctable error on a read by issuing a write to this memory location with all bytes masked. This will trigger a read/modify/write operation, where the Controller will read in the data from memory, discover and correct the error and then write the data and the check code to memory. Since all bytes of the data are written to memory, the error is overwritten - or scrubbed - from memory. ECC scrubbing requires that ECC detection and correction are both enabled (the ecc_enable parameter is programmed to 11b).
- Automatic corruption of ECC codes for read data errors
  If ECC detection is enabled (10b or 11b), the read data is verified for ECC prior to combining the read data with the new write data and calculating a new ECC. If an un-correctable error is detected on the read data, it is likely that this address space contains bad memory. As a result, the Controller will automatically corrupt the check code for this ECC data word so that all future accesses to this area will result in an error. This feature may be disabled. If disabled, the error will be ignored and new data and a new check code will be written to memory. A future access to this address may or may not result in an un-correctable error.
- Separate set of recording parameters for read/modify/write ECC errors
  In addition to the ECC signature register storage and interrupts, a new set of ECC signature parameters and interrupts is included for read/modify/write ECC errors.

57.4.3.5 Buffer Allocation / Optimization

A normal read operation with in-line ECC consists of two read operations: one to fetch memory data and one to fetch the ECC data. A normal write operation similarly consists of two write operations: a write to store memory data and a write to store the ECC data that is related. As a single transaction, this is meaningless, but multiplying every transaction by 2 results in a large overhead and system inefficiency.

There are 8 ECC buffers in this Controller. Each buffer stores one memory burst of ECC data which corresponds to 8 aligned, continuous memory bursts. Having multiple buffers allows each to operate independently and store ECC data

related to 8 different commands. This Controller takes advantage of having multiple buffers and attempts to re-use ECC data whenever possible through two user-controllable optimizations: read caching and write combining.

Once a read transaction is complete, the Controller will look ahead at the commands in the command queue. If any read commands are pending that will require the same ECC data that is currently in a buffer, the Controller can re-allocate the existing buffer to the pending read command and nullify the associated pending ECC data read. Since an ECC buffer may contain 8 transactions worth of read ECC values, it is possible that this option could eliminate 7 extra ECC data reads. Read caching is controlled through the programming of the ecc_read_caching_en parameter.

Write operations will compute ECC values on the data, then store the data and then the ECC values before proceeding to the next command. However, if a pending command will overwrite at least some of that ECC information in the buffer, then it may not be necessary to perform the ECC data write until both (or more, up to 7) transactions are complete. Write combining is controlled through the programming of the ecc_write_combining_en parameter.

57.4.3.6 In-line ECC Control

ECC functionality is controlled through a parameter named ecc_enable. This parameter enables ECC and sets the reporting and correcting behavior.

Table 57.18 shows the bit settings for the ecc_enable parameter.

Table 57.18 ECC control parameter settings

ecc_enable	ECC enabled	Error detection	Error correction	Description
00b	No	—	—	ECC is disabled. The ecc_dataout_corrected_X / ecc_dataout_uncorrected_X signals will never assert during read data transfers. The int_status_ecc parameter and the ECC reporting parameters will never be updated to reflect any ECC error events. Data will be written to the memory without ECC values, and data will be returned to the user interface without being verified for accuracy.
01b	Yes	No	No	ECC is enabled, but without detection or correction. ECC is computed on all transactions, but the ecc_dataout_corrected_X / ecc_dataout_uncorrected_X signals will never be asserted, the int_status_ecc parameter will not reflect ECC errors, and the ECC reporting parameters will not be updated.
10b	Yes	Yes	No	ECC is enabled with detection, but correction is not supported. When an un-correctable error is found on a read operation, the ecc_dataout_corrected_X / ecc_dataout_uncorrected_X signals will be asserted with the erring ECC data word. The int_status_ecc parameter and the ECC reporting parameters will be updated for read commands. Erroneous data will be returned to the user on read commands and written to the memory on write commands.
11b	Yes	Yes	Yes	ECC is enabled with detection and correction. When an un-correctable error is found on a read operation, the ecc_dataout_corrected_X / ecc_dataout_uncorrected_X signals will be asserted with the erring ECC data word. The int_status_ecc parameter and the ECC reporting parameters will be updated for read commands. Single bit errors will be corrected automatically by the Controller in both read and write commands.

Additional ECC-related parameters are listed in Table 57.19.

Table 57.19 ECC parameters

Controller parameter	Description
ecc_disable_w_uc_err	Disables automatic corruption of the ECC codes for an entire user word when the read portion of a read/modify/write operation has an un-correctable error. If this is not disabled, the corruption will occur even if the erroneous byte is overwritten with new data.
ecc_writeback_en	Enables automatic writing of corrected data on single bit correctable errors on read operations. This is only valid if the ecc_enable parameter is programmed to 11b.

57.4.3.7 Error Signature Parameters

Register storage is provided for the failing address, data, and syndrome for ECC single bit and double bit errors for read operations from port transactions when the ecc_enable parameter is programmed for detection (10b or 11b). There is one set of parameters for each type of error.

The signature information will be stored until the user clears the first correctable error or first un-correctable error interrupt by writing a 1 to the associated bit in the int_ack_ecc parameter. Subsequent errors that occur before this error interrupt is cleared will trigger an interrupt and an output signal, but the error signature will not be recorded. Table 57.20 lists the parameters in which the error signature is captured.

Table 57.20 ECC error signature parameters

Information	Correctable error	Un-correctable error
Address	ecc_c_addr	ecc_u_addr
Data	ecc_c_data	ecc_u_data
Syndrome	ecc_c_synd	ecc_u_synd

57.4.3.8 Interrupt Status Bits

The int_status_ecc parameter indicates the status of all interrupts in the Controller. The parameter is cleared on reset of the Controller.

There are 4 bits in the int_status_ecc parameter relating to ECC when the ecc_enable parameter is programmed for detection (10b or 11b).

Table 57.21 ECC Error Parameter Description

Bit	Description
int_mask_ecc [3:0]	Mask field to inhibit assertion of ECC interrupt signals.
int_status_ecc [0]	Set to 1 if a correctable ECC event has been detected on a read operation.
int_status_ecc [1]	Set to 1 if another correctable ECC event has been detected on a read operation, prior to the initial event being acknowledged.
int_status_ecc [2]	Set to 1 if an un-correctable ECC event has been detected on a read operation.
nt_status_ecc [3]	Set to 1 if another un-correctable ECC event has been detected on a read operation, prior to the initial event being acknowledged.
int_status_ecc [6]	Set to 1 if one or more ECC writeback commands could not be executed.
nt_status_ecc [7]	Set to 1 if the scrub operation triggered by setting the ecc_scrub_start parameter has completed.
int_status_ecc [8]	Set to 1 if an ECC correctable error has been detected in a scrubbing read operation.

On the first instance of an ECC error, the associated bit will be set in the int_status_ecc parameter. If the associated interrupt is not masked, the controller_int signal will also be asserted to the user interface. An interrupt is masked if the associated bit is set to 1 in the int_mask_ecc parameter. Bits [1] and [3] will only be set in the int_status_ecc parameter if a subsequent ECC error occurs before the initial ECC error was acknowledged. To acknowledge the first error, the user must set the associated bit in the int_ack_ecc parameter to 1.

57.4.3.9 Automatic In-line ECC Corruption

For ECC accuracy, data is verified for ECC during the read phase of a write prior to modifying and writing. If an un-correctable error is detected on the read data, it is likely that this address space contains bad memory. As a result, the Controller will automatically corrupt the check code for this ECC data word so that all future accesses to this area will result in an error. The corruption occurs during both the read and the write phases of the write transaction.

If desired, this feature may be disabled by setting the ecc_disable_w_uc_err parameter. If disabled, the un-correctable error will be ignored. The read data and write data will be combined, and a check code will be calculated to match the new ECC data word. New data and the new check code will be written to memory. A future access to this address may or may not result in an un-correctable error.

57.4.3.10 Clearing a Reported In-line ECC Event

To clear a reported ECC event, the user should follow these steps:

- Read the ECC data, address, and syndrome parameters to determine where the event occurred.
- Set the associated bit in the int_ack_ecc parameter to 1. This clears the ECC interrupt bit, and allows the ECC event data in the ECC parameters to be overwritten by a subsequent ECC event.

57.4.3.11 In-line ECC Scrubbing

The Controller's ECC logic includes a scrubbing feature which periodically reads each location of the memory, checks ECC, corrects any correctable errors found, and writes back the corrected data. By walking through the entire DRAM, the memory integrity is preserved, and the likelihood of an un-correctable error is reduced.

(1) ECC Scrubbing Parameters

Relevant parameters are listed in Table 57.22.

Table 57.22 ECC scrubbing parameters

Controller prameter	Description
ecc_scrub_end_addr	Defines the ending address at which scrubbing operations will stop and the ECC scrubbing operation complete interrupt (bit [7]) will be set to 1 in the int_status_ecc parameter. Once complete, the scrubbing address pointer will be reset to the start address so that when another scrubbing operation is initiated, it will scrub a full cycle from the start address. This parameter must be programmed to a non-zero value for the scrubbing logic to operate.
ecc_scrub_idle_cnt	Defines the number of controller clock cycles that the scrubbing engine will wait in the controller's idle state before starting scrubbing operations. This parameter is only valid when the ecc_scrub_mode parameter is set to 1. The controller is considered idle when the command queue is empty. When this condition is detected, an internal counter will load with the value programmed in this parameter and count down on each controller clock. When the counter expires, either the scrubbing operation will begin or the next address will be tested. The controller clock is based on the controller's operating frequency. Clearing this parameter to 0x0000 will disable idle operation.
ecc_scrub_in_progress	Reports the scrubbing operation status. This parameter is read-only. 0: Not actively performing a scrubbing operation 1: The controller is in the process of performing a scrubbing operation.
ecc_scrub_interval	Sets the minimum interval between two ECC scrubbing commands, in number of controller clock cycles. This parameter is only valid when the ecc_scrub_mode parameter is cleared to 0. The controller clock is based on the controller's operating frequency. Clearing this parameter to 0x0000 will disable interval operation.
ecc_scrub_len	Defines the length (in bytes) of the ECC scrubbing read command that the controller will issue. This value must be an integer multiple of the memory burst length, and the lowest 3 bits of this parameter must be cleared to 0.
ecc_scrub_mode	Specifies when ECC scrubbing operations will be performed. 0: ECC scrubbing operations will be performed at regular intervals as dictated by the ecc_scrub_interval parameter. 1: ECC scrubbing operations will be performed only when the controller is idle.
ecc_scrub_start	Initiates the ECC scrubbing operation. This parameter is write-only and will always read back as 0. 0: No action 1: Initiate the scrubbing operation.
ecc_scrub_start_addr	Defines the starting address from where scrubbing operations will begin. This value must be less than or equal to the value programmed into the ecc_scrub_end_addr parameter.

(2) Procedure

To initiate an ECC scrubbing operation, the user should follow these steps:

1. If the ecc_scrub_in_progress parameter is cleared to 0, program the ecc_scrub_start_addr, ecc_scrub_end_addr and ecc_scrub_len parameters with appropriate values and then set the ecc_scrub_start parameter to 1. The ecc_scrub_start parameter will be automatically cleared after 1 cycle.

2. When the user sets the ecc_scrub_start parameter, the ecc_scrub_in_progress parameter will be set to 1. If the ecc_scrub_mode parameter is cleared to 0, a timer will be loaded with the value in the ecc_scrub_interval parameter. If the ecc_scrub_mode parameter is set to 1, the scrubbing logic will wait for an indication that the Controller is actually idle and then the timer will be loaded with the value in the ecc_scrub_idle_cnt parameter.
3. In either case, when the counter expires, the ECC scrubbing operation will begin.
   The scrubbing engine will start reading the memory at the address defined in the ecc_scrub_start_addr parameter. The Controller will issue a dummy read command which pulls the data and the ECC code from memory into the read data FIFO, calculate the ECC on the read data and compare it to the stored ECC code. If the ECC codes do not match, and the syndrome indicates a correctable error, the Controller will correct the data, write the corrected data and new check code into memory, and set the ECC scrubbing correctable error interrupt (bit [8]) in the int_status_ecc parameter. The read data in the read data FIFO will be discarded if the ECC codes match or not.
   For interval operation (the ecc_scrub_mode parameter was cleared to 0), the timer will be re-loaded with the value in the ecc_scrub_interval parameter. When that timer expires, the ECC scrubbing operation will issue a dummy read to the next address. This process continues, with a delay of interval counts in between dummy reads, until the address defined in the ecc_scrub_end_addr parameter is reached.
   For idle operation (the ecc_scrub_mode parameter was set to 1), as long as the Controller is still idle and no commands have been accepted from the user interface, the ECC scrubbing operation will issue a dummy read to the next address without any delays or interval counts. If a user command is accepted during the scrubbing operation, the scrubbing operation will be halted after the current read/check/writeback is completed and the Controller will return to normal operation. When the scrubbing logic receives an indication that the Controller is idle again, the timer will be re-loaded with the value in the ecc_scrub_idle_cnt parameter. When that counter expires, the scrubbing logic will continue by issuing a dummy read to the next un-tested address until the address defined in the ecc_scrub_end_addr parameter is reached.
4. Once all addresses have been tested, the ecc_scrub_in_progress parameter will be cleared to 0 and the ECC scrubbing operation complete interrupt (bit [7]) in the int_status_ecc parameter.

Note: Software should monitor the ECC scrubbing correctable error interrupt (bit [8]) in the int_status_ecc parameter during the ECC scrubbing operation. This bit is intended as a diagnostic tool to keep track of the number of correctable errors. It can be used as an indication that memory may be slowly reaching the end of its life. No other ECC error information will occur on ECC scrub commands. The other ECC interrupts will NOT be set, the user interface ECC signal will never be asserted, and the ECC reporting parameters will never be filled.

57.4.4 Memory Initialization Programming

Memory initialization programming allows a selectable range of memory to be initialized with a programmable data value. The ECC feature can optionally be turned on or off when initializing the memory.

57.4.4.1 Overview

To select memory initialization programming, program the bist_test_mode parameter to 100b. The data pattern used in this mode is programmable and must be written to the bist_data_pattern parameter before initiating the BIST sequence. In this test mode, when the bist_go parameter is set to 1, the memory from the address defined in the bist_start_address parameter to the computed ending address will be initialized with the data from the bist_data_pattern parameter.

Note: This configuration supports in-line ECC. If in-line ECC is enabled (the ecc_enable parameter is programmed to a non-zero value and the ecc_type parameter is set to 1), then 1/8 of each memory device will be restricted from use and restricted from memory initialization programming. If the user wishes to complete memory initialization on the entire memory, then before running the test, the user should disable in-line ECC (the ecc_enable parameter should be cleared to 00b, or the ecc_type parameter should be cleared to 0) and re-calculate and program new values into the cs_size_X value and the cs_val_lower_X and cs_val_upper_X parameters.

57.4.4.2 Procedure

Memory initialization programming will program the specified address range. This procedure should be followed:

1. Define the data range to check by programming the bist_start_address and the addr_space parameters. When using in-line ECC option (the ecc_enable parameter is programmed to a non-zero value), the user should make sure that the start address and ending address are data and ECC group aligned.
2. Set the bist_data_check parameter to 1.

3. Program the bist_test_mode parameter to 100b.
4. Program the data pattern to be used into the bist_data_pattern parameter.
5. Set the bist_go parameter to 1 to begin the programming.
6. Wait for the BIST complete interrupt (bit [0]) in the int_status_bist parameter to be set to 1. This procedure will not verify any data or report any errors.
7. Clear the bist_go parameter to 0 to re-enable normal operation.

57.4.4.3 BIST Logic Description

The BIST logic is activated by setting the bist_go parameter to 1. This logic can only be activated when the controller is known to be idle. No read or write commands can be pending at the time the user initiates the BIST routine. If BIST checking is enabled after the start parameter is set to 1 but before initialization is complete, BIST traffic will stall in the command queue in the Controller core waiting for the DLL to lock and initialization to complete.

Once the bist_go parameter is set to 1, the internal BIST logic will issue read and write commands for a single aligned user word through the controller's command and data ports. While this operation is in progress, the controller will ignore any commands coming from the user interface.

Note: All programmable registers must be static for a minimum of 10 clock cycles prior to writing a "1" to the bist_go parameter.

Note: The BIST logic will be interrupted when a self-refresh is required (regardless of how the self-refresh is initiated) and resume at the interrupted point when the self-refresh completes.

57.4.5 Port Protection

The port protection feature allows the user to control access to the memory space. When configured and enabled, each port's sources will be restricted to access memory based on its protection level (secure/privileged).

The controller is configured with 16 address regions for each port. The system will default to restricting any type of access for the entire memory. For any one port, valid regions must be set up with unique addresses such that the regions do not overlap and with protection levels that will be supported. Failure to change protection level parameters prior to enabling protection will result in all commands failing. For ports that will not be using port protection, the user must define at least one region that will accept all read and write cycles, for any address.

Note that it is acceptable for address regions for one port to overlap access regions for another port.

The minimum granularity of an address region is 16K. Bit [13] of the incoming system address is used as the lowest bit for comparison. Internally, the Controller compares bit X of the address, where X is the defined as the 14th bit, left shifted by the number of bits used for the datapath. The user may define regions as any multiple of 16K.

Port protection works by checking incoming requests against the valid address ranges. If enabled by setting the port_addr_protection_en parameter to 1, an incoming command is checked based on the restrictions defined in the port protection parameters. Protection region Z is defined with the axiY_start_addr_Z and axiY_end_addr_Z parameters with axiY_address_range_enable = 1. ALL of the following tests must pass for a command to remain valid:

Address Check: The starting and ending addresses must both fall within a single region Z as defined by the axiY_start_addr_Z and axiY_end_addr_Z parameters. If both starting and ending addresses do not fall within a single region Z, the command will fail. The use of these parameters is defined by the axiY_address_range_enable parameters. If the axiY_address_range_enable parameter is set to 1 for a port, the start/end parameters associated with that port will be used.

Protection Check: The relevant protection signal (axiY_ARPROT or axiY_AWPROT) must match the settings for the associated axiY_range_prot_bits_Z parameter. The axiY_range_prot_bits_Z parameter defines the restrictions for region Z as privileged & secure access, privileged access, secure access, or full access. If the protection signal does not match the settings of region Z, the command will fail.

Commands that fail are processed through the controller, but do not corrupt memory and do not return valid data to the user interface. A failing write command will be processed internally as a flushed write command. In this case, the write data is cleared out of the controller FIFOs but the data stored in DRAM memory will NOT change.

A failing read will read the appropriate number of bytes in the memory. However, this data will be ignored and the appropriate number of 0x0 bytes will be returned to the user interface.

Note: In the event of a failure, a SLVERR will be reported on the AXI interface.

In the event of a failure, the port command error interrupt (bit [2]) will be set in the int_status_userif parameter, and reported to the user interface on the DDRC_INT signal (if the associated bit is not masked in the int_mask_userif parameter).

57.5 Interrupt

57.5.1 Interrupt List

Table 57.23 shows interrupt signals of this unit.

Table 57.23 Interrupt list

Interrupt signal	Description
DDRC_INT	Interrupt signal from the controller. This is a level-sensitive signal which will be asserted when the controller detects any interrupt conditions. An interrupt will only cause the DDRC_INT signal to be asserted if the associated bit is set in the int_status_master parameter and cleared in the int_mask_master parameter (and the uppermost bit of the int_mask_master parameter is cleared to 0).
perf_mon_data_status[1:0]	Performance monitor signal Single cycle pulse per event [0] = ECC data correction has occurred on read data. [1] = Uncorrectable ECC error detected on read data.

57.5.2 DDRC_INT

This unit supports a full system of interrupts to alert the user of any issues the Controller encounters. Interrupts are reported by the setting of bits in the int_status_master parameter and the individual int_status_<group> parameters. If any bits are set in the group parameter, the associated bit will be set in the int_status_master parameter. Individual interrupts must be cleared by writing to the bit in the int_ack_<group> parameter, which will then update the int_status_<group> parameter and may clear the bit in the int_status_master parameter. Interrupts may be masked individually in the int_mask_<group> parameter, groups of interrupts may be masked in the int_mask_master parameter, or all interrupts may be masked by setting the uppermost bit of the int_mask_master parameter to 1. See Table 57.24 for the correlation of bits in the master and group parameters, and all subsequent tables for the individual bits of each group.

Table 57.24 Groups of DDRC_INT

Logic Group	Bit in the int_status_master and int_mask_master Parameters	Name of the Group Status Parameter	Name of the Group Mask Parameter	Name of the Group Acknowledge Parameter
Reserved	31 to 15	—	—	—
Mode Registers	14	int_status_mode	int_mask_mode	int_ack_mode
Initialization	13	int_status_init	int_mask_init	int_ack_init
Reserved	12, 11	—	—	—
DFI	10	int_status_dfi	int_mask_dfi	int_ack_freq
Reserved	9	—	—	—
BIST Logic	8	int_status_bist	int_mask_bist	int_ack_bist
Miscellaneous Logic	7	int_status_misc	int_mask_misc	int_ack_misc
User Interface	6	int_status_userif	int_mask_userif	int_ack_userif
Training	5	int_status_training	int_mask_training	int_ack_training
Reserved	4, 3	—	—	—
Low Power Control	2	int_status_lowpower	int_mask_lowpower	int_ack_lowpower
ECC	1	int_status_ecc	int_mask_ecc	int_ack_ecc
Timeout	0	int_status_timeout	int_mask_timeout	int_ack_timeout

Table 57.25 Bits of the Mode Registers Group

Bit	Description
7 to 4	Reserved
3	The register interface-initiated mode register write has completed and another mode register write may be issued.
2	The requested mode register read has completed. The chip and data can be read in the peripheral_mrr_data parameter.
1	Reserved
0	An MRR error has occurred. Error information can be found in the mrr_error_status parameter.

Table 57.26 Bits of the Initialization Group

Bit	Description
7 to 4	Reserved
3	The state machine is in the power-on software initialization state during initialization.
2	Reserved
1	The MC initialization has been completed.
0	The memory reset is valid on the DFI bus.

Table 57.27 Bits of the DFI Group

Bit	Description
7, 6	Reserved
5	The DFI tINIT_COMPLETE value has timed out. This value is specified in the tdfi_init_complete parameter.
4	The user-initiated DLL resynchronization has completed.
3	A state change has been detected on the dfi_init_complete signal after initialization.
2	Error received from the PHY on the DFI bus.
1	A DFI PHY Master Interface error has occurred. Error information can be found in the phymstr_error_status parameter.
0	A DFI update error has occurred. Error information can be found in the update_error_status parameter.

Table 57.28 Bits of the BIST Logic Group

Bit	Description
7 to 1	Reserved
0	The BIST operation has been completed.

Table 57.29 Bits of the Miscellaneous Logic Group

Bit	Description
15 to 12	Reserved
11	The S/W requested refresh operation has resulted in a status bit being set.
10 to 8	Reserved
7	The refresh operation has resulted in a status bit being set.
6	A temperature alert condition (low or high temp) has been detected.
5	The last automatic MRR of MR4 indicated a change in the device temperature or refresh rate (TUF bit set).
4	The controller has entered the software-requested mode.
3	The assertion of the inhibit_dram_cmd parameter has successfully inhibited the command queue.
2 to 0	Reserved

Table 57.30 Bits of the User Interface Group (1 of 2)

Bit	Description
31 to 8	Reserved

Table 57.30 Bits of the User Interface Group (2 of 2)

Bit	Description
7	The user has programmed an invalid setting associated with core words per burst. Examples: Setting the mem_dp_reduction parameter when burst length = 2.
6	A wrap cycle crossing a DRAM page has been detected. This is unsupported and may result in memory data corruption.
5 to 3	Reserved
2	An error occurred on the port command channel.
1	Multiple accesses outside the defined PHYSICAL memory space have occurred.
0	A memory access outside the defined PHYSICAL memory space has occurred.

Table 57.31 Bits of the Training Group

Bit	Description
31 to 16	Reserved
15	A DQS oscillator measurement has been detected to be out of variance.
14	A DQS oscillator measurement overflow has been detected.
13	The DQS oscillator has updated the base values.
12	The software-requested DQS oscillator measurement has completed.
11	The ZQ calibration operation has resulted in a status bit being set. Refer to the zq_status_log parameter for more information.
10 to 0	Reserved

Table 57.32 Bits of the Low Power Control Group

Bit	Description
15 to 4	Reserved
3	A Low Power Interface (LPI) timeout error has occurred.
2, 1	Reserved
0	The low power operation has been completed.

Table 57.33 Bits of the ECC Group

Bit	Description
15 to 9	Reserved
8	An ECC correctable error has been detected in a scrubbing read operation.
7	The scrub operation triggered by setting the ecc_scrub_start parameter has completed.
6	One or more ECC writeback commands could not be executed.
5, 4	Reserved
3	Multiple uncorrectable ECC events have been detected.
2	An uncorrectable ECC event has been detected.
1	Multiple correctable ECC events have been detected.
0	A correctable ECC event has been detected.

Table 57.34 Bits of the Timeout Group (1 of 2)

Bit	Description
31 to 20	Reserved
19	The auto-refresh max deficit timeout has expired.
18	Reserved
17	The low power interface wakeup timeout has expired.
16	The DFI update FM timeout has expired.

Table 57.34 Bits of the Timeout Group (2 of 2)

Bit	Description
15	A DQS oscillator request timeout has been detected.
14	The MRR temperature check FM timeout has expired.
13 to 10	Reserved
9	The ZQ calstart FM timeout has expired.
8	The ZQ callatch FM timeout has expired.
7	The ZQ cal init, cs, cl, or reset FM timeout has expired.
6 to 0	Reserved

57.5.3 perf_mon_data_status[1:0]

This unit has a mechanism to notify the user that correctable or uncorrectable ECC error detect on the read data. These signals are available as error events.

- Bit [0] is asserted 1 for 1 DfiClk when ECC data correction has occurred on the read data.
- Bit [1] is asserted 1 for 1 DfiClk when uncorrectable ECC error detect on the read data.

57.6 Usage Notes

This unit has the following restrictions.

Table 57.35 Restrictions

Item	Restriction
Period of DRAM inaccessibility	It has the period of DRAM inaccessibility to execute the periodic training. It is 1,280 core clock cycles every 134,217,728 core clock cycles. So, it depends only on the Bit Rate (the frequency of the core clock). It is below: 3200 Mbps: 1.6 µs every 167 ms
Density when ECC function is enabled	1/8 area is used for the ECC code, so the usable density is 7/8.
Refresh Management (RFM) and Scaling Level (SCL)	No support
Coherent bufferable issue in memory controller	Incorrect read data may be returned to a read associated with a co-bufferable write when the read is from a different AXI port. It is possible to happen in case of enabling ECC. Workaround is to prioritize write rather than read. DDR parameter generation tool automatically sets axiY_fixed_port_priority_enable, axiY_w_priority and axiY_r_priority properly. Do not change these values from the generated ones.

58. Electrical Characteristics

Electrical characteristics of this LSI is defined with the following conditions unless otherwise described.

Conditions:

- Core voltage
  VDD08 = VDD08_PLLn (n = 0 to 4) = DVDD08A_TSU = OTPVDD08 = USB_USDVDD = PCIE_VDD08A_Ln (n = 0, 1) = AVDD_ADCn (n = 0 to 2) = 0.76 to 0.84 V
- LPDDR4 I/O voltage
  DDR_VDDQ = 1.06 to 1.17 V
- 1.8V I/O and analog voltage
  VDD1833_n (n = 0 to 7, 1.8 V mode) = VDDP_18_n (n = 0 to 7, 33, X) = VDD18_PLLn (n = 0 to 4) = AVDD18A_TSU = OTPVDD18 = USB_USVDD18 = PCIE_VDD18A_CMN = PCIE_VDD18A_L0 = PCIE_VDD18A_L1 = DDR_VAA = AVDDIO_ADCn (n = 0 to 2) = AVDDREF_ADCn (n = 0 to 2) = 1.71 to 1.89 V
- 3.3V I/O and analog voltage
  VDD33 = VDD1833_n (n = 0 to 7, 3.3 V mode) = VDD33_X = USB_USVDD33 = 3.135 to 3.465 V
- Ground
  VSS = VSS_PLLn (n = 0 to 4) = AVSS_ADCn (n = 0 to 2) = AVSSIO_ADCn (n = 0 to 2) = 0 V
- Operating Temperature
  Tj = −40 to 125ºC

58.1 Absolute Maximum Ratings

Table 58.1 Absolute maximum ratings (1 of 2)

Parameter	Symbol	Value	Unit
Power supply voltage (3.3-V I/O)	VDD33, VDD1833_0 to VDD1833_7 (3.3-V mode)	−0.3 to +3.8	V
Power supply voltage (1.8-V I/O)	VDD1833_0 to VDD1833_7 (1.8-V mode), VDDP_18_33, VDDP_18_0 to VDDP_18_7	−0.3 to +2.5	V
Power supply voltage (Core)	VDD08	−0.3 to +1.2	V
Input voltage	Vin (3.3-V logic)	−0.3 to smaller value of VDD33/ VDD1833_n (3.3-V mode) + 0.3 or 3.8	V
Input voltage	Vin (1.8-V logic)	−0.3 to smaller value of VDD1833_n (1.8-V mode) + 0.3 or 2.5	V
Oscillator power supply voltage	VDD33_X	−0.3 to +3.8	V
	VDDP_18_X	−0.3 to +2.5	V
PLL power supply voltage	VDD18_PLL0 to VDD18_PLL4	−0.3 to +2.5	V
	VDD08_PLL0 to VDD08_PLL4	−0.3 to +1.2	V
TSU power supply voltage	AVDD18A_TSU	−0.3 to +2.5	V
	DVDD08A_TSU	−0.3 to +1.2	V
OTP power supply voltage	OTPVDD18	−0.3 to +2.5	V
	OTPVDD08	−0.3 to +1.2	V
USB power supply voltage	USB_USVDD33	−0.3 to +3.8	V
	USB_USVDD18	−0.3 to +2.5	V
	USB_USDVDD	−0.3 to +1.2	V
PCI Express power supply voltage	PCIE_VDD18A_CMN, PCIE_VDD18A_L0, PCIE_VDD18A_L1	−0.3 to +2.5	V
	PCIE_VDD08A_L0, PCIE_VDD08A_L1	−0.3 to +1.2	V

Table 58.1 Absolute maximum ratings (2 of 2)

Parameter	Symbol	Value	Unit
LPDDR4 power supply voltage	DDR_VAA	−0.3 to +2.5	V
	DDR_VDDQ	−0.3 to +1.5	V
ADC12 power supply voltage	AVDDIO_ADC0 to AVDDIO_ADC2	−0.3 to +2.5	V
	AVDD_ADC0 to AVDD_ADC2	−0.3 to +1.2	V
ADC12 analog input voltage	VAN	−0.3 to smaller value of AVDDIO_ADCn + 0.3 or 2.5	V
ADC12 reference voltage	AVDDREF_ADC0 to AVDDREF_ADC2	−0.3 to smaller value of AVDDIO_ADCn + 0.3 or 2.5	V
Crystal oscillator pins input voltage	XTAL, EXTAL	−0.3 to + 2.5	V
Operating temperature (Junction temperature)	T_j	−40 to +125	ºC
Storage temperature	T_{stg}	−40 to +125	ºC

Caution: Permanent damage to the LSI might result if absolute maximum ratings are exceeded.

58.2 Power Supply

Table 58.2 Power supply (1 of 2)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Power supply voltage (3.3-V I/O)	VDD33, VDD1833_0 to VDD1833_7 (3.3-V mode)	3.135	3.3	3.465	V
Power supply voltage (1.8-V I/O)	VDD1833_0 to VDD1833_7 (1.8-V mode), VDDP_18_33, VDDP_18_0 to VDDP_18_7	1.71	1.8	1.89	V
Power supply voltage (Core)	VDD08	0.76	0.8	0.84	V
Ground	VSS	—	0	—	V
Oscillator power supply voltage	VDD33_X	3.135	3.3	3.465	V
	VDDP_18_X	1.71	1.8	1.89	V
PLL power supply voltage	VDD18_PLL0 to VDD18_PLL4	1.71	1.8	1.89	V
	VDD08_PLL0 to VDD08_PLL4	0.76	0.8	0.84	V
	VSS_PLL0 to VSS_PLL4	—	0	—	V
TSU power supply voltage	AVDD18A_TSU	1.71	1.8	1.89	V
	DVDD08A_TSU	0.76	0.8	0.84	V
OTP power supply voltage	OTPVDD18	1.71	1.8	1.89	V
	OTPVDD08	0.76	0.8	0.84	V
USB power supply voltage	USB_USVDD33	3.135	3.3	3.465	V
	USB_USVDD18	1.71	1.8	1.89	V
	USB_USDVDD	0.76	0.8	0.84	V
PCI Express power supply voltage	PCIE_VDD18A_CMN, PCIE_VDD18A_L0, PCIE_VDD18A_L1	1.71	1.8	1.89	V
	PCIE_VDD08A_L0, PCIE_VDD08A_L1	0.76	0.8	0.84	V
LPDDR4 power supply voltage	DDR_VAA	1.71	1.8	1.89	V
	DDR_VDDQ	1.06	1.1	1.17	V

Table 58.2 Power supply (2 of 2)

Parameter	Symbol	Min.	Typ.	Max.	Unit
ADC12 power supply voltage	AVDDIO_ADC0 to AVDDIO_ADC2	1.71	1.8	1.89	V
	AVDD_ADC0 to AVDD_ADC2	0.76	0.8	0.84	V
	AVSSIO_ADC0 to AVSSIO_ADC2	—	0	—	V
	AVSS_ADC0 to AVSS_ADC2	—	0	—	V

58.3 Power On/Off Sequence

Power on/off sequence and timing are shown in the figure and table below.

For power-up, 0.8-V (i.e. VDD08) must be supplied first, 1.8-V power (i.e. VDD18 and AVDD) must be supplied second, 1.1-V power must be supplied third, then 3.3-V power (i.e. DDR_VDDQ and VDD33) must be supplied. The power-up sequence must be completed within 100 ms. Reset signal (i.e. RES#) must be held to Low level during the power-up.

For Power-down, 1.1-V and 3.3-V power (i.e. DDR_VDDQ and VDD33) must go down first and then 0.8-V and 1.8-V power (i.e. VDD08, VDD18, and AVDD). The power-down sequence must be completed within 100 ms.

Rise time of each power supply for the power-up must be larger than 40 µs and fall time of each power supply for the power-down must be larger than 10 µs.

Power supply voltages and reset signal must be applied with monotonic increase.

Do not apply a negative voltage to power supply voltages.

Stable clock must be supplied to EXTAL/XTAL or EXTCLKIN pin when reset signal (i.e. RES#) is driven high.

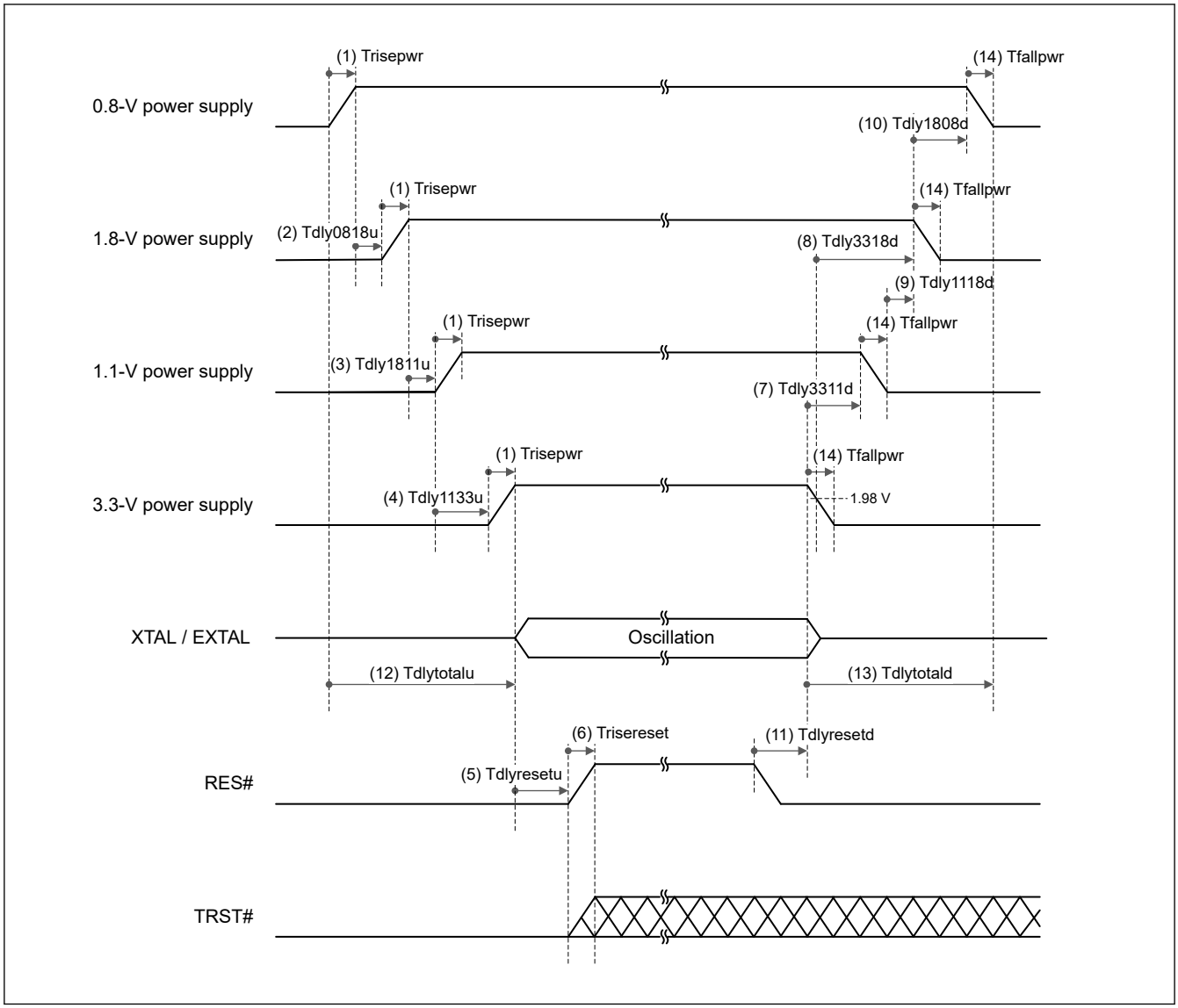


Figure 58.1 Power on/off sequence

Table 58.3 Power on/off sequence timing (1 of 2)

No	Symbol	Description	Value		
			Min.	Typ.	Max.
(1)	Trisepwr	Rising time of the power supply voltage	40 µs	—	30 ms
(2)	Tdly0818u	Delay time from completion of rising of the 0.8-V power supply voltage to start of rising of the 1.8-V power supply voltage	1 µs	—	100 ms
(3)	Tdly1811u	Delay time from completion of rising of the 1.8-V power supply voltage to start of rising of the 1.1-V power supply voltage	0	—	100 ms
(4)	Tdly1133u	Delay time from start of rising of the 1.1-V power supply voltage to start of rising of the 3.3-V power supply voltage	0	—	100 ms
(5)	Tdlyresetu	Delay time from completion of rising of the 3.3-V power supply voltage to start of rising of RES#	10 ms	—	—
(6)	Trisereset	Rising time of RES#	—	—	150 µs
(7)	Tdly3311d	Delay time from start of falling of the 3.3-V power supply voltage to start of falling of the 1.1-V power supply voltage	0	—	100 ms
(8)	Tdly3318d	Delay time from the time when 3.3-V power supply voltage drops below 1.98-V to start of falling of the 1.8-V power supply voltage	0	—	100 ms

Table 58.3 Power on/off sequence timing (2 of 2)

No	Symbol	Description	Value		
			Min.	Typ.	Max.
(9)	Tdly1118d	Delay time from completion of falling of the 1.1-V power supply voltage to start of falling of the 1.8-V power supply voltage	0	—	100 ms
(10)	Tdly1808d	Delay time from start of falling of the 1.8-V power supply voltage to start of falling of 0.8-V power supply voltage	0	—	100 ms
(11)	Tdlyresetd	Delay time from start of falling of RES# to start of falling of the 3.3-V power supply voltage	10 µs	—	—
(12)	Tdlytotalu	Startup time of all power supply voltage	0	—	100 ms
(13)	Tdlytotald	Shut down time of all power supply voltage	0	—	100 ms
(14)	Tfallpwr	Falling time of the power supply voltage	10 µs	—	30 ms

58.4 DC Characteristics

Table 58.4 DC Characteristics for Type A I/O buffer (VDD33 domain)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input High-level voltage	V_{IH33}	—	VDD33 × 0.7	—	VDD33 + 0.3	V
Input Low-level voltage	V_{IL33}	—	−0.3	—	VDD33 × 0.3	V
Hysteresis voltage	ΔV_{T33}	—	VDD33 × 0.08	—	—	V
Output High-level voltage	V_{OH33}	Low, IOH = −2 mA	VDD33 × 0.8	—	VDD33	V
		Middle, IOH = −4 mA	VDD33 × 0.8	—	VDD33	V
		High, IOH = −8 mA	VDD33 × 0.8	—	VDD33	V
		Ultra High, IOH = −12 mA	VDD33 × 0.8	—	VDD33	V
Output Low-level voltage	V_{OL33}	Low, IOL = 2 mA	0	—	VDD33 × 0.2	V
		Middle, IOL = 4 mA	0	—	VDD33 × 0.2	V
		High, IOL = 8 mA	0	—	VDD33 × 0.2	V
		Ultra High, IOL = 12 mA	0	—	VDD33 × 0.2	V
Input leakage current	$\lvert I_{in} \rvert$	Vin = 0 V, Vin = VDD33	—	—	12	µA
Three-State leakage current (off state)	$\lvert I_{TSI} \rvert$	Vin = 0 V, Vin = VDD33	—	—	12	µA
Input Pull-up resistors resistance	Rpu	Vin = 0 V	10	—	100	kΩ
Input Pull-down resistors resistance	Rpd	Vin = VDD33	10	—	100	kΩ
Input Capacitance	Cin	All input/output and input pins	—	—	10	pF

Table 58.5 DC Characteristics for Type B I/O buffer (VDD1833_n domain) (3.3-V mode) (1 of 2)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input High-level voltage	V_{IH33}	—	VDD1833 × 0.7	—	VDD1833 + 0.3	V
Input Low-level voltage	V_{IL33}	—	−0.3	—	VDD1833 × 0.3	V
Hysteresis voltage	ΔV_{T33}	—	0.1	—	—	V
Output High-level voltage	V_{OH33}	Low, IOH = −8 mA	VDD1833 × 0.8	—	VDD1833	V
		Middle, IOH = −11 mA	VDD1833 × 0.8	—	VDD1833	V
		High, IOH = −14 mA	VDD1833 × 0.8	—	VDD1833	V
		Ultra High, IOH = −17 mA	VDD1833 × 0.8	—	VDD1833	V

Table 58.5 DC Characteristics for Type B I/O buffer (VDD1833_n domain) (3.3-V mode) (2 of 2)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Output Low-level voltage	V_{OL33}	Low, IOL = 8 mA	0	—	VDD1833 × 0.2	V
		Middle, IOL = 11 mA	0	—	VDD1833 × 0.2	V
		High, IOL = 14 mA	0	—	VDD1833 × 0.2	V
		Ultra High, IOL = 17 mA	0	—	VDD1833 × 0.2	V
Input leakage current	$\|I_{in}\|$	Vin = 0 V, Vin = VDD1833	—	—	12	µA
Three-State leakage current (off state)	$\|I_{TSI}\|$	Vin = 0 V, Vin = VDD1833	—	—	12	µA
Input Pull-up resistors resistance	Rpu	Vin = 0 V	18	—	72	kΩ
Input Pull-down resistors resistance	Rpd	Vin = VDD1833	24	—	87	kΩ
Input Capacitance	Cin	All input/output and input pins	—	—	10	pF

Table 58.6 DC Characteristics for Type B I/O buffer (VDD1833_n domain) (1.8-V mode)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input High-level voltage	V_{IH18}	—	VDD1833 × 0.7	—	VDD1833 + 0.3	V
Input Low-level voltage	V_{IL18}	—	−0.3	—	VDD1833 × 0.3	V
Hysteresis voltage	ΔV_{T18}	—	0.1	—	—	V
Output High-level voltage	V_{OH18}	Low, IOH = −4 mA	VDD1833 × 0.8	—	VDD1833	V
		Middle, IOH = −5 mA	VDD1833 × 0.8	—	VDD1833	V
		High, IOH = −6 mA	VDD1833 × 0.8	—	VDD1833	V
		Ultra High, IOH = −7 mA	VDD1833 × 0.8	—	VDD1833	V
Output Low-level voltage	V_{OL18}	Low, IOL = 4 mA	0	—	VDD1833 × 0.2	V
		Middle, IOL = 5 mA	0	—	VDD1833 × 0.2	V
		High, IOL = 6 mA	0	—	VDD1833 × 0.2	V
		Ultra High, IOL = 7 mA	0	—	VDD1833 × 0.2	V
Input leakage current	|Iin|	Vin = 0 V, Vin = VDD1833	—	—	12	µA
Three-State leakage current (off state)	$\|I_{TSI}\|$	Vin = 0 V, Vin = VDD1833	—	—	12	µA
Input Pull-up resistors resistance	Rpu	Vin = 0 V	12	—	92	kΩ
Input Pull-down resistors resistance	Rpd	Vin = VDD1833	13	—	92	kΩ
Input Capacitance	Cin	All input/output and input pins	—	—	10	pF

Table 58.7 Supply Current (1 of 3)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Normal operation	I_{VDD08}	Cortex-A55 Clock = 1.2 GHz, Cortex-R52 Clock = 1.0 GHz, Tj ≤ 110 °C	—	—	4600	mA
	I_{VDD33}	*1	—	50	—	mA
	$I_{VDD1833\_0}$	*1	—	9	—	mA
	$I_{VDD1833\_1}$	*1	—	9	—	mA
	$I_{VDD1833\_2}$	*1	—	9	—	mA
	$I_{VDD1833\_3}$	*1	—	9	—	mA
	$I_{VDD1833\_4}$	*1	—	9	—	mA
	$I_{VDD1833\_5}$	*1	—	9	—	mA
	$I_{VDD1833\_6}$	*1	—	14	—	mA
	$I_{VDD1833\_7}$	*1	—	9	—	mA
	$I_{VDDP\_18\_33}$	—	—	12	—	mA
	$I_{VDDP\_18\_0}$	—	—	2	—	mA
	$I_{VDDP\_18\_1}$	—	—	2	—	mA
	$I_{VDDP\_18\_2}$	—	—	2	—	mA
	$I_{VDDP\_18\_3}$	—	—	2	—	mA
	$I_{VDDP\_18\_4}$	—	—	2	—	mA
	$I_{VDDP\_18\_5}$	—	—	2	—	mA
	$I_{VDDP\_18\_6}$	—	—	2	—	mA
	$I_{VDDP\_18\_7}$	—	—	2	—	mA
	I_{VDD33_X}	—	—	1	—	mA
	$I_{VDDP\_18\_X}$	—	—	10	—	mA
	I_{VDD18_PLL0}	—	—	—	2.2	mA
	I_{VDD18_PLL1}	—	—	—	2.2	mA
	I_{VDD18_PLL2}	—	—	—	2.2	mA
	I_{VDD18_PLL3}	—	—	—	2.2	mA
	I_{VDD18_PLL4}	—	—	—	2.2	mA
	I_{VDD08_PLL0}	—	—	—	2.6	mA
	I_{VDD08_PLL1}	—	—	—	2.6	mA
	I_{VDD08_PLL2}	—	—	—	2.6	mA
	I_{VDD08_PLL3}	—	—	—	2.6	mA
	I_{VDD08_PLL4}	—	—	—	2.6	mA
	$I_{AVDD18A\_TSU}$	—	—	—	1	mA
	$I_{DVDD08A\_TSU}$	—	—	—	0.07	mA
	$I_{OTPVD18}$	—	—	—	18	mA
	$I_{OTPVDD08}$	—	—	—	1.4	mA
	$I_{USB\_USVDD33}$	—	—	—	4.7	mA
	$I_{USB\_USVDD18}$	—	—	—	23	mA
	I_{USB_USDVDD}	—	—	—	8.4	mA

Table 58.7 Supply Current (2 of 3)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Normal operation	$I_{PCIE\_VDD18A\_CMN}$	—	—	—	19	mA
	$I_{PCIE\_VDD18A\_L0}$	—	—	—	27	mA
	$I_{PCIE\_VDD18A\_L1}$	—	—	—	27	mA
	$I_{PCIE\_VDD08A\_L0}$	—	—	—	42	mA
	$I_{PCIE\_VDD08A\_L1}$	—	—	—	42	mA
	I_{DDR_VAA}	—	—	—	5	mA
	I_{DDR_VDDQ}	—	—	—	300	mA
	I_{AVDDIO_ADC0}	—	—	—	0.06	mA
	I_{AVDDIO_ADC1}	—	—	—	0.06	mA
	I_{AVDDIO_ADC2}	—	—	—	0.06	mA
	I_{AVDD_ADC0}	—	—	—	0.7	mA
	I_{AVDD_ADC1}	—	—	—	0.7	mA
	I_{AVDD_ADC2}	—	—	—	0.7	mA
Low power consumption mode*2	I_{VDD08}	All modules inactive	—	105	—	mA
	I_{VDD33}	*1	—	5	—	mA
	$I_{VDD1833\_0}$	*1	—	1	—	mA
	$I_{VDD1833\_1}$	*1	—	1	—	mA
	$I_{VDD1833\_2}$	*1	—	1	—	mA
	$I_{VDD1833\_3}$	*1	—	1	—	mA
	$I_{VDD1833\_4}$	*1	—	1	—	mA
	$I_{VDD1833\_5}$	*1	—	1	—	mA
	$I_{VDD1833\_6}$	*1	—	1	—	mA
	$I_{VDD1833\_7}$	*1	—	1	—	mA
	$I_{VDDP\_18\_33}$	—	—	8	—	mA
	$I_{VDDP\_18\_0}$	—	—	1	—	mA
	$I_{VDDP\_18\_1}$	—	—	1	—	mA
	$I_{VDDP\_18\_2}$	—	—	1	—	mA
	$I_{VDDP\_18\_3}$	—	—	1	—	mA
	$I_{VDDP\_18\_4}$	—	—	1	—	mA
	$I_{VDDP\_18\_5}$	—	—	1	—	mA
	$I_{VDDP\_18\_6}$	—	—	1	—	mA
	$I_{VDDP\_18\_7}$	—	—	1	—	mA
	I_{VDD33_X}	—	—	1	—	mA
	$I_{VDDP\_18\_X}$	—	—	10	—	mA
	I_{VDD18_PLL0}	—	—	2	—	µA
	I_{VDD18_PLL1}	—	—	2.2	—	mA
	I_{VDD18_PLL2}	—	—	2	—	µA
	I_{VDD18_PLL3}	—	—	2	—	µA

Table 58.7 Supply Current (3 of 3)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Low power consumption mode*2	I_{VDD18_PLL4}	—	—	2.2	—	mA
	I_{VDD08_PLL0}	—	—	9	—	µA
	I_{VDD08_PLL1}	—	—	1.6	—	mA
	I_{VDD08_PLL2}	—	—	9	—	µA
	I_{VDD08_PLL3}	—	—	9	—	µA
	I_{VDD08_PLL4}	—	—	1.6	—	mA
	$I_{AVDD18A\_TSU}$	—	—	11	—	µA
	$I_{DVDD08A\_TSU}$	—	—	13	—	µA
	$I_{OTPVD18}$	—	—	28	—	µA
	$I_{OTPVDD08}$	—	—	63	—	µA
	$I_{USB\_USVDD33}$	—	—	54	—	µA
	$I_{USB\_USVDD18}$	—	—	18	—	µA
	I_{USB_USDVDD}	—	—	1	—	µA
	$I_{PCIE\_VDD18A\_CMN}$	—	—	0.9	—	mA
	$I_{PCIE\_VDD18A\_L0}$	—	—	0.05	—	mA
	$I_{PCIE\_VDD18A\_L1}$	—	—	0.05	—	mA
	$I_{PCIE\_VDD08A\_L0}$	—	—	1.3	—	mA
	$I_{PCIE\_VDD08A\_L1}$	—	—	1.3	—	mA
	I_{DDR_VAA}	—	—	0.2	—	mA
	I_{DDR_VDDQ}	—	—	0.2	—	mA
	I_{AVDDIO_ADC0}	—	—	0.3	—	µA
	I_{AVDDIO_ADC1}	—	—	0.3	—	µA
	I_{AVDDIO_ADC2}	—	—	0.3	—	µA
	I_{AVDD_ADC0}	—	—	3	—	µA
	I_{AVDD_ADC1}	—	—	3	—	µA
	I_{AVDD_ADC2}	—	—	3	—	µA

Note: These values are reference values. The actual operating current greatly depends on the system (such as unsharpened waveforms due to I/O load and toggle frequency). Be sure to measure these current values in the system.

Note 1. IO supply current (I_{VDD33}, $I_{VDD1833\_n\ (n = 0\ to\ 7)}$) should be 50 mA or less. (ΣIOH in Table 58.8)

Note 2. All applicable modules are stopped or standby mode with the lowest clock frequency setting, no pull-up/down or operation for all I/O ports, and room temperature.

Table 58.8 Permissible Output Currents (1 of 2)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Permissible output low current (max. value per pin)	IOL	All output pins	Low	—	—	2.5	mA
			Middle	—	—	5.0	
			High	—	—	9.0	
			Ultra High	—	—	11.8	
Permissible output low current (total)	ΣIOL	Sum of all output pins		—	—	50	mA

Table 58.8 Permissible Output Currents (2 of 2)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Permissible output high current (max. value per pin)	IOH	All output pins	Low	—	—	2.5	mA
			Middle	—	—	5.0	
			High	—	—	9.0	
			Ultra High	—	—	11.8	
Permissible output high current (total)	ΣIOH	Sum of all output pins		—	—	50	mA

Table 58.9 Thermal Resistance value (Reference)

Item	Symbol	Package	Max.	Unit
Thermal Resistance	Θja	729 pin FCBGA 23 × 23 mm, 0.8-mm pitch	10.9	°C/W
		576 pin FCBGA 21 × 21 mm, 0.8-mm pitch	11.2	°C/W
	Ψjt	729 pin FCBGA 23 × 23 mm, 0.8-mm pitch	0.02	°C/W
		576 pin FCBGA 21 × 21 mm, 0.8-mm pitch	0.02	°C/W

58.5 AC Characteristics

Table 58.10 Operating frequency

Parameter		Symbol	Min.	Max.	Unit
Operating frequency	Cortex-A55 Core clock (CA55CnCLK, n = 0 to 3)	f	600	1200	MHz
	Cortex-A55 SCU clock (CA55SCLK)		500	1000	
	Cortex-R52 Core clock (CR52CnCLK, n = 0, 1)		500	1000	
	Peripheral module clock AH (PCLKAH)		400		
	Peripheral module clock AM (PCLKAM)		200		
	Peripheral module clock AL (PCLKAL)		100		
	Peripheral module clock H (PCLKH)		250		
	Peripheral module clock M (PCLKM)		125		
	Peripheral module clock L (PCLKL)		62.5		
	Peripheral module clock for SCIn (PCLKSCIn, n = 0 to 5)		75	125	
	Peripheral module clock for SCIEn (PCLKSCIEn, n = 0 to 11)		75	125	
	Peripheral module clock for SPIn (PCLKSPIn, n = 0 to 3)		75	125	
	xSPI serial clock (XSPI_CLKn) (n = 0, 1)		12.5	133	
	LCDC clock		5	100	
	DDR controller DFI clock (DFICLK)		800		
	External bus clock output (CKIO)		31.25	125	
	Ethernet PHY reference clock (ETHn_REFCLK, n = 0 to 3)		25		
	Ethernet PHY reference clock (RMIIn_REFCLK, n = 0 to 3)		50		

AC Characteristics are defined in condition of the IO setting (DRCTLm register setting) show in Table 58.11.

Table 58.11 IO setting (DRCTLm register setting) condition (1 of 2)

Module	Signal	IO type	Voltage	DRCTLm register		
				DRVn	SRn	SMTn
Bus, DMAC	CKIO	—	3.3 V	High	Fast	—
	Other than the above	Type A	3.3 V	Middle	Fast	Disable
		Type B	3.3 V	Low	Fast	Disable

Table 58.11 IO setting (DRCTLm register setting) condition (2 of 2)

Module		Signal	IO type	Voltage	DRCTLm register		
					DRVn	SRn	SMTn
MTU3, GPT, IIC, CANFD, DSMIF, ENCIF, ENDAT, HDSL		All signals	Type A	3.3 V	Middle	Slow	Disable
			Type B	3.3 V	Low	Slow	Disable
SCI, SCIE, SPI		All signals	—	3.3 V	High	Fast	Disable
xSPI (n = 0, 1; m = 0, 1)		XSPIn_CKP, XSPIn_CKN, XSPIn_IO[7:0], XSPIn_CSm#, XSPIn_DS	—	1.8 V	High	Fast	Disable
			—	3.3 V	High	Fast	Enable
		Other than the above	—	—	Low	Slow	Disable
Ethernet Interface (n = 0 to 3, m = 0 to 2)		ETHn_TXCLK	—	1.8 V/3.3 V	Ultra-high	Fast	Disable
		ETHn_TXD[3:0], ETHn_TXEN	—	1.8 V (RGMII)	Ultra-high	Fast	—
			—	3.3 V (RMII)	High	Fast	—
			—	3.3 V (MII)	Middle	Fast	—
		ETHn_TXER	—	3.3 V (MII)	Middle	Fast	—
		ETHn_RXCLK, ETHn_RXD[3:0], ETHn_RXDV	—	1.8 V/3.3 V	—	—	Disable
		ETHn_RXER, ETHn_COL, ETHn_CRS	—	3.3 V	—	—	Disable
		ETHn_REFCLK	—	1.8 V/3.3 V	Middle	Fast	—
		RMIIn_REFCLK	—	1.8 V/3.3 V	High	Fast	—
		GMACm_MDC, GMACm_MDIO, ETHSW_MDC, ETHSW_MDIO, ESC_MDC, ESC_MDIO	—	1.8 V/3.3 V	Middle	Slow	Disable
		Other than the above	—	3.3 V	Middle	Slow	Disable
SHOSTIF, MBXSEM		HSPI_CK, HSPI_CS#, HSPI_IO[7:0]	—	3.3 V	High	Fast	Disable
		HSPI_INT#, MBX_HINT#	Type A	3.3 V	Middle	Slow	—
			Type B	3.3 V	Low	Slow	—
LCDC		All signals	—	3.3 V	Ultra-high	Fast	—
SDHI (n = 0, 1)	SDR104, SDR50, HS200	SDn_CLK	—	1.8 V	Ultra-high	Fast	—
		Other than the above	—	1.8 V	High	Fast	Disable
	DDR50, High Speed DDR	All signals	—	1.8 V/3 .3 V	High	Fast	Disable
	Other than the above	SDn_CLK	—	1.8 V/3 .3 V	High	Fast	—
		Other than the above	—	1.8 V/3 .3 V	Middle	Fast	Disable
Debug Interface		TDO, TMS	—	3.3 V	High	Fast	Disable
		TCK, TDI	—	3.3 V	—	—	Enable
GPIO		All signals in 1.8 V or 3.3 V selectable domain (VDD1833_n (n = 0 to 7))	—	1.8 V/3.3 V	Any	Any	Any
		All signals in 3.3 V fixed domain (VDD33)	—	3.3 V	Any	Any	Any
Other than the above		All signals	—	3.3 V	Low	Slow	Disable

58.5.1 Clock Timing

58.5.1.1 CKIO Pin Output Timing

Table 58.12 CKIO pin output timing

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
CKIO pin output cycle time	t_{CKcyc}	Figure 58.2		8*1	—	32	ns
CKIO pin output high level pulse width	t_{CKH}			t_{CKcyc} / 2 - t_{CKr}	—	—	ns
CKIO pin output low level pulse width	t_{CKL}			t_{CKcyc} / 2 - t_{CKf}	—	—	ns
CKIO pin output rising time 1	t_{CKr}	V_{OH33} = VDD33 – 0.5 V, V_{OL33} = 0.4 V	C = 30 pF	—	—	4.0	ns
			C = 15 pF	—	—	3.5	ns
CKIO pin output falling time 1	t_{CKf}	V_{OH33} = VDD33 – 0.5 V, V_{OL33} = 0.4 V	C = 30 pF	—	—	4.0	ns
			C = 15 pF	—	—	3.5	ns
CKIO pin output rising time 2	t_{CKr}	V_{OH33} = 2.0 V, V_{OL33} = 0.8 V	C = 30 pF	—	—	2.3	ns
			C = 15 pF	—	—	1.5	ns
CKIO pin output falling time 2	t_{CKf}	V_{OH33} = 2.0 V, V_{OL33} = 0.8 V	C = 30 pF	—	—	2.3	ns
			C = 15 pF	—	—	1.5	ns

Note 1. Condition is C=15 pF. In case of C = 30 pF, Min. is 12.

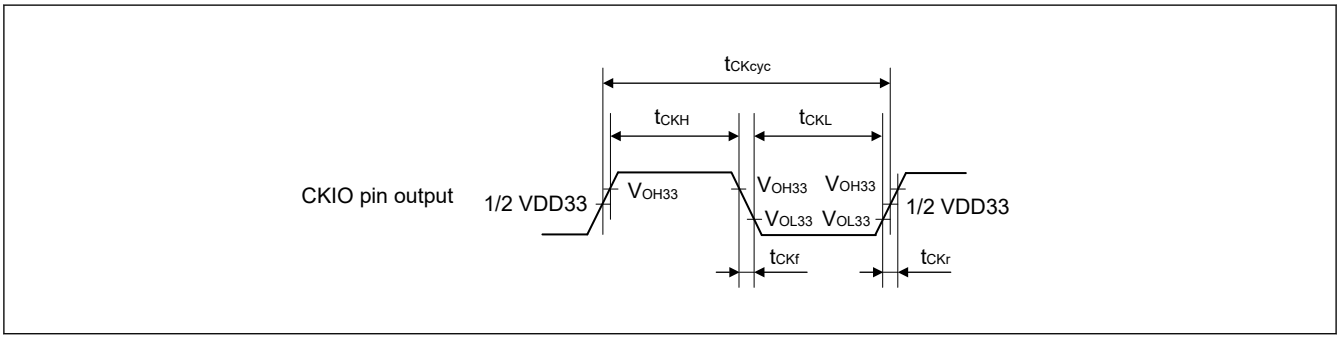


Figure 58.2 CKIO pin output timing

58.5.1.2 Ethernet PHY Reference Clock Output Timing

Conditions:

C = 30 pF (ETHn_REFCLK)

C = 20 pF (RMIIn_REFCLK)

Table 58.13 Ethernet PHY reference clock output timing

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
ETHn_REFCLK cycle time	t_{CK}	Figure 58.3	40	—	—	ns
ETHn_REFCLK frequency	—		25.00 ± 50 ppm			MHz
ETHn_REFCLK duty	—		45	—	55	%
ETHn_REFCLK rising/falling time	t_{CKr} / t_{CKf}		0.5	—	4.0	ns
RMIIn_REFCLK cycle time	t_{CK}		20	—	—	ns
RMIIn_REFCLK frequency	—		50.00 ± 50 ppm			MHz
RMIIn_REFCLK duty	—		45	—	55	%
RMIIn_REFCLK rising/falling time	t_{CKr} / t_{CKf}		0.5	—	3.5	ns

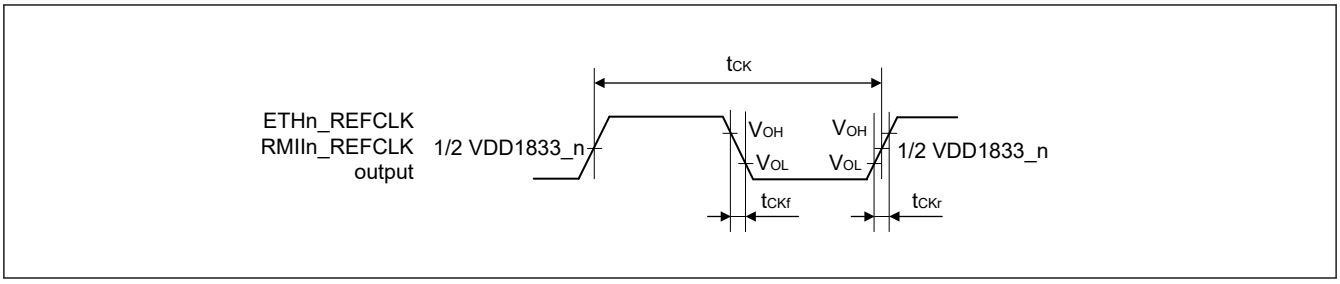


Figure 58.3 Ethernet PHY reference clock output timing

58.5.1.3 EXTCLKIN External Clock Input

Table 58.14 EXTCLKIN clock timing

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
EXTCLKIN external clock frequency	$f_{EXTCLKIN}$	—	25.00 ± 50 ppm			MHz
		EtherCAT in use	25.00 ± 25 ppm			MHz
EXTCLKIN duty	$r_{EXTCLKIN}$	—	±5%			—
EXTCLKIN rising time	$t_{rEXTCLKIN}$	—	0	—	5	ns
EXTCLKIN falling time	$t_{fEXTCLKIN}$	—	0	—	5	ns

Note: XTALSEL and EXTAL should be driven low. Leave XTAL open-circuit.
Note: When using crystal resonator (i.e. EXTA/XTAL clock is used), EXTCLKIN should be driven low and XTALSEL should be driven high.

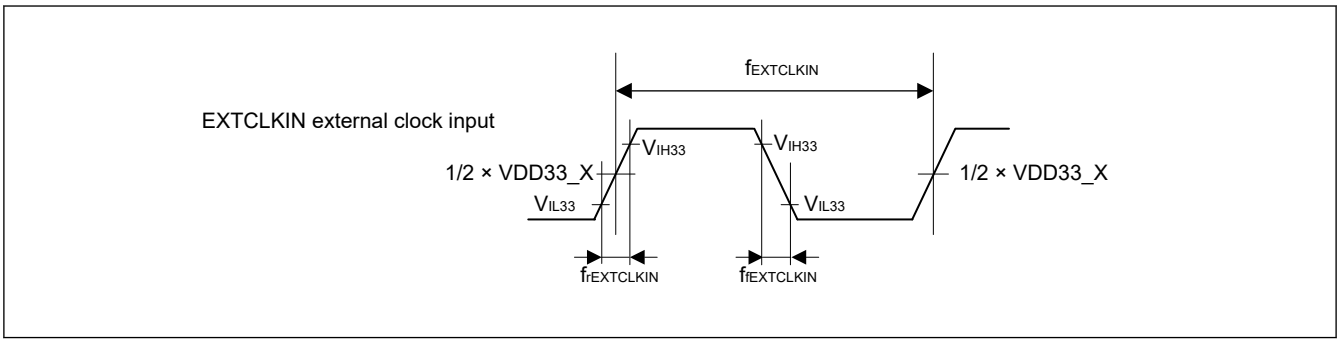


Figure 58.4 EXTCLKIN external clock input timing

58.5.1.4 EXTAL/XTAL Clock Timing

Table 58.15 EXTAL/XTAL clock timing

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
EXTAL/XTAL clock frequency*1	f_{XTAL}	—	25.00 ± 50 ppm			MHz
		EtherCAT in use	25.00 ± 25 ppm			MHz

Note: XTALSEL should be driven high and EXTCLKIN should be driven low.
Note: When using an external oscillator, be sure to leave XTAL open-circuit and make sure that XTALSEL and EXTAL are driven low.
Note 1. When using the EXTAL/XTAL clock (i.e. crystal resonator), ask the oscillator manufacturer to evaluate oscillation of the oscillator. For the oscillation stabilization time, see the evaluation result provided by the oscillator manufacturer.

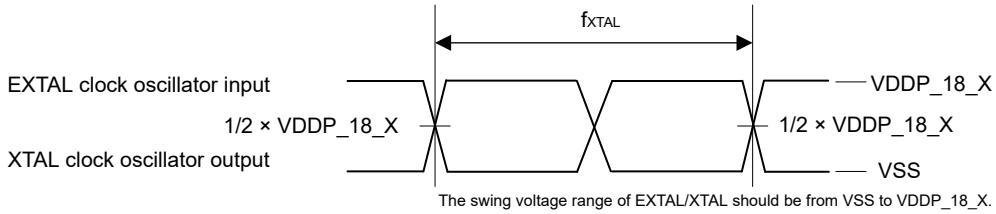


Figure 58.5 EXTAL clock oscillator input and XTAL clock oscillator output timing

58.5.1.5 LOCO Clock Timing

Table 58.16 LOCO clock timing

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
LOCO clock cycle time	t_{Lcyc}	—	0.83	1	1.25	µs
LOCO clock oscillation frequency	t_{LOCO}	—	0.8	1	1.2	MHz
LOCO clock oscillation stabilization wait time	t_{LOCOWT}	—	—	—	5	µs

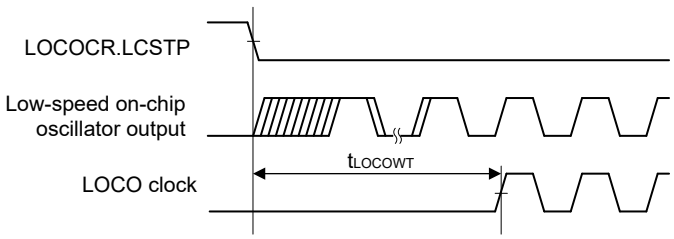


Figure 58.6 LOCO clock oscillation start timing

58.5.2 Reset, Interrupt, and Mode Timing

Table 58.17 Reset, interrupt, and mode timing

Parameter		Symbol	Conditions	Min.*1	Typ.	Max.	Unit
RES# pulse width	At power on	$t_{dlyreset}$	Figure 58.7	10	—	—	ms
	Other than above	$t_{dlyreset2}$		1	—	—	ms
RES# rising time		$t_{risereset}$		—	—	150	µs
TRST# pulse width	At power on	$t_{dlyreset}$		10	—	—	ms
	Other than above	$t_{dlyreset2}$		1	—	—	ms
TRST# rising time		$t_{risereset}$		—	—	150	µs
SEI pulse width		t_{SEIW}	Figure 58.8	t_{PHcyc} × 2	—	—	ns
IRQ pulse width		t_{IRQW}	Figure 58.9	t_{PHcyc} × 2	—	—	ns
Mode hold time (to RES#)	At power on	t_{MDH}	Figure 58.10	250	—	—	ns

Note 1. t_{PHcyc}: PCLKH cycle

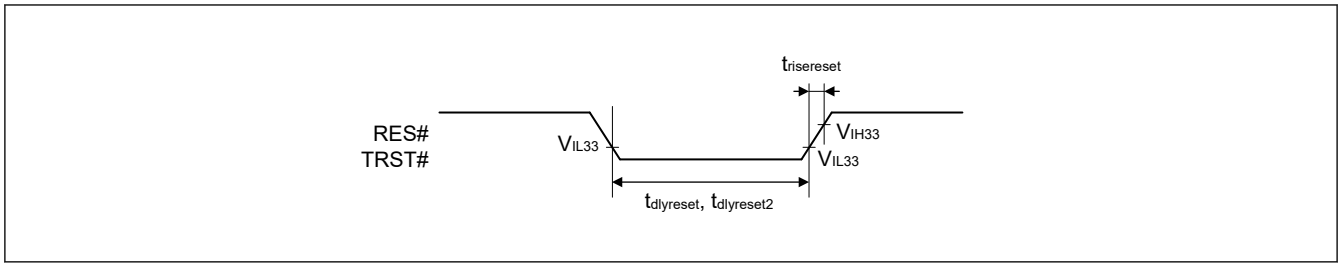


Figure 58.7 Reset input timing

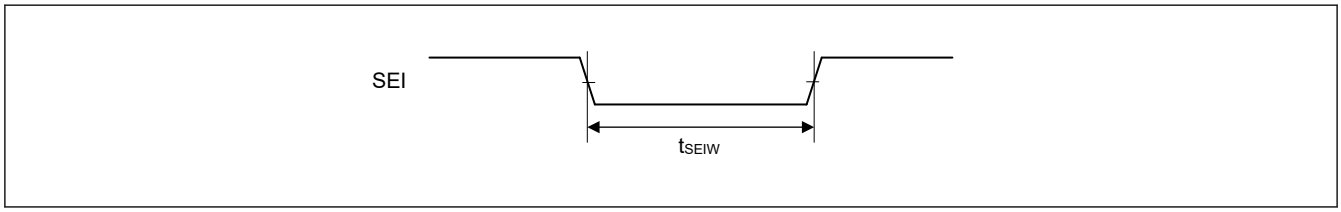


Figure 58.8 SEI interrupt input timing

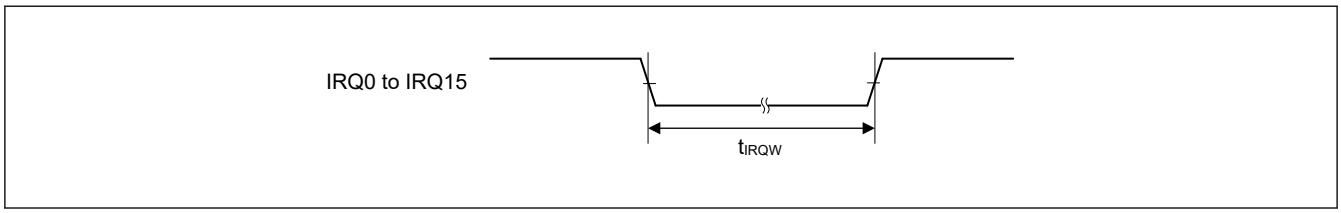


Figure 58.9 IRQ interrupt input timing

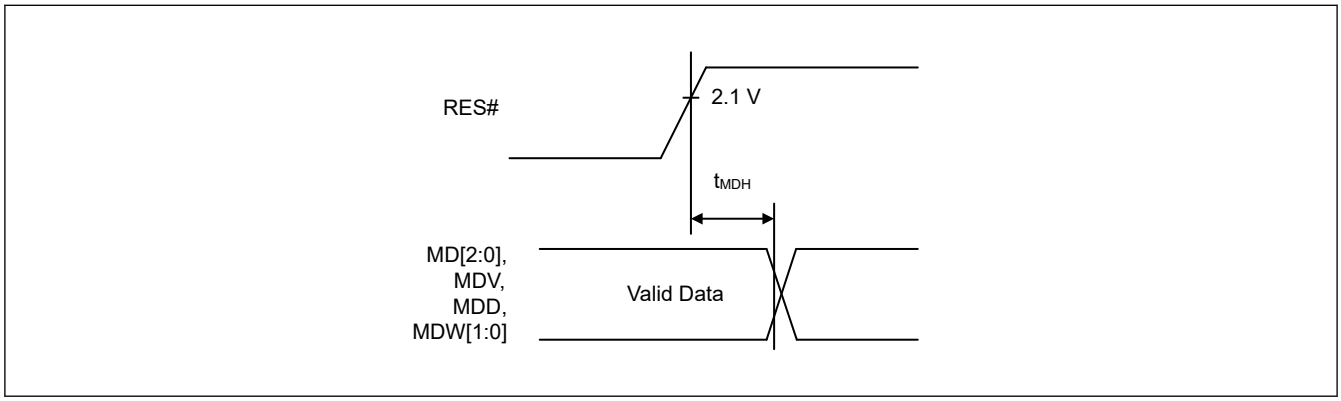


Figure 58.10 Mode input timing

58.5.3 Bus Timing

Table 58.18 Bus timing (1 of 2)

Conditions: V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 30 pF (CKIO), 30 pF (others), $T_j$min = −40°C

Parameter	Symbol	CKIO = 1/tCKcyc (Max 83.3 MHz)		Unit	Reference Figure
		Min.	Max.		
Address delay time 1	t_{AD1}	0	8	ns	Figure 58.11 to Figure 58.18
Address delay time 2	t_{AD2}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 8	ns	Figure 58.18
Address setup time	t_{AS}	0	—	ns	Figure 58.11 to Figure 58.14, Figure 58.18
Chip enable setup time	t_{CS}	0	—	ns	Figure 58.11 to Figure 58.14, Figure 58.18
Address hold time	t_{AH}	0	—	ns	Figure 58.11 to Figure 58.14
BS delay time	t_{BSD}	—	8	ns	Figure 58.11 to Figure 58.18

Table 58.18 Bus timing (2 of 2)

Conditions: V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 30 pF (CKIO), 30 pF (others), $T_j$min = −40°C

Parameter	Symbol	CKIO = 1/tCKcyc (Max 83.3 MHz)		Unit	Reference Figure
		Min.	Max.		
CSn# delay time 1	t_{CSD1}	0	8	ns	Figure 58.11 to Figure 58.18
Read/write delay time 1	t_{RWD1}	0	8	ns	Figure 58.11 to Figure 58.18
Read strobe delay time	t_{RSD}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 8	ns	Figure 58.11 to Figure 58.18
Read data setup time 1	t_{RDS1}	$1/2t_{CKcyc}$ + 4	—	ns	Figure 58.11 to Figure 58.17
Read data setup time 3	t_{RDS3}	$1/2t_{CKcyc}$ + 4	—	ns	Figure 58.18
Read data hold time 1	t_{RDH1}	0	—	ns	Figure 58.11 to Figure 58.17
Read data hold time 3	t_{RDH3}	0	—	ns	Figure 58.18
Write enable delay time 1	t_{WED1}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 8	ns	Figure 58.11 to Figure 58.16
Write enable delay time 2	t_{WED2}	—	8	ns	Figure 58.17
Write data delay time 1	t_{WDD1}	—	8	ns	Figure 58.11 to Figure 58.17
Write data hold time 1	t_{WDH1}	0	—	ns	Figure 58.11 to Figure 58.17
Write data hold time 4	t_{WDH4}	0	—	ns	Figure 58.11 to Figure 58.15
WAIT# setup time	t_{WTS}	$1/2t_{CKcyc}$ + 3.5	—	ns	Figure 58.12 to Figure 58.18
WAIT# hold time	t_{WTH}	$1/2t_{CKcyc}$	—	ns	Figure 58.12 to Figure 58.18
AH# delay time	t_{AHD}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 8	ns	Figure 58.15
Multiplex address delay time	t_{MAD}	—	8	ns	Figure 58.15
Multiplex address hold time	t_{MAH}	0	—	ns	Figure 58.15
Address setup time to AH#	t_{AVVH}	$1/2t_{CKcyc}$ − 2	—	ns	Figure 58.15
DACK/TEND delay time	t_{DACD}	See section 58.5.4. DMAC Timing		ns	Figure 58.11 to Figure 58.18

Note: Notation of $1/2t_{CKcyc}$ in the delay time, setup time, and hold time shows 1/2 cycles from the clock rising edge, that is, the reference of clock falling.

Table 58.19 Bus timing (1 of 2)

Conditions: V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 15 pF (CKIO), 30 pF (others), $T_j$min = −40°C

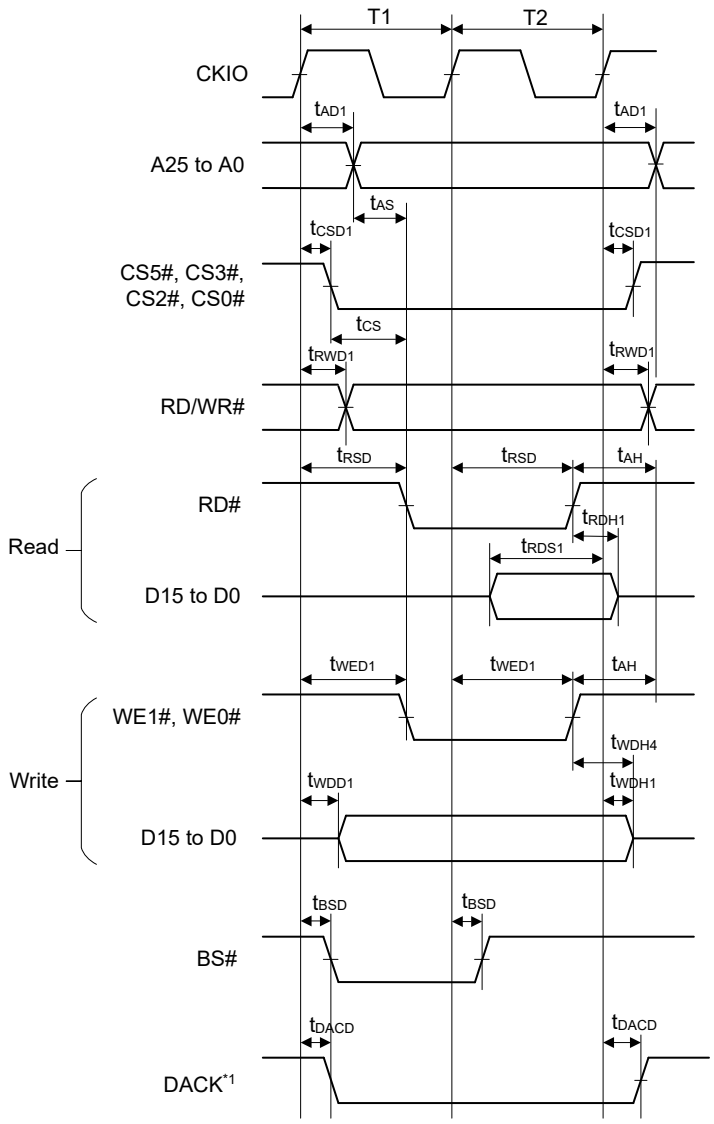
Parameter	Symbol	CKIO = 1/tCKcyc (Max 125 MHz)		Unit	Reference Figure
		Min.	Max.		
Address delay time 1	t_{AD1}	0	6	ns	Figure 58.11 to Figure 58.18
Address delay time 2	t_{AD2}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 6	ns	Figure 58.18
Address setup time	t_{AS}	0	—	ns	Figure 58.11 to Figure 58.14, Figure 58.18
Chip enable setup time	t_{CS}	0	—	ns	Figure 58.11 to Figure 58.14, Figure 58.18
Address hold time	t_{AH}	0	—	ns	Figure 58.11 to Figure 58.14
BS delay time	t_{BSD}	—	6	ns	Figure 58.11 to Figure 58.18
CSn# delay time 1	t_{CSD1}	0	6	ns	Figure 58.11 to Figure 58.18
Read/write delay time 1	t_{RWD1}	0	6	ns	Figure 58.11 to Figure 58.18
Read strobe delay time	t_{RSD}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 6	ns	Figure 58.11 to Figure 58.18
Read data setup time 1	t_{RDS1}	$1/2t_{CKcyc}$ + 3.5	—	ns	Figure 58.11 to Figure 58.17
Read data setup time 3	t_{RDS3}	$1/2t_{CKcyc}$ + 3.5	—	ns	Figure 58.18
Read data hold time 1	t_{RDH1}	0	—	ns	Figure 58.11 to Figure 58.17

Table 58.19 Bus timing (2 of 2)

Conditions: V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 15 pF (CKIO), 30 pF (others), $T_j$min = −40°C

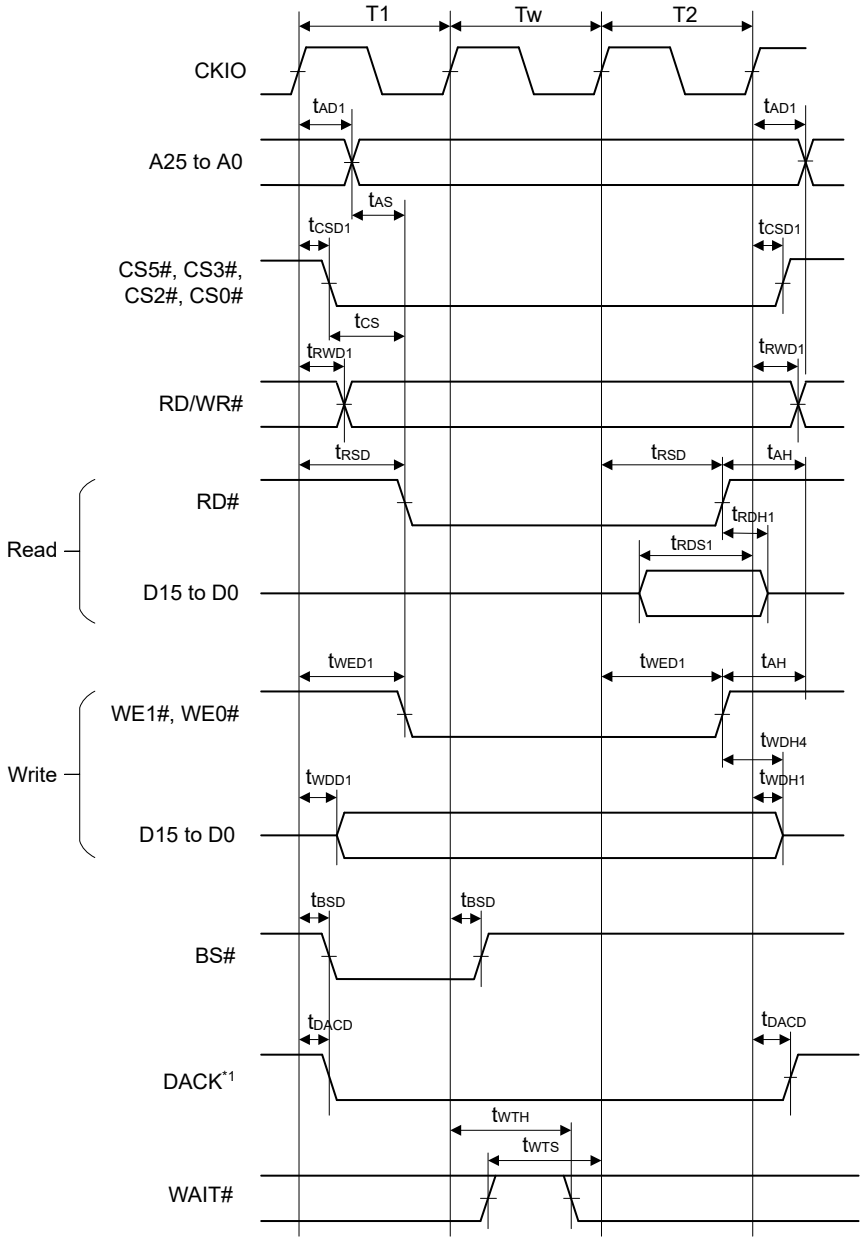
Parameter	Symbol	CKIO = 1/tCKcyc (Max 125 MHz)		Unit	Reference Figure
		Min.	Max.		
Read data hold time 3	t_{RDH3}	0	—	ns	Figure 58.18
Write enable delay time 1	t_{WED1}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 6	ns	Figure 58.11 to Figure 58.16
Write enable delay time 2	t_{WED2}	—	6	ns	Figure 58.17
Write data delay time 1	t_{WDD1}	—	6	ns	Figure 58.11 to Figure 58.17
Write data hold time 1	t_{WDH1}	0	—	ns	Figure 58.11 to Figure 58.17
Write data hold time 4	t_{WDH4}	0	—	ns	Figure 58.11 to Figure 58.15
WAIT# setup time	t_{WTS}	$1/2t_{CKcyc}$ + 3.5	—	ns	Figure 58.12 to Figure 58.18
WAIT# hold time	t_{WTH}	$1/2t_{CKcyc}$	—	ns	Figure 58.12 to Figure 58.18
AH# delay time	t_{AHD}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 6	ns	Figure 58.15
Multiplex address delay time	t_{MAD}	—	6	ns	Figure 58.15
Multiplex address hold time	t_{MAH}	0	—	ns	Figure 58.15
Address setup time to AH#	t_{AVVH}	$1/2t_{CKcyc}$ − 2	—	ns	Figure 58.15
DACK/TEND delay time	t_{DACD}	See section 58.5.4. DMAC Timing		ns	Figure 58.11 to Figure 58.18

Note: Notation of $1/2t_{CKcyc}$ in the delay time, setup time, and hold time shows 1/2 cycles from the clock rising edge, that is, the reference of clock falling.



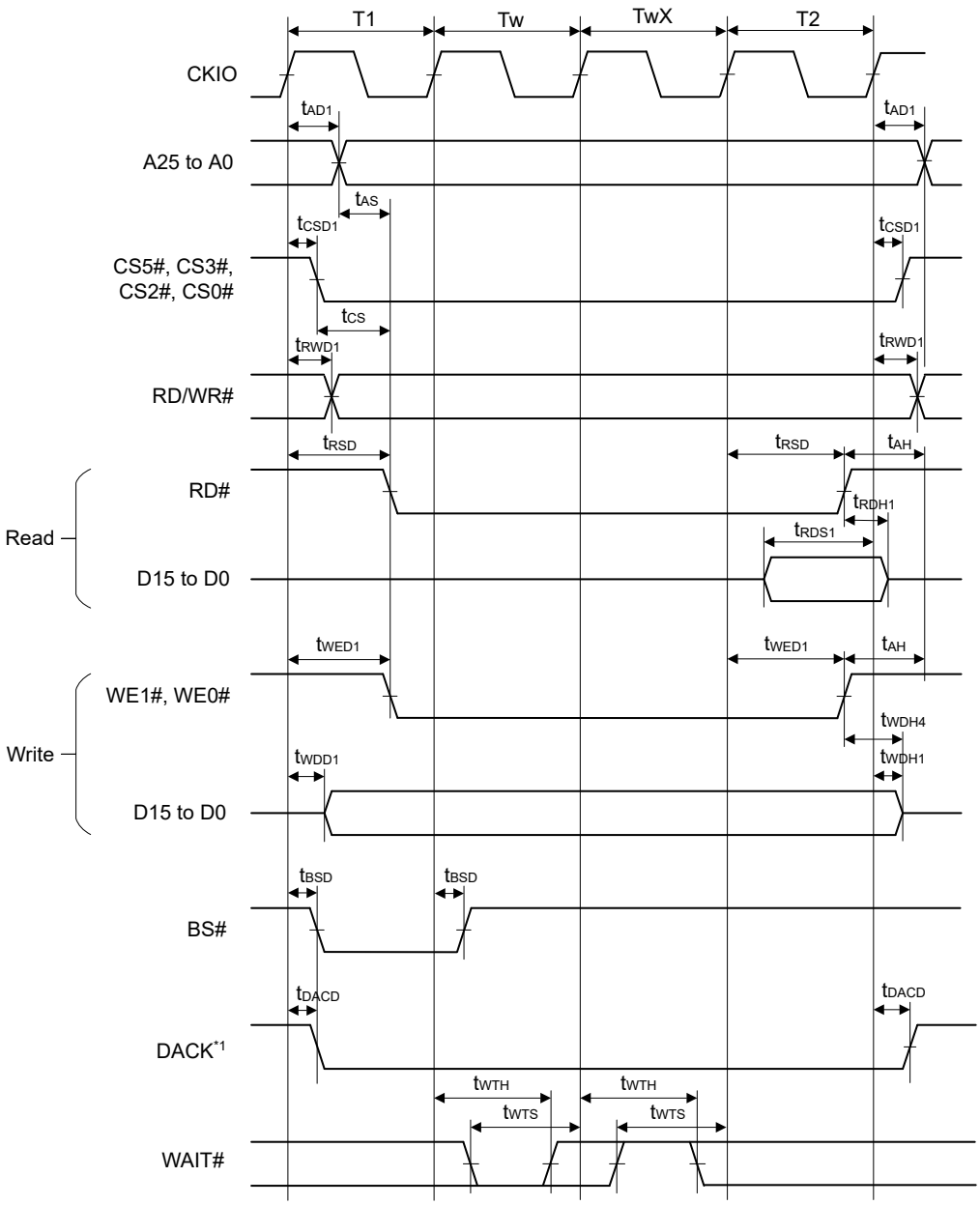
Note 1. DACK is waveform when active-low is specified.

Figure 58.11 SRAM interface basic bus cycle (no wait)



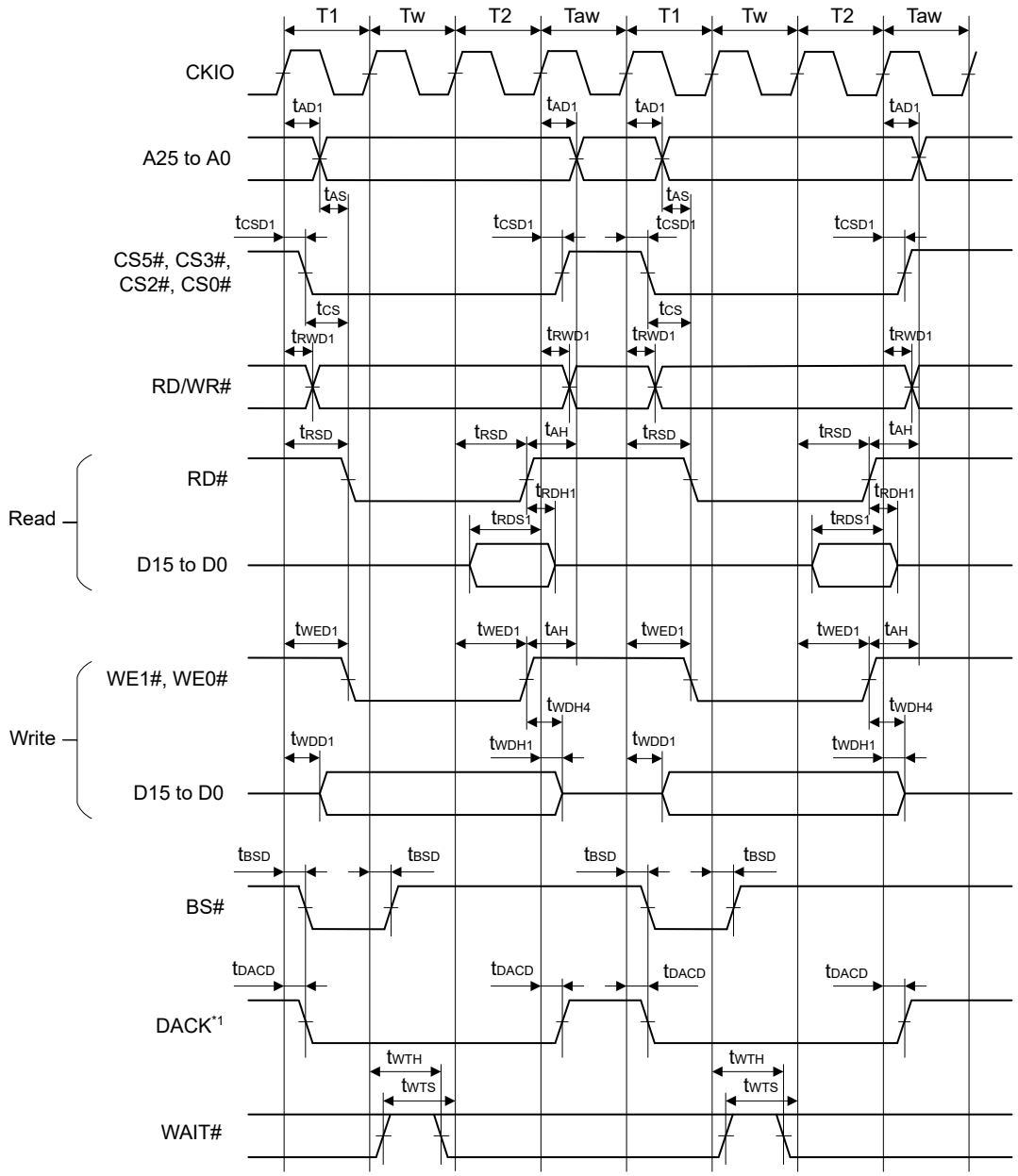
Note 1. DACK is waveform when active-low is specified.

Figure 58.12 SRAM interface basic bus cycle (software wait 1)



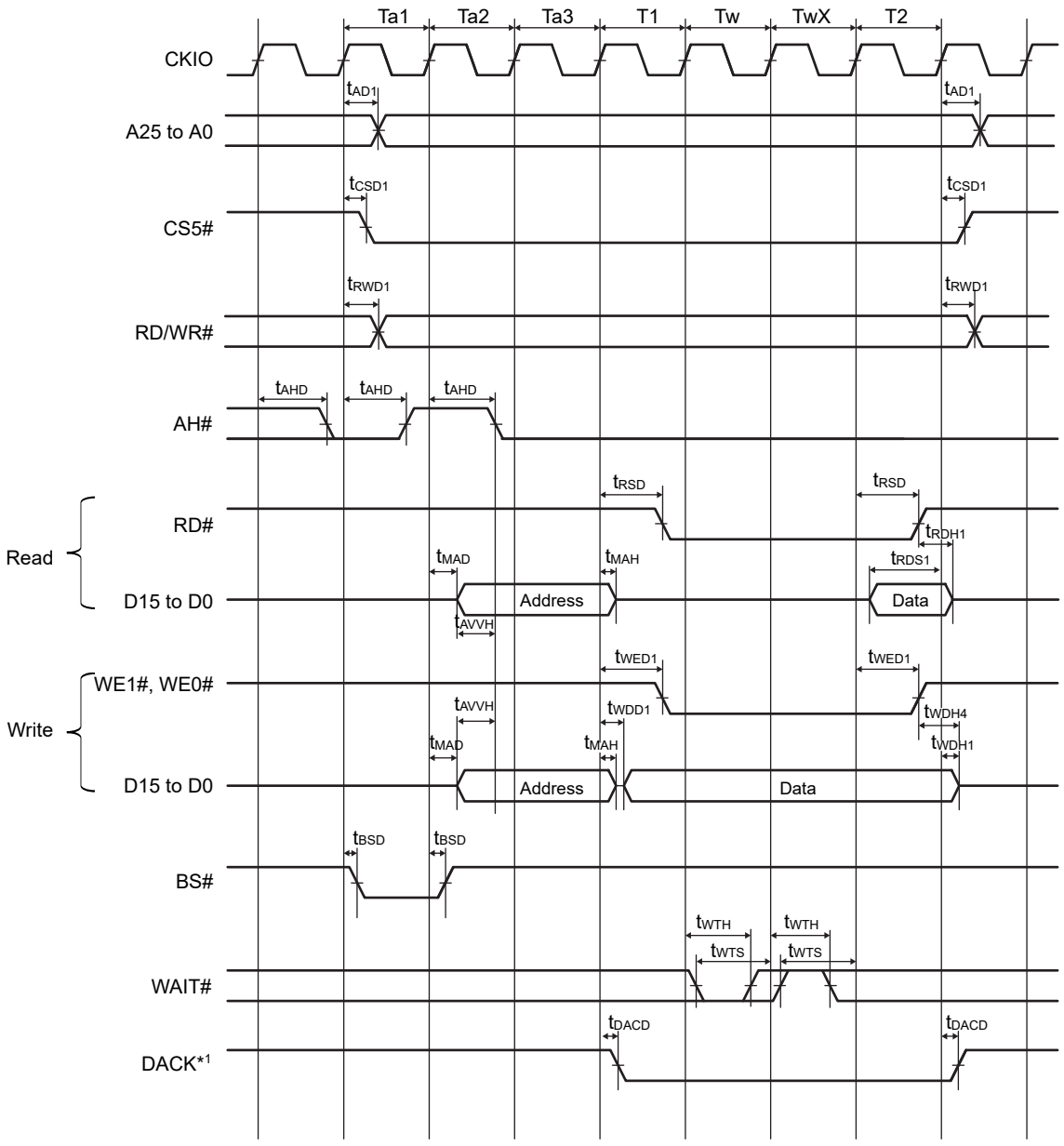
Note 1. DACK is waveform when active-low is specified.

Figure 58.13 SRAM interface basic bus cycle (software wait 1, external wait 1 inserted)



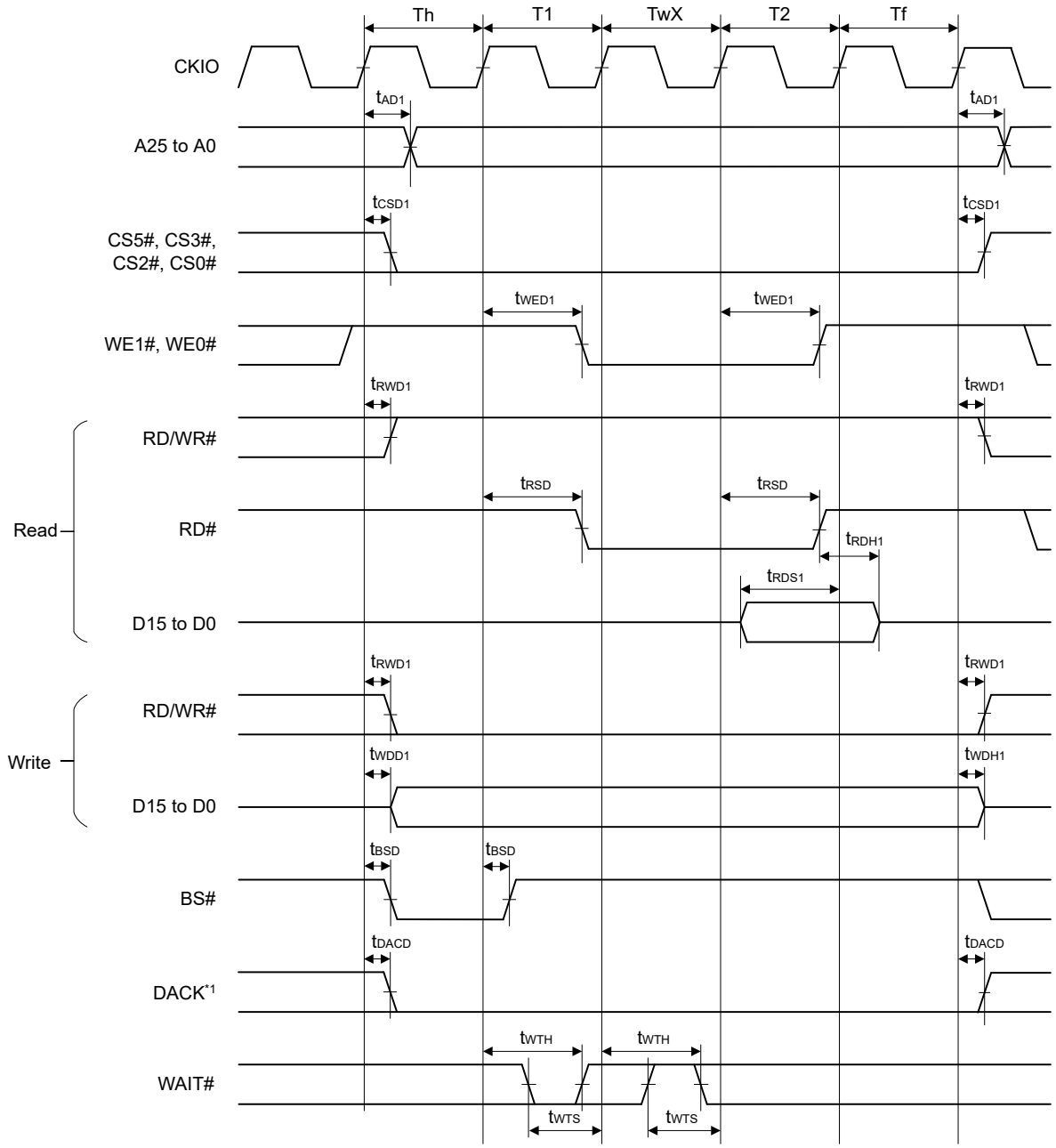
Note 1. DACK is waveform when active-low is specified.

Figure 58.14 SRAM interface basic bus cycle (software wait 1, external wait enabled (WM = 0), no idle cycle)



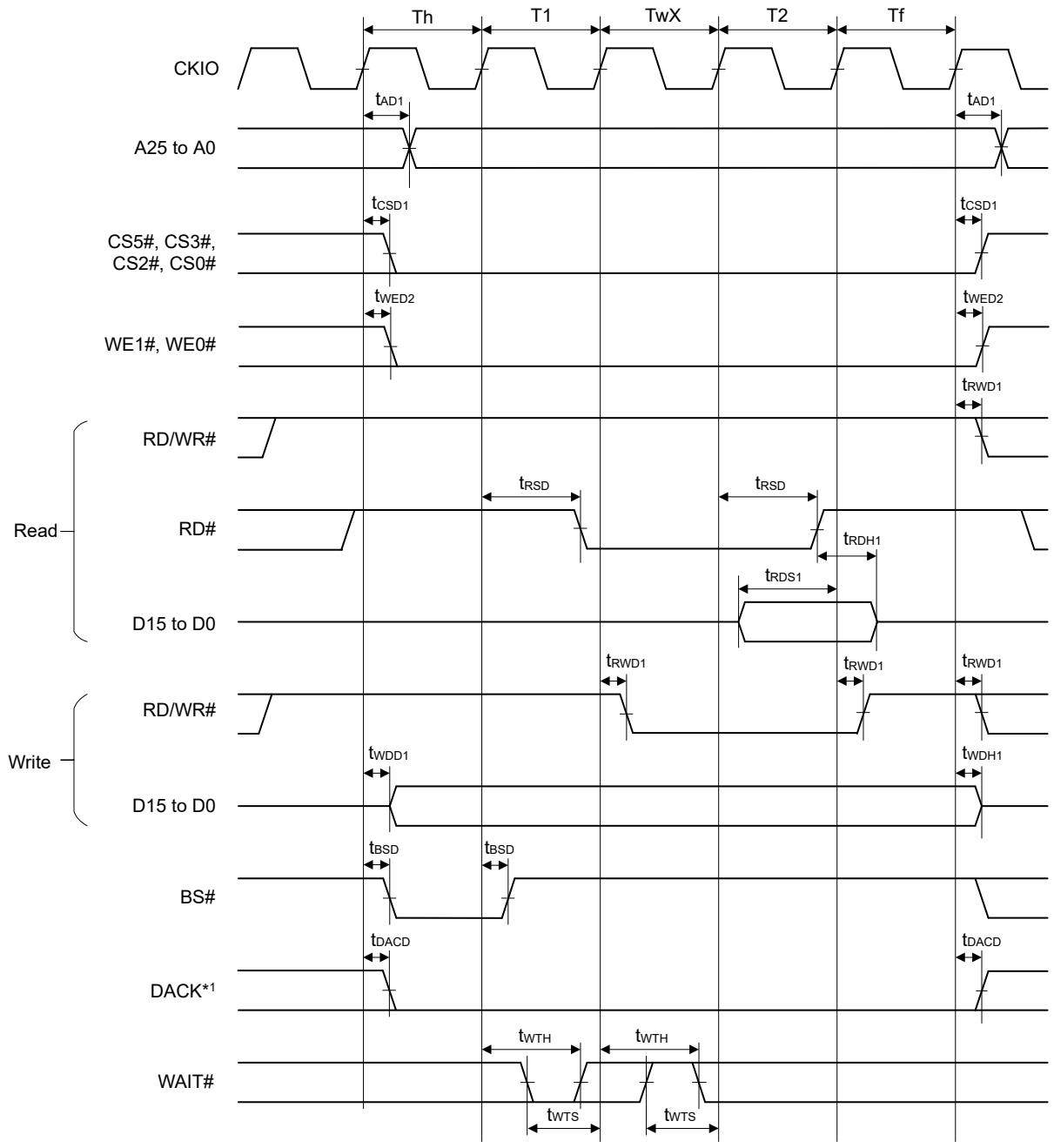
Note 1. DACK is waveform when active-low is specified.

Figure 58.15 MPX-I/O interface bus cycle (address cycle 3, software wait 1, external wait 1 inserted)



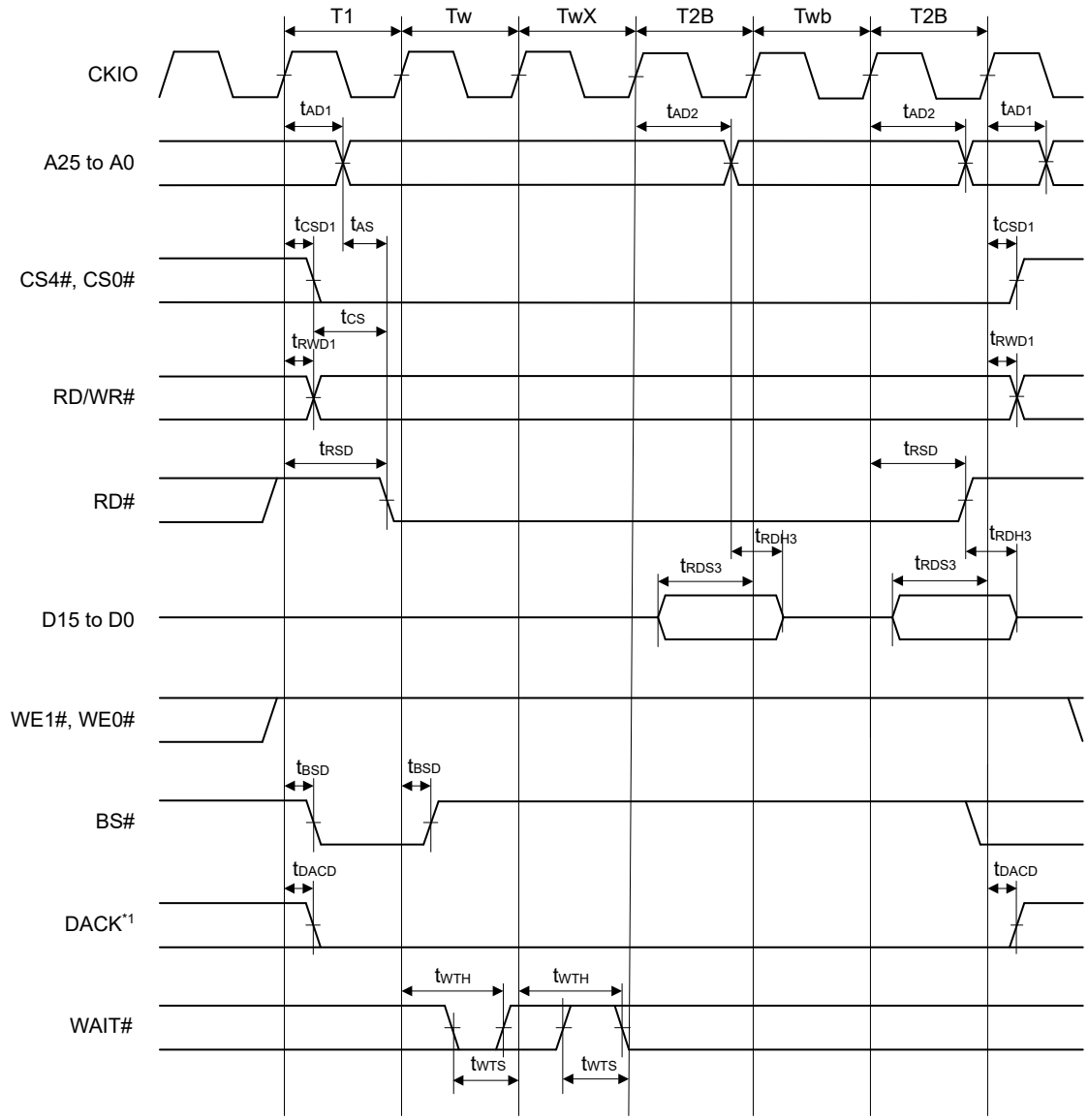
Note 1. DACK is waveform when active-low is specified.

Figure 58.16 SRAM bus cycle with byte selection (SW = 1 cycle, HW = 1 cycle, asynchronous external wait 1 inserted, BAS = 0 (write cycle UB/LB control))



Note 1. DACK is waveform when active-low is specified.

Figure 58.17 SRAM bus cycle with byte selection (SW = 1 cycle, HW = 1 cycle, asynchronous external wait 1 inserted, BAS = 1 (write cycle WE control))



Note 1. DACK is waveform when active-low is specified.

Figure 58.18 Burst ROM read cycle (software wait 1, asynchronous external wait 1 inserted, burst wait 1, 2)

58.5.4 DMAC Timing

Table 58.20 DMAC timing

Conditions: V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 30 pF (CKIO), 30 pF (others), $T_j$min = −40°C

Parameter		Symbol	Min.*1	Max.	Unit	Reference figure
DMAC	DREQ pulse width	t_{DRQW}	t_{PLcyc} × 2	—	ns	Figure 58.19
	DACK and TEND delay time	t_{DACD}	0	8	ns	Figure 58.20

Note 1. t_{PLcyc}: PCLKL cycle

Table 58.21 DMAC timing

Conditions: V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 15 pF (CKIO), 30 pF (others), $T_j$min = −40°C

Parameter		Symbol	Min.*1	Max.	Unit	Reference figure
DMAC	DREQ pulse width	t_{DRQW}	t_{PLcyc} × 2	—	ns	Figure 58.19
	DACK and TEND delay time	t_{DACD}	0	6	ns	Figure 58.20

Note 1. t_{PLcyc}: PCLKL cycle

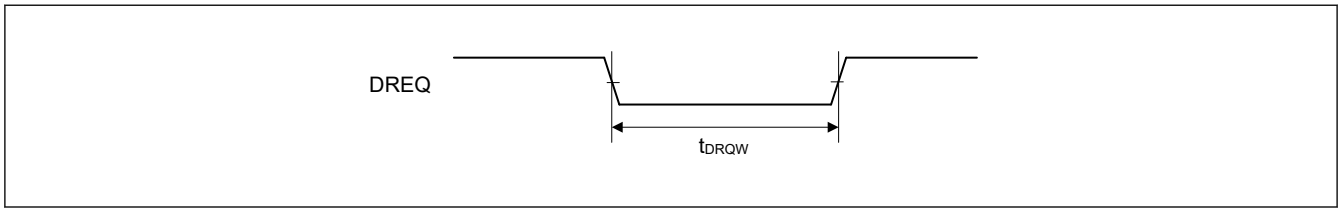


Figure 58.19 DREQ input timing

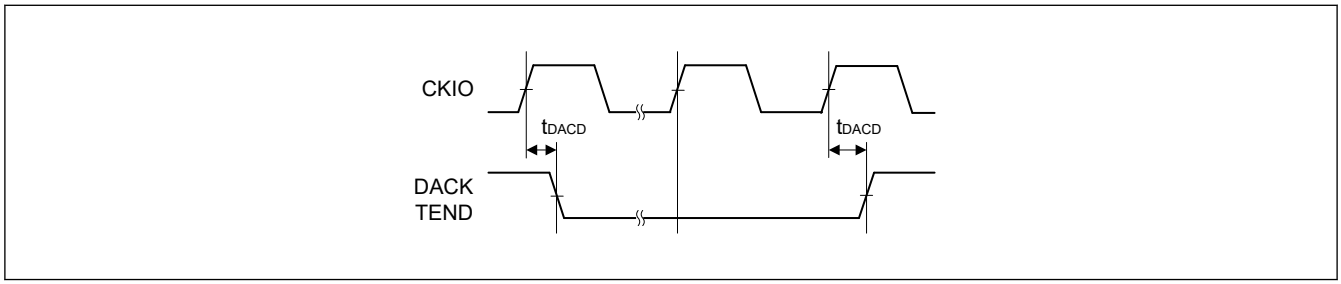


Figure 58.20 DACK and TEND output timing

58.5.5 On-Chip Peripheral Module Timing

58.5.5.1 I/O Port Timing

Table 58.22 I/O port timing

Parameter		Symbol	Min.	Max.	Unit*1	Reference figure
I/O port	Input data pulse width	t_{PRW}	1.5	—	t_{PLcyc}	Figure 58.21

Note 1. t_{PLcyc}: PCLKL cycle

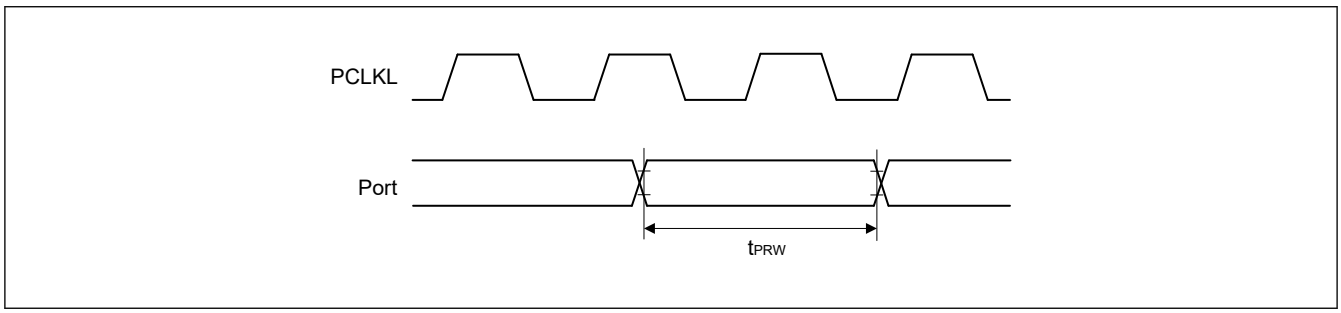


Figure 58.21 I/O port input timing

58.5.5.2 CMTW Timing

Table 58.23 CMTW timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
CMTW	Input capture input pulse width	Single-edge setting	$t_{CMTWICW}$	1.5	—	t_{PLcyc}	Figure 58.22
		Both-edge setting		2.5	—		

Note 1. t_{PLcyc}: PCLKL cycle

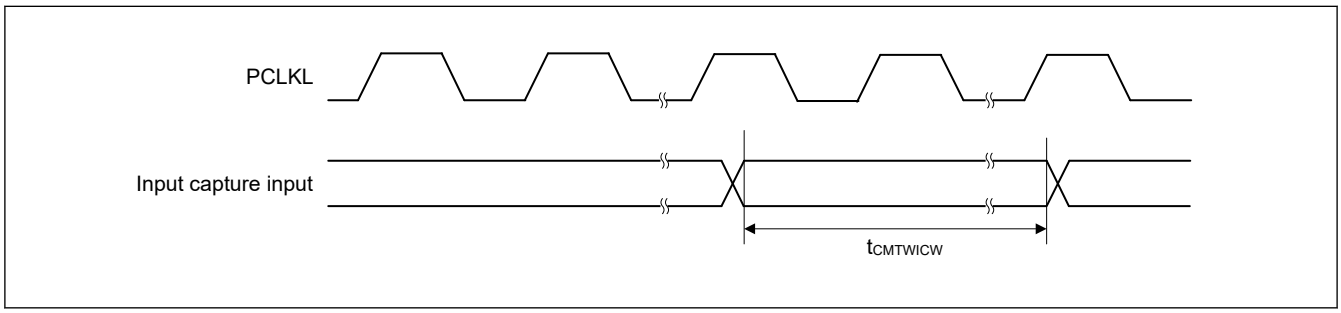


Figure 58.22 CMTW input capture input timing

58.5.5.3 MTU3 Timing

Table 58.24 MTU3 timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
MTU3	Input capture input pulse width	Single-edge setting	t_{MTICW}	2.5	—	t_{PHcyc}	Figure 58.23
		Both-edge setting		3.5	—		
	Timer clock pulse width	Single-edge setting	t_{MTCKWH}, t_{MTCKWL}	2.5	—	t_{PHcyc}	Figure 58.24
		Both-edge setting		3.5	—		
		Phase counting mode		3.5	—		

Note 1. t_{PHcyc}: PCLKH cycle

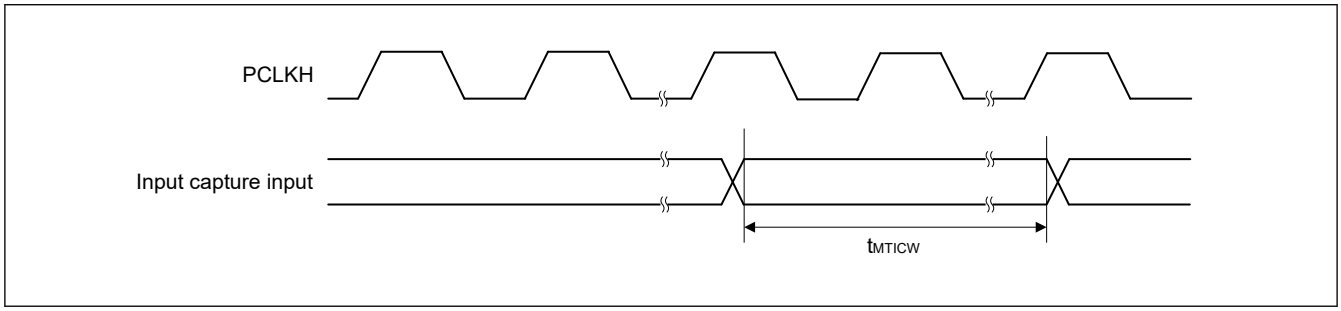


Figure 58.23 MTU3 input capture input timing

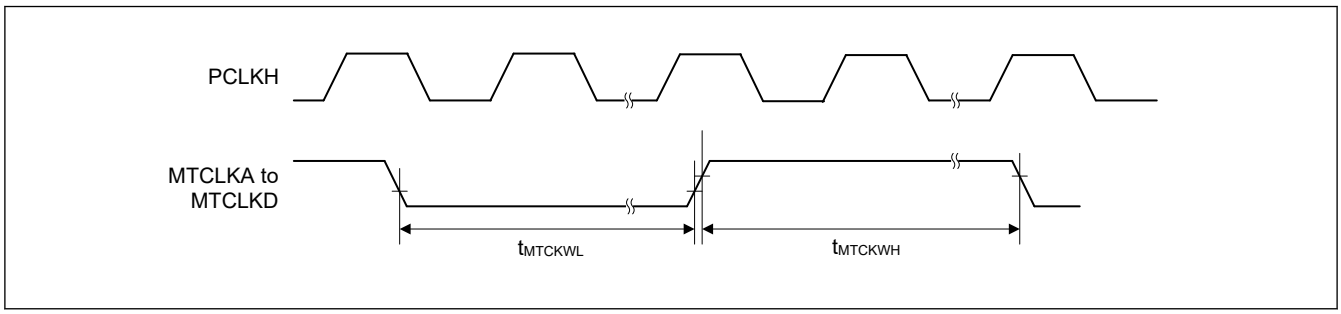


Figure 58.24 MTU3 clock input timing

58.5.5.4 POE3 Timing

Table 58.25 POE3 timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
POE3	POEn# input pulse width		t_{POEW}	2.5	—	t_{PHcyc}	Figure 58.25
	Output disable time	Transition of the POEn# signal level	t_{POEDI}	—	5 × PCLKH + 0.1	μs	Figure 58.26
		Simultaneous conduction of output pins	t_{POEDO}	—	3 × PCLKH + 0.1	μs	Figure 58.27
		Register setting	t_{POEDS}	—	PCLKH + 0.1	μs	Figure 58.28
		Oscillation stop detection	t_{POEDOS}	—	74	μs	Figure 58.29

Note 1. t_{PHcyc}: PCLKH cycle

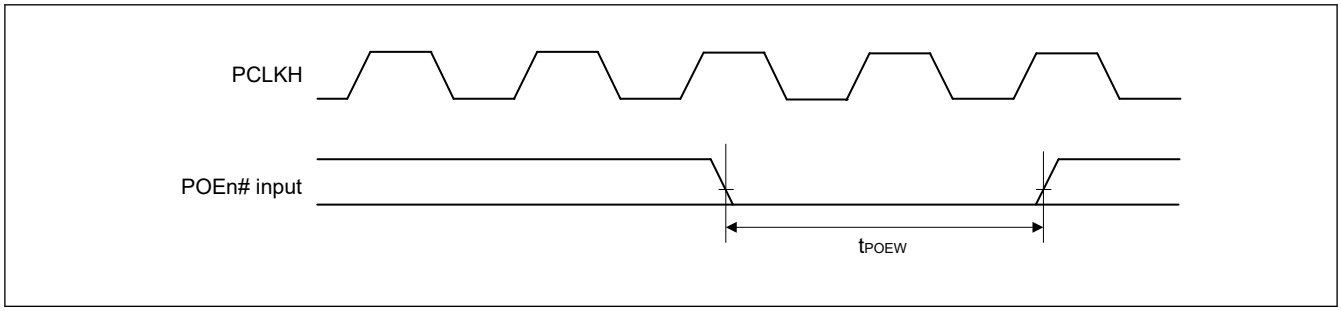


Figure 58.25 POEn# input pulse timing

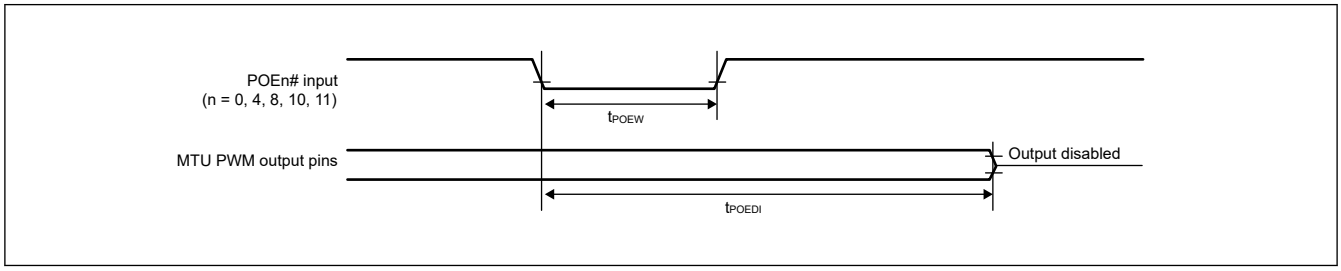
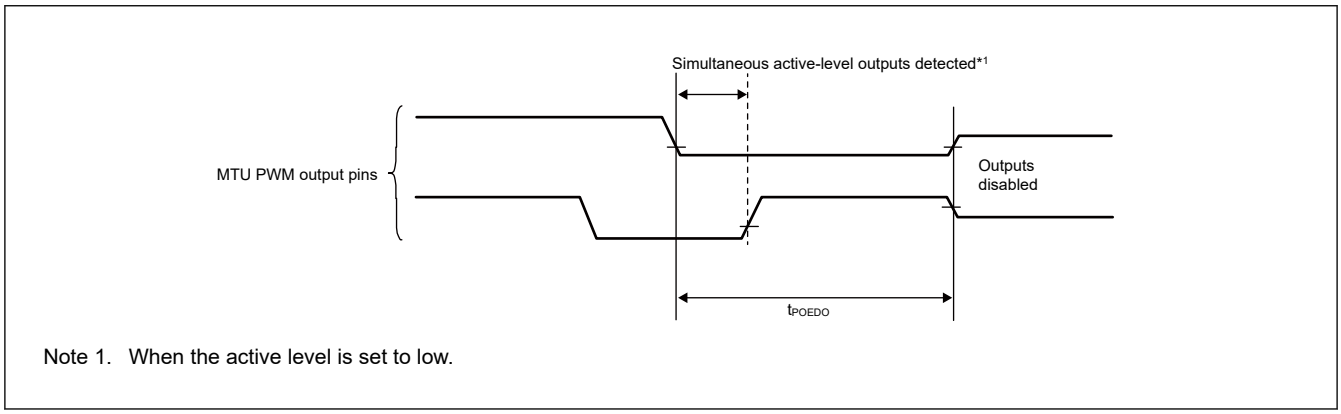


Figure 58.26 Output disable time for POE in response to transition of the POEn# signal level



Note 1. When the active level is set to low.

Figure 58.27 Output disable time for POE in response to the simultaneous conduction of output pins

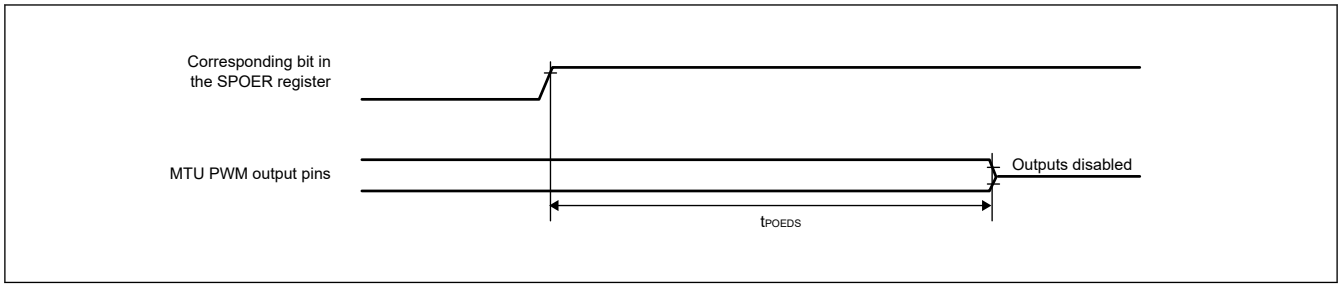


Figure 58.28 Output disable time for POE in response to the register setting

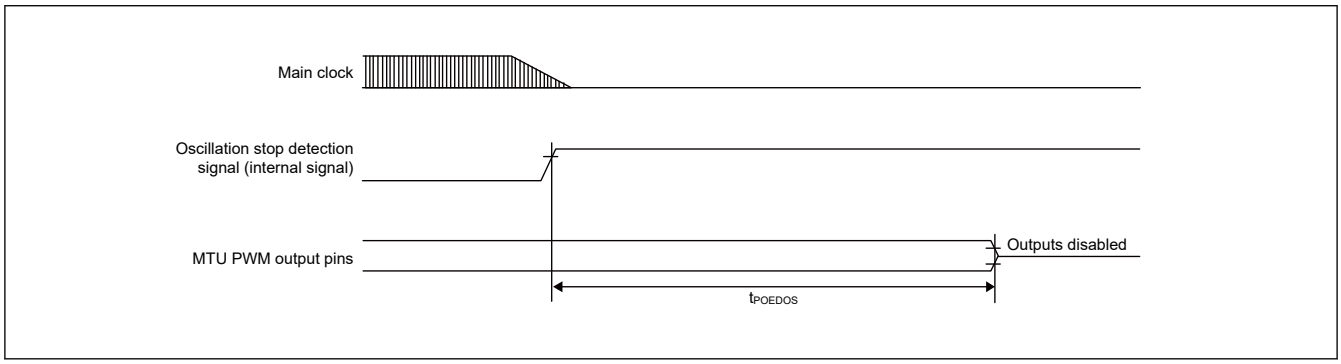


Figure 58.29 Output disable time for POE in response to the oscillation stop detection

58.5.5.5 GPT Timing

Table 58.26 GPT timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
GPT	Input capture input pulse width	Single-edge setting	t_{GTICW}	2.5	—	t_{PHcyc}	Figure 58.30
		Both-edge setting		3.5	—		
	External trigger input pulse width	Single-edge setting	t_{GTEW}	2.5	—	t_{PHcyc}	Figure 58.31
		Both-edge setting		3.5	—		

Note 1. t_{PHcyc}: PCLKH cycle (LLPP channels), PCLKM cycle (Other channels)

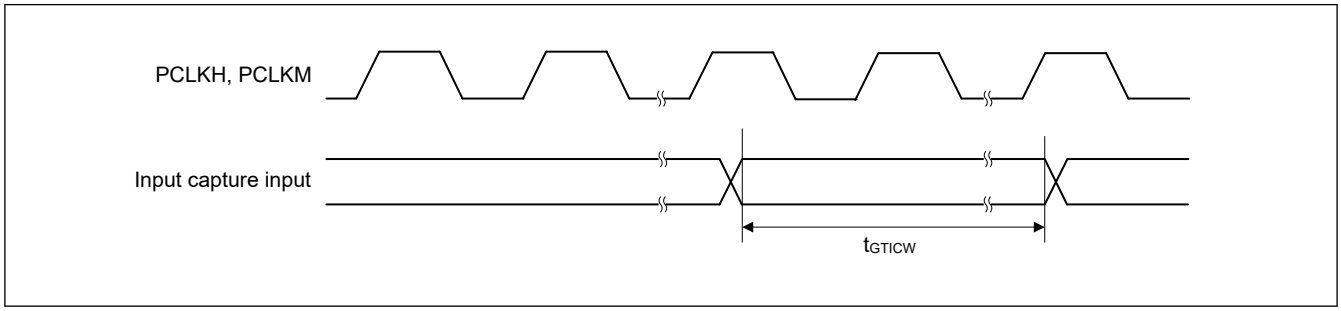


Figure 58.30 GPT input capture input timing

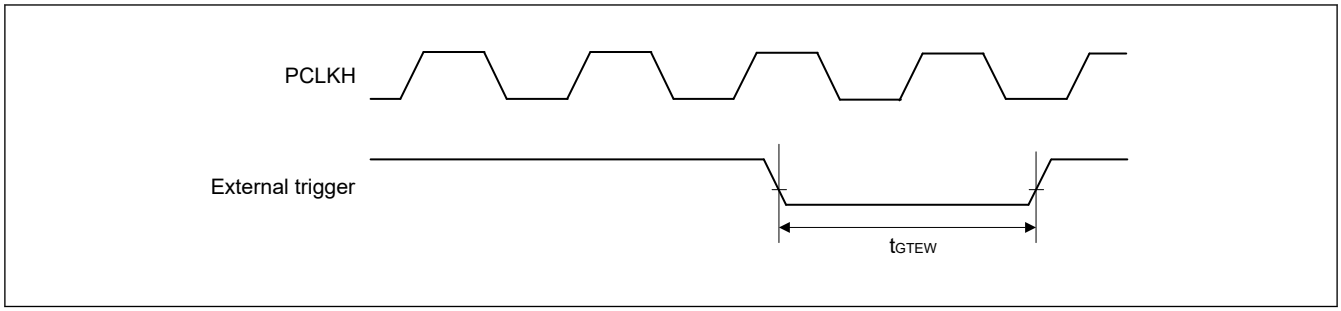


Figure 58.31 GPT external trigger input timing

58.5.5.6 POEG Timing

Table 58.27 POEG timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
POEG	GTETRGn input pulse width (n = A to D)		t_{POEGW}	2.5	—	t_{PHcyc}	Figure 58.32
	Output disable time	Input level detection of the GTETRGn pin (via flag)	t_{POEGDI}	—	3 × PCLKH + 0.1	μs	Figure 58.33
		Detection of the output stopping signal from GPT (dead time error, simultaneous high output, or simultaneous low output)	t_{POEGDO}	—	0.1	μs	Figure 58.34
		Register setting	t_{POEGDS}	—	PCLKH + 0.1	μs	Figure 58.35
		Oscillation stop detection	$t_{POEGDOS}$	—	74	μs	Figure 58.36

Note 1. t_{PHcyc}: PCLKH cycle (LLPP channels), PCLKL cycle (Other channels)

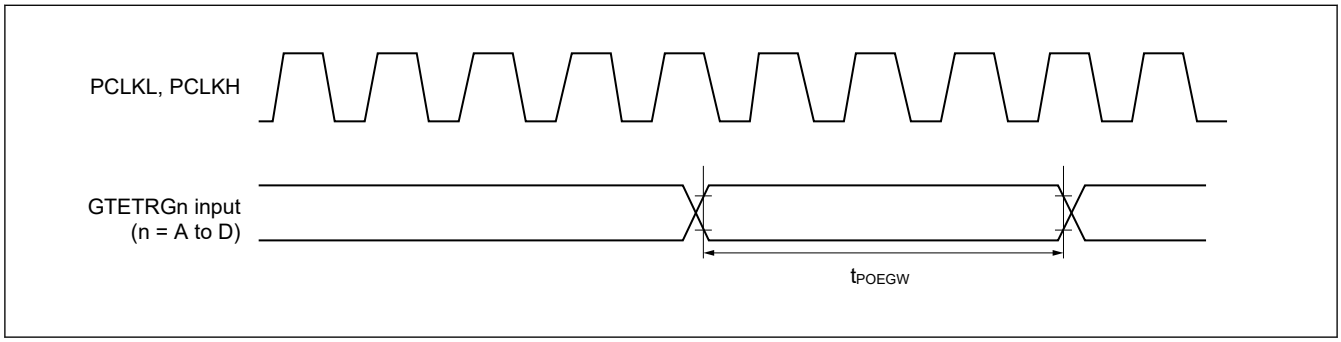


Figure 58.32 POEG input timing

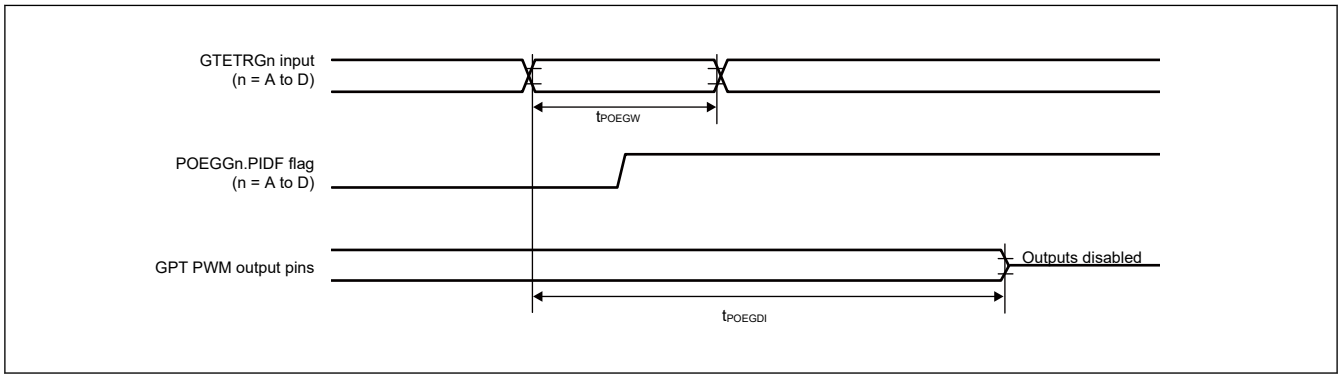
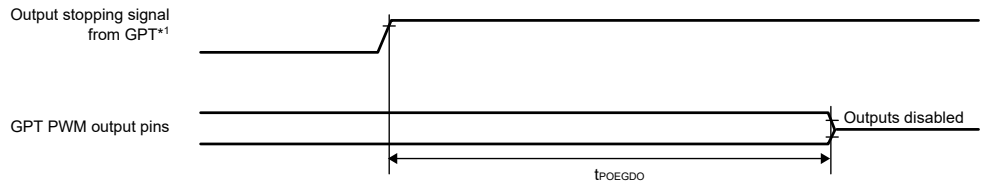


Figure 58.33 Output disable time for POEG via detection flag in response to the input level detection of the GTETRGn pin



Note 1. GPTn.GTST.DTEF (dead time error flag), GPTn.GTST.OABLF (simultaneous low out flag), or GPTn.GTST.OABHF (simultaneous high output flag)

Figure 58.34 Output disable time for POEG in response to detection of the output stopping signal from GPT

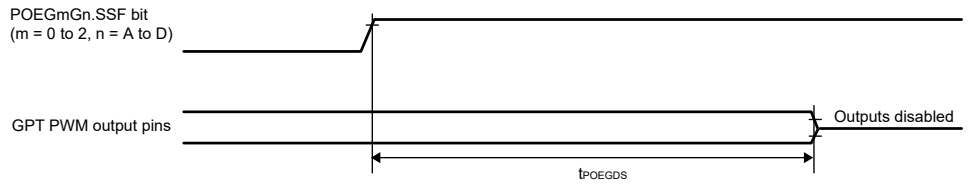


Figure 58.35 Output disable time for POEG in response to the register setting

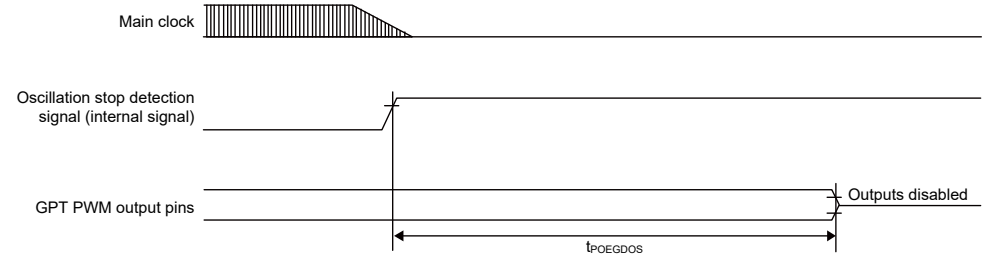


Figure 58.36 Output disable time for POEG in response to the oscillation stop detection

58.5.5.7 A/D Converter Trigger Timing

Table 58.28 A/D converter trigger timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
A/D converter	A/D converter trigger input pulse width	ADTRG0#, ADTRG1#, ADTRG2#	t_{TRGW}	1.5	—	$t_{PCLKLcyc}$	Figure 58.37

Note 1. $t_{PCLKLcyc}$: PCLKL cycle

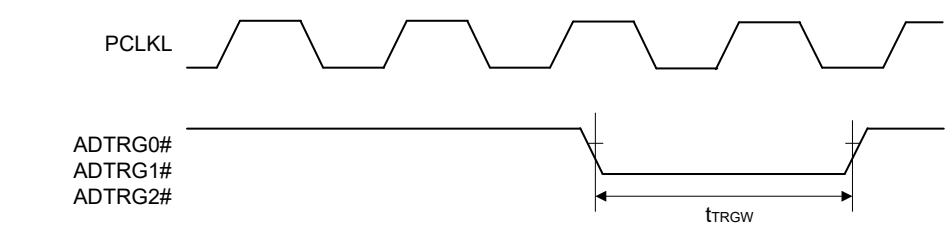


Figure 58.37 A/D converter trigger input timing (ADTRG0#, ADTRG1#, ADTRG2#)

58.5.5.8 SCI, SCIE Timing

Conditions: V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 30 pF (except Simple I2C)

Table 58.29 SCI, SCIE timing (1 of 2)

Parameter		Symbol	Min.	Max.	Unit	Reference figure
SCI, SCIE (Asynchronous)	Input clock cycle	t_{Scyc}	4	—	$t_{PSCIcyc}$	Figure 58.38
	Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time	t_{SCKr}	—	3	ns	
	Input clock fall time	t_{SCKf}	—	3	ns	
	Output clock cycle	t_{Scyc}	6	—	$t_{PSCIcyc}$	
	Output clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time	t_{SCKr}	—	4	ns	
	Output clock fall time	t_{SCKf}	—	4	ns	
SCI (Simple I2C, Standard mode)	SDA input rise time	t_{Sr}	—	1000	ns	Figure 58.39
	SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	2 × NF_{cyc} *1	ns	
	Data input setup time	t_{SDAS}	250	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	
SCI (Simple I2C, Fast mode)	SDA input rise time	t_{Sr}	—	300	ns	Figure 58.39
	SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	2 × NF_{cyc} *1	ns	
	Data input setup time	t_{SDAS}	100	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	

Table 58.29 SCI, SCIE timing (2 of 2)

Parameter			Symbol	Min.	Max.	Unit	Reference figure
SCI (Clock sync, Simple SPI) SCIE (Clock sync)	SCK output clock cycle (master)		t_{SPcyc}	2*2	65536	$t_{PSCIcyc}$	Figure 58.40 to Figure 58.45
	SCK input clock cycle (slave)			2*2	65536		
	SCK clock high pulse width		t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width		t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise/fall time		t_{SPCKR}, t_{SPCKF}	—	4	ns	
	Data input setup time	Internal clock	t_{SU}	7	—	ns	
		External clock		3	—		
	Data input hold time	Internal clock	t_{H}	3	—	ns	
		External clock		3	—		
	Data output delay time	Internal clock	t_{OD}	—	3	ns	
		External clock		—	12		
	Data output hold time	Internal clock	t_{OH}	-3	—	ns	
		External clock		0	—		
	Data rise/fall time		t_{DR}, t_{DF}	—	4	ns	
	Slave access time	Internal clock	t_{SA}	—	3 × $t_{PSCIcyc}$ + 12	ns	
		External clock		—	3 × $t_{PSCIcyc}$ + 12		
	Slave output release time	Internal clock	t_{REL}	—	3 × $t_{PSCIcyc}$ + 12	ns	
		External clock		—	3 × $t_{PSCIcyc}$ + 12		
SCI (Simple SPI)	SS input setup time		t_{LEAD}	1	—	t_{SPcyc}	
	SS input hold time		t_{LAG}	1	—	t_{SPcyc}	
	SS input rise/fall time		t_{SSR}, t_{SSF}	—	3	ns	

Note: $t_{PSCIcyc}$: PCLKSCIn cycle

Note 1. $N_{Fcyc} = 4^{n} \times 2^{m-1} \times t_{PSCIcyc}$
n: CCR2.CKS[1:0] (n = 0, 1, 2, 3)
m: CCR1.NFCS[2:0] (m = 1, 2, 3, 4)

Note 2. In case of PCLKSCIn = 125 MHz, Min. is 4.

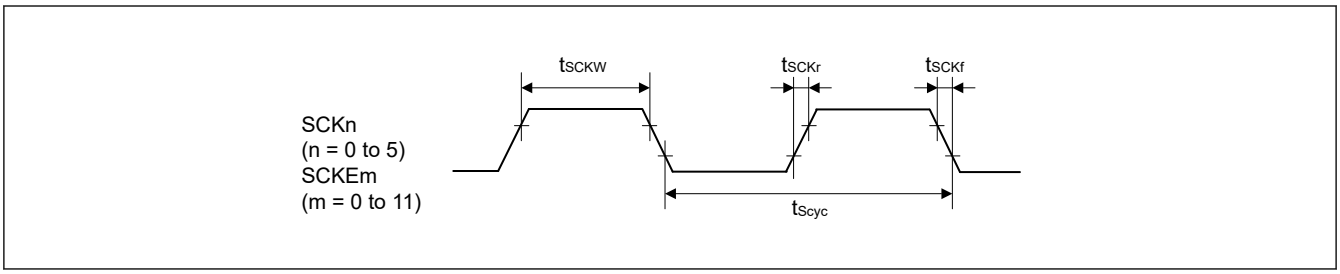
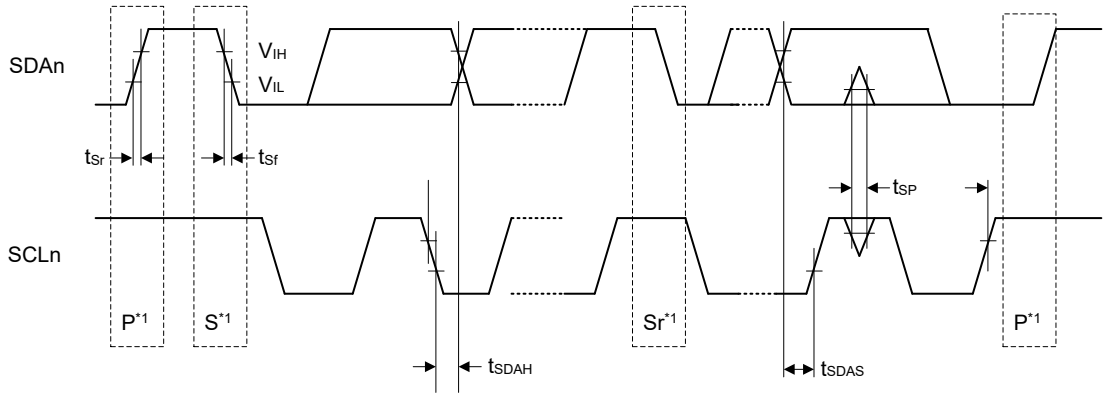


Figure 58.38 SCK clock input/output timing



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 58.39 SCI simple I2C mode timing

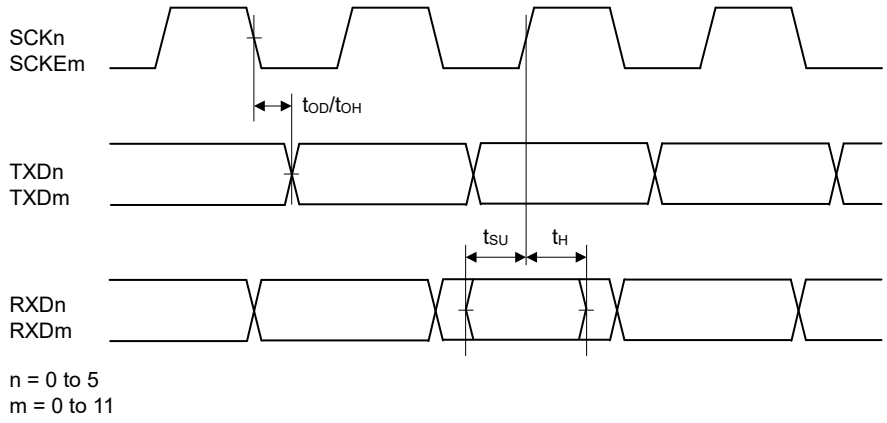
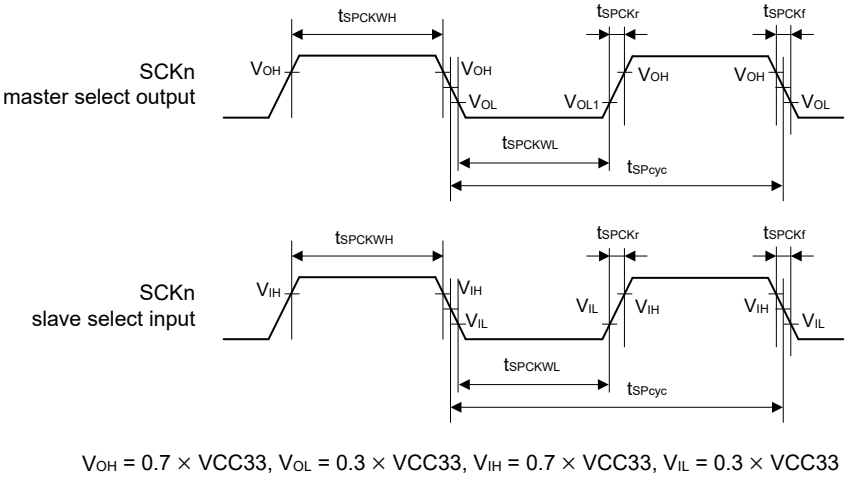


Figure 58.40 SCI input/output timing in clock synchronous mode



V_{OH} = 0.7 × VCC33, V_{OL} = 0.3 × VCC33, V_{IH} = 0.7 × VCC33, V_{IL} = 0.3 × VCC33

Figure 58.41 SCI simple SPI mode clock timing

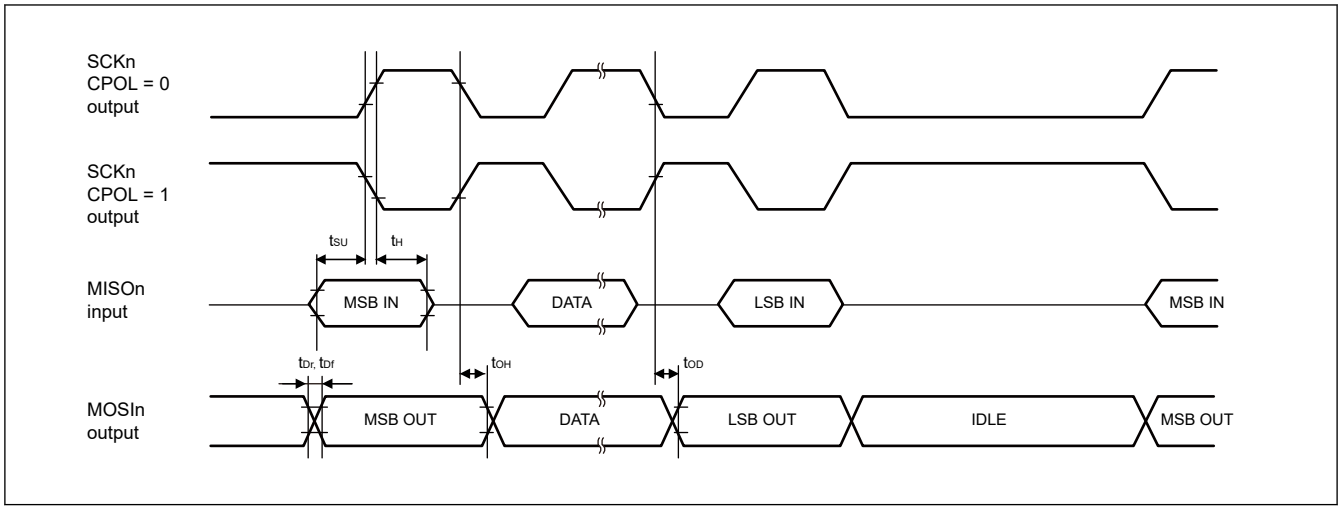


Figure 58.42 SCI simple SPI mode timing for master when CPHA = 0

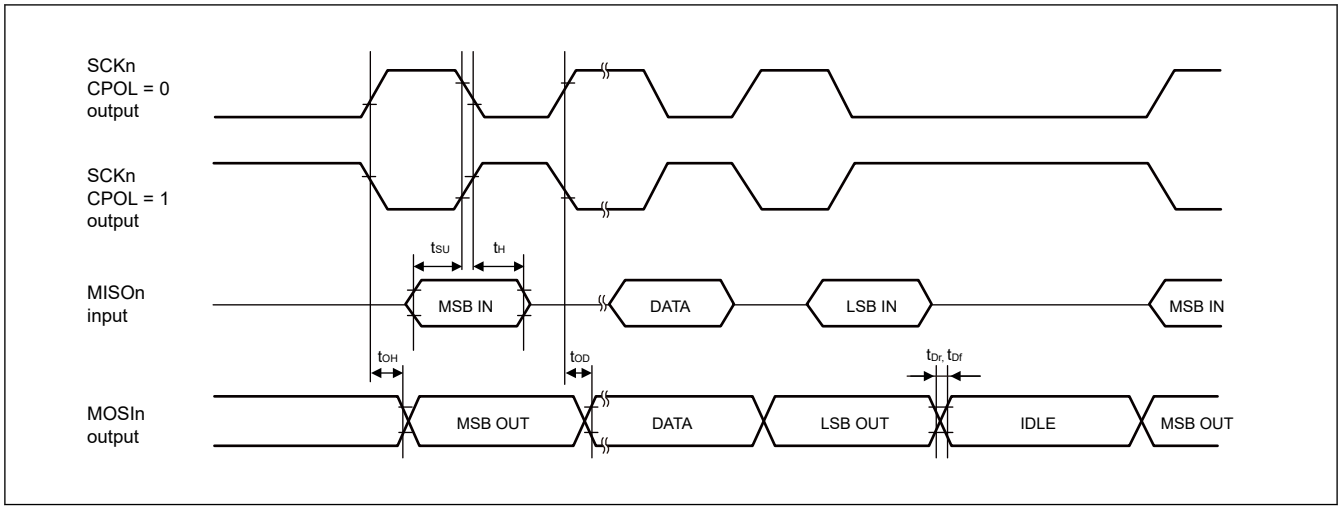


Figure 58.43 SCI simple SPI mode timing for master when CPHA = 1

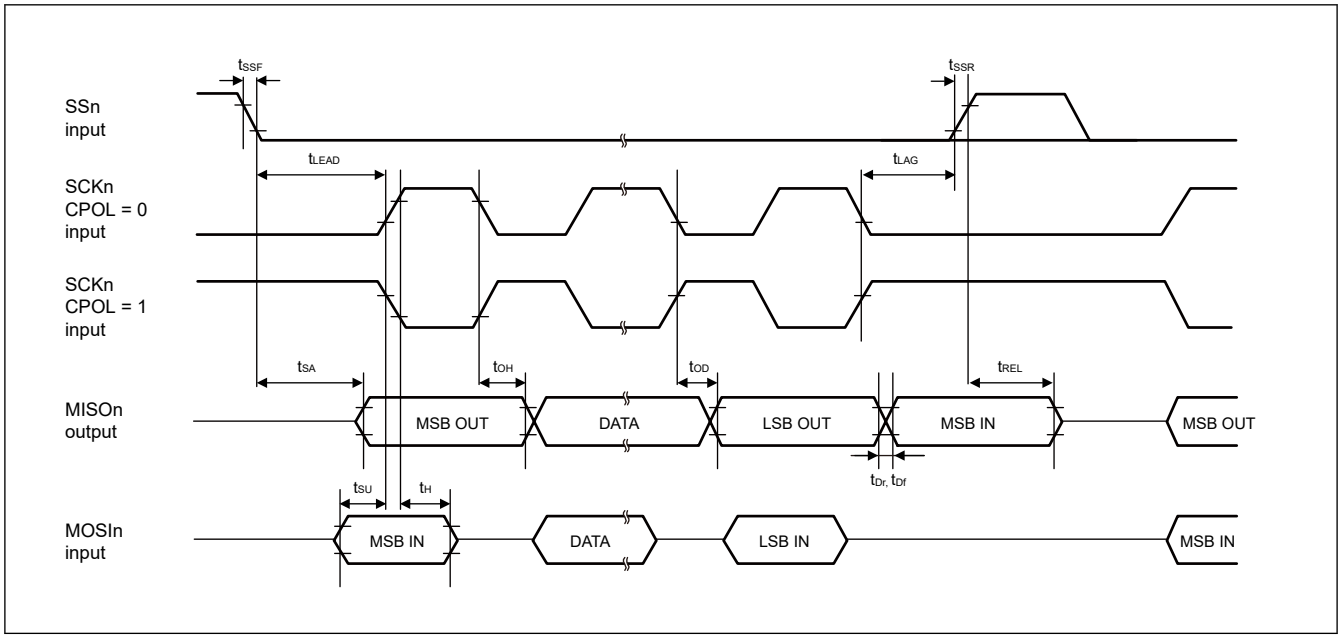


Figure 58.44 SCI simple SPI mode timing for slave when CPHA = 0

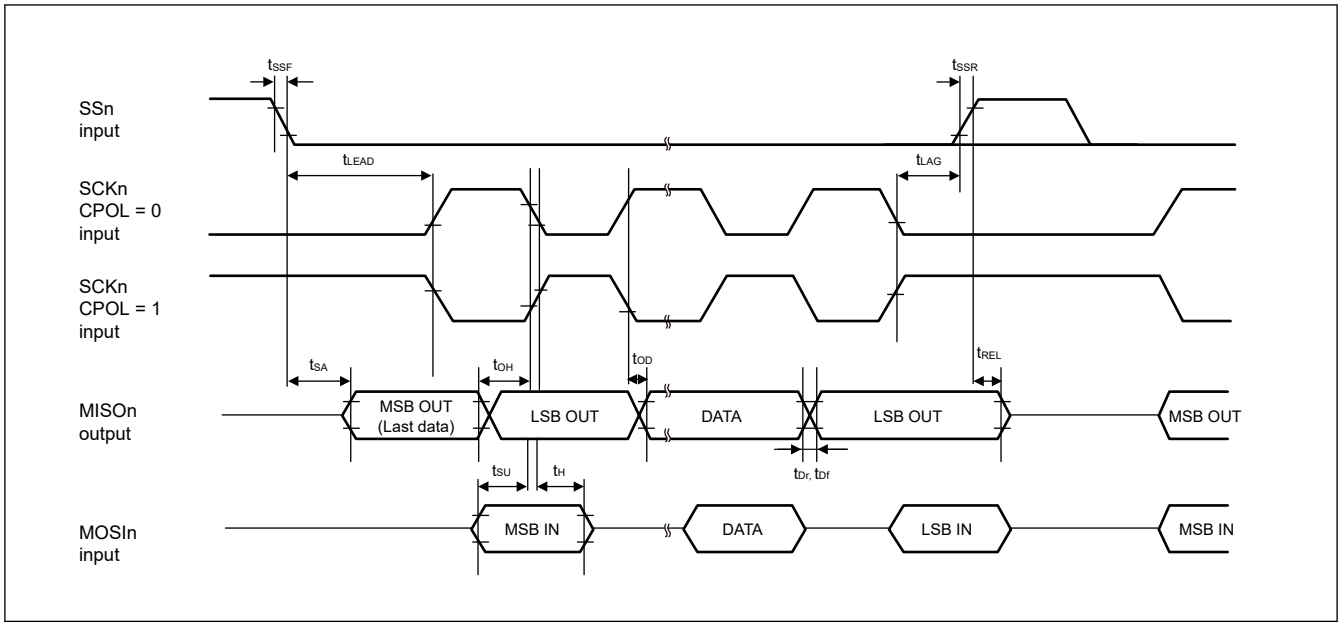


Figure 58.45 SCI simple SPI mode timing for slave when CPHA = 1

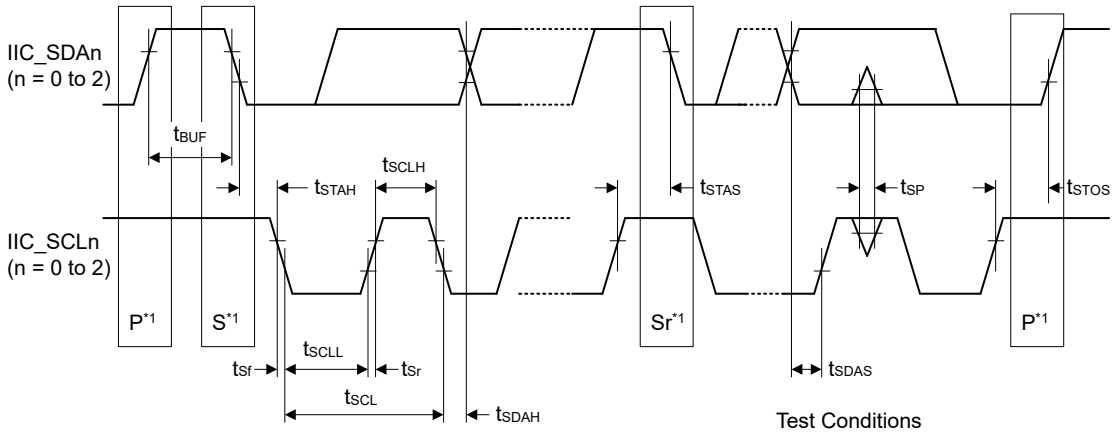
58.5.5.9 IIC Timing

Conditions: V_{OL} = 0.4 V, I_{OL} = 4 mA

Table 58.30 IIC timing

Parameter		Symbol	Min.*1 *2	Max.*1 *2	Unit	Reference figure
IIC (Standard-mode)	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 1300	—	ns	Figure 58.46
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rising time	t_{sr}	—	1000	ns	
	SCL, SDA input falling time	t_{sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	1000	—	ns	
	Stop condition input setup time	t_{STOS}	1000	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	
IIC (Fast-mode)	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 600	—	ns	
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rising time	t_{sr}	—*4	300	ns	
	SCL, SDA input falling time	t_{sf}	—*4	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	300	—	ns	
	Stop condition input setup time	t_{STOS}	300	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load*3	C_b	—	400	pF	

Note 1. t_{IICcyc}: IIC internal reference clock (IICΦ) cycle
Note 2. The value out of parentheses is applicable when the value of the ICMR3.NF[1:0] bits is 00b while the digital filter is enabled by setting ICFER.NFE = 1. The value within parentheses is applicable when the value of the ICMR3.NF[1:0] bits is 11b while the digital filter is enabled by setting ICFER.NFE = 1.
Note 3. Cb is the total capacitance of the bus lines.
Note 4. The minimum values are not specified for t_{sr} and t_{sf} in Fast-mode.



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 58.46 IIC bus interface input/output timing

58.5.5.10 CANFD Timing

Table 58.31 CANFD timing

Parameter		Symbol	CAN		CANFD		Unit	Reference figure
			Min.	Max.	Min.	Max.		
CANFD	Internal delay time	t_{node}	—	100	—	50	ns	Figure 58.47
	Transmission rate	—	—	1	—	8	Mbps	

Note: Internal delay time (t_{node}) = Internal transmission delay time (t_{output}) + Internal reception delay time (t_{input})

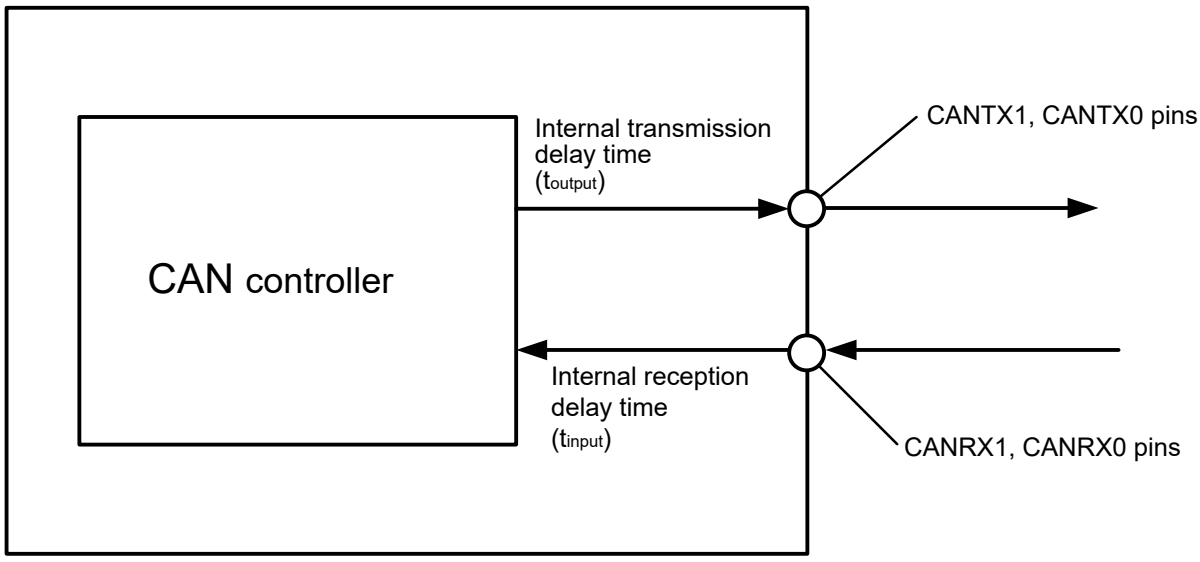


Figure 58.47 CAN interface condition

58.5.5.11 SPI Timing

Table 58.32 SPI timing (1 of 2)

Conditions: V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 30 pF

Parameter		Symbol	Min.*1	Max.*1	Unit*1	Reference figure
RSPCK clock cycle	Master	t_{SPcyc}	2*5	4096	t_{SPIcyc}	Figure 58.48
	Slave		2*5	4096		
RSPCK clock high level pulse width	Master	t_{SPCKWH}	(t_{SPcyc} − t_{SPCKr} − t_{SPCKf}) / 2 − 2.5	—	ns	
	Slave		1	—	t_{SPIcyc}	
RSPCK clock low level pulse width	Master	t_{SPCKWL}	(t_{SPcyc} − t_{SPCKr} − t_{SPCKf}) / 2 − 2.5	—	ns	
	Slave		1	—	t_{SPIcyc}	
RSPCK clock rising/falling time	Output	t_{SPCKr}, t_{SPCKf}	—	4	ns	
	Input		—	1	ns	
Data input setup time	Master	t_{SU}	5	—	ns	Figure 58.49 to Figure 58.55
	Slave		3	—		
Data input hold time	Master	t_H	3	—	ns	
	Slave		3	—		
SSL setup time	Master	t_{LEAD}	N × t_{SPcyc} − 3*2	N × t_{SPcyc} + 3*2	ns	Figure 58.49 to Figure 58.52
	Slave		4	—	t_{SPIcyc}	
SSL hold time	Master	t_{LAG}	N × t_{SPcyc} − 3*3	N × t_{SPcyc} + 3*3	ns	
	Slave		4	—	t_{SPIcyc}	
Continuous transmission delay	Master	t_{TD}	t_{SPcyc} + 2 × t_{SPIcyc}	8 × t_{SPcyc} + 2 × t_{SPIcyc}	ns	
	Slave		t_{SPcyc} + 5 × t_{SPIcyc}	—		
TI-SSP SS input setup time		t_{TISS}	3	—	ns	Figure 58.53 to Figure 58.55
TI-SSP SS input hold time		t_{TISH}	3	—	ns	
TI-SSP next access time		t_{TIND}	M*4	—	t_{SPIcyc}	
TI-SSP Master SS output delay		t_{TISSOD}	-3	3	ns	
TI-SSP Master OE delay 1		$t_{TIMOED1}$	—	2	ns	
TI-SSP Master OE delay 2		$t_{TIMOED2}$	—	2	ns	
TI-SSP Slave OE delay 1		$t_{TISOED1}$	—	12	ns	
TI-SSP Slave OE delay 2		$t_{TISOED2}$	—	8	ns	
Data output delay time	Master	t_{OD}	—	3	ns	Figure 58.49 to Figure 58.55
	Slave		—	12	ns	
Data output hold time	Master	t_{OH}	-3	—	ns	
	Slave		3	—		
MOSI, MISO rising/falling time	Output	t_{Dr}, t_{Df}	—	4	ns	
	Input		—	1	µs	
SSL rising/falling time	Output	t_{SSLr}, t_{SSLf}	—	4	ns	Figure 58.49, Figure 58.50
	Input		—	1	µs	

Table 58.32 SPI timing (2 of 2)

Conditions: V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 30 pF

Parameter	Symbol	Min.*1	Max.*1	Unit*1	Reference figure
Slave access time	t_{SA}	—	12	ns	Figure 58.51, Figure 58.52
Slave output release time	t_{REL}	—	12	ns	

Note 1. t_{SPIcyc}: PCLKSPIn cycle
Note 2. SPCKD set value + 1 (1 to 8)
Note 3. SSLND set value + 1 (1 to 8)
Note 4. SSLND set value + 2 (2 to 9)
Note 5. In case of PCLKSPIn = 125 MHz, Min. is 4.

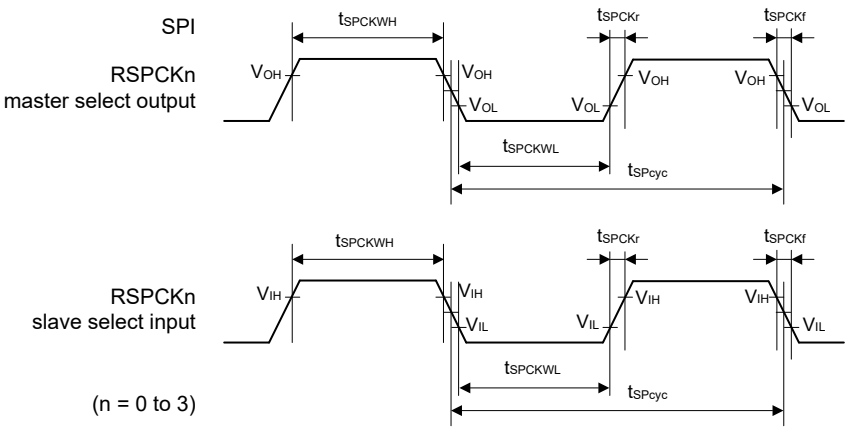


Figure 58.48 SPI clock timing

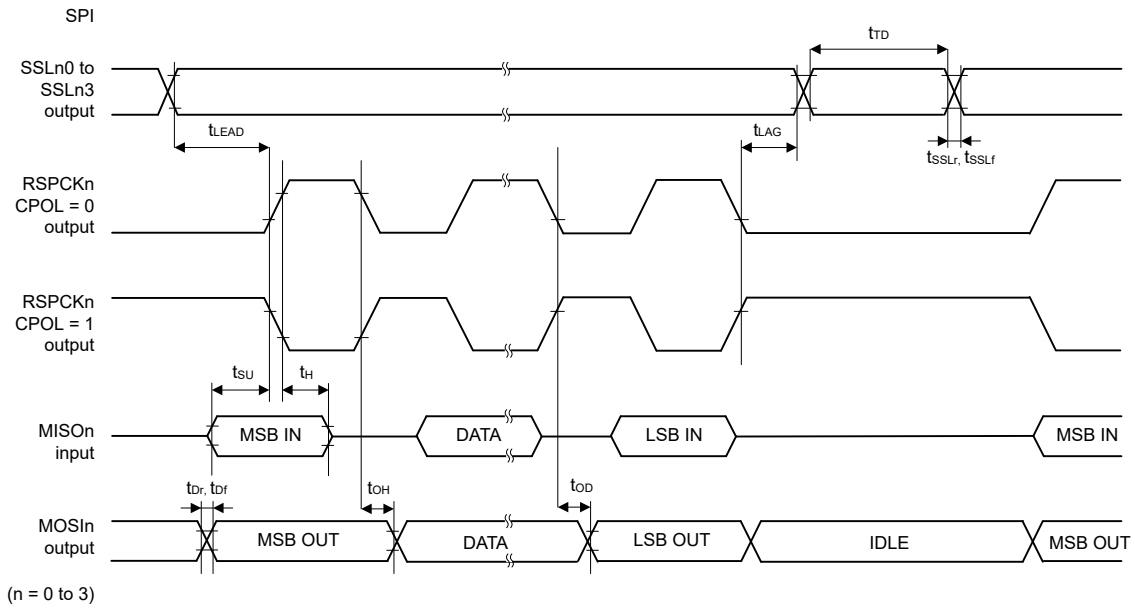


Figure 58.49 SPI timing (Master, Motorola SPI, CPHA = 0)

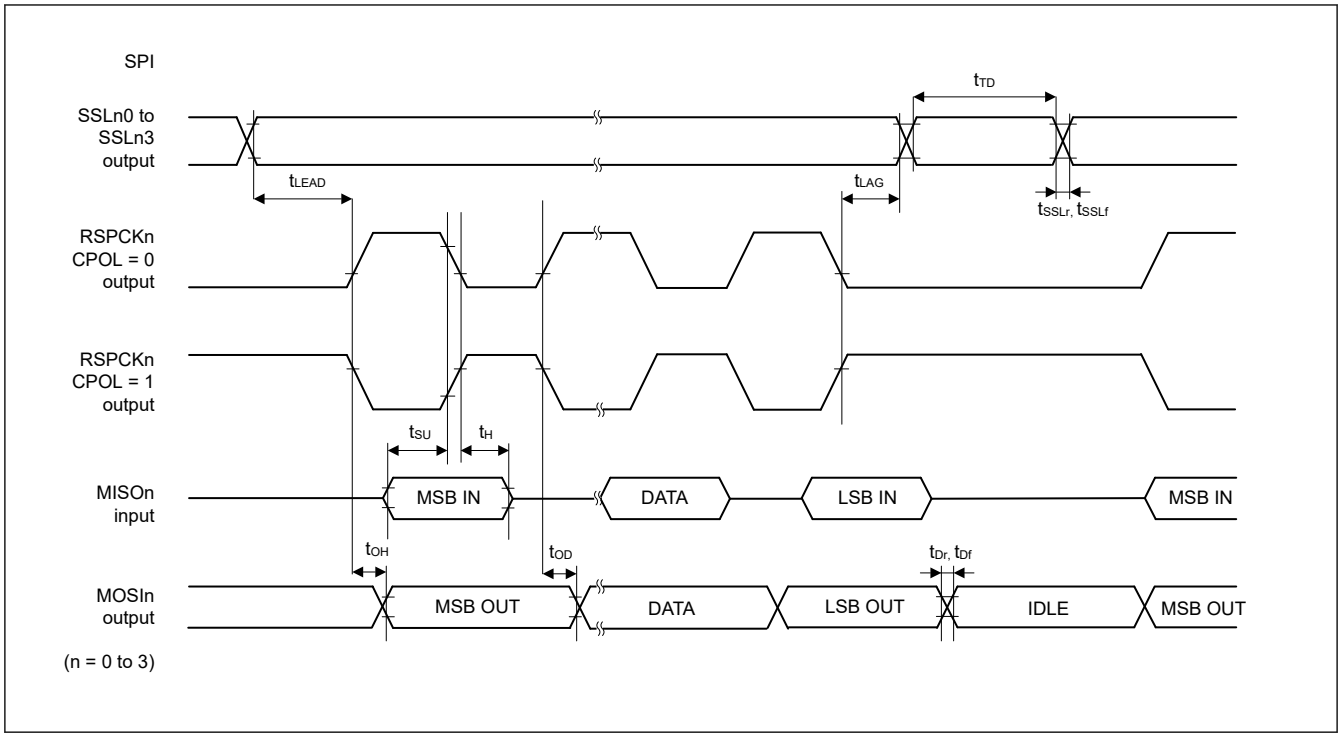


Figure 58.50 SPI timing (Master, Motorola SPI, CPHA = 1)

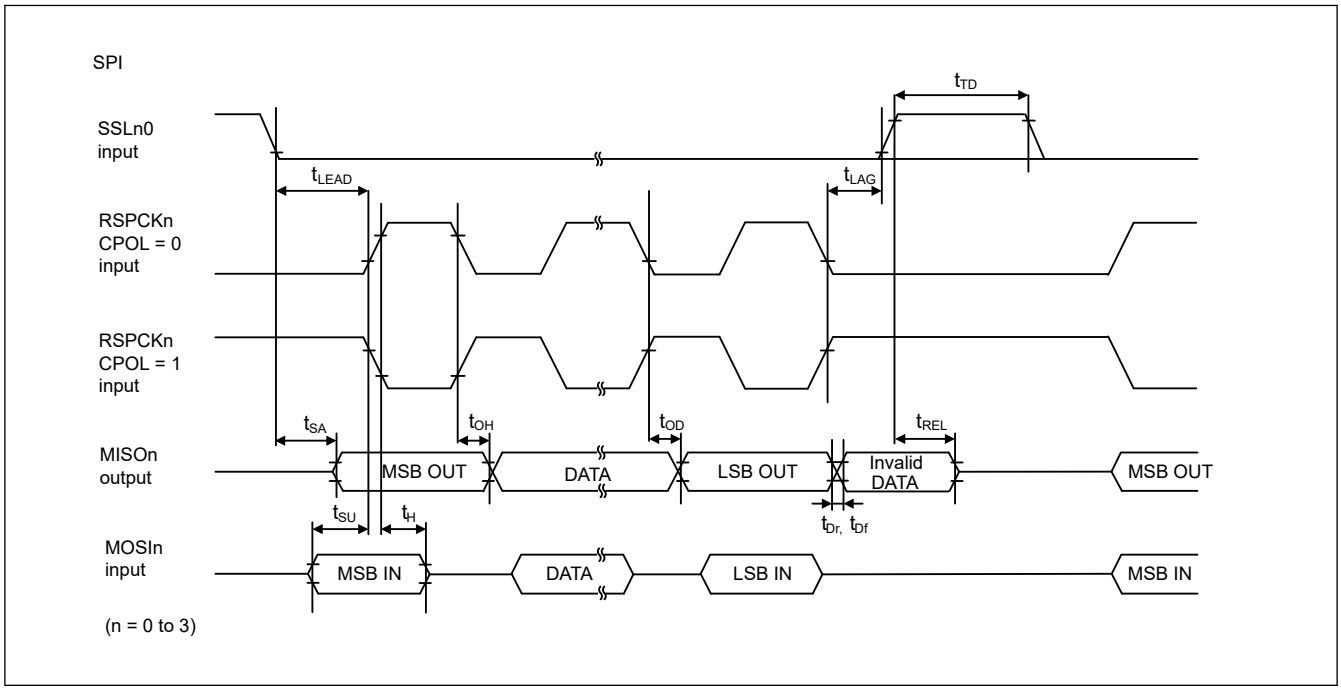


Figure 58.51 SPI timing (Slave, Motorola SPI, CPHA = 0)

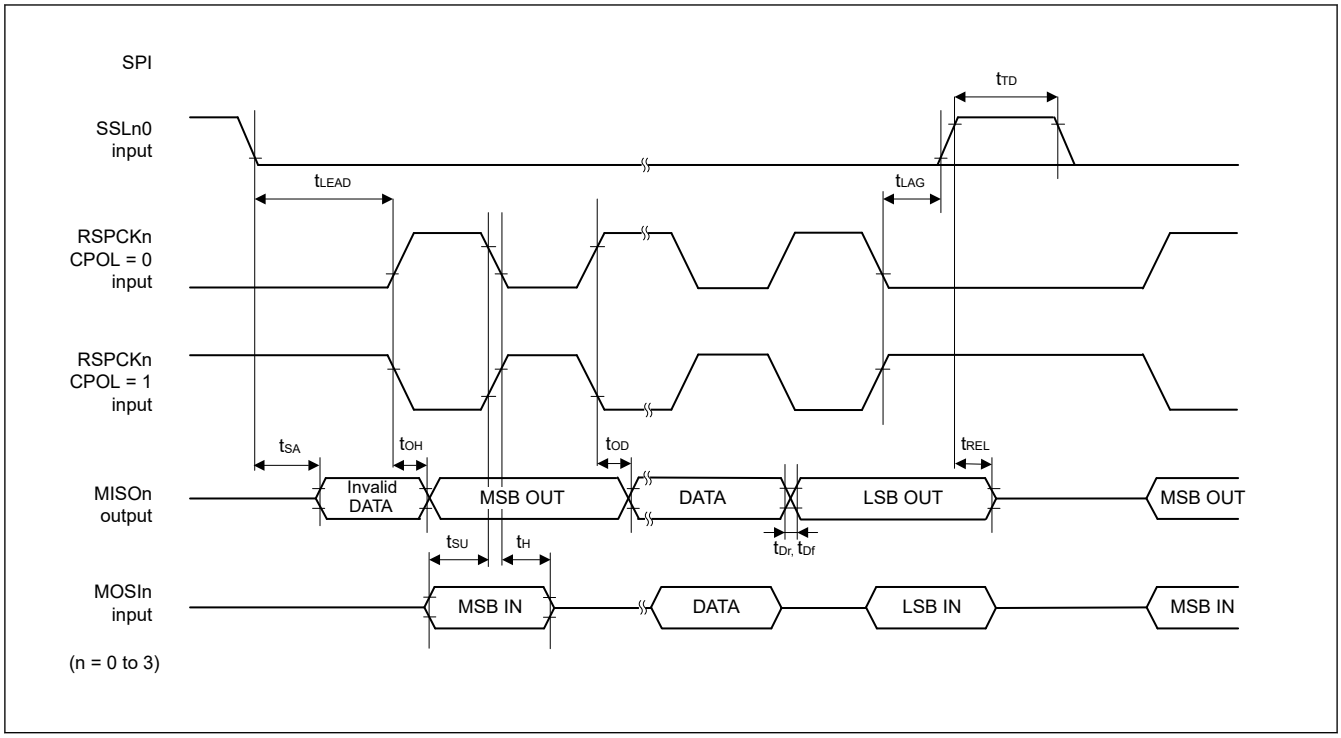


Figure 58.52 SPI timing (Slave, Motorola SPI, CPHA = 1)

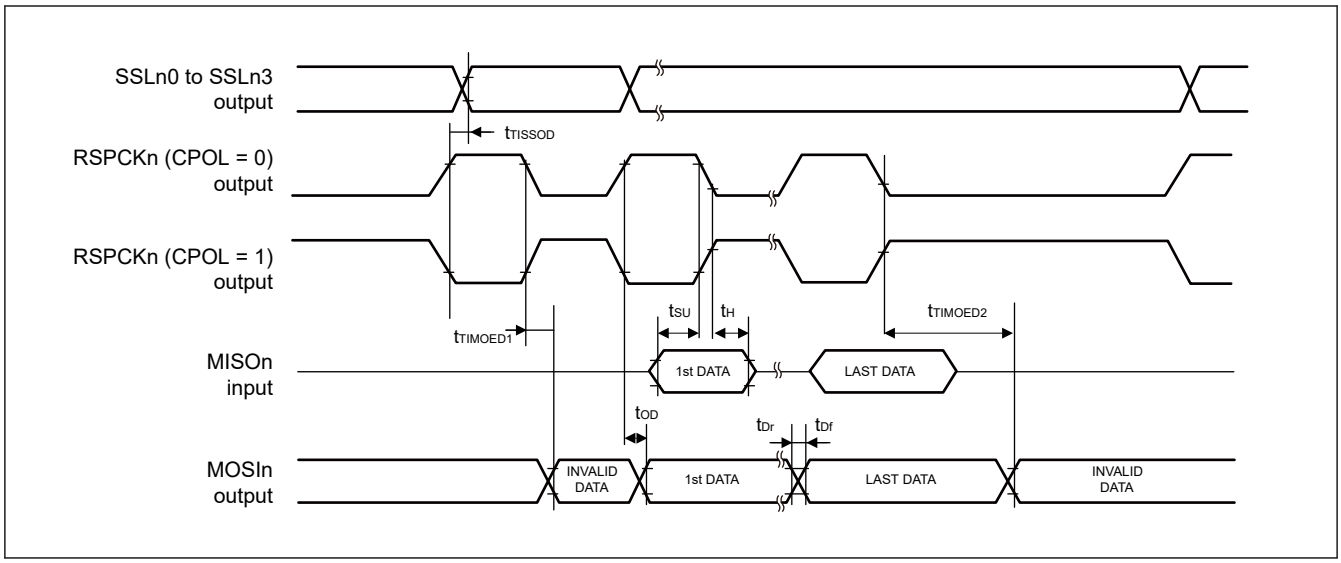


Figure 58.53 SPI timing (Master, TI SSP)

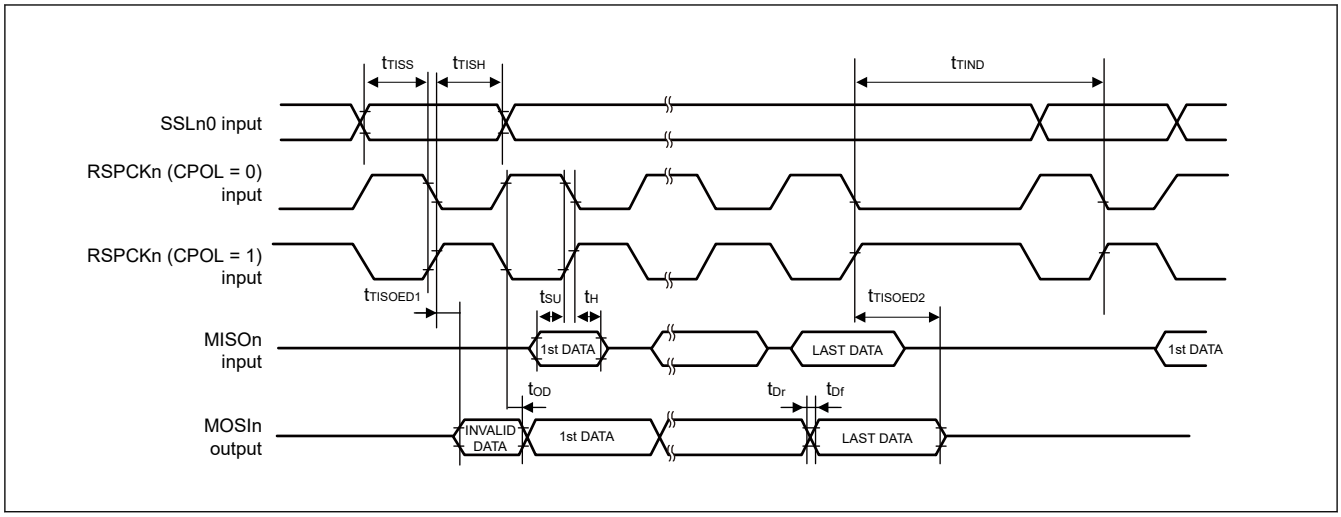


Figure 58.54 SPI timing (Slave, TI-SSP, with delay in burst transfer)

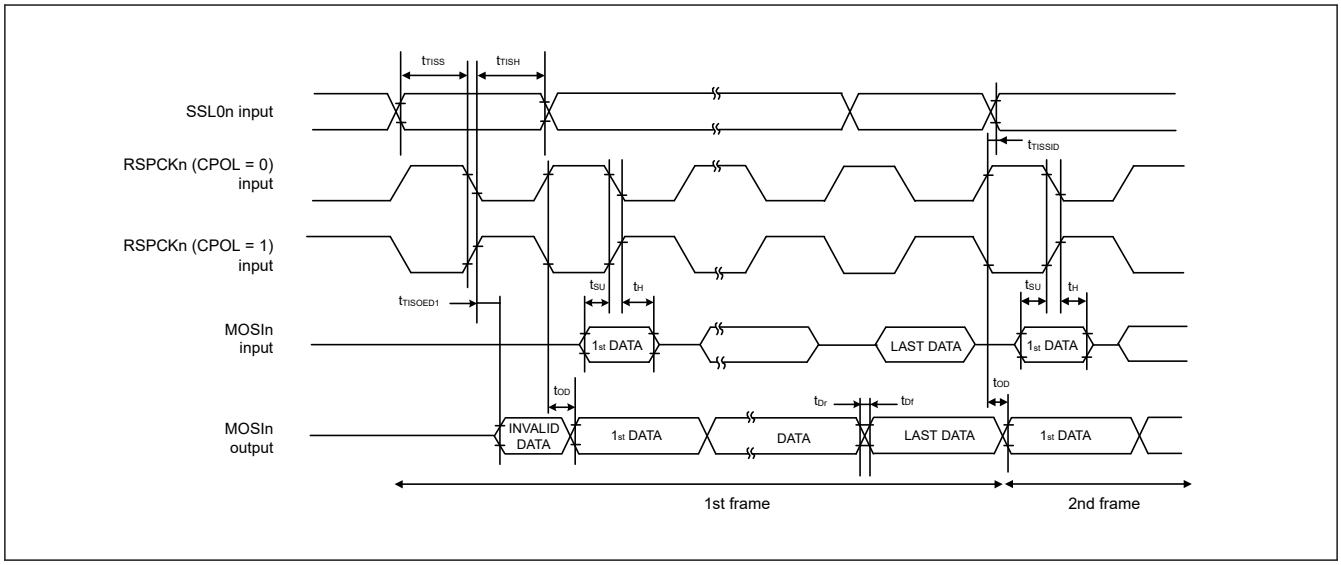


Figure 58.55 SPI timing (Slave, TI-SSP, without delay in burst transfer)

58.5.5.12 xSPI Timing

Table 58.33 xSPI timing

Conditions:

V_{OH} = VDD1833 × 0.5, V_{OL} = VDD1833 × 0.5, C = 15 pF (VDD1833 = 1.8 V or 3.3 V)

Parameter		Symbol	1.8 V		3.3 V		Unit	Reference figure
			Min.	Max.	Min.	Max.		
Cycle time	SDR	t_{PERIOD}	7.5	—	10.0	—	ns	Figure 58.56
	DDR		7.5	—	10.0	—	ns	
Clock output slew rate		t_{SRck}	0.75/0.56*2	—	0.56	—	V/ns	
Clock Duty cycle distortion		t_{CKDCD}	0.0	t_{PERIOD} × 0.05	0.0	t_{PERIOD} × 0.05	ns	
Clock Minimum Pulse width		t_{CKMPW}	t_{PERIOD} × 0.45	—	t_{PERIOD} × 0.45	—	ns	
Differential clock crossing voltage		$V_{OX(AC)}$	0.4 × VCC18	0.6 × VCC18	—	—	V	
DS Duty cycle distortion		t_{DSDCD}	0.0	t_{PERIOD} × 0.04	0.0	t_{PERIOD} × 0.04	ns	
DS Minimum Pulse width		t_{DSMPW}	t_{PERIOD} × 0.41	—	t_{PERIOD} × 0.41	—	ns	
Data input/output slew rate		t_{SR}	0.75/0.56*2	—	0.56	—	V/ns	
Data input setup time (to CK)	SDR	t_{SU}	2.0	—	2.0	—	ns	Figure 58.57
Data input hold time (to CK)		t_{H}	1.0	—	1.0	—	ns	
Data output delay time		t_{OD}	—	1.0*3	—	2.0*3	ns	
Data output hold time		t_{OH}	-1.0	—	-2.0	—	ns	
Data output buffer off time		t_{BOFF}	-1.0	—	-2.0	—	ns	
Data input setup time (to DS)	DDR*1 *3	t_{SU}	-0.4/-0.6*2	—	-0.3	—	ns	Figure 58.58, Figure 58.59
Data input hold time (to DS)		t_{H}	t_{PERIOD} × 0.41 - 0.4/0.6*2	—	t_{PERIOD} × 0.41 - 0.3	—	ns	
Data output setup time (to CK)		t_{SUO}	0.8/1.0*2	—	1.0	—	ns	
Data output hold time (to CK)		t_{HO}	0.8/1.0*2	—	1.0	—	ns	
CS Low to Clock High		t_{CSLCKH}	6.0/8.0*2 *4	—	8.0*4	—	ns	Figure 58.57 to Figure 58.59
Clock Low to CS High		t_{CKLCSH}	6.0/8.0*2	—	8.0	—	ns	
CS High time		t_{CSTD}	1	16	1	16	t_{PERIOD}	
DS Low to CS High		t_{DSLCSH}	6.0/8.0*2	—	10.6	—	ns	Figure 58.60
CS High to DS Tri-State		t_{CSHDST}	0.0	t_{PERIOD}	0.0	t_{PERIOD}	ns	
CS Low to DS Low		t_{CSLDSL}	0.0	—	0.0	—	ns	
DS Tri-State to CS Low		t_{DSTCSL}	0.0	—	0.0	—	ns	

Note 1. The DS shift setting (WRAPCFG.DSSFTCSx[4:0]) is 01001b for 133 MHz and 01100b for 100 MHz.
Note 2. Specification at 133 MHz / Specification at 100 MHz
Note 3. These are values when the OEN assertion is extended in the Output Enable Asserting extension bit (COMCFG.OEASTEX = 1).
Note 4. These are the values when the CS assertion is extended in the CS asserting extension bit (LIOCFGCSn.CSASTEX = 1).

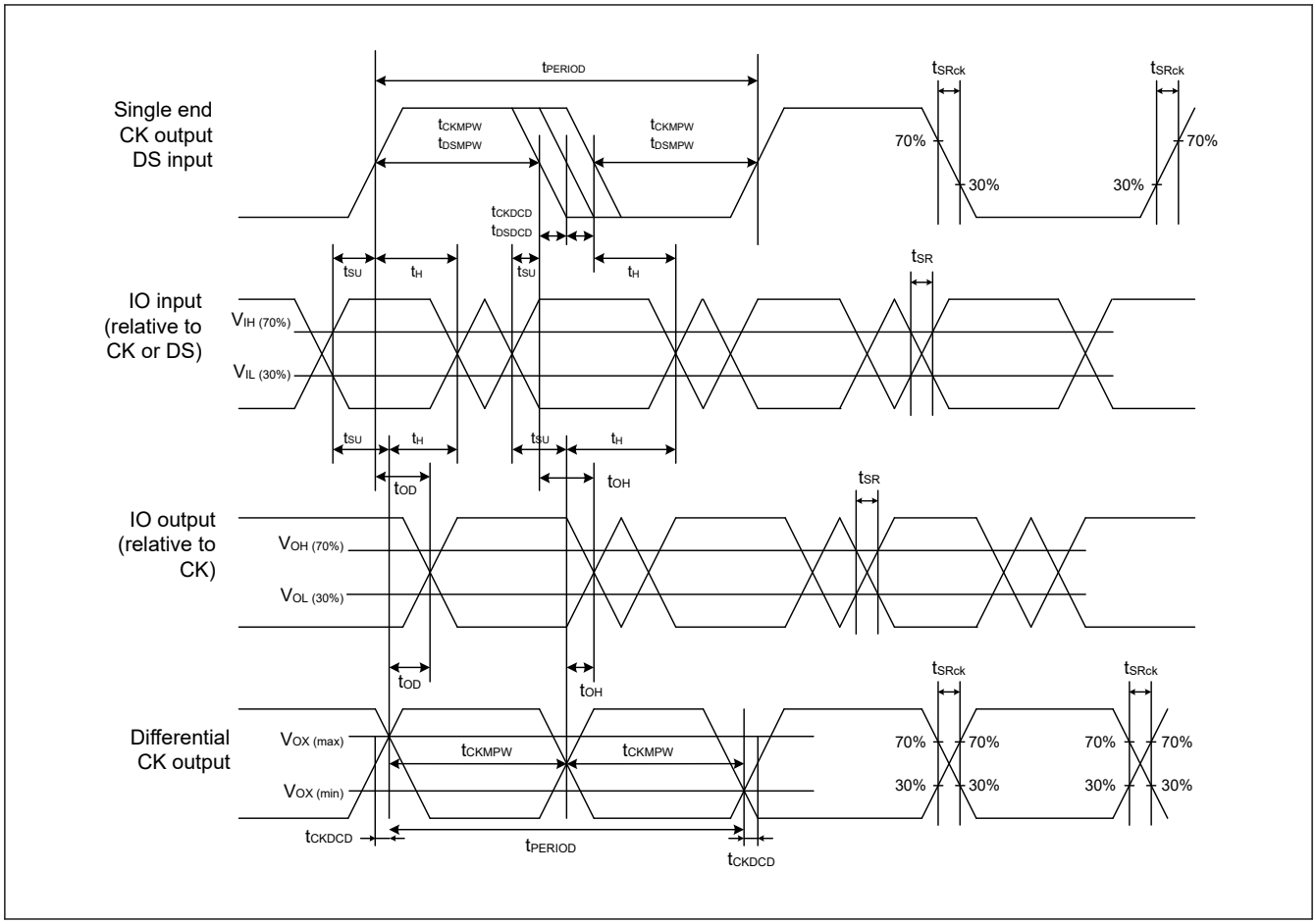


Figure 58.56 xSPI clock / DS timing

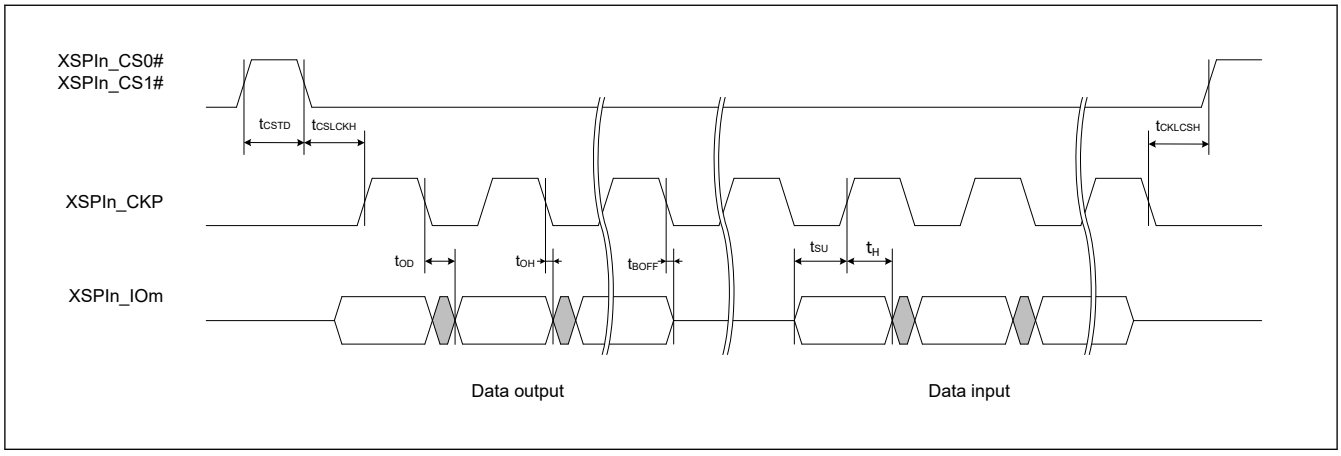


Figure 58.57 SDR transmit/receive timing (1S-1S-1S, 1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S)

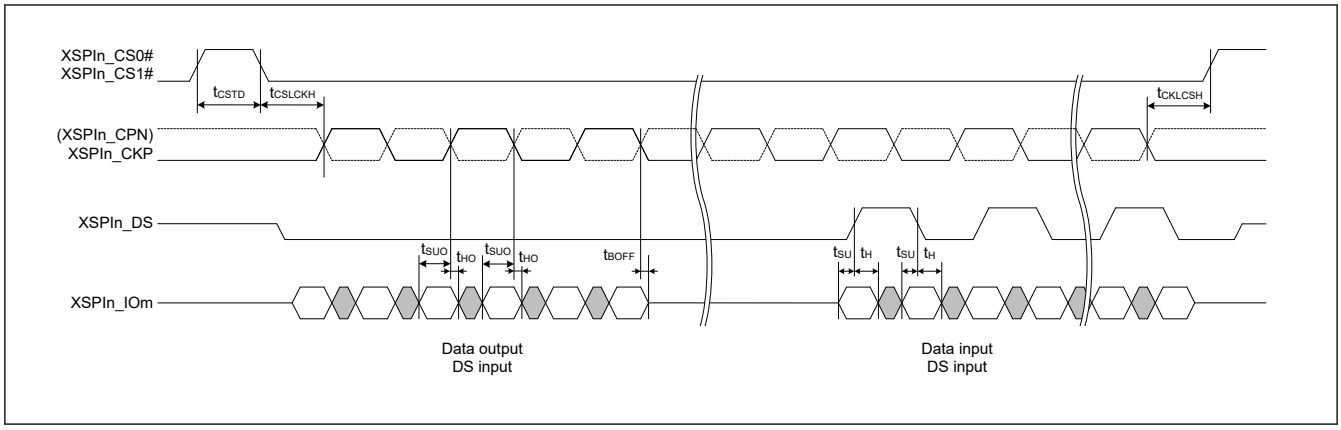


Figure 58.58 DDR transmit/receive timing (4S-4D-4D, 8D-8D-8D)

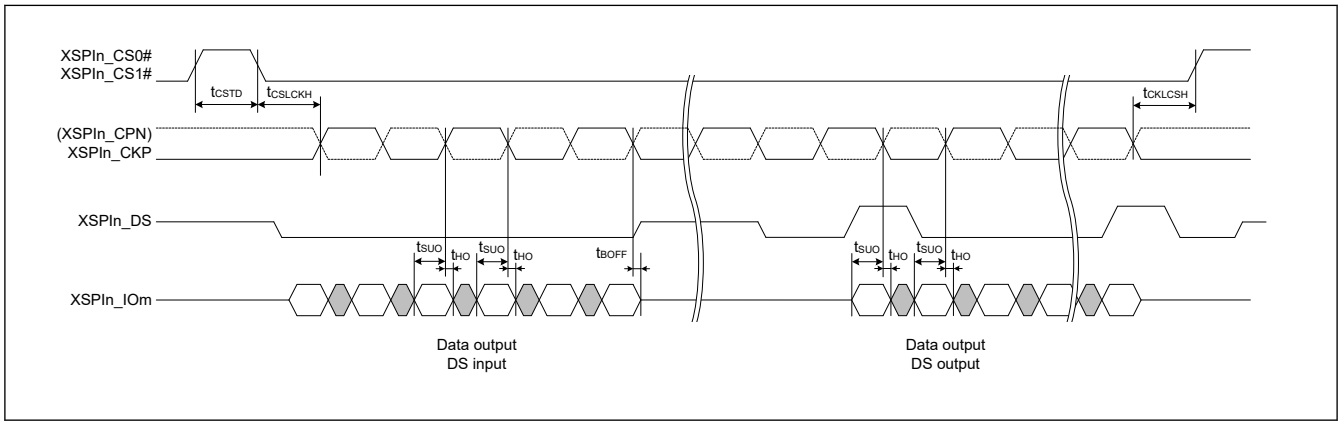


Figure 58.59 DDR transmit/receive timing (HyperRAM write)

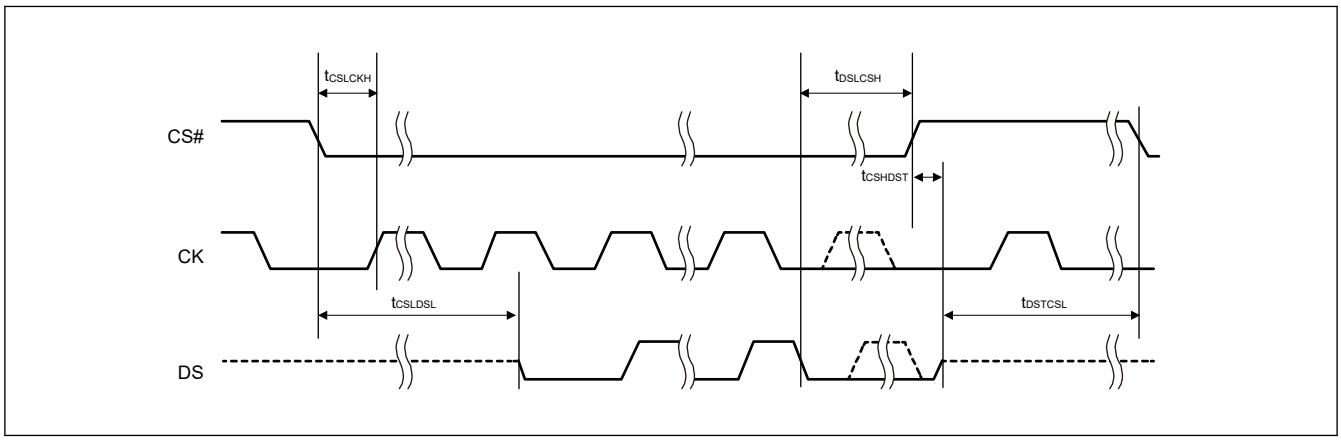


Figure 58.60 DS to CS signal timing

58.5.5.13 Delta-Sigma Interface Timing

Conditions: V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 30 pF

Table 58.34 ΔΣ interface timing

Parameter			Symbol	Min.	Max.	Unit	Reference figure
DSMIF	Clock cycle	Master	t_{DScyc}	40	200	ns	Figure 58.61
		Slave		40	200		
	Clock high level	Master	t_{DSCKWH}	16	—	ns	
		Slave		16	—		
	Clock low level	Master	t_{DSCKWL}	16	—	ns	
		Slave		16	—		
	Setup time	Master	t_{SU}	15	—	ns	Figure 58.62, Figure 58.63
		Slave		5	—		
	Hold time	Master	t_H	0	—	ns	
		Slave		5	—		

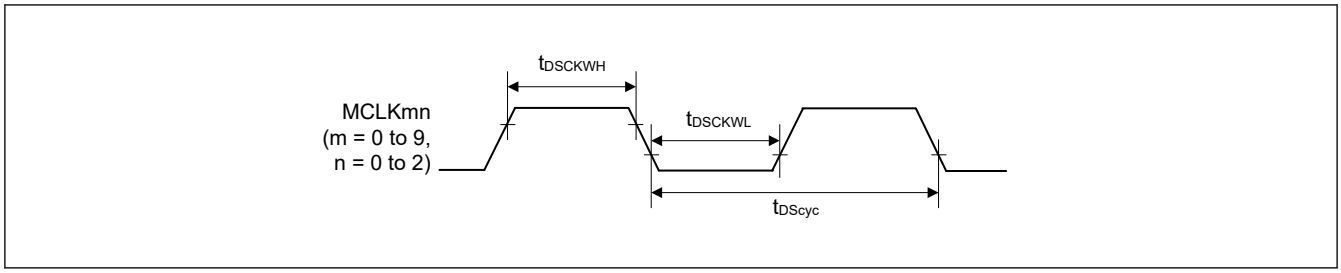


Figure 58.61 Clock input/output timing

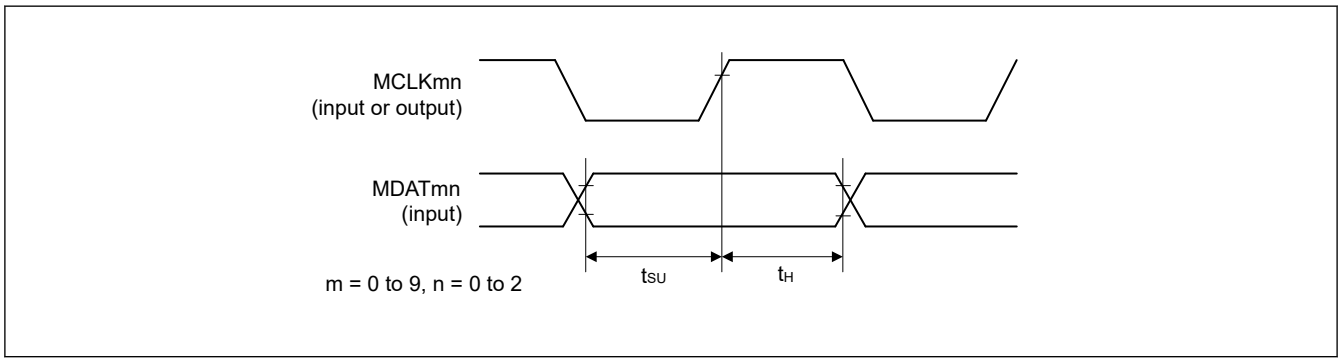


Figure 58.62 Reception timing (MCLKn rising synchronous)

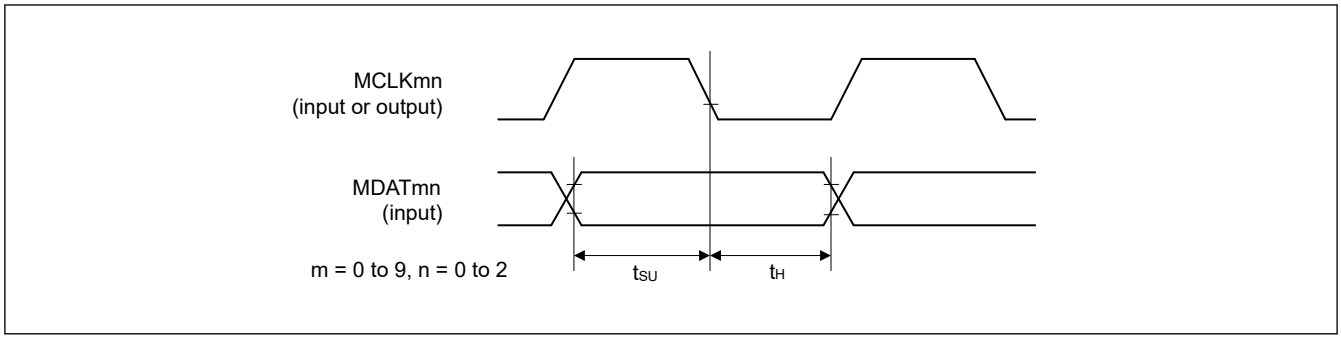


Figure 58.63 Reception timing (MCLKn falling synchronous)

58.5.5.14 Ethernet Interface Timing

Conditions:

V_{OH} = VDD1833 × 0.5, V_{OL} = VDD1833 × 0.5, C = 15 pF (RGMII, VDD1833 = 1.8 V)

V_{OH} = VDD1833 × 0.5, V_{OL} = VDD1833 × 0.5, C = 25 pF (RMII, VDD1833 = 3.3 V)

V_{OH} = VDD1833 × 0.5, V_{OL} = VDD1833 × 0.5, C = 30 pF (MII, VDD1833 = 3.3 V)

Table 58.35 Ethernet interface timing

Parameter			Symbol	Min.	Max.	Unit	Reference figure
Ethernet (RGMII)	ETHn_TXCLK, ETHn_RXCLK cycle time duration	1 Gbps	$t_{RGMIIck}$	7.2	8.8	ns	Figure 58.64
		100 Mbps		36	44		
		10 Mbps		360	440		
	ETHn_TXCLK, ETHn_RXCLK frequency	1 Gbps	—	125 − 50 ppm	125 + 50 ppm	MHz	
		100 Mbps		25 − 50 ppm	25 + 50 ppm		
		10 Mbps		2.5 − 50 ppm	2.5 + 50 ppm		
	ETHn_TXCLK, ETHn_RXCLK duty cycle	1 Gbps	—	45	55	%	
		100 Mbps 10 Mbps		40	60		
	ETHn_TXCLK, ETHn_TXD0 to ETHn_TXD3, ETHn_TXEN (TX_CTL), ETHn_RXCLK, ETHn_RXD0 to ETHn_RXD3, ETHn_RXDV (RX_CTL) rise/fall time		t_{RGMIIr}, t_{RGMIIf}	—	0.75	ns	
	ETHn_TXD0 to ETHn_TXD3, ETHn_TXEN (TX_CTL) to ETHn_TXCLK output skew		$t_{RGMIIos}$	−0.5	0.5	ns	
	ETHn_RXD0 to ETHn_RXD3, ETHn_RXDV (RX_CTL) setup time		t_{RGMIIs}	1	—	ns	
	ETHn_RXD0 to ETHn_RXD3, ETHn_RXDV (RX_CTL) hold time		t_{RGMIIh}	1	—	ns	
Ethernet (RMII)	ETHn_RXCLK cycle time		t_{RMIIck}	20	—	ns	Figure 58.65
	ETHn_RXCLK frequency Typ. 50 MHz		—	50 − 50 ppm	50 + 50 ppm	MHz	
	ETHn_RXCLK duty		—	35	65	%	
	ETHn_RXCLK rise/fall time		$t_{RMIIckr}$, $t_{RMIIckf}$	0.5	3.5	ns	
	ETHn_TXD0, ETHn_TXD1, ETHn_TXEN output delay time		t_{RMIId}	2.5	12	ns	
	ETHn_RXD0, ETHn_RXD1, ETHn_RXER, ETHn_RXDV (CRS_DV) setup time		t_{RMIIs}	4	—	ns	
	ETHn_RXD0, ETHn_RXD1, ETHn_RXER, ETHn_RXDV (CRS_DV) hold time		t_{RMIIh}	2	—	ns	
	ETHn_TXD0, ETHn_TXD1, ETHn_TXEN, ETHn_RXD0, ETHn_RXD1, ETHn_RXER, ETHn_RXDV (CRS_DV) rise/fall time		t_{RMIIr}, t_{RMIIf}	0.5	4	ns	
Ethernet (MII)	ETHn_TXCLK, ETHn_RXCLK cycle time	100 Mbps	t_{MIIck}	40	—	ns	Figure 58.66
		10 Mbps		400	—		
	ETHn_TXCLK, ETHn_RXCLK frequency	100 Mbps	—	25 − 50 ppm	25 + 50 ppm	MHz	
		10 Mbps		2.5 − 50 ppm	2.5 + 50 ppm		
	ETHn_TXD0 to ETHn_TXD3, ETHn_TXEN, ETHn_TXER output delay time		t_{MIId}	1	20	ns	
	ETHn_RXD0 to ETHn_RXD3, ETHn_RXDV, ETHn_RXER setup time		t_{MIIs}	10	—	ns	
	ETHn_RXD0 to ETHn_RXD3, ETHn_RXDV, ETHn_RXER hold time		t_{MIIh}	10	—	ns	

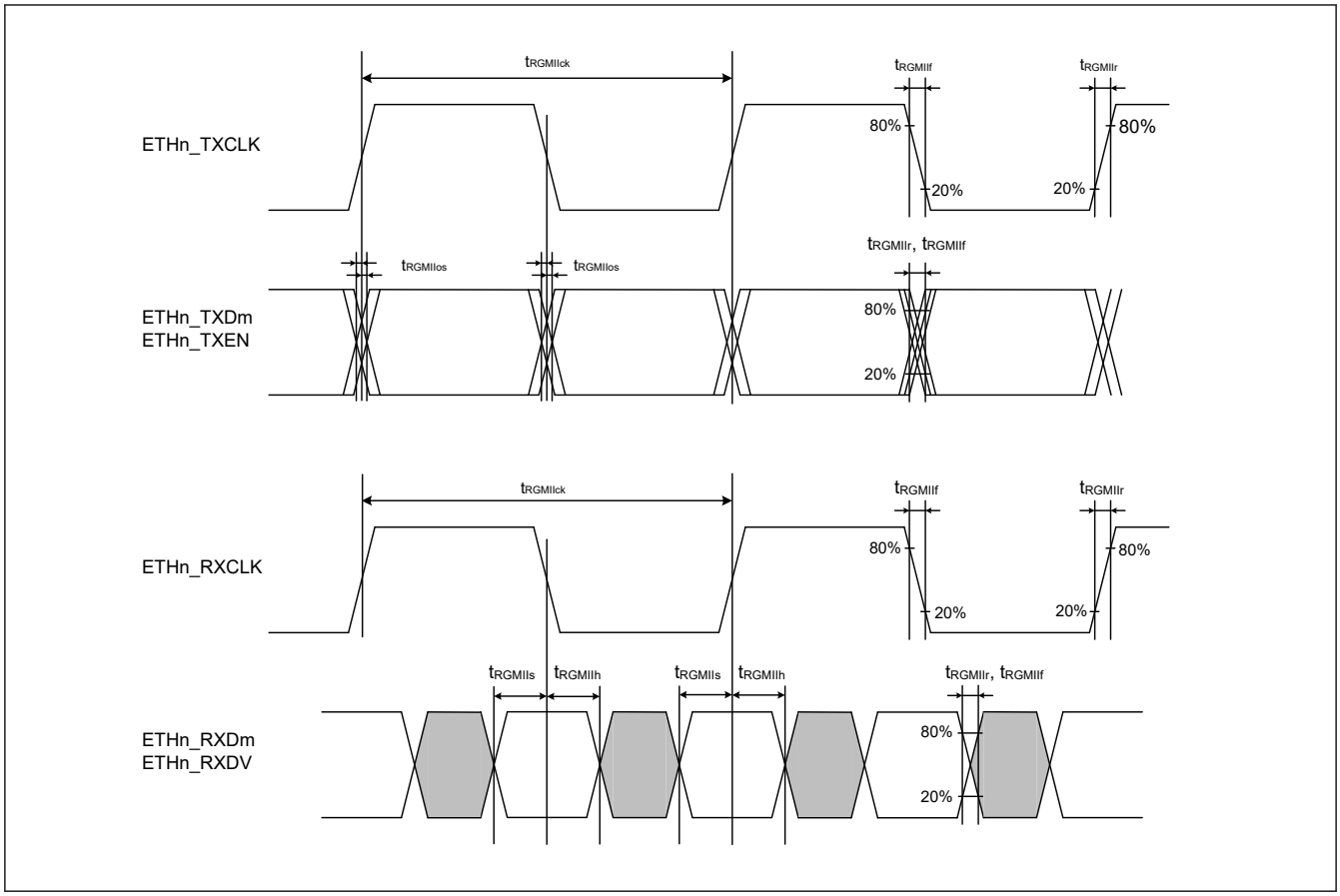


Figure 58.64 RGMII transmission and reception timing (n = 0 to 3, m = 0 to 3)

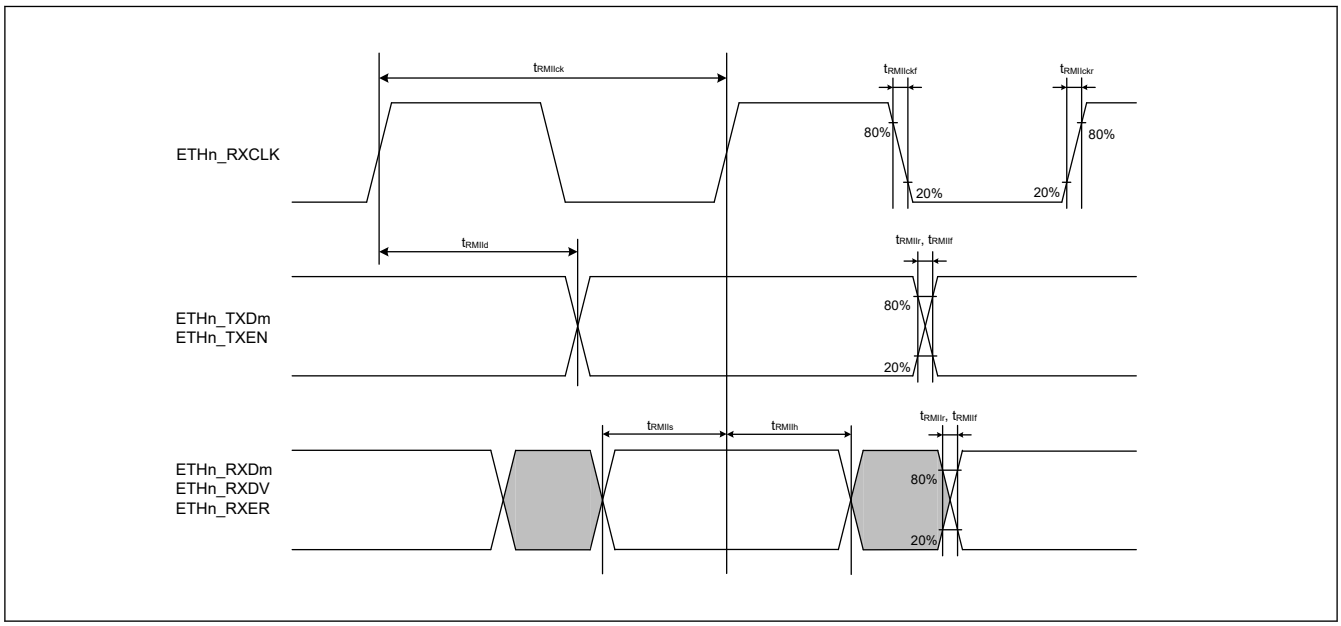


Figure 58.65 RMII transmission and reception timing (n = 0 to 3, m = 0 to 1)

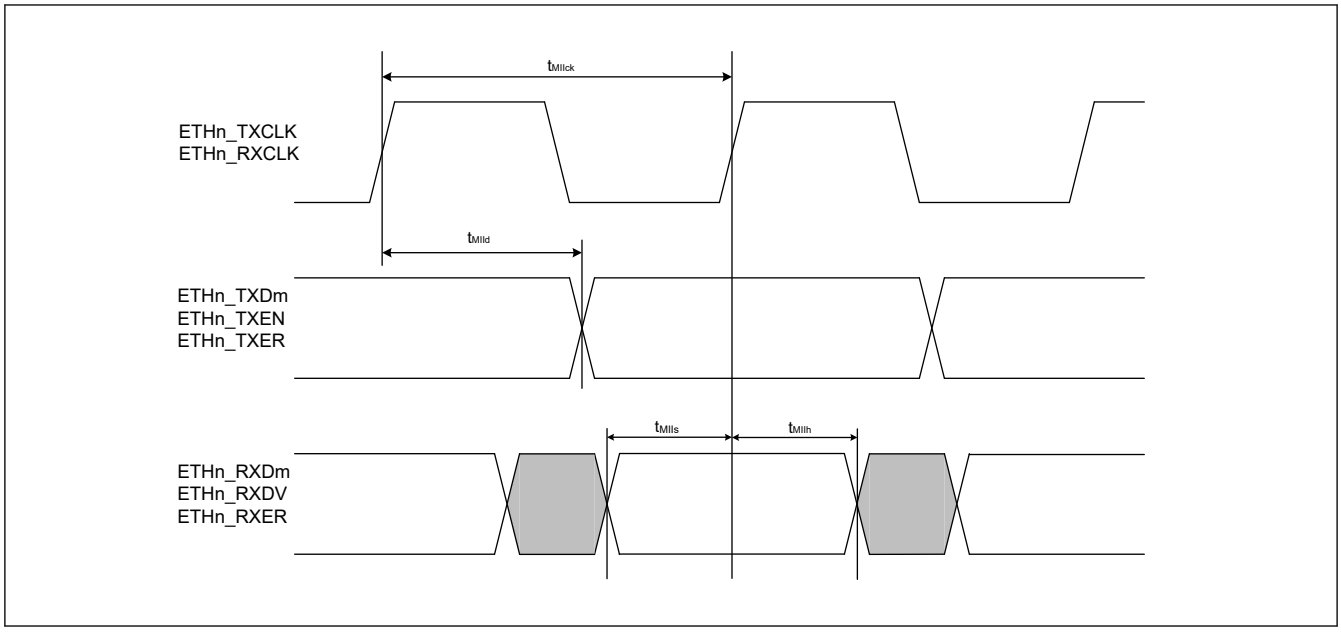


Figure 58.66 MII transmission and reception timing (n = 0 to 3, m = 0 to 3)

58.5.5.15 Serial Management Interface Timing

Conditions:

V_{OH} = VDD1833 × 0.5, V_{OL} = VDD1833 × 0.5, C = 30 pF (VDD1833 = 1.8 V or 3.3 V)

Table 58.36 Serial management interface timing

Parameter			Symbol	Min.	Max.	Unit	Reference figure
MDIO	MDC output cycle time	GMACn_MDC, ETHSW_MDC	T_{MDCck}	80	—	ns	Figure 58.67
		ESC_MDC		400	—	ns	
	MDIO output delay time (for MDC fall)*1		T_{MDIOd}	—	20	ns	
	MDIO input setup time (for MDC rise)	GMACn_MDC, ETHSW_MDC	T_{MDIOs}	18	—	ns	
		ESC_MDC		70	—	ns	
	MDIO input hold time (for MDC rise)		T_{MDIOh}	0	—	ns	

Note 1. The output timing from ETHSW is based on the rising edge of MDC, and the output delay can be set in the register.

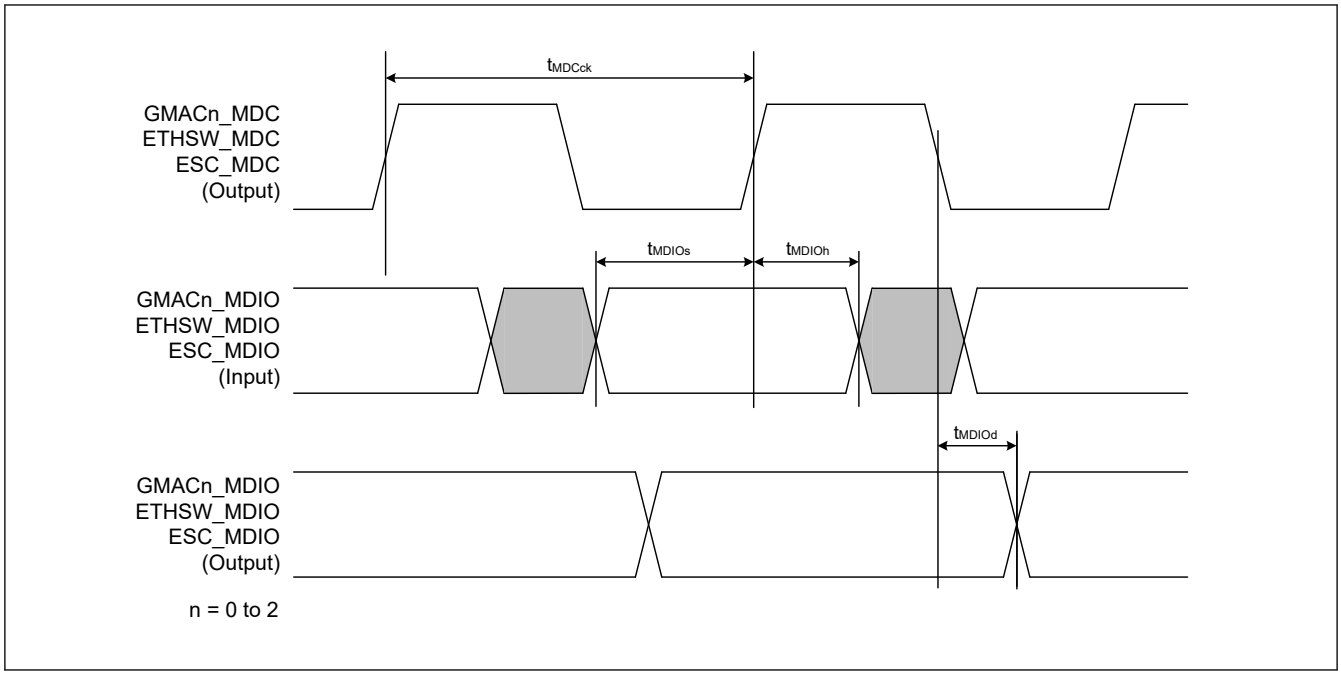


Figure 58.67 Serial management interface timing

58.5.5.16 SHOSTIF Timing

Conditions:

V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 30 pF

Table 58.37 SHOSTIF timing

Parameter		Symbol	Min.	Max.	Unit	Reference figure
SHOSTIF	Clock cycle time	t_{SHck}	25	—	ns	Figure 58.68
	Clock high time	t_{CH}	0.45	0.55	t_{SHck}	
	Clock low time	t_{CL}	0.45	0.55	t_{SHck}	
	Clock rise slew rate	t_{CRT}	0.1	—	V/ns	
	Clock fall slew rate	t_{CFT}	0.1	—	V/ns	
	CS# high time	t_{CS}	2	—	t_{SHck}	Figure 58.69, Figure 58.70
	CS# active setup time	t_{CSS}	15	—	ns	
	CS# active hold time	t_{CSH}	15	—	ns	
	Data input setup time	t_{SU}	3	—	ns	
	Data input hold time	t_{HD}	10.5	—	ns	
	Clock low to output valid	t_{V}	—	15.5	ns	
	Data output hold time	t_{HO}	6	—	ns	
	Data output disable time	t_{DIS}	—	18	ns	

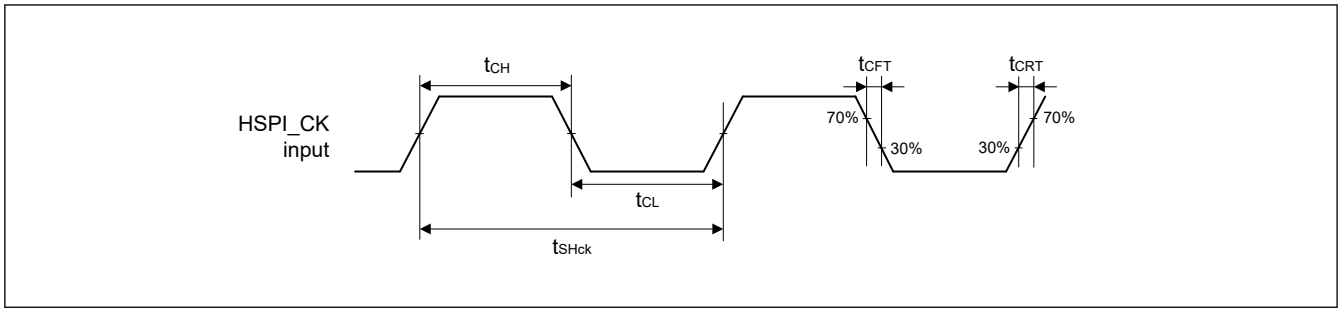


Figure 58.68 SHOSTIF clock timing

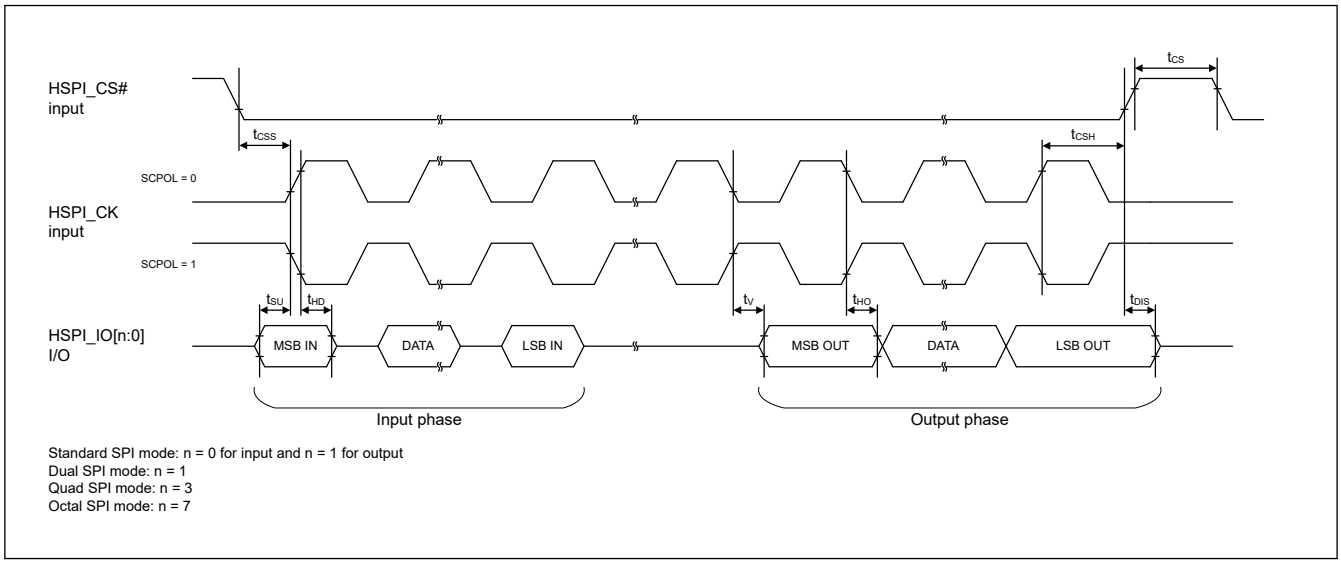


Figure 58.69 SHOSTIF timing (SCPH = 0)

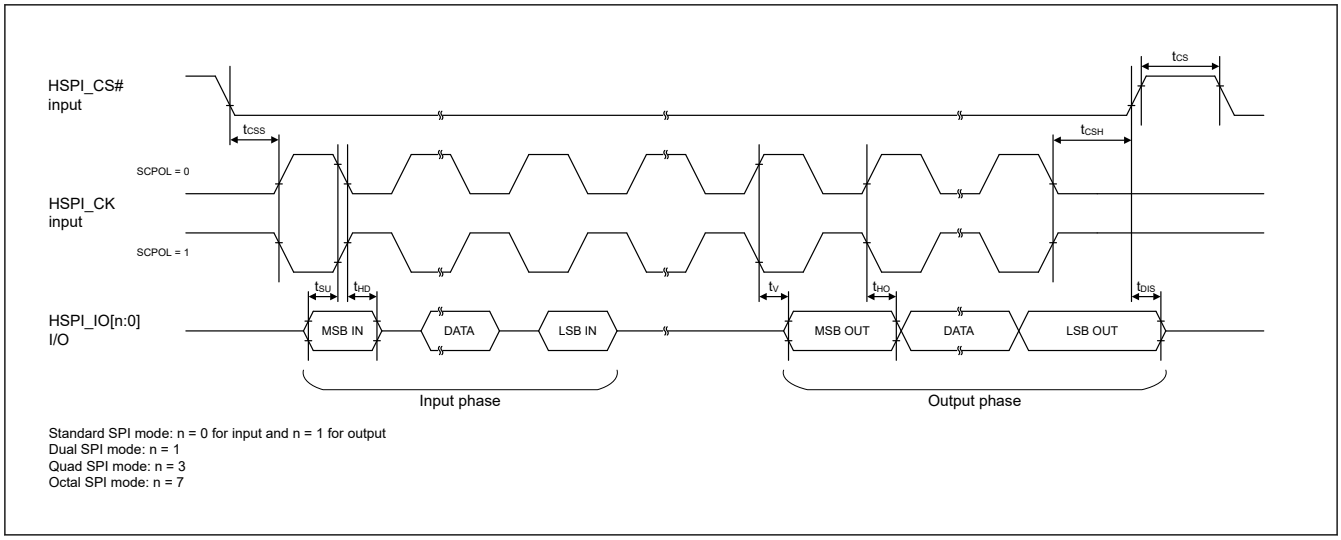


Figure 58.70 SHOSTIF timing (SCPH = 1)

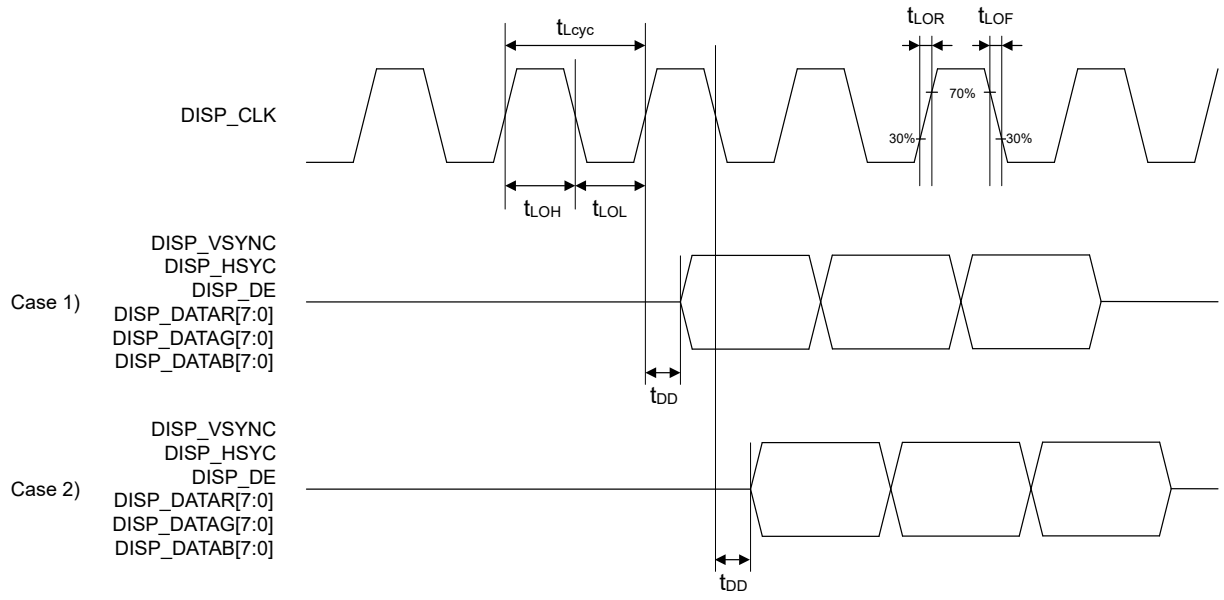
58.5.5.17 LCDC Timing

Conditions:

V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5, C = 30 pF

Table 58.38 LCDC timing

Parameter		Symbol	Min.	Typ.	Max.	Unit	Reference figure
LCDC	DCLK output period	t_{Lcyc}	10	—	200	ns	Figure 58.71
	DCLK output low pulse width	t_{LOL}	$t_{Lcyc}/2 - 1$	—	$t_{Lcyc}/2 + 1$	ns	
	DCLK output high pulse width	t_{LOH}	$t_{Lcyc}/2 - 1$	—	$t_{Lcyc}/2 + 1$	ns	
	DCLK output rise time	t_{LOR}	—	—	3	ns	
	DCLK output fall time	t_{LOF}	—	—	3	ns	
	Data output delay	t_{DD}	−1.5	—	1.5	ns	



Note:
Case 1) DISP_CLK forward rotation output mode when DPI_CLKMD bit in DU_DITR0 register is 0.
Case 2) DISP_CLK inverted output mode when DPI_CLKMD bit in DU_DITR0 register is 1.

Figure 58.71 LCDC timing

58.5.5.18 SDHI Timing

Table 58.39 1.8 V SDHI timing (1 of 2)

Conditions: V_{OH} = VDD1833 × 0.5, V_{OL} = VDD1833 × 0.5 (VDD1833 = 1.8 V)

Parameter		Symbol	Conditions	Min.	Max.	Unit	Reference figure
(SD) SDR104 (eMMC) HS200	SD_CLK clock cycle	T_{SDCYC}	C = 15 pF Drive Strength CLK: Ultra-high Others: High	5	10	ns	Figure 58.72
	SD_CLK clock high level width	T_{SDWH}		1.5	—	ns	
	SD_CLK clock low level width	T_{SDWL}		1.5	—	ns	
	SD_CLK clock rise time	T_{SDLH}		—	1	ns	
	SD_CLK clock fall time	T_{SDHL}		—	1	ns	
	SD_CMD, SD_DATA output delay	T_{SDODLY}		−1.7	0.9	ns	
	SD_CMD, SD_DATA input set up time	T_{SDIS}		—	—	ns	
	SD_CMD, SD_DATA input hold time	T_{SDIH}		—	—	ns	
	SD_CMD, SD_DATA input data width	T_{SDIDW}		2.88	—	ns	

Table 58.39 1.8 V SDHI timing (2 of 2)

Conditions: V_{OH} = VDD1833 × 0.5, V_{OL} = VDD1833 × 0.5 (VDD1833 = 1.8 V)

Parameter		Symbol	Conditions	Min.	Max.	Unit	Reference figure
(SD) SDR50, SDR25, SDR12 (eMMC) High Speed SDR, Backwards Compatibility	SD_CLK clock cycle	T_{SDCYC}	C = 20 pF Drive Strength CLK: Ultra-high Others: High	10	—	ns	Figure 58.72
	SD_CLK clock high level width	T_{SDWH}		3	—	ns	
	SD_CLK clock low level width	T_{SDWL}		3	—	ns	
	SD_CLK clock rise time	T_{SDLH}		—	2	ns	
	SD_CLK clock fall time	T_{SDHL}		—	2	ns	
	SD_CMD, SD_DATA output delay	T_{SDODLY}		−4.2	1.6	ns	
	SD_CMD, SD_DATA input set up time	T_{SDIS}		1.1	—	ns	
	SD_CMD, SD_DATA input hold time	T_{SDIH}		1.8	—	ns	
	SD_CMD, SD_DATA input data width	T_{SDIDW}		—	—	ns	
(SD) DDR50 (eMMC) High Speed DDR	SD_CLK clock cycle	T_{SDCYC}	C = 25 pF Drive Strength CLK: High Others: High	20	—	ns	Figure 58.73
	SD_CLK clock high level width	T_{SDWH}		9	11	ns	
	SD_CLK clock low level width	T_{SDWL}		9	11	ns	
	SD_CLK clock rise time	T_{SDLH}		—	3	ns	
	SD_CLK clock fall time	T_{SDHL}		—	3	ns	
	SD_CMD output delay (SDR)	T_{SDODLY}		−6	3	ns	
	SD_CMD input set up time (SDR)	T_{SDIS}		4.8	—	ns	
	SD_CMD input hold time (SDR)	T_{SDIH}		2.5	—	ns	
	SD_DATA output delay (DDR)	T_{SDODLY_DDR}		2.5	6	ns	
	SD_DATA input set up time (DDR)	T_{SDIS_DDR}		1.5	—	ns	
	SD_DATA input hold time (DDR)	T_{SDIH_DDR}		1.5	—	ns	

Table 58.40 3.3 V SDHI timing (1 of 2)

Conditions: V_{OH} = VDD1833 × 0.5, V_{OL} = VDD1833 × 0.5 (VDD1833 = 3.3 V)

Parameter		Symbol	Conditions	Min.	Max.	Unit	Reference figure
(SD) High Speed (eMMC) High Speed SDR,	SD_CLK clock cycle	T_{SDCYC}	C = 40 pF Drive Strength CLK: High Others: Middle	20	—	ns	Figure 58.72
	SD_CLK clock high level width	T_{SDWH}		7	—	ns	
	SD_CLK clock low level width	T_{SDWL}		7	—	ns	
	SD_CLK clock rise time	T_{SDLH}		—	3	ns	
	SD_CLK clock fall time	T_{SDHL}		—	3	ns	
	SD_CMD, SD_DATA output delay	T_{SDODLY}		−6.2	2.5	ns	
	SD_CMD, SD_DATA input set up time	T_{SDIS}		4	—	ns	
	SD_CMD, SD_DATA input hold time	T_{SDIH}		2	—	ns	
	SD_CMD, SD_DATA input data width	T_{SDIDW}		—	—	ns	

Table 58.40 3.3 V SDHI timing (2 of 2)

Conditions: V_{OH} = VDD1833 × 0.5, V_{OL} = VDD1833 × 0.5 (VDD1833 = 3.3 V)

Parameter		Symbol	Conditions	Min.	Max.	Unit	Reference figure
(SD) Default Speed (eMMC) Backwards Compatibility	SD_CLK clock cycle	T_{SDCYC}	C = 40 pF Drive Strength CLK: High Others: Middle	40	—	ns	Figure 58.72
	SD_CLK clock high level width	T_{SDWH}		10	—	ns	
	SD_CLK clock low level width	T_{SDWL}		10	—	ns	
	SD_CLK clock rise time	T_{SDLH}		—	10	ns	
	SD_CLK clock fall time	T_{SDHL}		—	10	ns	
	SD_CMD, SD_DATA output delay	T_{SDODLY}		−7.5	2.5	ns	
	SD_CMD, SD_DATA input set up time	T_{SDIS}		4	—	ns	
	SD_CMD, SD_DATA input hold time	T_{SDIH}		2	—	ns	
	SD_CMD, SD_DATA input data width	T_{SDIDW}		—	—	ns	
(eMMC) High Speed DDR	SD_CLK clock cycle	T_{SDCYC}	C = 30 pF Drive Strength CLK: High Others: High	20	—	ns	Figure 58.73
	SD_CLK clock high level width	T_{SDWH}		9	11	ns	
	SD_CLK clock low level width	T_{SDWL}		9	11	ns	
	SD_CLK clock rise time	T_{SDLH}		—	3	ns	
	SD_CLK clock fall time	T_{SDHL}		—	3	ns	
	SD_CMD output delay (SDR)	T_{SDODLY}		−6	6	ns	
	SD_CMD input set up time (SDR)	T_{SDIS}		4.8	—	ns	
	SD_CMD input hold time (SDR)	T_{SDIH}		2.5	—	ns	
	SD_DATA output delay (DDR)	T_{SDODLY_DDR}		2.5	6.5	ns	
	SD_DATA input set up time (DDR)	T_{SDIS_DDR}		1.4	—	ns	
	SD_DATA input hold time (DDR)	T_{SDIH_DDR}		1.5	—	ns	

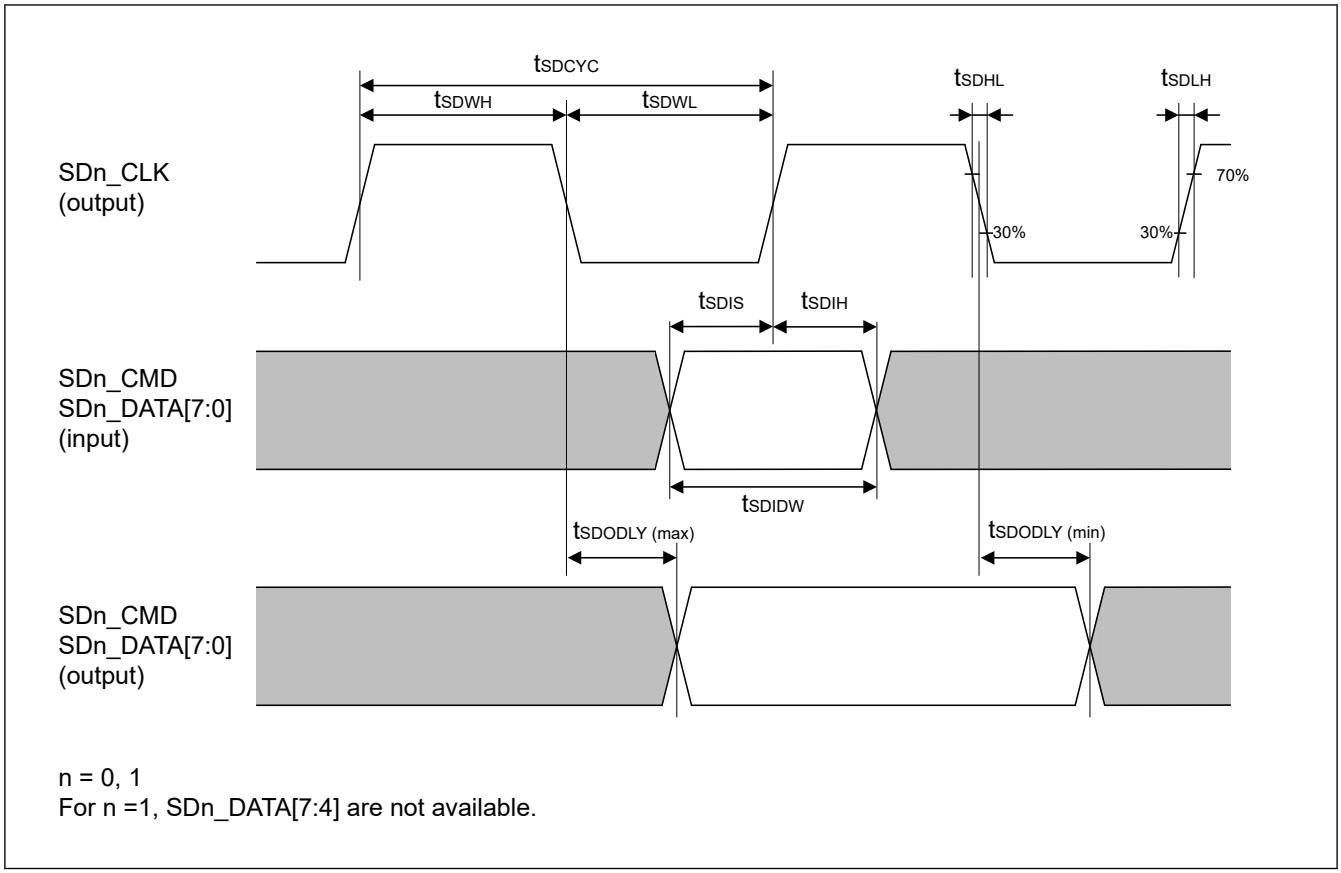
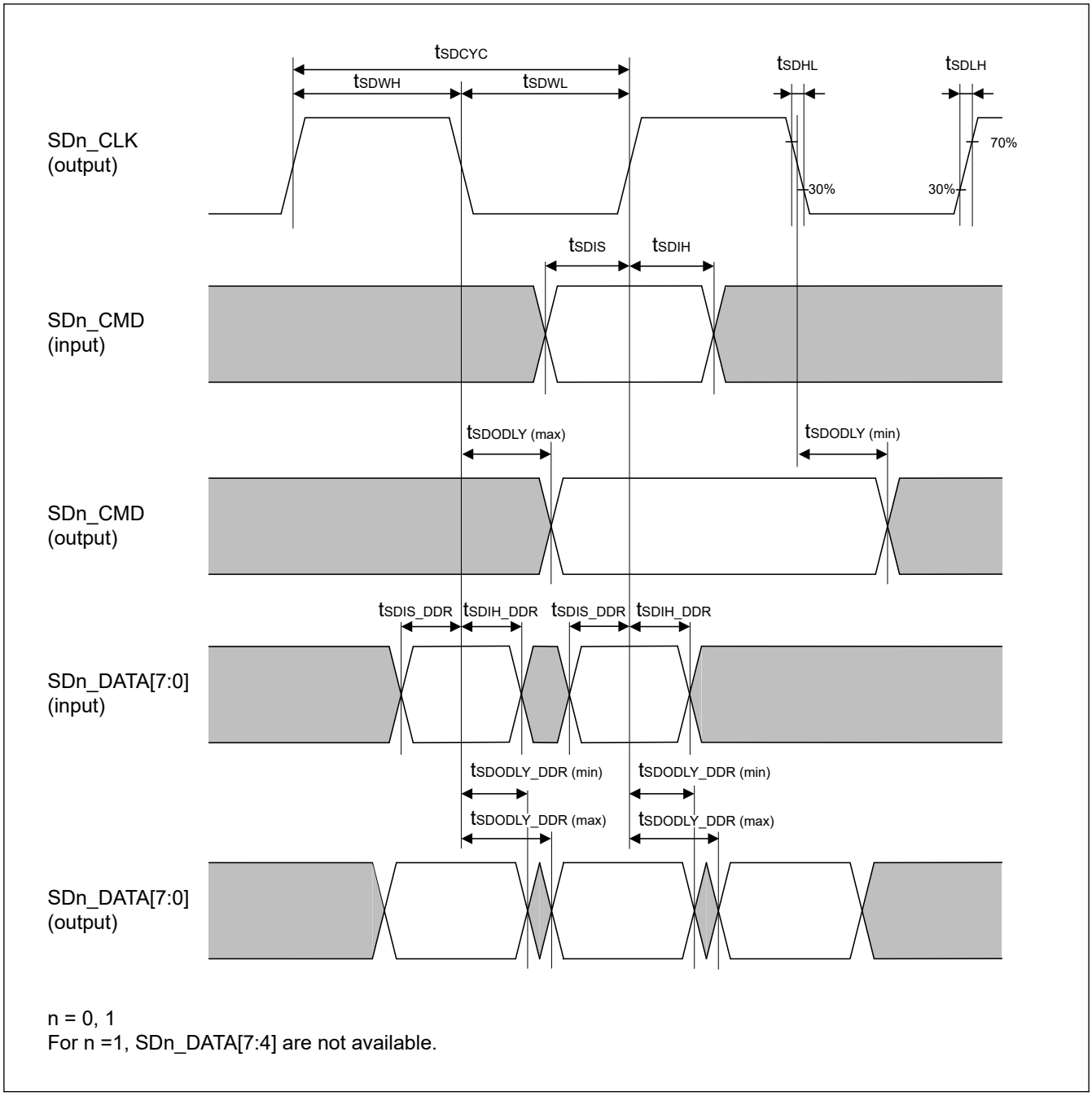


Figure 58.72 SDHI timing (SDR)



n = 0, 1
For n =1, SDn_DATA[7:4] are not available.

Figure 58.73 SDHI timing (DDR)

58.6 USB Characteristics

The USB PHY of this LSI is compliant with the Universal Serial Bus Specification, Revision 2.0.

Note: 30 (±1%) kΩ external resistor must be connected between USB_VUBUSIN pin and VBUS voltage supply.
200 (±1%) Ω external resistor must be connected between USB_TXRTUNE pin and VSS.

58.7 A/D Conversion Characteristics

Table 58.41 12-Bit A/D conversion characteristics (1 of 2)

Parameter	Min.	Typ.	Max.	Unit	Reference figure
Resolution			12	bits	—

Table 58.41 12-Bit A/D conversion characteristics (2 of 2)

Parameter		Min.	Typ.	Max.	Unit	Reference figure
Analog input capacitance		—	—	15	pF	—
Conversion time*1 Permissible signal source impedance Max. = 1.0 kΩ	Channel-dedicated sample-and-hold circuits in use	0.64	—	—	μs	—
	Channel-dedicated sample-and-hold circuits not in use	0.32	—	—		—
Offset error		—	—	±50	LSB	—
Full-scale error		—	—	±50	LSB	—
Quantization error		—	±0.5	—	LSB	—
Absolute accuracy		—	—	±55	LSB	—
DNL differential non-linearity error		—	—	±2	LSB	—
INL integral non-linearity error		—	—	±3.5	LSB	—
Dynamic range		0.02	—	AVDDREF_ADCn - 0.02	V	—

Note: The specified values in the table apply when there is no access to the external bus during A/D conversion. If access proceeds during A/D conversion, values may not fall within the specified ranges.

Note 1. The conversion time is the total of the sampling time and the comparison time.

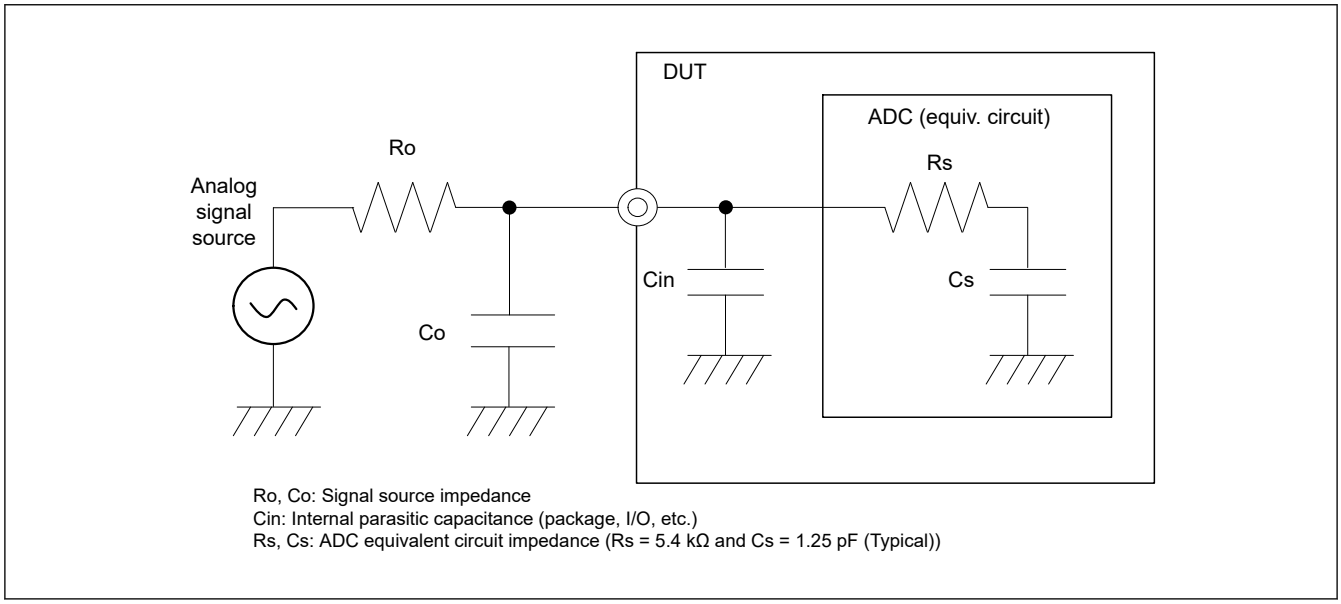


Figure 58.74 A/D converter equivalent circuit and peripheral configuration diagram

58.8 Temperature Sensor Characteristics

Table 58.42 Temperature sensor characteristics

Parameter	Min.	Typ.	Max.	Unit	Test Conditions
Relative accuracy	—	±2	—	°C	*1
Temperature slope	—	0.0625	—	°C/LSB	—
Output code (at 25°C)	—	1751 (decimal)	—	—	SCRR register

Note 1. 2-point calibration (Tj = −40°C and Tj = 125°C) and 8 times averaging.

58.9 PCI-Express Characteristics

The PCI Express PHY of this LSI is compliant with the PCI Express® Base Specification 3.1 (Gen1/Gen2/Gen3).

Table 58.43 PCI Express REFCLK input characteristics

Item	Symbol	Condition	Min	Typ	Max	Unit
Input single-end termination impedance	$Z_{RX-SINGLEEND}$	—	10	—	—	kΩ
Differential input peak-to-peak voltage	$V_{RXCLK-DPP}$	—	120	—	—	mV
Input common mode voltage	$V_{RXCLK-DC-CM}$	—	0.25	0.6	0.95	V
Absolute single-end input voltage	$V_{RXCLK-SE}$	—	−0.3	—	1.1	V

58.10 DDRSS (LPDDR4) Characteristics

The LPDDR4 PHY of this LSI is compliant with the JEDEC 209-4D standard.

Note: 120 (±1%) Ω external resistor must be connected between DDR_ZN and VSS.

58.11 Debug Interface Timing

Condition: V_{OH} = VDD33 × 0.5, V_{OL} = VDD33 × 0.5

Table 58.44 Debug interface timing

Parameter		Symbol	Min.	Max.	Unit	Reference figure
TCK cycle time	With an ICE connected	t_{TCKcyc}	30*1	—	ns	Figure 58.75
	For use in BSCAN		80	—		
TCK high pulse width		t_{TCKH}	0.4	0.6	t_{TCKcyc}	
TCK low pulse width		t_{TCKL}	0.4	0.6	t_{TCKcyc}	
TDI setup time		t_{TDIS}	5	—	ns	Figure 58.76 Output load: 30 pF
TDI hold time		t_{TDIH}	5	—	ns	
TMS/SWDIO setup time		t_{TMSS}	5	—	ns	
TMS/SWDIO hold time		t_{TMSH}	5	—	ns	
SWDIO delay time		t_{SWDO}	—	15	ns	
TDO delay time	With an ICE connected	t_{TDOD}	—	15	ns	
	For use in BSCAN		—	22		
Capture register setup time		t_{CAPTS}	5	—	ns	Figure 58.77
Capture register hold time		t_{CAPTH}	5	—	ns	
Update register delay time		$t_{UPDATED}$	—	15	ns	

Note 1. This value is the minimum cycle time for the normal operation of internal circuits.
The actual cycle time should be determined in consideration of the TCK capture edge timing and cable length of the connected ICE.

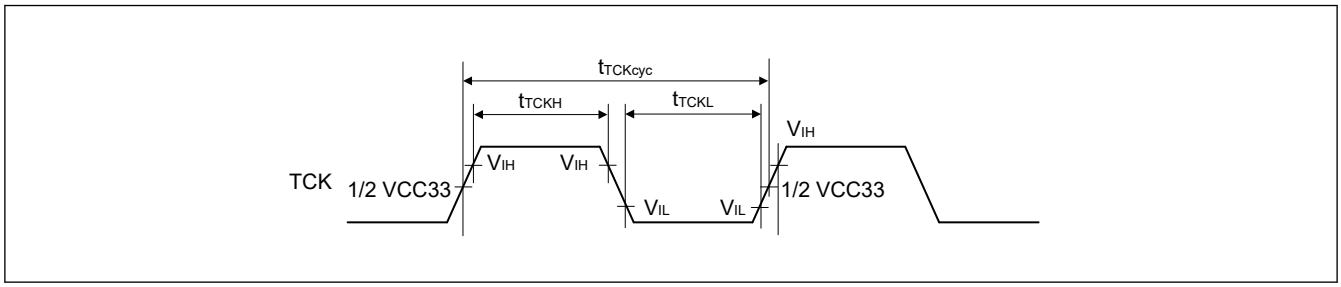


Figure 58.75 TCK input timing

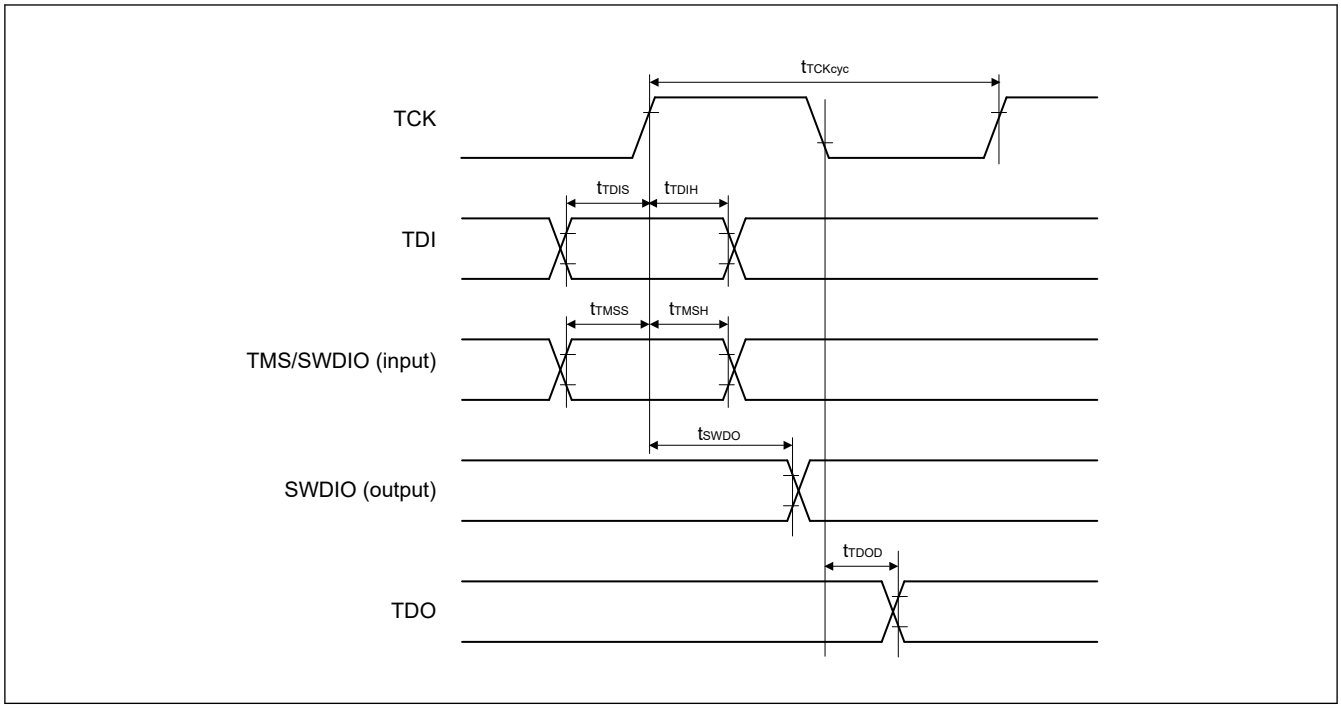


Figure 58.76 Data transfer timing

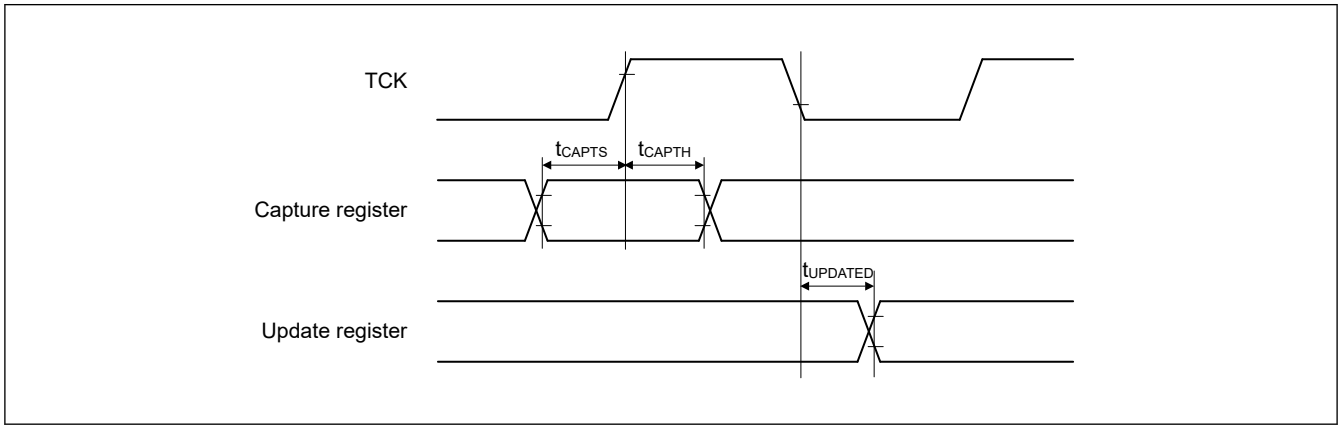
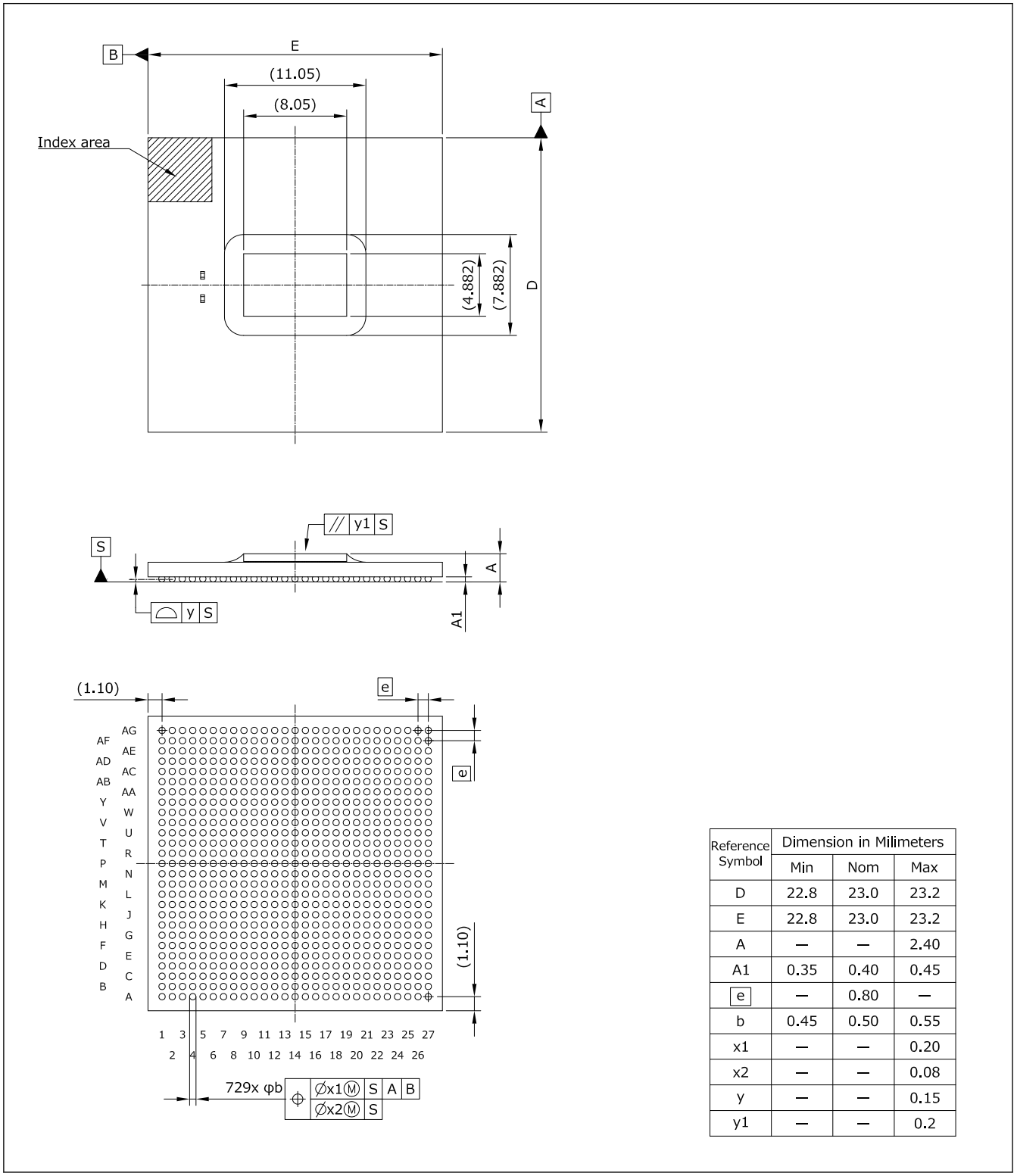


Figure 58.77 Boundary scan input/output timing

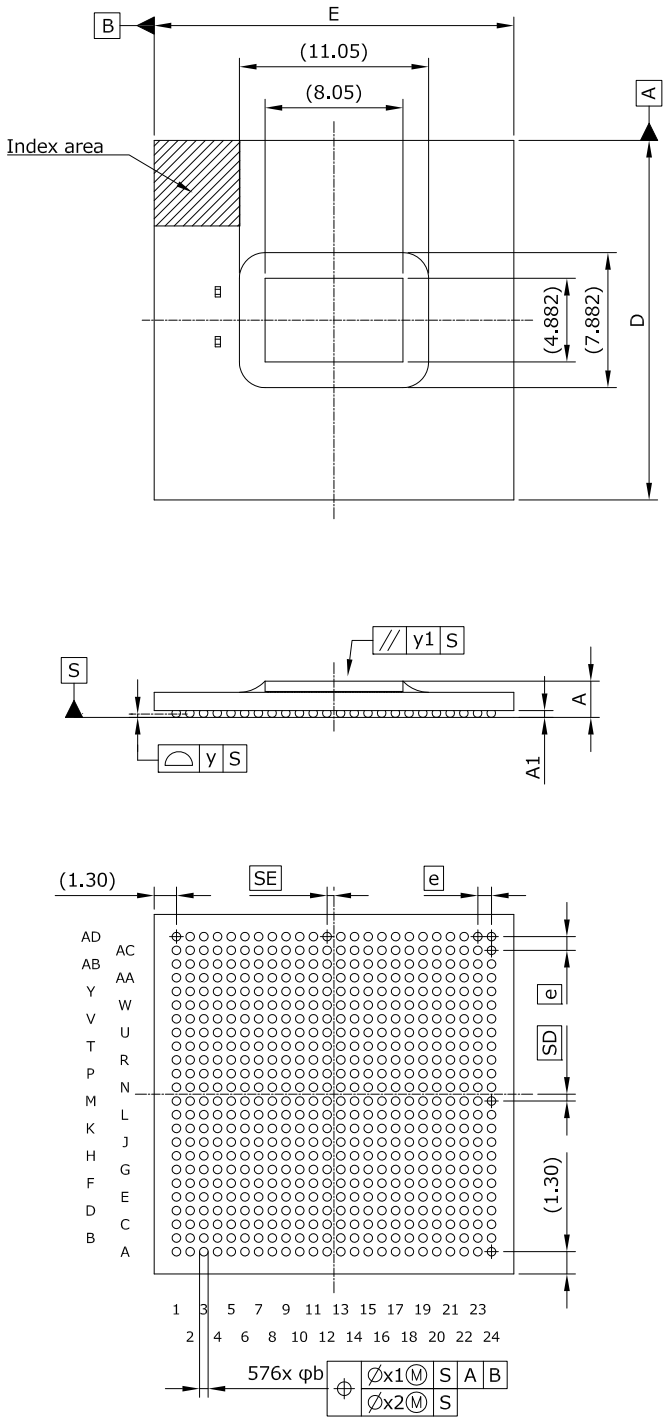
Appendix 1. Package Dimensions

Information on the latest version of the package dimensions or mountings is displayed in "Packages" on the Renesas Electronics Corporation website.



Reference Symbol	Dimension in Milimeters		
	Min	Nom	Max
D	22.8	23.0	23.2
E	22.8	23.0	23.2
A	—	—	2.40
A1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x1	—	—	0.20
x2	—	—	0.08
y	—	—	0.15
y1	—	—	0.2

Figure 1.1 729-pin FCBGA



Reference Symbol	Dimension in Milimeters		
	Min	Nom	Max
D	20.85	21.0	21.15
E	20.85	21.0	21.15
A	—	—	2.30
A1	0.35	0.40	0.45
e	—	0.80	—
SD	—	0.4	—
SE	—	0.4	—
b	0.45	0.50	0.55
x1	—	—	0.20
x2	—	—	0.08
y	—	—	0.15
y1	—	—	0.2

Figure 1.2 576-pin FCBGA

Revision History

Revision 1.00 — June 28, 2024

Initial release

Rvision 1.10 — October 25, 2024

4. Operating Modes:
- Added 4.6.11 SCI Authentication Boot Procedure with Plaintext ID.
- Added 4.6.12 USB Authentication Boot Procedure with Plaintext ID.

18. Multi-Function Timer Pulse Unit 3 (MTU3):
- Updated Table 18.1 MTU specifications.
- Updated Table 18.2 MTU functions.
- Removed 18.4.12 Dead Time Compensation and 18.4.12.1 Example of Dead Time Compensation Setting Procedure.

28. Ethernet MAC (GMAC):
- Added No. 3 and No. 4 to Table 28.104 Known issue and workaround.

46. Serial Host Interface (SHOSTIF):
- Updated Figure 46.4 WRITESTATUSm instruction (in case of m = 0).

56. PCI Express 3.0 Interface (PCIE):
- Updated Figure 56.9 Descriptor chain.

57. LPDDR4 SDRAM Subsystem (DDRSS):
- Updated the footnote in the DDR_MEMC_DENALI_CTL_x registers (x = 00, 318, 319, 382-387, 716-731).
- Updated the footnote in the DDR_MEMC_MCAR_AXCTL0-4 registers.
- Updated the footnote int the DDR_MEMC_MCAR_CTL and DDR_MEMC_MCAR_MON registers.
- Updated Table 57.35 Restrictions.

58. Electrical Characteristics:
- Updated Table 58.29 SCI, SCIE timing.

RZ/T2H and RZ/N2H Groups User’s Manual: Hardware

Publication Date: Rev.1.10 Oct. 25, 2024
Rev.1.00 Jun. 28, 2024

Published by: Renesas Electronics Corporation

RZ/T Series for Real-Time Control
RZ/T2H and RZ/N2H Groups

Renesas Electronics Corporation

R01UH1039EJ0110